Gary Tan
Gee Kin Yeo
Stephen John Turner
Yong Meng Teo (Eds.)

Communications in Computer and Information Science 402

AsiaSim 2013

13th International Conference on Systems Simulation
Singapore, November 2013
Proceedings

Springer

Communications in Computer and Information Science 402

Gary Tan Gee Kin Yeo
Stephen John Turner Yong Meng Teo (Eds.)

AsiaSim 2013

13th International Conference on Systems Simulation
Singapore, November 6-8, 2013
Proceedings

Springer

Volume Editors

Gary Tan
Yong Meng Teo
National University of Singapore, School of Computing
13 Computing Drive, Singapore 117417
E-mail: {gtan, teoym}@comp.nus.edu.sg

Gee Kin Yeo
Society of Simulation and Gaming of Singapore
c/o National University of Singapore, School of Computing
13 Computing Drive, Singapore 117417
E-mail: yeogk@ssagsg.org

Stephen John Turner
Nanyang Technological University, School of Computer Engineering
Nanyang Avenue, Singapore 639798
E-mail: steve@pmail.ntu.edu.sg

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-45036-5 e-ISBN 978-3-642-45037-2
DOI 10.1007/978-3-642-45037-2
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013952949

CR Subject Classification (1998): I.6, H.1, F.2, C.4

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The AsiaSim conference is an annual international conference that started in 1999, and has primarily been organized by the three Asian simulation societies: Chinese Association for System Simulation (CASS), Japanese Society for Simulation Technology (JSST), and Korea Society for Simulation (KSS). In 2011, the Federation of Asia Simulation Societies (ASIASIM) was set up to promote the advancement of modelling and simulation in industry, research and development in Asia and beyond. After a full zodiac cycle of 12 years, the AsiaSim series finally left the "Golden Triangle" of China, Japan, and Korea, and was held in Singapore for the first time. We were proud to host AsiaSim 2013, and on behalf of the Organizing Committee of AsiaSim 2013, we welcome you to the proceedings of AsiaSim 2013.

Asiasim 2013 was organized by the Society of Simulation and Gaming of Singapore, the National University of Singapore, and Nanyang Technological University. The Society of Simulation and Gaming of Singapore is a non-profit professional organization set up to contribute to the development of simulation and gaming in Singapore and the region. It is a focused community for researchers, practitioners, and developers who are keen in furthering their professional knowledge through learning and working together and promoting experiential activities in the public.

We received 95 full papers and six short papers. Submissions came from China, Japan, Korea, India, France, UK, Germany, Sweden, Kuwait, Morocco, Malaysia, and of course Singapore. After an intensive review process by a carefully assembled International Program Committee, where each paper was reviewed by no less than three reviewers, we finally accepted 45 full papers and 18 short papers. Six of the 45 papers were shortlisted for the best paper award.

These papers are now consolidated in this volume of the *Communications in Computer and Information Science* series, and are divided into many relevant topics, including Agent-Based Simulation, Simulation Methods and Tools, Visualization, Modeling Methodology, Simulation in Science and Engineering, High-Performance Computing and Simulation, and Parallel and Distributed Simulation. The diversity of topics presented in this conference will certainly make for healthy exchange of research ideas and technical exchanges.

We would like to take this opportunity to thank the ASIASIM Federation for allowing us to host AsiaSim 2013 in Singapore.

We also thank the members of the Program Committee for their valuable effort in the review of the submitted papers. Finally, we would also like to thank

our technical co-sponsors and sponsors. Your contributions and support have helped to make AsiaSim 2013 a reality and a success.

October 2013

Gary Tan
Gee Kin Yeo
Stephen John Turner
Yong Meng Teo

AsiaSim 2013 Organization

Conference Chairs

Gary Tan	National University of Singapore, Singapore
Gee Kin Yeo	Society of Simulation and Gaming of Singapore

Program Chairs

Stephen John Turner	Nanyang Technological University, Singapore
Yong Meng Teo	National University of Singapore, Singapore

International Program Committee

Tiranee Achalakul	King Mongkut's University of Technology Thonburi, Thailand
Rassul Ayani	Royal Institute of Technology (KTH), Sweden
Heiko Aydt	TUM-CREATE, Singapore
Azzedine Boukerche	University of Ottawa, Canada
Agostino G. Bruzzone	University of Genoa, Italy
Cristina Carbunaru	National University of Singapore, Singapore
Ian Chai	Multimedia University, Malaysia
Huah Yong Chan	Universiti Sains Malaysia, Malaysia
Dan Chen	China University of Geosciences, China
Franco Cicirelli	University of Calabria, Italy
Lorenzo Donatiello	University of Bologna, Italy
Alois Ferscha	Johannes Kepler Universität, Austria
Richard M. Fujimoto	Georgia Institute of Technology, USA
Ross Gore	Old Dominion University, USA
Kyoko Hasegawa	Ritsumeikan University, Japan
Seth Hetu	SMART, USA
Terence Hung	Institute of High Performance Computing, Singapore
Taku Itoh	Tokyo University of Technology, Japan
Helen Karatza	Aristotle University of Thessaloniki, Greece
Myoung-Hee Kim	Ewha Womans University, South Korea
Yun-Bae Kim	Sung Kyun Kwan University, South Korea
Akinori Kimura	Ashikaga Institute of Technology, Japan
Noriyuki Komine	Tokai University, Japan
Michael Lees	Nanyang Technological University, Singapore

Axel Lehmann	Universität de Bundeswehr München, Germany
Bo Hu Li	Beihang University, China
Malcolm Low	D-SIMLAB Technologies, Singapore
Zaharuddin Mohamed	Universiti Teknologi Malaysia, Malaysia
Farshad Moradi	Swedish Defence Research Agency, Sweden
Navonil Mustafee	University of Exeter, UK
Libero Nigro	University of Calabria, Italy
Ernie H. Page	The MITRE Corp, USA
Soo-Hyun Park	Kookmin University, South Korea
Irena Patasiene	Kaunas University of Technology, Lithuania
Kalyan Perumalla	Oak Ridge National Laboratory, USA
J. Mark Pullen	George Mason University, USA
Francesco Quaglia	University of Rome, Italy
Andreas Riener	Johannes Kepler Universität, Austria
George F. Riley	Georgia Institute of Technology, USA
Yahya Mohd Sam	Universiti Teknologi Malaysia, Malaysia
Thomas Schulze	University of Magdeburg, Germany
Rob Simmonds	University of Calgary, Canada
Xiao Song	Beihang University, China
Yingwen Song	Shanghai Advanced Research Institute, China
Yuen Jien Soo	National University of Singapore, Singapore
Steffen Strassburger	Technical University of Limenau, Germany
Claudia Szabo	University of Adelaide, Australia
Gary Tan	National University of Singapore, Singapore
Sun Teck Tan	National University of Singapore, Singapore
Satoshi Tanaka	Ritsumeikan University, Japan
Pichaya Tandayya	Prince of Songkla University, Thailand
Simon Taylor	Brunel University, UK
Georgios Theodoropoulos	Durham University, UK
Nam Thoai	Ho Chi Minh University of Technology, Vietnam
Andreas Tolk	SimIS Inc., USA
Carl Tropper	McGill University, Canada
Bimlesh Wadhwa	National University of Singapore, Singapore
Norhaliza Abd Wahab	Universiti Teknologi Malaysia, Malaysia
Gabriel Wainer	University of Carleton, Canada
Xianbing Wang	Wuhan University, China
Zijian Wang	Chinese Academy of Sciences, China
Philip A. Wilsey	University of Cincinnati, USA
Tianyuan Xiao	Tsinghua University, China
Yiping Yao	National University of Defense Technology, China
Gee Kin Yeo	Society of Simulation and Gaming, Singapore
Lin Zhang	Beihang University, China

AsiaSim 2013 Organizing Committee

Best Paper Chair

Axel Lehmann	Universität de Bundeswehr München, Germany

Best Paper Sub-Committee

Yun-Bae Kim	Sung Kyun Kwan University, South Korea
Gary Tan	National University of Singapore, Singapore
Satoshi Tanaka	Ritsumeikan University, Japan
Yong Meng Teo	National University of Singapore, Singapore
Stephen John Turner	Nanyang Technological University, Singapore
Lin Zhang	Beihang University, China

Finance Chair

Sun Teck Tan	National University of Singapore, Singapore

Local Arrangements Chairs

Bimlesh Wadhwa	National University of Singapore, Singapore
Yuen Jien Soo	National University of Singapore, Singapore
Yew Khim Tan	Taknet Systems Pte. Ltd., Singapore

Industry Chair

Malcolm Low	D-SIMLAB Technologies, Singapore

Publicity Chair

Michael Lees	Nanyang Technological University, Singapore

Publications Chair

Xiao Song	Beihang University, China

Web Masters

Chun Saen Lok	Nanyang Technological University, Singapore
Irene Goh	Nanyang Technological University, Singapore

ASIASIM Council 2013

President

Bo Hu Li

Executive Vice-President

Lin Zhang (Vice-President of CASS)

Council Members

Satoshi Tanaka (Vice-President of JSST)
Osamu Ono (JSST board member, former President of JSST)
Yun-Bae Kim (President of KSS)
Kang Sun Lee (Executive Director of Foreign Affairs of KSS)
Gary Tan (Vice-President, SSAG Singapore)
Stephen John Turner (Professor of NTU)
Yahaya Md Sam (Associate Professor of UTM)

Honorary Member

Axel Lehmann (Professor of Universität de Bundeswehr München, Germany)

Sponsors

ASIASIM: Federation of Asia Simulation Societies
Society of Simulation and Gaming of Singapore
Advent2 Labs Consultation Pte. Ltd.
HGST, a Western Digital company
Lee Foundation

Technical Co-sponsors

IEEE Singapore Computer Chapter
Chinese Association for System Simulation (CASS)
Japanese Society for Simulation Technology (JSST)
Korea Society for Simulation (KSS)
Society for Modeling and Simulation International (SCS)
Federation of European Simulation Societies (EUROSIM)

Organizers

Society of Simulation and Gaming of Singapore
National University of Singapore, Singapore
Nanyang Technological University, Singapore

Keynote Speakers

Keynote I: Making Sense of a Complex World

Professor Peter Sloot

Distinguished Research Professor and Professor of Computational Science, University of Amsterdam, The Netherlands
Professor of Advanced Computing, St. Petersburg State University, Russia
Visiting Professor of Complex Systems, Nanyang Technological University, Singapore

Abstract

We live in a complex world and are surrounded by complex systems: from a biological cell, made of thousands of different molecules that seamlessly work together, to millions of computer systems that should work together, to our society, a collection of seven billion individuals that try to work and live together. These complex systems display endless signatures of order, disorder, self-organization, and self-annihilation. Understanding, quantifying, and handling this complexity is one of the biggest scientific challenges of our time. Using examples from our recent research, I will introduce a new paradigm of information processing in natural systems that guide the design and evaluation of models for simulating complex dynamical systems. I will show how this paradigm can bridge the uncomfortable gap between Popperian deductivism and Baconian inductivism.

Biography

Professor Peter Sloot is a Distinguished Research Professor and Professor of Computational Science at the University of Amsterdam, The Netherlands. He received his PhD (Computer Science) in 1988 and MSc (Chemistry) and MSc (Physics) in 1983, all from the University of Amsterdam. Peter writes, "I try to understand how nature processes information. I study this 'natural information processing' in complex systems by computational modeling and simulation as well as through formal methods. My work is applied to a large variety of disciplines with a focus on – but not limited to – biomedicine. Recent work is on modeling the virology and epidemiology of infectious diseases, notably HIV, through complex networks, cellular automata and multi-agents. Recently in my work I try to build bridges to socio-dynamics."

Professor Peter Sloot currently leads two large EU projects: ViroLab and DynaNets and supervises research from various NIH, NSF and NWO and Royal Academy projects. He has supervised over 38 PhD dissertations. He has published over 470 papers, books, chapters and edited volumes. He has given over 20 radio and TV interviews on various scientific topics, including two documentaries on his work. He is Editor in Chief of *Journal of Computational Science* and *Future Generation Computing Systems*, both published by Elsevier. He is General Chair of the ICCS series of Conferences on Computational Science. He has given an average of eight international keynote talks and invited lectures per year over the past 5 years.

Keynote II: Challenges in Three-Level Parallelization-Based Analytic Simulation

Professor Yiping Yao

Vice-Director, Institute of Simulation Engineering
Professor, School of Information Systems, and Management
National University of Defense Technology, Changsha, Hunan, China

Abstract

Analytic simulation is an effective approach to study and analyze complex systems. As a peer methodology to experiment and theory, it is used more and more widely in the area of defense and economy. With the in-depth development of the applications of analytic simulation, their scale is becoming larger and the models are becoming more complicated. As a result, much more computing resources are required than ever before. To shorten the execution time of simulation has become an urgent job. Parallel computing is an effective way to solve the problem. In this presentation, we will introduce the characteristics of computing of analytic simulation such as multi-sample, as fast as possible, complex model calculation and synchronization for constraint of causality. According to these characteristics, we present a three-level parallelization solution: multi-sample parallelization, multi-entity parallelization, and complex model calculation parallelization.

Multi-sample parallelization is job-level parallelization: as there are no dependencies among samples, it is relatively easy to implement. Multi-entity parallelization, is MPI task-level parallelization; it is the core issue of parallel simulation. There are several ways to achieve calculation parallelization of complex

models now: multicore CPU, GPGPU, FPGA, DSP and MIC etc. In this presentation, we will discuss the challenges in these three levels and introduce our solutions as well as give a perspective on each of them.

Biography

Dr. Yiping Yao is Professor of the School of Information System and Management and the school of Computer Science at the National University of Defense Technology (NUDT). He is also the Vice-Director of the Institute of Simulation Engineering at NUDT. He received his PhD and MS degrees from NUDT in Computer Science in 2004 and 1987. His research interests include: parallel and distributed simulation, agent-based and component-based modeling and simulation, and hardware-in-the-loop real-time simulation. He has authored or co-authored over 130 technical papers in the above areas. He led the development of parallel/distributed/real-time simulation software systems including YH-SUPE (support environment for parallel discrete event simulation), Starlink (HLA Run-Time Infrastructure), and YHSIM (hardware-in-the-loop real-time modeling and simulation platform), all of which have been widely used in China. He won numerous Science and Technology Progress Awards of China: two second prizes at the state level, three first prizes, and seven second prizes at the ministerial level.

Dr. Yao is a member of the IEEE and ACM, and a member of the board of directors of the Chinese Association for System Simulation (CASS). He was the general co-chair of PADS 2012 and CGSS 2011 (the first Sino-German Symposium on "Parallel and Distributed Discrete-event Simulation, Experimental Validation and Multi-Scale Applications in Computational Systems Biology"). He has served as Program Committee member for several international conferences. The most recent ones were: PADS 2013, 2012, DS-RT 2012, 2011, 2010, 2009, SIMUL 2013, 2012, 2011, 2010, 2009, SimTools 2010, SummerSim 2009, SIMULTECH 2013, 2012, 2011, DCIA 2012, 2011, 2010.

Table of Contents

Session 1: Best Paper Session I

Session 2A: Agent-Based Simulation

Session 2B: Scheduling Algorithms

Session 2C: Simulation Methods and Tools I

Session 3A: Simulation and Visualization

Session 3B: Modeling Methodology

Session 3C: Simulation Methods and Tools II

Session 4: Best Paper Session II

Session 5A: Simulation in Science and Engineering I

Session 5B: Parallel and Distributed Simulation

Session 6A: Short Paper Session I

Session 6B: Short Paper Session II

Session 6C: Short Paper Session III

Session 7A: Simulation in Science and Engineering II

Session 7B: High Level Architecture

Session 7C: Aircraft Simulation

Session 8A: Simulation in Science and Engineering III

Session 8B: High Performance Computing and Simulation

Session 8C: Simulation Applications

A Multi-threaded Execution Model for the Agent-Based SEMSim Traffic Simulation*

Heiko Aydt[1], Yadong Xu[1,2], Michael Lees[3], and Alois Knoll[4]

[1] TUM CREATE Ltd., Singapore
[2] Nanyang Technological University, Singapore
[3] University of Amsterdam, The Netherlands
[4] Technical University of Munich, Germany

Abstract. An efficient simulation execution engine is crucial for agent-based traffic simulation. Depending on the size of the simulation scenario the execution engine would have to update several thousand agents during a single time step. This update may also include route calculations which are computationally expensive. The ability to dynamically re-calculate the route of agents is a feature often not required in classical microscopic traffic simulations. However, for the agent-based traffic simulation which is part of the Scalable Electro-Mobility Simulation (SEMSim) platform, the routing ability of agents is an important feature. In this paper, we describe a multi-threaded simulation engine that explicitly supports routing capabilities for every agent. In addition, we analyse the efficiency and performance of our execution model in the context of a Singapore-based simulation scenario.

1 Introduction

In the context of electro-mobility research we are currently developing the SEMSim (Scalable Electro-Mobility Simulation) platform. This platform will provide the capability to simulate the various aspects related to electro-mobility for an entire city. In particular, this includes microscopic simulation of the traffic system (SEMSim Traffic) and microscopic simulation of the power system (SEMSim Power). SEMSim Traffic is more precisely an agent-based traffic simulation where each agent represents a driver-vehicle unit. This agent not only integrates typical driver behaviour models (e.g., car following, gap acceptance and lane changing models) but also explicit vehicle component models (e.g., drive train, battery, air conditioning).

Integrating these models and the ability to couple SEMSim Traffic with SEMSim Power allows us to study the impact of the disruptive technology 'electric vehicle' on the traffic and power infrastructure of an entire city. For example, the SEMSim platform will allow us to study the impact of the electric vehicle

* This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 1–12, 2013.

population on the power system under various electro-mobility scenarios (e.g., placement of charging infrastructure, vehicle design, regulations and operating policies). The goal of this research is to develop the capability to analyse an entire city, such as Singapore, from a complex systems perspective.

Coupling of the different simulation entities (e.g., SEMSim Traffic and SEMSim Power) will be done by means of the High Level Architecture (HLA)[1]. An electro-mobility simulation will thus be distributed across multiple nodes. In addition, SEMSim Traffic will also be parallelised to deal with the large number of agents and the variety of models they incorporate. Parallelisation can be done in multiple ways which we will further discuss in Section 2. In this paper we focus on parallelisation by means of using a multi-threaded execution engine.

We follow a bottom-up modelling approach for developing the SEMSim platform. As already mentioned, for SEMSim Traffic this means that individual agents will be equipped with driver behaviour and vehicle component models. We therefore also refer to this kind of simulation as nanoscopic traffic simulation, as opposed to microscopic traffic simulation which does not consider vehicle component models for individual agents. Large-scale microscopic traffic simulations are known to be very compute intensive. The additional vehicle models make a nanoscopic traffic simulation even more computationally demanding. An efficient simulation execution engine is thus very important.

In general, the agents in an agent-based simulation have the ability to react upon perceived changes in the environment. In the context of agent-based traffic simulation, for example, this refers to the ability of an agent to plan routes and drive from its current location to the destination. Driving requires the agent to accelerate/decelerate to achieve the desired velocity. In addition, it may have to change lanes at appropriate locations, avoiding collisions with other vehicles at any time. While dynamic re-routing may not be needed in many microscopic traffic simulations, this is an important feature in SEMSim Traffic. For example, it enables agents to change their routes towards the nearest charging station if necessary. In this paper, we describe a multi-threaded simulation execution engine that explicitly supports routing capabilities for every agent.

The remainder of this paper is structured as follows. Section 2 gives an overview of the different parallelisation techniques used for microscopic traffic simulations. Section 3 and Section 4 explain the underlying SEMSim Traffic simulation model and execution engine, respectively. We evaluate the proposed execution engine and discuss the experimental results in Section 5. We present our conclusions and discuss future work in Section 6.

2 Related Work

Parallel computing methods are used by a variety of existing microscopic traffic simulators. In general, parallelisation can be achieved in two ways: (1) by distributing the simulation on multiple nodes by using some form of partitioning and (2) by using multi-threading to exploit the advantages of shared-memory systems. Most work published in the literature is concerned with distributing

a simulation on multiple nodes. For this purpose, it is generally distinguished between functional decompositioning and domain decompositioning. Functional decompositioning can be found in DYNAMIT/MITSIM [3,4]. However, according to [9] functional decompositioning may be easier to implement but poses limitations on the achievable speed-up.

Using domain compositioning for parallel traffic simulation has been described in various works. Nagel and Rickert describe how domain decompositioning has been used in a parallel implementation of TRANSIMS [9] which is based on cellular automata for representing driving dynamics. Different geographical regions are processed by multiple CPUs. Various key challenges for domain decompositioning, concerned with dividing the network, is explained by Klefstad et al. [8]. Yet another parallel traffic simulation that uses domain decompositioning is FastTrans. Thulasidasan et al. describe different strategies to perform domain decompositioning and show how different strategies affect the simulation performance [13].

Distributing a simulation on multiple nodes has several advantages. For example, large-scale simulations may be too large to fit into the memory of a single compute node. If memory limitations are not an issue, it is in principle also possible to distribute the simulation on multiple cores on the same compute node. However, in this case, the advantages of a shared memory system may not be fully utilised. In principle, a distributed simulation is not limited to the number of processors (or cores). Of course, scalability limits the amount of cores that can effectively be used.

In contrast to distributed simulation, one of the main disadvantages of the multi-threading approach is the total number of available processors (or cores) which is limited to the number of processors available on a single compute node. As of 2013, high-end processors (e.g., Intel Xeon E7 family) provide up to 10 cores with support for hyper-threading which effectively doubles the number of threads that be executed concurrently. Another limitation is the memory available. Large-scale simulations that require large amounts of memory may be too big to fit into the memory of one compute node. Nevertheless, one important advantage of multi-threading solutions is the significantly reduced communication overhead needed for synchronisation. Given the current trend towards many-core processors, multi-threading solutions deserve serious consideration.

The work by Barceló et al. [2] is very relevant to the work presented in this paper. They describe the multi-threaded execution engine used for AIMSUN2 and analyse its performance. Their execution engine is based on the idea of grouping parts of the road network (i.e., lanes) together in such a way as to minimise the need for synchronisation. This is the same approach taken by us in this paper. A key difference between AIMSUN2 and SEMSim Traffic is the way how traffic is generated. According to [2], vehicles in AIMSUN2 do not have knowledge about their complete path along the network. In contrast, the agents in our traffic simulation have complete routes to follow in order to reach the destination. This is an important difference because route calculation is computationally expensive and the execution engine presented here takes this explicitly into consideration.

A noteworthy and very recent development is the IBM Mega Traffic Simulator (Megaffic), a microscopic traffic simulation that aims for being used in large-scale traffic simulation of mega cities [10]. Megaffic is based on a platform for massive agent-based simulation: X10-based Agent eXecutive Infrastructure for Simulation (XAXIS) [11]. Suzumura et al. evaluate the performance in terms of scalability, including an analysis of the scalability on a single node with a different number of threads [12], showing that Megaffic achieves a five-fold speed-up when using 12 threads.

3 Simulation Model

The road network is a directed graph in which edges represent lanes and vertices represent the start and end points of a lane in terms of longitude and latitude coordinates. The length L of a lane follows from the geographical distance between the start and end point of this lane. A lane is modelled by a spatial queue, i.e., a queue of agents in which each agent has also a spatial location relative to the start point in addition to its logical position in the queue. Figure 1 illustrates the concept of spatial queues. Agents are not directly placed in the queue. Instead, place holder objects that have the same size as the agent are placed in the queue. These place holder objects have a reference to the corresponding agent.

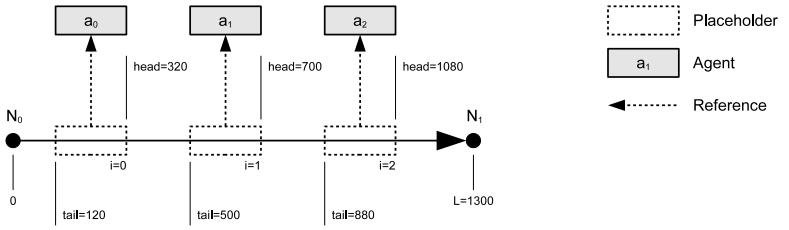


Fig. 1. A lane with an origin N_0 and destination N_1 is represented by a spatial queue with length L. This queue can contain an arbitrary number of agents. The index i indicates the logical position of an agent within the queue. In addition, the spatial position of the agent is indicated as distance from the origin of the lane N_0 to the head and tail of the vehicle.

When moving forward, agents can crossover from one lane to the next. During that transition period, an agent can be partially located on one lane and partially on the next (see Figure 2(a)). Technically this is done by having two place holder objects in either queue, both of which are referring to the same agent (see Figure 2(b)). Each time step, the place holder objects of the various agents in the simulation are moved forward by some distance Δs which depends on the agents current speed. Once an agent is leaving a lane, i.e., when its tail reaches the end of the lane, the place holder object is removed. Similarly, a new place holder object is created when an agent is entering a lane when its head reaches the beginning of the lane.

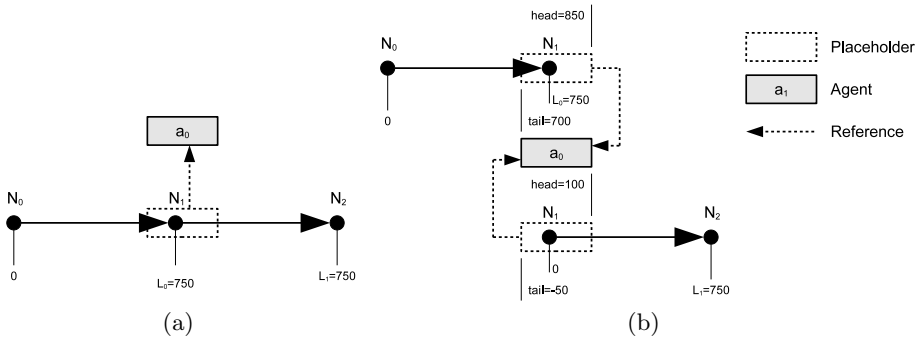


Fig. 2. An agent a_0 which is overlapping on two lanes (a) has place holder objects in each lane that reserve the required space for the agent (b). When the agent is moved forward, all place holder objects associated with this agent need to be moved. If a place holder object leaves the lane entirely, it is deleted and the agent is entirely part of the next lane.

This kind of crossing over is not to be confused with lane changes to adjacent lanes to the left or right for which there is no transition period.

Agent-based traffic simulation requires an agent to perceive its environment and executing certain actions which include route calculation, lane changes (if necessary or desired) and moving forward with a certain acceleration and speed. There are a number of standard models for lane changing and acceleration which can be found in the literature. For example, car following models (e.g., [7] and [14]) are used to determine the acceleration of the agent at every time step. This typically depends on the current speed and the distance to the next car in front as well as the speed of the car in front. In this paper, we focus on the following three operations an agent can perform: (1) routing, (2) lane changing and (3) moving forward (by accelerating and decelerating).

4 Execution Model

Microscopic traffic simulators typically generate fixed traffic before executing the simulation. Two common practices are using origin-destination matrices (e.g., PARAMICS [5]) and stochastic turning ratios at road junctions (e.g., AIMSUN [2] and VisSim [6]). For example, in case of AIMSUN, agents do not have knowledge about the complete route. Instead their driving directions are based on some probabilistic decision model. While this execution model avoids calculating entire routes, it would be too restrictive in the context of nanoscopic traffic simulation where energy consumption of individual agents is being considered. An important feature of the execution model presented in this section is the routing feature which is also the key difference to the multi-threaded execution engine used in AIMSUN/MT [2]

In principal, a time-stepped execution models is used as the underlying models involved in moving agents (such as the car following model) are inherently time-stepped. However, for any other behavioural models (e.g., decision models) or vehicle component models this may not be case. Therefore, in SEMSim Traffic we use a discrete-event engine that allows to schedule events at arbitrary moments of time. The event triggering of the execution model will be scheduled recurringly with a certain time interval (typically less or equal to 1 second), thus effectively emulating time-stepped behaviour. This approach enables to use time-stepped models alongside with model components that schedule events infrequently.

Since the scope of this paper is the execution model only, we consider time-stepped execution of the simulation where agents are updated once every time step. During each of these updates, an agent may re-calculate its route, change lanes if necessary and move forward. In a multi-threaded by-agent parallel execution model, all agents are distributed to multiple threads each of which is executing the various operations. This naive form of parallel execution may cause significant synchronisation overhead: initialising an agent (i.e., placing an agent on a lane), changing lanes or moving forward requires information of the state of multiple lanes. In order to maintain the integrity of a spatial queue, it is important that only one thread is operating on a queue at a time.

There are five synchronisation cases that requires threads to have varying mutually exclusive access to lanes: (1) inserting an agent into a queue (i.e., placing an agent onto a lane), (2) changing lanes, (3) moving forward without overlaps, (4) moving forward with overlap to the next queue and (5) moving forward with overlap to the previous queue. These cases are illustrated in Figure 3. A thread that operates on the granularity of agents needs mutually exclusive access to various spatial queues. During this operation, none of the other agents that are contained by the affected lanes can be processed as the exclusive access concerns the entire spatial queue.

Efficient parallelisation requires to minimise the need for synchronisation. This can be achieved by forming blocks of queues that are direct or indirect neighbours to each other. Neighbouring lanes are those lanes that can be reached by changing lanes. For example, if a road splits into two different directions, then the lanes of those two alternatives are not considered neighbours since lane changing between them is impossible. Figure 4 shows an example how several spatial queues are grouped together to form blocks.

Based on the concept of blocks, we can now introduce an alternative execution engine which performs block processing rather than processing by agents. The execution model is thus referred to as by-block execution model. The by-block parallel execution model has the advantage that threads operate on the granularity of blocks rather than agents. As a consequence, the above-mentioned synchronisation cases (1), (2) and (3) are completely eliminated as agent placement on lanes, lane changing and moving forward without overlaps is done within the boundaries of a single block. Since only one thread can process a block, there are only potential needs for synchronisation in cases (4) and (5).

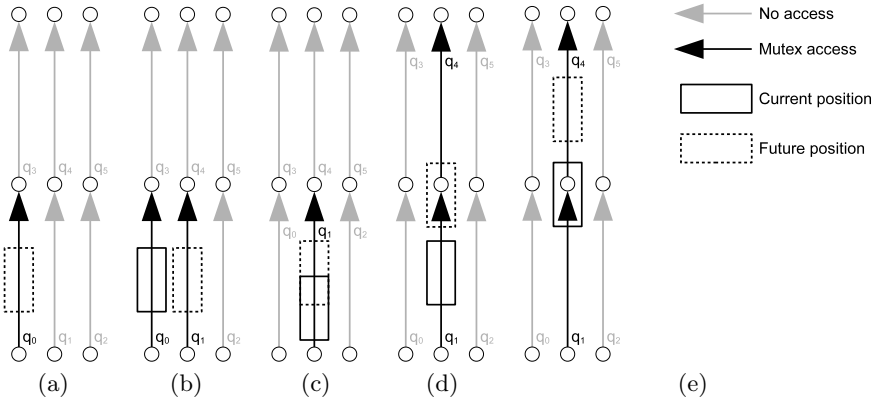


Fig. 3. Overview of the various synchronisation cases where threads need mutually exclusive access to spatial queues. Inserting an agent into a queue requires mutually exclusive access of this queue (see 3(a)). Changing lanes requires mutually exclusive access to the current lane and the target lane (see 3(b)). Moving forward requires mutually exclusive access at least to the current lane (see 3(c)). In case the agent is moving from one queue to another, mutually exclusive access to the next or previous queue is also needed (see 3(d) and 3(e), respectively).

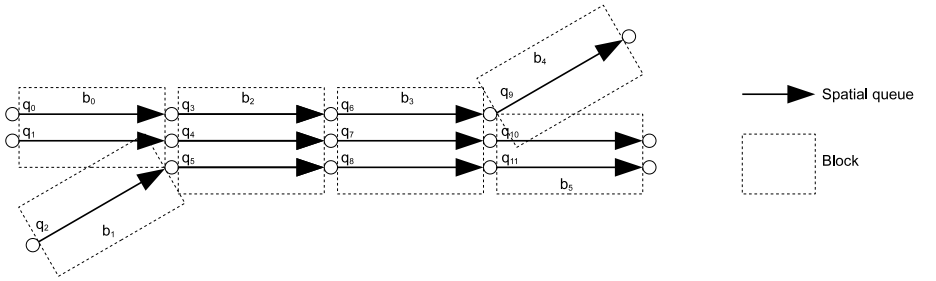


Fig. 4. Overview of grouping spatial queues to blocks. Spatial queues that belong to different branches are not grouped into one block because lane changing between them is impossible. For example, spatial queues q_0 and q_1 belong to a different block than spatial queue q_2.

We further reduce the probability of synchronisation by declaring blocks either as 'even' blocks or 'odd' blocks and processing them in batches. Which block is declared 'even' or 'odd' is arbitrary and not important so as long it is guaranteed that adjacent blocks are never of the same group. For example, consider Figure 4. If block b_0 is, say, 'even' then it follows that blocks b_1, b_2, b_3, b_4 and b_5 are 'even', 'odd', 'even', 'odd' and 'odd', respectively. Even and odd blocks are also kept separately in different sets B_0 and B_1. The first batch will process even blocks and the second batch will process odd blocks.

The by-block execution model, illustrated in Algorithm 1, performs two different steps. Step 1 is concerned with updating the routes of newly created agents or existing agents that need to change their route. Step 2 processes the blocks and updates the position of the agents in the various blocks. Step 2 is executed twice, first for the even blocks in B_0 and then for the odd blocks in B_1. A multi-threading parallel execution engine can be realised by having n threads performing the necessary operations during Step 1 and Step 2. For this purpose, each Step 1 worker thread removes one agent a from A and processes it. Similarly, Step 2 worker threads remove a block b from B and process it. Worker threads will continue until agents in A or all blocks in B have been processed. The processed agents and blocks are then consolidated and represent the set of agents and blocks that are being processed in the next cycle.

Algorithm 1. By-block execution model

```
A ← ∅;
B_0 ← ∅;
B_1 ← ∅;
repeat
    t ← t + δt;
    A ← A∪ generate_new_agents(t);
    perform_step1(A);
    perform_step2(B_0);
    perform_step2(B_1);
until A = ∅;
```

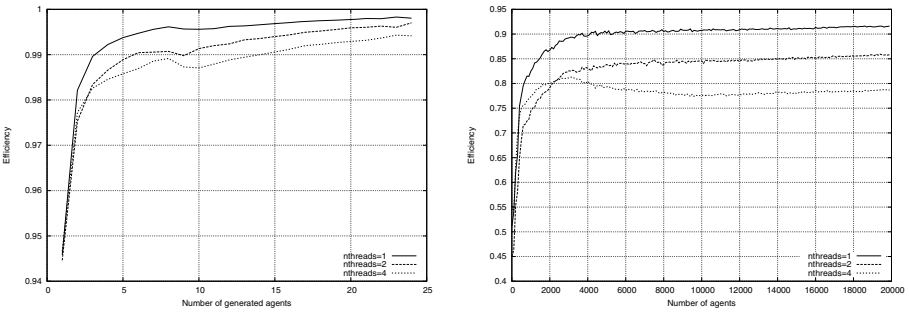
5 Experimental Evaluation

We evaluate the proposed parallel by-block execution model in two ways. First, we analyse the efficiency during the execution of Step 1 and Step 2 depending on the number of agents for an increasing number of threads. Second, we analyse the achieved speed-up depending on the number of agents for an increasing number of threads. For evaluation purposes, a Java implementation of the execution engine has been used. For the simulation scenario, we use road network of Singapore and generate traffic up to 20,000 agents in a 2-stage experiments. In Stage 1, the simulation generates agents up to a limit of 20,000 agents at which point Stage 2 begins. In Stage 2 no new agents are generated and no route calculations take place. The 2-stage approach is used in order to see the impact of Step 1 on the overall performance. The experiments have been performed on a system equipped with one Intel i5-2520M CPU with 4 cores, running at 2.5 GHz, and 8 GB memory.

We compare the efficiency E_1 and E_2 of Step 1 and Step 2, respectively, by measuring the time spent on useful computation t_u in comparison to the total time t_s spent to perform a step:

$$E = \frac{t_u}{t_s} \tag{1}$$

Overhead is caused by thread synchronisation and the time required to initialise the worker threads and the idling time by worker threads that need to wait for other threads to finish. In particular the latter can cause significant overhead if all threads except one have already finished their execution and need to wait for the last thread to finish. The overhead can be expected to increase with the number of threads. This is also supported by our experimental results, shown in Figure 5.



(a) (b)

Fig. 5. Efficiency of Step 1 (5(a)) and Step 2 (5(b)) execution depending on the number of threads

Figure 5 illustrates the efficiency of Step 1 and Step 2 depending on the number of threads used. The efficiency for Step 1 is significantly better than the efficiency for Step 2. This indicates that Step 2 is of greater importance in order to achieve good scalability. Figure 6 shows the average number of thread synchronisations during a single Step 2 execution. These results indicate that synchronisation is not a significant obstacle in order to achieve good scalability. Inefficiencies in Step 2 are thus due to thread initialisation and idling.

Regardless the issue of efficiency, the actual time spent on Step 1 and Step 2 is very different. Figure 7 illustrates the execution speed of the simulation during Stage 1 (with routing and thus with Step 1) and Stage 2 (without routing and thus without Step 1). Simulation execution during Stage 2 is faster than during Stage 1 by the order of roughly one magnitude. This somewhat reduces the impact of lower efficiency for Step 2 as Step 1 is more important in terms of execution time. However, as we explained in Section 1, SEMSim Traffic will

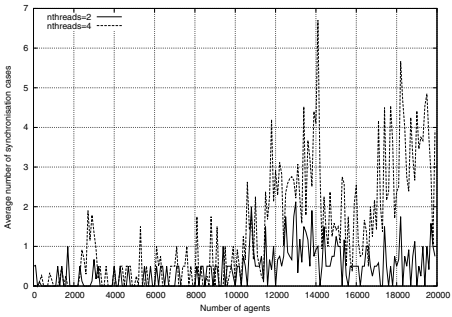


Fig. 6. Average number of synchronisation cases during Step 2 depending on the number of threads and agents

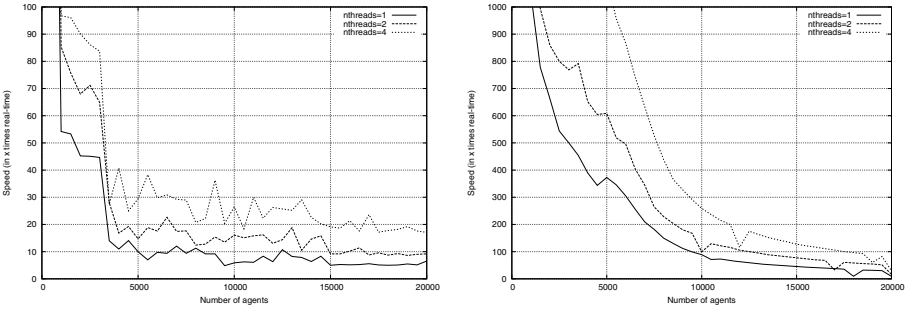


Fig. 7. Execution speed of the simulation in terms multiple to real-time for Stage 1 (7(a)) and Stage 2 (7(b))

incorporate vehicle component models that need updating which may be triggered during Step 2. This may change the proportions of Step 1 and 2 to the overall execution time significantly.

6 Conclusions and Future Work

An efficient execution model is crucial for large-scale agent-based traffic simulation of an entire city. An important feature of any agent-based simulation is the ability of agents to react to perceived changes in the environment. In the context of agent-based traffic simulation, for example, this refers to the ability of the agent to change its route. This feature is often not supported by many microscopic traffic simulations that use a more or less static form of traffic assignment. The AIMSUN2 engine, to which our approach is most closely related, does not consider explicit routing. In contrast, SEMSim Traffic requires agents to be able to change their routes. Our execution engine thus explicitly includes the case of routing in a two-step execution model.

Our results indicate that the simulation time is dominated by Step 1 (routing) for which a high degree of efficiency can be achieved. Future work will focus on improvements for the efficiency of Step 2 as well as the evaluation of our multi-threading execution engine on more cores. Furthermore, we will investigate possible hybrid approaches that combine the advantages of distributing the simulation on multiple multi-core shared memory nodes.

A by-agent execution model, in which all agents are updated concurrently by different threads, may be the more natural execution model for an agent-based simulation. However, it will also inevitably lead to more synchronisation cases. In contrast, the by-block execution model minimises the need for synchronisation and does not alter the agent-based model itself, i.e., agents are updated in exactly the same way in both execution models. The only difference is the actual sequence in which agents are being updated. This may affect the behaviour of the model. Future work will thus investigate the qualitative differences between the two execution models.

References

1. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Framework and Rules. IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000), pp. 1–38 (2010)
2. Barceló, J., Ferrer, J.L., Garcia, D.: Microscopic traffic simulation for ATT systems analysis: a parallel computing version. In: 25th Anniversary of CRT, pp. 1–16 (1998)
3. Ben-Akiva, M., Koutsopoulos, H.N., Antoniou, C., Balakrishna, R.: Traffic Simulation with DynaMIT. In: Barceló, J. (ed.) Fundamentals of Traffic Simulation. International Series in Operations Research & Management Science, vol. 145, pp. 363–398. Springer, New York (2010)
4. Ben-Akiva, M., Koutsopoulos, H.N., Toledo, T., Yang, Q., Choudhury, C.F., Antoniou, C., Balakrishna, R., Barceló, J.: Traffic Simulation with MITSIMLab. In: Barceló, J. (ed.) Fundamentals of Traffic Simulation. International Series in Operations Research & Management Science, vol. 145, pp. 233–268. Springer, New York (2010)
5. Cameron, G.D.B.: PARAMICS–Parallel Microscopic Simulation of Road Traffic. The Journal of Supercomputing 53, 25–53 (1996)
6. Fellendorf, M., Vortisch, P., Barceló, J.: Microscopic Traffic Flow Simulator VISSIM. In: Barceló, J. (ed.) Fundamentals of Traffic Simulation. International Series in Operations Research & Management Science, vol. 145, pp. 63–93. Springer, New York (2010)
7. Gipps, P.G.: A behavioural car-following model for computer simulation. Transportation Research Part B: Methodological 15(2), 105–111 (1981)
8. Klefstad, R., Zhang, Y.: A Distributed, Scalable, and Synchronized Framework for Large-Scale Microscopic Traffic Simulation. In: IEEE Conference on Intelligent Transportation Systems, pp. 813–818 (2005)
9. Nagel, K., Rickert, M.: Parallel implementation of the TRANSIMS. Parallel Computing 27, 1611–1639 (2001)
10. Osogami, T., Imamichi, T., Mizuta, H., Morimura, T., Raymond, R., Suzumura, T., Takahashi, R., Id, T.: Research Report IBM Mega Traffic Simulator. Technical report (2012)

11. Saraswat, V.A., Sarkar, V., von Praun, C.: X10: concurrent programming for modern architectures. In: Proceedings of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pp. 271–271. ACM (2007)
12. Suzumura, T., Kanezashi, H.: Highly Scalable X10-Based Agent Simulation Platform and Its Application to Large-Scale Traffic Simulation. In: 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, pp. 243–250. IEEE (October 2012)
13. Thulasidasan, S., Kasiviswanathan, S., Eidenbenz, S., Galli, E., Mniszewski, S., Romero, P.: Designing systems for large-scale, discrete-event simulations: Experiences with the FastTrans parallel microsimulator. In: 2009 International Conference on High Performance Computing (HiPC), pp. 428–437. IEEE (December 2009)
14. Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Physical Review E 62(2), 1805 (2000)

A Latency-Hiding Scheme for Adjacent Interaction Simulation on Multi-core/Many-Core Clusters

Chen Li-li, Li Wei, Zhang Jing, Shi Shuai, and Huang Jian-xin

Science and Technology on Complex System Simulation Laboratory, Beijing, China
chenlili8209@nudt.edu.cn

Abstract. As the processor has entered the multi-core/many-core era, the parallel processing capability of a single processor can be improved in scale with increasing cores. However, for the high performance computation (HPC) clusters, the improvement of inter-node communication latency is far behind of the performance improvement of processors. As a result, communication latency often becomes the performance bottleneck of most HPC applications. This paper focuses on solving the communication latency problem of adjacent inter-action simulation on multi-core/many-core clusters, and pro-poses an optimized algorithm for adjacent interaction simula-tion on modern general purpose graphic many-core architec-tures and an O(B+2R) algorithm for inter-node latency-hiding. The theoretical analysis and experimental result show that the techniques proposed in this paper can effectively improve the performance of adjacent interaction simulation on multi-core/many-core clusters.

Keywords: adjacent interaction simulation, communication latency-hiding, parallel processing, multi-core/many-core cluster.

1 Introduction

The computation model of adjacent interaction simulation is widely used in physical, biological, economical and many other areas' simulated computations. As the scales of the simulation entities are growing with the simulation scale requirements of these applications, the performance demands on the computation platforms which are used for these adjacent interaction simulations is growing too. As for the memory and performance limits, the sequential execution method can only be used for adjacent interaction simulations in small scales. Thus, parallel simulation is a natural choice to meet the large scale requirement of adjacent inter-action simulations. However, one major problem with parallel adjacent interaction simulations is that the computations of entity's state updates in each time step are very fast com-pared with the communication latencies between different processing nodes on different hierarchies. Therefore, the state update synchronization cost can become a severe performance bottleneck, when distributing the simulation entities to different processing nodes. As a result, the speedup achieved by parallel execution of adjacent interaction simulation may not scale well with increasing processing nodes, and the communication latency is one of the factors that limit the scalability and performance speedup of parallel adjacent interaction simulations on distributed computation environments [1], [2].

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 13–24, 2013.

The simulation entities in adjacent interaction simulations are usually organized in grid layout. A simulation entity in a grid cell interacts with the entities in adjacent cells, as showed in Fig. 1. For parallel execution, the entities on border cells need to read states from the entities on remote processing nodes and send their own states to remote nodes correspondingly. The reading/sending entities' states must be completed in one time step for the correctness of the computations. When the simulations are executed on a hierarchical heterogeneous cluster, communication and synchronization cost can seriously affect the speedup achieved by parallel execution, which prolongs the synchronization time in the conservative simulation algorithm, or increases roll-back ratio in the optimism algorithm. As a result, latency hiding mechanism must be designed to alleviate the communication cost on different hierarchies.

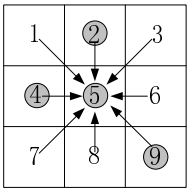


Fig. 1. An illustration for adjacet interaction simulation model

In recent years, the leading graphics processors have already reached the processing ability of TFLOPS, as they use many-core design for massive parallel processing compared with multi-core general purpose CPUs. Furthermore, using GPUs for accelerating general purpose parallel computations can be very easy with aid of GPU programming frameworks. As a result, equipping processing nodes with GPUs has become a trend in today's HPC clusters. However, on the other hand, as for the special processor architecture design of GPU, exerting the high performance of GPU is highly depending on the program's implantation.

This paper focuses on parallel adjacent interaction simulation, and proposes an optimized algorithm for adjacent interaction simulation on modern general purpose graphic many-core architectures and an O(B+2R) algorithm for inter-node latency-hiding. The theoretical analysis and experimental result show that the techniques proposed in this paper can effectively improve the performance of adjacent interaction simulation on multi-core/many-core clusters.

2 Related Work

The adjacent interaction simulation model is used in many areas' simulated computations. The most commonly used among them are fluid flow simulation, massive adjacent interacting agent simulation, stencil computation, and so on.

In the fluid flow simulations based on the Lattice Boltzmann Method [3], each cell is associated with several attributes. At discrete time steps, the attributes are updated to determine the dynamics of a flow field. Stam proposed a widely used "stable fluids" method for the computation of fluid flow speed in [4]. Harris proposed a shading language based 2D fluid flow implementation for Stam's method in [5].

Among the well known agent modeling and simulation systems are Repast [6], Swarm [7], JAMES [8], and so on. However, most of these agent simulation systems use sequential execution mechanism. Until recent years, parallel executed agent simulation systems have been developed, such as SEAS [9], and the agent simulation systems described in [10], [11].

Aaby's work [12] is the most related to this paper. In [12] a B+2R latency hiding algorithm was proposed for parallel agent simulations, which achieved remarkable performance speedups on both CPU and GPU clusters. But there are still three defects. First, on the CPU clusters, R should be very large to get the optimal speedup. Second, GPU storage capacity restrictions make the R value can't be too large, so it is difficult to reach the maximum speedup. Third, all processing nodes must synchronize with each other after every R time steps, so 1/R of the communication latencies can't be hidden.

The researches on stencil computation in recent years include: Automatic parallelization of stencil computation was reported in [13] in the context of a one dimensional Jacobi code on a 32 node (single core) platform; a detailed empirical study of stencil computation optimization on several multi-core CPU based architecture (but restricted to a single GPU) was reported in [14].

3 Methodology

3.1 Memory Access Latency-Hiding for GPU

1) A Primary GPU Implementation Based on CUDA

Current widely used general programming frameworks for GPUs are CUDA [15] and OpenCL [16]. As the two programming models are similar to each other, the GPU algorithms proposed in this paper are using CUDA frame-work for implementation.

Take the computation of two-dimension adjacent inter-action simulation for an example. A direct and primary im-plementation for adjacent interaction simulation on GPU based on CUDA programming framework is updating the value of a single cell by a single GPU thread. That is, each thread using thread indices (i.e. threadIdx.x and threadIdx.y) indexes a corresponding cell and updates its value in each time step. This way of implementation has two major short-comings: first, the on chip memory storage is not effectively utilized; second, if a thread only updates a single cell, a great deal of GPU threads need to be spawn, which will put heavy burden on the GPU threads scheduling. Different from CPU, current GPUs have no hardware cache mecha-nism for automated caching off chip memory accesses. In-stead, current GPUs use so called Shared Memory for re-ducing the cost of off chip memory access. The data in Shared Memory (SM) are manually moved from or to off chip memory, and are shared between threads that are in the same Streaming Processor (SP). Accessing data in SM is much faster than direct accessing from off chip memory. Effectively utilizing SM can greatly improve the perform-ance of GPU based algorithms. The algorithm in Fig. 2 presents a primary implementation for adjacent interaction simulation on GPU with the utilizing of SM. For the clarity of this algorithm, the computations of borderline cells are not shown.

Algorithm 1. A primary implementation for two-dimension adjacent interaction simulation on GPU

```
//TH_WIDTH: number of threads within a thread block
__shared__ ds_cell[TH_WIDTH][3];

//tx0 is used for indexing memory reading position•
//tx1 is used for indexing memory updating position
initialize tx0, tx1;
ds_cell[threadIdx.x][0]..[2] • d_cell[tx0],d_cell[tx0 +
SIZE],d_cell[tx0 + 2*SIZE];
__syncthreads();
//Precomputing the data in the first row
Update(d_cell[tx1], ds_cell[threadIdx.x][0]..[1]);
idx0 = 0; idx1 = 1; idx2 = 2;
for i = 1 to SIZE −1 do
  update tx0, tx1;
  __syncthreads();
//Preloading
ds_cell[threadIdx.x][idx0] = d_cell[tx0 + SIZE * 2];
Update(d_cell[tx1], ds_cell[threadIdx.x][0]..[2]);
idx0 ++; idx1 ++; idx2 ++;
idx0 = idx0 % 3; idx1 = idx1 % 3; idx2 = idx2 % 3;
endfor
```

Fig. 2. A primary algorithm for two-dimension adjacent interaction simulation on GPU

The central idea of algorithm in Fig. 2 is utilizing three rows of SM to cache the cell values which need updating computation. At the beginning of the algorithm, the threads within a thread block work together to load three rows of data to SM from off chip memory. As threads load data from contiguous memory addresses, the GPU memory access bandwidth can be fully utilized, which can greatly improve memory accessing efficiency. After loading data and before entering the main loop, the first row of data is pre-computed. In the main loop, each thread is responsible for updating one cell. When writing the updated value back to memory, the accessing addresses can be contiguous again. At the end of the main loop, the *idx0*, *idx1* and *idx2* are updated in a circular way. As a result, at the beginning of the next loop, the oldest row of SM can be used to cache the new data, whereas the other two rows of SM can be used for computation.

After simple analysis, it is not hard to see that a thread utilizes three SM cells on average in Algorithm 1, and accesses off chip memory two times (a loading and a saving) for each loop iteration. After all the cells have been updated in a time step, a global synchronization is needed. This global synchronization can regarded as communication synchronization between different SPs within a GPU. As the values of border cells must be updated by the threads on other SPs before a new time step, which can only be carried out by synchronization on CPU side under current CUDA programming framework.

2) An Optimized GPU Implementation Based on CUDA

In Algorithm 1, the number of utilized SM cells is three times the number of threads, which is far from reaching the maximum SM cells can be utilized. An important way of optimizing Algorithm 1 is fully utilizing the SM to minimize memory access latency. The Algorithm 2 in Fig. 3 presents an improved implementation which fully utilizes the SM. Again, for the clarity of this algorithm, the computations of borderline cells are not shown.

Algorithm 2. An optimized implementation for two-dimension adjacent interaction simulation on GPU

```
// TH_WIDTH: number of threads within a thread block
__shared__ ds_cell[TH_WIDTH][TH_WIDTH];
__shared__ ds_tmp[2][TH_WIDTH];

//Loading data from off chip memory to SM
for i = 0 to TH_WIDTH - 1 do
  Load ds_cell[i][threadIdx.x] from d_cell
__syncthreads();
for r = 0 to R - 1do
  for i = r + 2 to TH_WIDTH - (r + 1) -1 do
    update(ds_tmp[i % 2][threadIdx.x]);
     __syncthreads();
    ds_cell[i - 1][threadIdx.x] = ds_tmp[(i - 1) %
2][threadIdx.x];
  endfor
endfor
__syncthreads();
//Writing result back to off chip memory
for i = R to TH_WIDTH - R - 1 do
  Save ds_cell[i][threadIdx.x] to d_cell
```

Fig. 3. A primary algorithm for adjacent interaction simulation on GPU

The central idea of Algorithm 2 can be depicted by Fig. 4. To maximize the utilization of SM and reducing the number of global synchronization, for a certain SP, the whole sub-grid data which will be updated by this SP and the cell data on the R rows and R columns which are neighboring to this sub-grid are loaded into the SM of this SP. As a result, TH_WIDTH*TH_WIDTH units of SM will be occupied. In addition, TH_WIDTH*2 units of SM is claimed as swap space for cell data updating. At the time of first iteration, the cells within row 2^{nd} to row (TH_WIDTH-1)th and column 2^{nd} to column (TH_WIDTH-1)th are updated. As the border data are pre-loaded, the computation correctness of first iteration can be guaranteed. At the time of second iteration, the cells within row 3^{rd} to row (TH_WIDTH-1)th and column 3^{rd} to column (TH_WIDTH-1)th are updated. As the border data of this iteration are the results of the first iteration, the computation correctness can also be guaranteed. The outmost loop will do R-1 iterations in total. During these R-1 iterations, the computation correctness of block $B_{1,1}$ can be guaranteed.

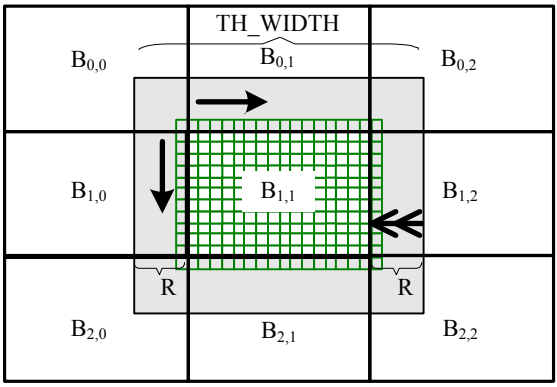


Fig. 4. An illustration for Algorithm 2

The central idea of Algorithm 2 is using extra computations to compensate memory access latencies. It is not hard to see that, in Algorithm 2, only two memory accesses (a loading and a saving) are needed on every R updates. Compared with Algorithm 1, (R-1)*(TH_WIDTH-R) *(TH_WIDTH-R) off chip memory accesses can be saved for each kernel function call. Thus, Algorithm 2 is more efficient than Algorithm 1.

3.2 O(B+R) Communication Latency Hiding Algorithm for Multi-node Platform

Here, we proposed an optimized B+2R algorithm, named O(B+2R) for short, to overcome some of the defects of B+2R algorithm, which discussed in the above related work.

1) The Principle of O(B+2R) Algorithm

The structure of our O(B+2R) scheme is similar to that of B+2R scheme, in which sub-grids are padded on the sides by R layers of surrounding data. In order to make it more clearly to illustrate the principle of the O(B+2R) algorithm, we simplify the model here, which only consider one-dimensional. We cut out the horizontal direction part of the neighborhood interaction model as shown in Fig. 5. The agent in 5^{th} cell will read the states of the agents in the 4^{th} cell and 6^{th} cell to do the next state computation on each time step. Fig. 6 shows the extracted part, which is framed by bold lines, from the (B+R)×(B+R) block for illustration.

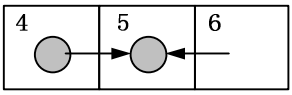


Fig. 5. Simplified agent interaction model

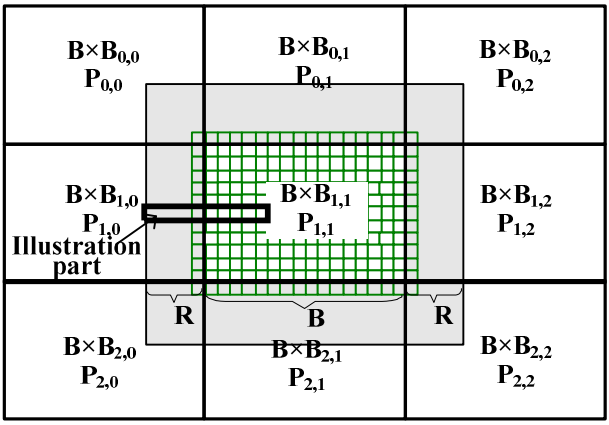


Fig. 6. Extracted part for illustration

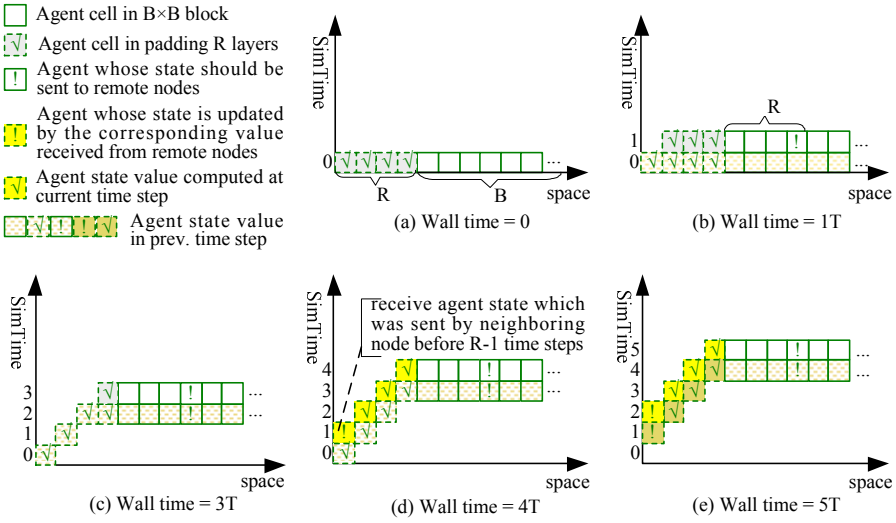


Fig. 7. Promoting process of O(B+2R) algorithm

Fig. 7 illustrates the communication and computation at successive (up to R+1) simulation time steps. Vertical axis represents the simulation time [2], i.e. logical time step. Each sub-figure shows system status at different wall time in which T is a symbol representing physical time overhead for each time step. At simulation initialization, the external R layers load the status of remote agents to be simulated by neighboring blocks assigned to other independent processing elements as their initial state values, as shown in Fig. 7(b)-(c). In the first R-1 time steps, the external R layers

compute their next states based on these initial values. At each time step, the inner R-th agent, which is marked by ① in Fig. 7, is sent to the neighboring remote parallel node which contains its adjacent block. If network transmission time is equal to the computation time overhead of R-1 time steps, the remote node will receive the value of the agent state after the simulation advancing R-1 time steps. Similarly, at R-th time step, local node receives the agent state value sent by its neighboring node before R-1 time steps, as shown in Fig. 7(d). Local node then updates the outermost agent state to this received value, which is marked by ① in Fig. 7(d). Based on this updated state and the history states, the remaining outer R-1 layers can compute their next states correctly, which marked by ② in Fig. 7(d). As a result, at the next time step, the inner B x B block can read correct adjacent agent state values. If a proper R value is selected, the simulation system can advance correctly with no network latency waiting overhead, as shown in Fig. 7(e).

2) Analytical Model

Assuming that the total number of simulation time steps is much larger than 1, so here time overhead during initialization are ignored, i.e., only consider the performance analysis in the main loops.

Let F_{c1} be the computation time for Line 1.5.1 in Algorithm 1 at one iteration (i.e., one simulation time step) and F_{c2} be the computation time for Line 1.5.4 and Line 1.5.4.1. Note that $f_c(\mathrm{X})$ is the computation cost function for agents in a given region X. F_{c1} and F_{c2} is calculated as follows:

$$F_{c1} \equiv f_c(\mathrm{B})$$

$$F_{c2} \equiv f_c(<t_{li}\text{-}1, t_{lj}\text{-}1, b_{ri}\text{+}1, b_{rj}\text{+}1, 1>) + f_c(<t_{li}\text{-}2, t_{lj}\text{-}2, b_{ri}\text{+}2, b_{rj}\text{+}2, 1>) + \cdots$$
$$+ f_c(<t_{li}\text{-}(R\text{-}1), t_{lj}\text{-}(R\text{-}1), b_{ri}\text{+}(R\text{-}1), b_{rj}\text{+}(R\text{-}1), 1>)$$

If a is an implementation-specific computation constant and N is the degree of parallelism in the computation for B×B block, then

$$F_{c1} \equiv \frac{aB^2}{N}$$

$$F_{c2} \equiv a\sum_{i=1}^{R}[4(B+i)-4] = 4a(B-\frac{1}{2})R - 4aB + 4a$$

If the constant C_1 is the local overhead (not including network delay) for Send(<t_{li}+R-1, t_{lj}+R-1, b_{ri}-R+1, b_{rj}-R+1, 1>) operation, then the total time overhead for Block 1 in Algorithm 1 is calculated by $C_1 + F_{c1}$. If the constant w and C_2 are the waiting overhead and copy overhead respectively for receive operation in Line 1.5.3, then the total time overhead for Block 2 in Algorithm 1 is calculated by $w + C_2 + F_{c2}$.

Let F be the total run time for one simulation time step. If code Block 1 and Block 2 in Algorithm 1 run in parallel, then

$$F = \max\{w + C_2 + F_{c2}, C_1 + F_{c1}\}$$

$$= \max\{w + C_2 + 4a(B - \frac{1}{2})R - 4aB + 4a, C_1 + \frac{aB^2}{N}\}$$

The waiting time of *Receive* function, i.e., the value of w, depends on how much the network transmission delay is hidden by the local computation for R-1 time steps. Their relationship can be described by $w = f_{nw} - (R-1) \times F_{c1}$. In order to make w equal zero, we make derivation from $w = f_{nw} - (R-1) \times F_{c1} = 0$ and get $R = \frac{f_{nw}}{F_{c1}} + 1 = \frac{f_{nw} N}{aB^2} + 1$. So we can obtain the following conclusion from this equation: the optimal value of R depends on network delay, and the finer the granularity of the agent model is, the greater the R value should be taken.

3.3 B+2(R×r) Communication Latency Hiding Algorithm for CPU+GPU Cluster

Because the memory access and execution of CPU and GPU are in different ways, the communication latency hiding schemas are also different. Due to data transmission between the GPU memory and main memory require relatively large time overhead. So, if GPU platform adopts the O(B+2R) algorithm which is introduce in the previous section for CPU cluster, it would have to do data transmission with the outside main memory at each time step, which will take large time overhead. Here we introduce an r to make GPU communicate with main memory once in r time step. The corresponding communication latency hiding algorithm is called B+2(R×r) algorithm.

Fig. 8 shows B+2(R×r) algorithm at the 5-th time step and the 7-th time step for the R value with 3 and r value with 2. It can be observed that initial values of the external R×r layer simulation entities are read at initialization, which is different from O(B+2R) algorithm. What's more, communication between local and remote node is done at each time step in O(B+2R) algorithm, but in B+2 (R×r) algorithm it is done every r time steps. At each communication, state-values of the simulation entities from the inner $(R \times r\text{-}r)^{th}$ layer to the $(R \times r)^{th}$ layer are sent. The received remote emulation entities' state are used to update the states of the external r layer simulation entities. Then based on these received states and the historical states of the simulation entities in outer layers, the states of the r layers simulation entities adjacent to the internal B×B block can be computed correctly. With these r layers simulation entities' states, simulation on GPU can continue to move forward r time steps correctly.

For B+2(R×r) algorithm, due to the introduced r, theoretical analysis shows that when R equals $\frac{\frac{f_{nw}}{aB^2}}{r\frac{aB^2}{N} + L} + 1$, F can achieve the optimal value, where L is the time overhead of data exchange between GPU's global memory with main memory.

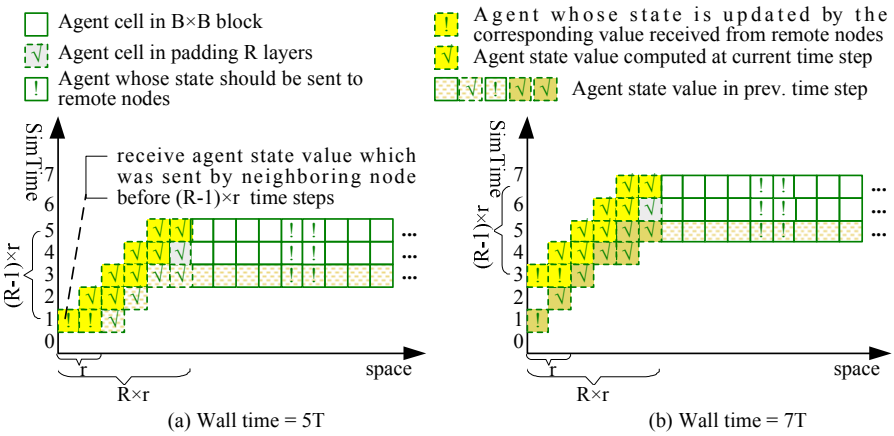


Fig. 8. Promoting process of B+2(R×r) algorithm (R=3, r=2)

4 Experiment and Evaluation

Our experiments use POSIX threads for thread-level parallelism and MPI for inter-node communication. Operating system is Kylin Server 3.1 with GCC version 4.1.2. We choose CUDA for GPU computing. Test cases are the classical Game of Life (GOL) model and the leadership relationship (LDR) model. The GOL model is more fine-grain than the LDR model.

Fig. 9 shows the experimental results of 160,000,000 scale GOL model running on GPU cluster with 16 nodes. We can observe that proper values assignment to R and r can reach a hundred speedup. Note that, the speedup here is a relative speedup to the execution without any latency-hiding scheme, rather than parallel speedup relative to serial execution.

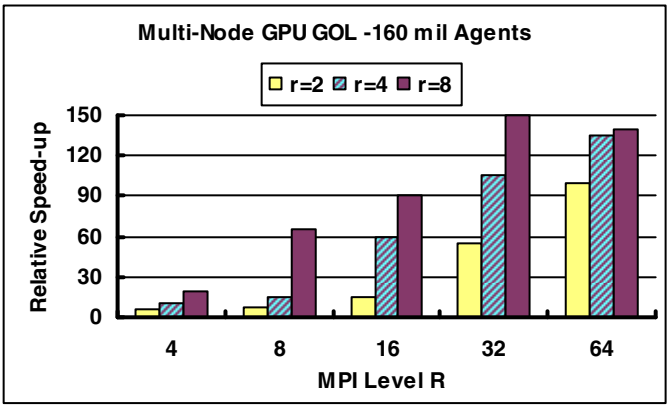


Fig. 9. Relative speedup of GOL on GPU cluster using latency-hiding scheme

Parallel speedup of GOL on GPU cluster compared with sequential execution is given in Fig. 10. Comparing Fig. 10 with Fig. 9, we can find that the parallel speedup relative to serial execution is less than the speedup relative to the parallel execution without any latency-hiding scheme. This is due to that without the communication latency hiding techniques, the performance of parallel execution is even worse than the serial execution for some fine-grain models. We can draw the following conclusions: our communication latency hiding technology makes achieving parallel speedup become possible for some fine-grain adjacent interaction application. What's more, in general, parallel speedup will not exceed 16 on 16-nodes cluster, but Fig. 10 shows the speedup can reach 30 under certain configures of R and r. This is because that the B+2(R× r) communication latency hiding techniques improve the performance of sub-blocks' serial execution on single-node GPU.

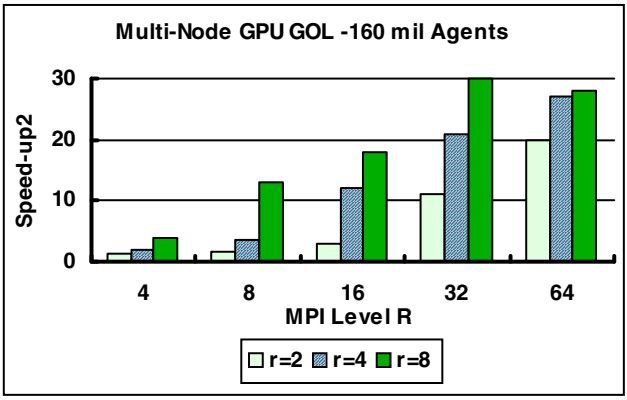


Fig. 10. Parallel speedup of GOL on GPU cluster compared with sequential execution

5 Conclusions

For large scale adjacent interactive simulation application with fine-grained model on multi-core/many-core cluster, communication optimization techniques still can not achieve sufficient communication latency hiding. To overcome some of the defects of B+2R algorithm, we proposed O(B+2R) algorithm to improve parallel performance. The principle of this algorithm is "calculation trade communication". An analytical model for this algorithm is presented to tell how to select R value to reach the best speedup. Compared to B+2R algorithm, theoretical analysis and experimental result show that the algorithm proposed in this paper has the following advantages when running on cluster: 1) it can hide more communication latency; 2) a smaller R can satisfy the best communication latency hiding.

Acknowledgement. This research was supported under the National Science Foundation of China (Grant No.61203140).

References

[1] Tu, B.B., Zou, M., Zhan, J.F., Zhao, X.F., Fan, J.: Research on Parallel Computation Model with Memory Hierarchy on Multi-Core Clusters. Chinese Journal of Computers 11 (2008) (in Chinese with English abstract)
[2] Tang, Y.: Research on the Communication Problem of Large Scale Scientific Computing on High Performance Cluster Environment (Ph.D. Thesis). Chinese Academy of Sciences, Beijing (2004) (in Chinese with English abstract)
[3] Chen, S., Doolean, G.D.: Lattice Boltzman Method for Fluid Flows. Annual Review of Fluid Mechanics 30, 329–364 (1998)
[4] Stam, J.: Stable Fluids. In: Proceedings of SIGGRAPH 1999, pp. 121–128 (1999)
[5] Harris, M.J.: Fast Fluid Dynamics Simulation on the GPU. In: Fernando, R. (ed.) GPU Gems, pp. 637–665. Addison-Wesley (2004)
[6] North, M.J., et al.: Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. ACM Transactions on Modeling and Computer Simulation 16, 1–25 (2006)
[7] Walter, B., et al.: UAV Swarm Control: Calculating Digital Phermone Fields with the GPU. In: Interservice/Industry Training, Simulation and Education Conference (IITSEC), Orlando, FL (2005)
[8] Uhrmacher, A.M., Gugler, K.: Distributed, parallel simulation of multiple, deliberative agents. In: Proceedings of the Fourteenth Workshop on Parallel and Distributed Simulation, Bologna, Italy (2000)
[9] Chaturvedi, A., et al.: Bridging Kinetic and Non-kinetic Interactions over Time and Space Continua. In: Interservice/Industry Training, Simulation and Education Conference, Orlando, FL, USA (2005)
[10] Parker, J.: A Flexible, Large-scale, Distributed Agent-based Epidemic Model. In: Winter Simulation Conference, Piscataway, NJ (2007)
[11] Armstrong, R.C., et al.: Parallel Computing in Enterprise Modeling. Sandia National Laboratory. Techincal Report SAND2008-6172, 2008/08/01 (2008)
[12] Aaby, B.G., Perumalla, K.S., Seal, S.K.: Efficient Simulation of Agent-Based Models on Multi-GPU and Multi-Core Clusters. In: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, vol. (29), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels (2010)
[13] Krishnamoorthy, S., et al.: Effective Automatic Parallelization of Stencil Computations. In: Programming Languages Design and Implementation (PLDI), San Diego, California, USA (2007)
[14] Datta, K., et al.: Stencil Computation Optimization and Auto-tuning on State-of-the-Art Multicore Architectures. In: Supercomputing, Austin, Texas (2008)
[15] NVIDIA Corporation. NVIDIA CUDA SDK code samples, http://developer.download.nvidia.com/compute/cuda/sdk/website/samples.html
[16] Khronos. Opencl - the open standard for parallel programming of heterogeneous systems, `http://www.khronos.org/opencl/`
[17] Fujimoto, R.M.: Parallel and Distributed Simulation Systems. John Wiley&Sons, Inc. (2000)

To Execute the C4ISR Architecture Based on DoDAF and Simulink

Xiaokai Xia, Kaipeng Zhao, Luo Xu, and Chao Liu

School of Computer Science and Engineering, Beihang University, Beijing, China
{xiaxiaokai,zhaokaipeng,xuluo}@sei.buaa.edu.cn,
liuchao@buaa.edu.cn

Abstract. In order to verify and evaluate the C4ISR systems before they are built, this paper proposes an approach to make the architecture developed by DoDAF executable. The model transformation technologies in model driven architecture are used to transform the architecture products, such as composite structure diagram, state machine diagram, activity diagram and sequence diagram, to single Simulink models and comprehensive Simulink models. The proposed approach can also effectively reuse the existing simulation blocks in the Simulink library to strengthen the ability of the generated Simulink model from architecture. Through the execution of the generated Simulink models, the data from the simulation can be used for the verificaiton and evaluation of the CRISR architecture. The case study shows the feasibility of the proposed approach.

Keywords: Executable C4ISR architecture, DoDAF, Simulink, Model transformation.

1 Introduction

Architecture, which is defined as the fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution [1], plays an important role in the development cycle of the complex interactive systems, such as C4ISR systems. Department of Defense Architecture Framework (DoDAF) and other architecture frameworks have been mandated as the guide for the development of the C4ISR systems' architecture in many countries. However, one problem existing in the current frameworks is that they prescribe only static representations of the architecture description. These static representations include Unified Modeling Language (UML), ICAM Definition Languages (IDEF), text, and table, and so on. As a result, they cannot effectively support the simulation and validation of the architecture in the design stage.

Some efforts have been done to make the architecture developed by DoDAF executable through transforming the DoDAF products/models to some executable models, such as Petri Nets, Markov Chain, and System Dynamics. Reference [2] proposes the method of transforming activity diagram of UML to Colored Petri Net (CPN) based on the semantic information extracted. Reference [3] proposes the method of

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 25–36, 2013.

transforming sequence model of SysML to CPN. Reference [4] presents an approach called 4-dimension modeling approach to support the derivation of the HCPN (Hierarchical CPN) executable model from a set of DoDAF operational products, which helps to support the validation and evaluation of the integrated system architecture's dynamic behavior. Reference [5] proposes the issues and concept of automatic transformation from activity diagram to CPN model supported by CPN Tools. Reference [6] presents the methodology of transforming OV-2, OV-5, OV-6b and OV-6c (using UML class diagram, activity diagram, state diagram and sequence diagram separately) in DoDAF to an extended CPN separately. Reference [7] describes how to use xUML to describe architecture in model driven architecture (MDA). Reference [8] gives the approach of transforming activity diagram with swim lane to ExtendSim for validating some measures of performance. Reference [9] summarizes the use of DoDAF to create four types of common executable models including Markov Chain, Petri Nets, System Dynamics and Mathematical graphs. References [10-12] extend DoDAF with DEVS to allow the DoDAF having the ability to generate DEVS models supported by simulation tools of DEVS. References [13, 14] also propose how to build the executable architecture enviroment during the conceptual analysis and design phases.

We found that many researches mentioned above about how to make the DoDAF executable have the following characteristics: 1) focused on validating the correctness of the logic and behavior for certain models in the architecture; 2) transformed models or products in architecture to executable models separately (e.g., activity or state diagram to Petri Net), which constrained the ability of the generated executable models to evaluate more user concerns such as measures of performance (MoP) and measures of effectiveness (MoE), which indicates the integral propriety of the C4ISR Systems; 3) established the executable models manually, which would easily lead to the extra workload and inconsistency; 4) could not combine the legacy simulation components effectively to reduce the effort and improve the accuracy of simulation. And the reasons are that the chosen executable models used in the above research always have rigid and formal mathematical representations, such as Petri Nets, which are fit for verifying the logic and behavior cf one single model, but not fit for simulating complex situations.

As one widespread used modeling and simulation tool, Simulink has been concerned in the design stages of some kind of systems. Reference [15] summarizes how UML and Matlab/Simulink can be associated and what is the impact of SysML. Reference [16] couples a UML based CASE tool (Rose RealTime) and Simulink to allow simultaneous simulation. Reference [17] proposes to map composite structure diagram and activity diagrams of UML to Simulink for the verification of real time embedded systems in the design stage. References [18, 19] also propose the combination of UML/SysML and Simulink for verification of embedded control systems. And these works are the basis of our research.

Our previous work [20] has proposed the idea of using Simulink to make the DoDAF architecture executable, but the details are not introduced. In this paper, we propose the approach of executing the DoDAF architecture of C4ISR systems based on Simulink. The MoPs and MoEs of C4ISR can be evaluated through the proposed approach, but that does not mean the logic and behavior of the DoDAF models cannot be verified through the approach.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction of DoDAF and Simulink. Section 3 introduces the framework of our approach and details the transformation of the DoDAF products to Simulink models. Section 4 uses a case study of C4ISR system to validate the feasiblity of the proposed approach. And section 5 concludes the paper.

2 DoDAF and Simulink

2.1 DoDAF

All departments of United States of America that participating in the development of defense systems have been obliged to use DoDAF as the framework for architecture description. Because DoDAF can guarantee that all architecture descriptions developed by different departments of defense could correspond with each other and they should be integrated easily. In this paper, we use DoDAF 1.5, which is released in 2007, as the guide for describing architecture. Operational View (OV), System View (SV), Technical View (TV) and All View (AV) are included in DoDAF 1.5. The Operational View describes tasks, activities, participating nodes, and associated information exchanges required to perform a mission. The System View describes the systems of concern and their connections in the context of the OV. The Technical View describes a profile of the minimum standards and rules that govern the implementation, arrangement, interaction and interdependence of the systems described in the SV. The All View describes the overarching aspects of an architecture than relate to all three other views. Each view also includes some products, and most of these products can be described using UML or its profiles like SysML. The products that related with the implementation of executable models of DoDAF in our approach and their UML presentations are listed as follows:

- Operational Node Connectivity Description (OV-2): describes operational nodes, operational activities performed at each node, connectivity and information exchange needlines between nodes. Composite structure diagram can be used;
- Operational Activity Model (OV-5): describes capabilities, operational activities, relationships among activities, inputs, and outputs; overlays can show cost, performing nodes, or other pertinent information. Activity diagram can be used;
- Operational State Transition Description (OV-6b): identifies business process responses to events. State machine diagram can be used;
- Operational Event Trace Description (OV-6c): traces actions in a scenario or sequence of events. Sequence diagram can be used;
- Systems Interface Description (SV-1): identifies systems nodes, systems, system items, services, and service items and their interconnections, within and between nodes. Composite structure diagram can be used;
- Systems Communications Description (SV-2): describes systems nodes, systems, system items, services, and service items and their related communications laydowns. Composite structure diagram can be used;
- Systems State Transition Description (SV-10b): identifies responses of a system/service to events. State machine diagram can be used;

- Systems Event-Trace Description (SV-10c): identifies system-specific refinements of critical sequences of events described in the OV. Sequence diagram can be used;

The DoDAF 2.0, which is proposed in 2009, is a slightly different from DoDAF 1.5 in the organization of views and data. But the views of DoDAF 1.5 used in our approach all have the corresponding views (called viewpoints) in the DoDAF 2.0. And the approach proposed in this paper can be easily adapted to DoDAF 2.0.

2.2 Simulink

Simulink is a block diagram environment for multi-domain simulation based on Matlab. It provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. The reasons that we choose Simulink to make the DoDAF executable are that the Simulink has a strong modeling and simulation ability for various complex systems, and many legacy systems or components of C4ISR systems were built based on it, so we can effectively reuse them for the better simulation and verification of the C4ISR systems.

Stateflow is one of the predefined blocks contained in Simulink that we will use in our approach. It is an interactive design and simulation tool for event-driven systems. Stateflow provides the language elements required to describe complex logic in a natural, readable, and understandable form. It is tightly integrated with MATLAB and Simulink, providing an efficient environment for designing systems that contain control, supervisory, and mode logic.

3 Implementation of the Executable DoDAF Based on Simulink

3.1 Framework

In order to make C4ISR architecture executable, and verify and evaluate the C4ISR design based on the execution results, we propose to transform the architecture products of C4ISR Systems to Simulink model. The framework of our appraoch is shown in Fig. 1.

In our approach, we use ATLAS Transformation Languages (ATL) [21] for the transformation. ATL are a model transformation language and toolkit proposed by ATLAS research group based on the QVT (Query/View/Transformation). ATL implements the transformation at the meta-model level, which could easily ensure the consistency between the source model and target model, and improve the standardization of transformation. ATL rules describe the relationships of how the source model can be mapped to the target model based on their meta-models. The ATL transformation engine takes ATL rules, source model and its meta-model and the meta-model of target model as inputs, and takes the target model as output. In Fig.1, the Intermediate Architecture Model and Intermediate Simulink Model are the source model and target model respectively (Their meta-models can be seen in our previous work [20]). These two models are not the initial DoDAF products/models and the final Simulink models, because the initial DoDAF products/models and Simulink models do not conform to the standard used in the ATL. So the Input Adaptor and Output Adaptor are developed to do the transformation. In MDA, the initial DoDA products/models developed

by modeling tools and Simulink models running in Simulink are platform-specific models (PSM), while the Intermediate Architecture Model and Intermediate Simulink Model in Fig.1 are platform-independent models (PIM). One advantage of using input adaptor is that when the architecture modeling tool is changed, only the input adaptor needs to be revised.

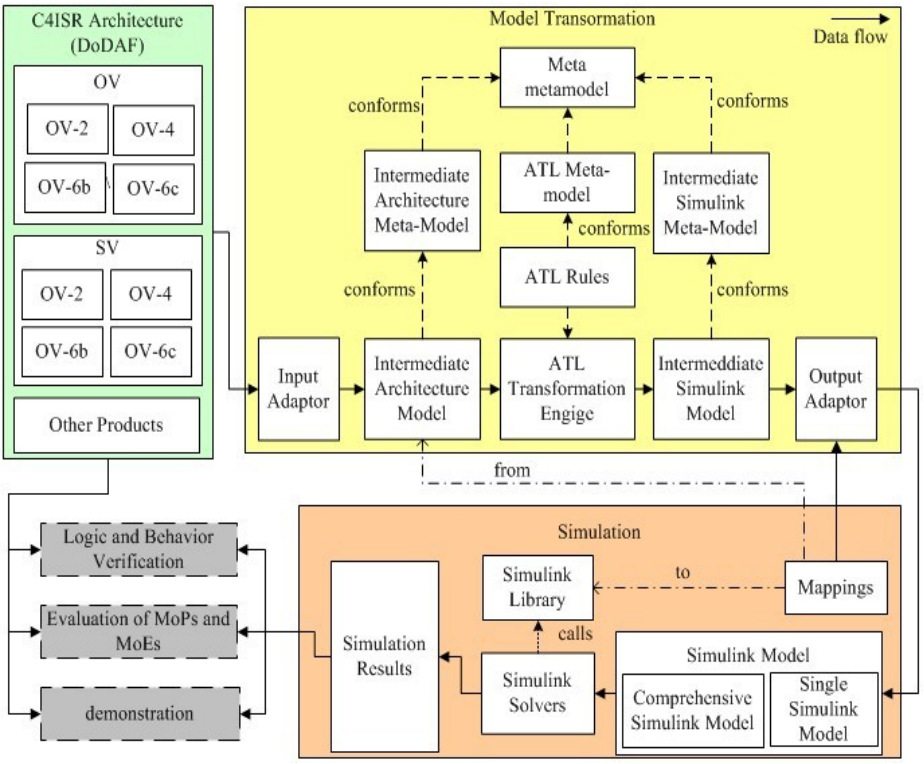


Fig. 1. Framework of our approach

In the following section, we will see that some generated Simulink blocks do not have concrete logic that can be executed directly. In this situation, we need to develop the concrete logic or reuse the legacy blocks that already implemented the logic in the legacy systems or Simulink library. The Mappings in Fig.1 list which components in the architecture will be mapped to already existing Simulink blocks in the Simulink library. And the output adaptor will substitute the corresponding generated blocks using the existing Simulink blocks in the Simulink library when the final Simulink model are generating. Through reusing the existing Simulink blocks in the Simulink library, the generated Simulink model can be executed directly, and the simulation ability will be greatly improved, which can not only support the verification of the logic and behavior of the products/models, but can support the evaluation of the MoPs and MoEs of the C4ISR systems.

After the Simulink model is generated, some parameters need to be configured, such as the simulation time, the inputs for driving the simulation. Then we can execute the Simulink model under the support of Simulink solvers. Two types of Simulink models can be generated: Single Simulink Model and Comprehensive Simulink Model. Usually Single Simulink Model is transformed from one single DoDAF product, while Comprehensive Simulink Model is transformed from many types of DoDAF products. Various data can be gathered from the running of the simulation, which can be used for the verification of logic and behavior of the architecture models, used for the evaluation of MoPs and MoEs, and used for the demonstration of some measures.

The core part of our approach is how to transform DoDAF products to Simulink models. In the following section, we will introduce the transformation between them, including the transformation from single DoDAF product to Single Simulink Model and the transformation from types of DoDAF products to one comprehensive Simulink Model.

3.2 Transformation of DoDAF Products to Simulink Models

Transform State Machine Diagram to Matlab/Stateflow. State machine diagram can be used to describe OV-6b and SV-10b in DoDAF. As introduced above, Stateflow is an important block existing in Simulink library for processing complex logic. State machine diagram and Stateflow are all based on finite state machine and they have similar meta-models, so the transformation between them is apparent. The state machine diagram needs to be transformed to a Stateflow block in the Simulink model diagram. Each state in the state machine diagram needs to be transformed to a state in the Stateflow model diagram. Each transition between two states in the state machine diagram need to be transformed a transition between the corresponding generated states in the Stateflow. The actions of a state and the conditions of a transition can also be transformed to corresponding elements in Stateflow as shown in Fig. 2. After the Stateflow model is generated, and the inputs of the stateflow are configured, the Stateflow model can be executed. So the logic of the state machine diagram can be verified through the execution of the generated Stateflow model.

Transform Activity Diagram to Simulink Model. Activity diagram is used to describe OV-5 in DoDAF. Each activity in the activity diagram can have a subactivity model, while the Simulink also support the hierarchical modeling through the block SubSystem, so the transformation from activity diagram to Simulink model is hierarchical as shown in Fig.3. Each activity is transformed to a SubSystem in the Simulink model. Each input/output port of an activity is transformed to the corresponding SubSystem's input/output port. The control flow between two activities is transformed to the corresponding communication channel in Simulink model.

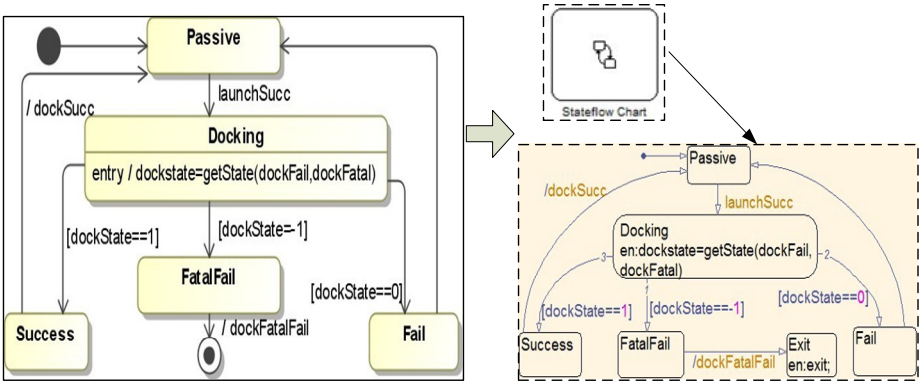


Fig. 2. Mapping of state machine diagram to Simulink/Stateflow

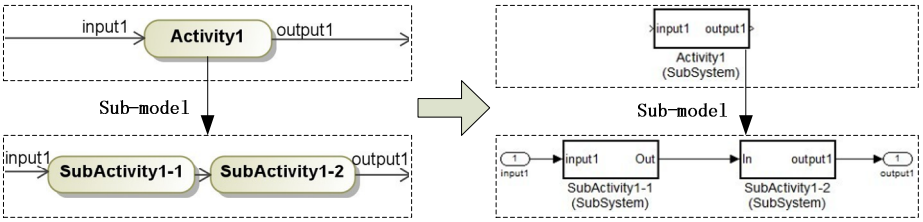


Fig. 3. Mapping of activity diagram to Simulink model

Transform Composite Structure Diagram to Simulink Model. Composite structure diagram can be used to describe OV-2, SV-1 and SV-2 in DoDAF. Class is transformed to a SubSystem. The ports are transformed to the corresponding input/output ports of the SubSystem. The Connectors are transformed to the corresponding communication channels. An example of transformation from composite structure diagram to Simulink model is shown in Fig.4 (The abbreviations in Fig.4 are explained in the case study). The transformation of composite structure diagram can also be hierarchical as the activity model shown above.

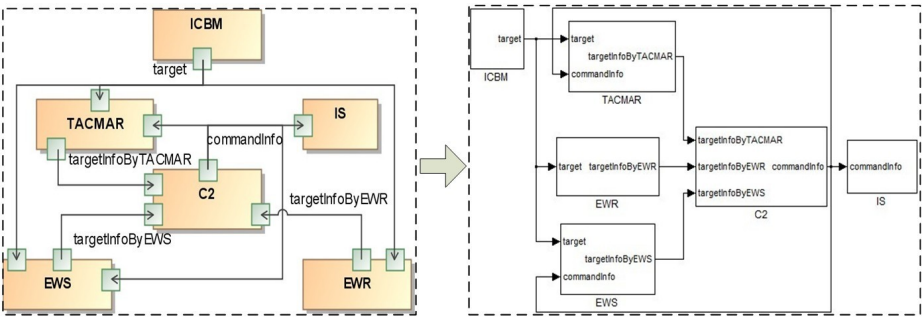


Fig. 4. Mapping of structure diagram to Simulink model

Transform Sequence Diagram to Simulink Model. Sequence diagram can be used to describe OV-6c and SV-10c in DoDAF. As shown in Fig.5, each object (called operational/system node in DoDAf) is transformed to a SubSystem, and the message between two objects need to be transformed to the corresponding communication channel between the generated SubSystems.

Transform DoDAF Products to Comprehensive Simulink Model. In the above four transformations, only the generated Stateflow from state machine diagram can be executed directly. While the other generated Simulink models from activity diagram, composite diagram and sequence diagram can not be executed directly, the reason is that there are no the concrete logic or behavior in the generated atomic SubSystems of the models. In order to make them executable, we need to combine the above four transformations and exisiting Simulink library together. For example, after the hierarchical transformation from the activity diagram to the Simulink model are completed. We need to transform the state machine diagram that describes the logic of the atomic activity to a Stateflow model, and substitute the atomic activity's correponding SubSystem with the generated Stateflow model. Then the generated Simulink model from activity diagram has the complete logic and behavior, and can be executed by invoking the Simulink solvers. Besides, the atomic activity's corresponding SubSystem can be substituted with an existing block in the Simulink library as shown in Fig. 1, in which way, the legacy simulation blocks can be reused. The generated Simulink models from composite structure diagram and sequence diagram can also be made executable using the same way like activity model.

After the comprehensive Simulink model is generated, the results from the simulation can be used for verification, evaluation and demonstration.

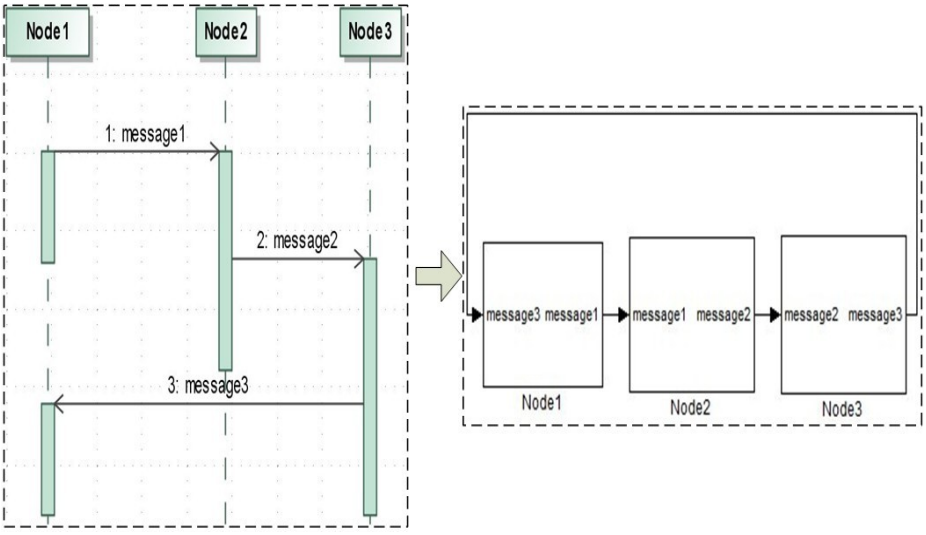


Fig. 5. Mapping of sequene diagram to Simulink model

4 Case Study

The proposed approach has been applied to evaluate the architecture of an anti-missile system in our project. The composite structure diagram shown in Fig.4 is the OV-2 product of the anti-missile system under evaluation. The system includes six related operational nodes, which are early warning satellite system (EWS), early warning radar (EWR), tactical multifunction array radar (TACMAR), command and control center (C2), intercept system (IS), and incoming ballistic missile (ICBM). Among them, EWS, EWR and TACMAR form the integrated warning system (WS), while ICBM is the external operational node of the anti-missile system.

Fig. 6 shows the overall operational activity model (OV-5) of the system. It includes four top-level activities: Warning and Detecting, Analyzing Situation, Commanding and Controlling and Operating. The flows between the activities represent how the operational information moves in the system.

The activity model in Fig.6 is hierarchical, and each top-level activity can be described in a detailed activity model. The atomic activities of the hierarchical activity model have their corresponding state machine diagrams that describe their logic. The state machine diagram shown in Fig. 2 describes the logic of one of the atomic activities.

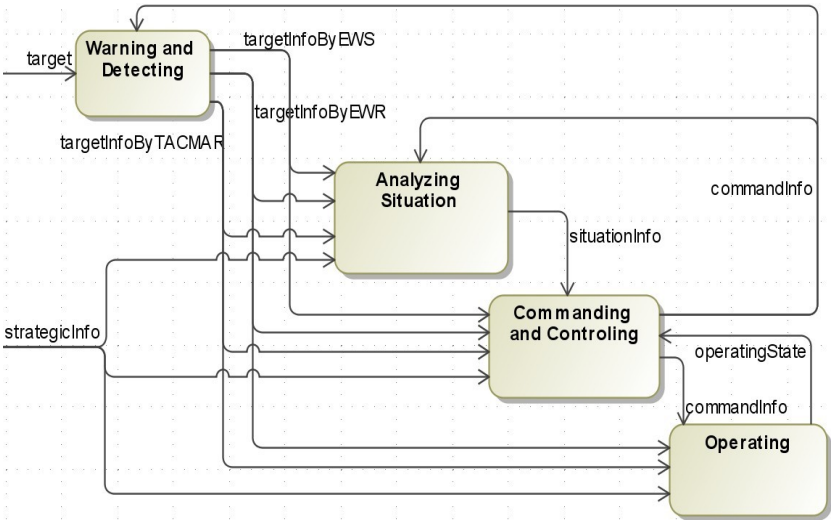


Fig. 6. Overall operatioanl acitivty model of the anti-missile system

Based on the proposed transformation method shown in Fig.2 and the developed supporting tool, a comprehensive Simulink model is generated semi-automatically as shown in Fig.7. The manual work we did was that we defined the mapping file, in which the mappings of some atomic activity to the exisitng simulation blocks in the Simulink library are listed, such as that we used an exisiting missile in Simulink library as the input of the anti-missile system. While the other activity diagrams and state machine diagrams that describing the atomic activities are transformed automatically.

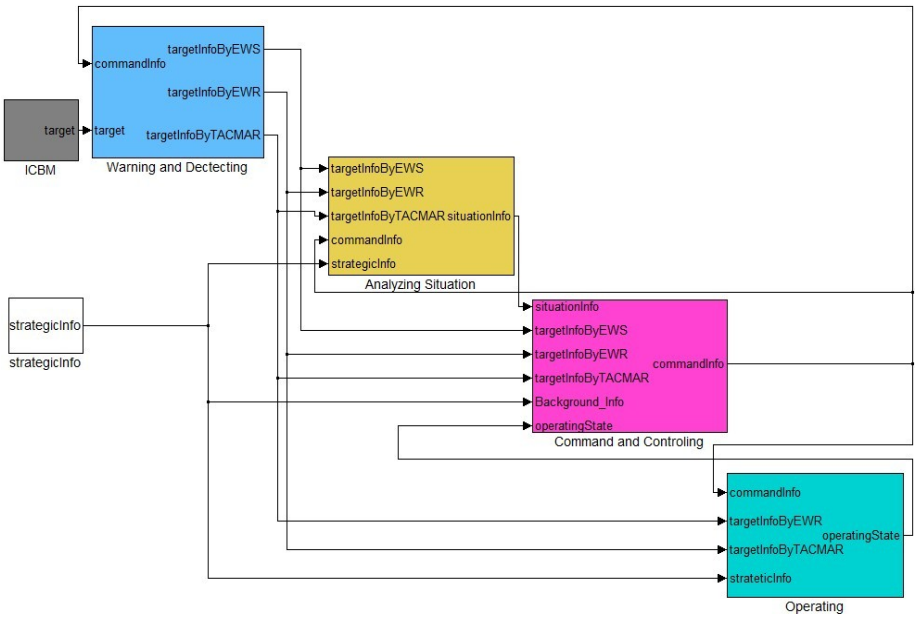


Fig. 7. Generated Simulink model of the anti-missile system

By running the generated Simulink model, the measures, such as the warning time, decision time and sucesess ratio of the system, could be gathered or calculated based on other gathered data. Then we could evaluate whether the design of the anti-missile system conforms to the requirements by comparing the measures with the design requirement.

5 Conclusion

This paper has proposed the approach of making the DoDAF architecture executable. The framework of the approach was firstly proposed. The key part of the approach is to transform DoDAF products to Simulink models automatically. The transformation technology ATL is used in our approach for the automatic transformation. Each composite structure diagram, state machine diagram, activity diagram and sequence diagram in DoDAF can be transformed to single Simulink model. And these DoDAF products can also be transformed to comprehensive Simulink models, which can simulate more characteristics of the C4ISR Systems. Besides, the approach allows reusing the existing simulation blocks of the legacy C4ISR systems stored in the Simulink library, which will make the generated Simulink model more powerful and valuable for the verification and evaluation of C4ISR architecture.

References

1. DoD Architecture Framework Working Group: DoD architecture framework version 1.5, Volume I: Introduction. U.S. Department of Defense (2007)
2. Staines, T.S.: Intuitive mapping of UML 2 activity diagrams into fundamental modeling concept Petri net diagrams and colored Petri nets. In: 15th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, pp. 191–200. Institute of Electrical and Electronics Engineers Inc., Belfast (2008)
3. Wang, R.Z., Dagli, C.H.: An Executable System Architecture Approach to Discrete Events System Modeling Using SysML in Conjunction with Colored Petri Net. In: 2nd Annual IEEE Systems Conference, pp. 1–8. IEEE Computer Society, Montreal (2008)
4. Ni, F., Wang, M.Z., Liao, J.J., Zhou, J.D.: Enhancing DoDAF with a HCPN executable model to support validation. In: 2th International Symposium on Computational Intelligence and Design, pp. 283–287. IEEE Press, Changsha (2009)
5. Wagenhals, L.W., Liles, S.W., Levis, A.H.: Toward executable architectures to support evaluation. In: 2009 International Symposium on Collaborative Technologies and Systems, pp. 502–511. IEEE Computer Society, Baltimore (2009)
6. Bai, X.L., Luo, X.S., Bai, X.H., Chen, H.H., Guo, D.K.: Study of DoD architecture simulation validation based on UML and extended colored Petri nets. In: 2008 IEEE International Conference on Networking, Sensing and Control, pp. 61–66. IEEE Computer Society, Sanya (2008)
7. Raistrick, C., Francis, P., Wright, J.: Model driven architecture with executable UML. Cambridge University Press, New York (2006)
8. Ge, B.F., Ren, C.S., Zhao, Q.S., Yang, K.W., Chen, Y.W.: Executable architecture modeling and analysis for system of systems. Systems Engineering and Electronics 31, 2191–2201 (2011)
9. Griendling, K., Mavris, D.N.: Development of a DoDAF-based executable architecting approach to analyze system-of-systems alternatives. In: IEEE Aerospace Conference, pp. 1–15. IEEE press, Big Sky (2011)
10. Mittal, S.: Extending DoDAF to allow integrated DEVS based modeling and simulation. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology 3, 95–123 (2006)
11. Mittal, S., Mitra, A., Gupta, A., Zeigler, B.P.: Strengthening OV-6a semantics with rule-based meta-models in DEVS/DoDAF based life-cycle architectures development. In: 2006 IEEE International Conference on Information Reuse and Integration, pp. 80–85. IEEE Computer Society, Waikoloa Village (2006)
12. Zeigler, B.P., Mittal, S.: Enhancing DoDAF with a DEVS-based system lifecycle development process. In: 2005 IEEE International Conference on Systems, Man and Cybernetics, pp. 3244–3251. IEEE, Waikoloa Village (2005)
13. Helle, P.S., Giblett, I., Levier, P.: An integrated executable architecture framework for System of Systems development. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology (2013)
14. Garcia, J.J., Tolk, A.: Executable Architectures in Executable Context enabling Fit-for-Purpose and Portfolio Assessment. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology (2013)
15. Vanderperren, Y., Dehaene, W.: From UML/SysML to Matlab/Simulink: current state and future perspectives. In: DATE 2006 Proceedings of the Conference on Design, Automation and Test in Europe, p. 93. European Design and Automation Association, Belgium (2006)

16. Hooman, J., Mulyar, N., Posta, L.: Coupling Simulink and UML Models. In: Proc. Symposium FORMS/FORMATS (2004)
17. Brisolara, L., Oliveira, M., Nascimento, F.A., Carro, L., Wagner, F.R.: Using UML as a front-end for an efficient Simulink-based multithread code generation targeting MPSoCs. In: UML for SoC Design (UML-SoC 2007): 4th International UML DAC Workshop (2007)
18. Sjöstedt, C.J., Shi, J., Törngren, M., Servat, D., Chen, D., Ahlsten, V., Lönn, H.: Mapping Simulink to UML in the design of embedded systems: Investigating scenarios and transformations. In: OMER4 Workshop: 4th Workshop on Object-Oriented Modeling of Embedded Real-Time Systems (2007)
19. Boldt, R.: Combining the Power of MathWorks Simulink and Telelogic UML/SysML-based Rhapsody to Redefine the Model Driven Development Experience. Telelogic (2007)
20. Kai, X.X., Ji, W., Luo, X., Chao, L.: A Model-Driven Approach for Evaluating Systems of Systems. In: International Conference on Complex Computer Systems, Singapore (in press, 2013)
21. Jouault, F., Kurtev, I.: Transforming models with ATL. In: Bruel, J.-M. (ed.) MoDELS 2005. LNCS, vol. 3844, pp. 128–138. Springer, Heidelberg (2006)

An Obesity Agent Based Model: A New Decision Support System for the Obesity Epidemic

Ali K. Bourisly

Biomedical Engineering Unit
Department of Physiology
Faculty of Medicine
Kuwait University
Kuwait
ali.b@hsc.edu.kw

Abstract. An agent based model (ABM) has been designed, developed, and implemented for the obesity epidemic. The Obesity ABM has been shown to serve as a decision support system as well as for running in-silico experiments. Eight in-silico experiment were run with different experimental parameter setups. The results suggest that food prices is an effective strategy to reduce obesity compared to exercise, individual encounter, number of food source allocations, and advertisements.

Keywords: agent-based model, multi-agent system, obesity, modeling, health, healthcare, actors.

1 Introduction

Obesity across the world has more than doubled since the 1980s and more than 60% of the world's population lives in countries where overweight and obesity kills more people than underweight. In the United States, where there are continuous research efforts to monitor and measure obesity rates and statistics, obesity is on the rise [1][2]. The obesity epidemic in the US has almost doubled to more than 30% between 1970 and 2000, with almost two-thirds of the US population being overweight [3][4]. Also, Obesity has been shown to have tremendous impact on the health, economic, psychological and social status of individuals and society as a whole[5]. If no proper intervention is placed to control the obesity epidemic, the current US generation may have shorter life expectancy than their parents and even grandparents [6].

There also exists a direct relationship between risk of death and disease on one axis and obesity on the other. In addition to a significant increased risk of type II diabetes and cardiovascular diseases, evidence suggests that there exists a direct relationship between excess body weight and risk of cancer in the following organ sites: breast, colon, oesophagus, endometrium, and kidney[7].

Obesity and causes of obesity are complex as well as complexly related [3][8] and there has been limited efforts towards in-silico experiments for obesity that address such complexity. Furthermore, current models, for the most part, do not reflect real world dynamics and do not incorporate interactions among members of the population

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 37–48, 2013.

during the process of obesity diffusion. For that we have designed and implemented an agent based model to study through in-silico experiments obesity as a complex adaptive system, and how members interact within a complex system to effect adoption of decisions to become or stay obese as well as the resulting diffusion of obesity in a population. Given the model's factors, our agent-based model allows for study of diffusion of obesity at the micro and macro levels as well as serves as a decision support system for intervention to reduce obesity rate and prevalence.

2 Background

2.1 Obesity as a Complex System

The obesity epidemic has three attributes that make it an especially challenging complex adaptive system problem (excellent problem for modeling) [3][8]: Firstly, the diverse and wide range in the involved scale levels [9]. There is empirical evidence that the obesity epidemic is affected by genes [10][11], neurobiology [12][13], psychology [14][15], familial structures and influences [16][17], social norms[18][19], environment [20][21], markets [10][22], and public policy[3][23]. Secondly, diversity of actors who have the potential of affecting a persons food intake. These actors may include families, retailers, industry, media, and sources of information[3].

Thirdly, multiplicity of mechanisms in obesity. Such mechanisms may include the mesocorticolimbic reward pathway which has been shown to be influenced by food [24][25]. Also, choice of food is influenced by neurobiological systems [11][26], and is measurable by psychological factors such as dietary disinhibition [15][27]and sensitivity to reward[28][29].

There exists feedback and interconnected relationship between obesity key factors. For example although exercise patterns on the individual level have been shown to be linked to the risk of becoming obese[30], obesity in itself has been shown to be a determinant of individual exercise patterns[31]. Similarly, although there is a clear association between risk of obesity and diet [31], dietary habits are in turn significantly affected by social networks[32] distance and ease of access to food[33]. Therefore it is likely that the relationship between all key factors cannot be easily parameterized especially given the non-linear relation among many key factors of obesity[8]. Moreover, a solution to the complicating causes and key factors associated with obesity is to adopt complex system dynamic computational models. A complex systems approach itself allows for consideration and inclusion of a network of heterogenous cause or factors that may not necessarily be linearly related and can interact non-linearly[34].

2.2 Complex Systems and Agent Based Modeling

A complex system is a system characterized by sensitivity to initial positions, strong interdependence, feedback, multiple metastable states and not necessarily having a Gaussian distributed output. Complex systems also consist of interacting agents with adaptive behavior(s) that are not in equilibrium and interact through both positive and

negative feedback and can result in emergent behavior and characteristics [35][36][37][38][39].

Agent based modeling allows researchers to model components in real-world and natural systems in a straightforward manner. ABMs retain much of the flexibility, consistency, and precision of other modeling techniques[40]. They also allow for a unique modeling approach that incorporates adaptive algorithms; providing a viable way to study adaptive actors rather than just fully rational ones. Another important characteristic of ABM is its ability to produce emergent results whereby emergent phenomena refers to results at the level of the aggregate system that may not be anticipated, discovered, expected, nor predicted by isolated examination of elements of the system[41][42][43]. Therefore in this work we propose to design and implement an agent based model that will serve as a decision support system to guide decision making towards proper intervention(s) to control the spread of the obesity epidemic as well as provide solutions to allow for individuals to become non-obese. Our ABM is also designed to study the spread of obesity as a complex adaptive system through in-silico experiments.

3 Methods

3.1 Model Overview

The agent based model for the obesity epidemic is programmed using Netlogo 4.1 [44]. For the purposes of this research, we have included several environmental parameters that have been shown to affect obesity rates[2][45][23][30][46][47] [48][49][50][3]. These environmental parameters are provided in Figure 1 and include healthy food and non-healthy food, health advocate and health antagonist agents, free and fee-based exercise facilities, as well as connections to obese and non-obese

Fig. 1. Diagram depicting the factors (surrounding circles) that can affect an obese or non-obese individual (center circle) to stay obese/non-obese or become obese/non-obese in the ABM.

individuals. These environmental parameters are governed by ease of access and price elasticity indexes as well as exposure with respect to the obese/non-obese individual. All these factors and interactions determine the spread of obesity within the population governed by the adoption criteria.

3.2 Environmental Parameters

The number of agents in the environment can be set prior to each simulation iteration. The concept of time (referred to as "tick" in Netlogo) is modeled according to movement of obese and non-obese individuals in the environment. A tick is recorded after every obese and non-obese individual makes a random walk in the environment. The following are the environmental parameters for our model:

Obese individual: An agent representing a person who is obese and randomly placed in the environment. The number of such individuals can be set prior to each iteration. The number of obese individuals can increase or decrease across iterations based upon the individual's adoption threshold, which is affected by other environmental parameters.

Non-obese individual: An agent representing a person who is not obese and randomly placed in the environment. The number of such individuals can be set prior to each iteration. The number of non-obese individuals can increase or decrease across iterations based upon the individual's adoption threshold, which is affected by other environmental parameters.

Healthy food: An agent representing a source of healthy food from which an individual can purchase healthy food. The number of such sources can be set prior to each iteration and are randomly distributed in the environment.

Non-healthy food: An agent representing a source of non-healthy and obesity-causing food from which an individual can purchase such food. The number of such sources can be set prior to each iteration and are randomly distributed in the environment.

Health advocate: An agent representing media, health campaign, or individual advocating healthy life style and food sources that do not cause obesity. The number of health advocate agents can be set prior to each iteration and are randomly distributed in the environment.

Health antagonist: An agent representing media, campaign, or individual antagonizing healthy life style or advocate food sources that cause obesity. The number of health antagonist agents can be set prior to each iteration and are randomly distributed in the environment.

Fee-based exercise facility: An agent representing a recreational or fitness facility that promotes exercise and movement, and a reoccurring subscription fee governs use of such a facility. The fee-based exercise facilities are randomly distributed across the environment and their count can be set prior to each iteration.

Free exercise facility: An agent representing a recreation or fitness facility that promotes exercise and movement without any associated fees. The number of free exercise facilities can be set prior to each iteration and are randomly distributed in the environment.

3.3 Adoption Criteria

In this ABM, that allows for in-silico experiments as well as serve in decision support for intervention, both obese and non-obese agents (individuals) interact with each other as well as with other agents in the environment (i.e. healthy food, non-healthy food, health advocate, health antagonist, fee-based exercise facility, free exercise facility). For example, a non-obese agent is more likely to adopt obesity or become obese if that respective agent is exposed to and is in close proximity to other obese agents, non-healthy food, health antagonist, and with limited exposure to exercise facilities. On the other hand, an obese agent is more likely to become non-obese and adopt "non-obesity" if that respective agent is exposed to as well as in close proximity to other non-obese agents, healthy food, health advocates, and exercise facilities.

Since healthy food, non-healthy food, and fee-based exercise facility require purchasing power, we have included a price elasticity variable in our model. The food price elasticity is set at 0.6 which was derived from demand for major food categories and assessment of mean elasticity by food category[46], while the price elasticity index for fee-based exercise facilities was set at 0.9 based on income incentive critical evaluation to promote physical activity[47]. As an example for price elasticity, a 10% increase in price of food (0.6 price elasticity) should reduce consumption by 6%.

Furthermore, in our model, in order for an individual to become or stay obese; the non-healthy food purchase count should be greater than the healthy food purchase count for each respective individual, as well as : Interaction with health antagonist + interaction with other obese individuals should be greater than or equal to 5. This criteria was set as the adoption threshold for obesity whereby interaction is determined by close proximity to respective obese/non-obese individual. On the other hand, the threshold for an individual to become or stay non-obese was determined according to the following formulation: the healthy food purchase count should be greater than the healthy food purchase count for each respective individual, as well as: interaction with health advocate + interaction with other non-obese individuals should be greater than or equal to 5.

3.4 Experiments

Eight experiments are conducted to analyze the evolution of non-obese populations towards reduction of obesity in the population under different scenarios. Each experiment at each multiple of 10 count of the respective variable was run five times and the average of the 5 runs are reported and analyzed. Experiments 1-8 (Table 1) vary the number of healthy food sources (10-100) across percent-change in food price and fee-based exercise facilities, as well as across both equal obese and non-obese populations, and 35.7% obesity (equal to recent obesity rate in the United States[4]).

4 Results and Discussion

We designed and implemented an agent based model for the obesity epidemic. The ABM was also designed to serve as a decision support system to reduce obesity in a given population. The ABM was also designed to run in-silico experiments with the goal of intervention to reduce obesity rate and obesity prevalence. In this work we ran 8 in-silico experiments.

Table 1. Initial experimental parameters for experiments 1-8

Experimental Parameters	*Experiment 1*	*Experiment 2*	*Experiment 3*	*Experiment 4*	*Experiment 5*	*Experiment 6*	*Experiment 7*	*Experiment 8*
Obese Population	**1000**	**1000**	**1000**	**1000**	**555**	**555**	**555**	**555**
Non-Obese Population	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**
Obese % of Population (%)	**50%**	**50%**	**50%**	**50%**	**35.70**	**35.70**	**35.70**	**35.70**
Non-Obese % of Population (%)	**50%**	**50%**	**50%**	**50%**	**64.30**	**64.30**	**64.30**	**64.30**
Healthy Food	**Variable***	**Variable***	**Variable***	**Variable***	**Variable***	**Variable***	**Variable***	**Variable***
Non-Healthy Food	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**
Health Advocate	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**
Health Antagonist	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**
Fee-based exercise facility	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**
Free exercise facility	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**
%change healthy food price	**0%**	**-25%**	**-25%**	**-25%**	**0%**	**-25%**	**-25%**	**-25%**
%change non-healthy food price	**0%**	**0%**	**10%**	**10%**	**0%**	**0%**	**10%**	**10%**
%change fee-based exercise facility	**0%**	**0%**	**0%**	**-25%**	**0%**	**0%**	**0%**	**-25%**

Results of Experiments 1-4 (Table 1) vary in convergence points (point at which obese population = non-obese population and after which non-obese population becomes greater than obese population). Experiment 1 (Figure 2) convergence occurs between 80 and 90 healthy food sources (healthy food count), while Experiment 2 (Figure 3) which reduced the price of healthy food by 25% shows a convergence when healthy food sources are between 60 and 70 in number. Figure 4 shows the results of experiment 3 by which the price of healthy food remained at -25% while the price of healthy food is increased by 10%. The convergence of experiment 4 occurred at 50 healthy food count (healthy food sources). When the price of fee-based exercise facility was reduced by 25% in experiment 4 (Figure 5), the convergence value increased to be between 70-80 healthy food count.

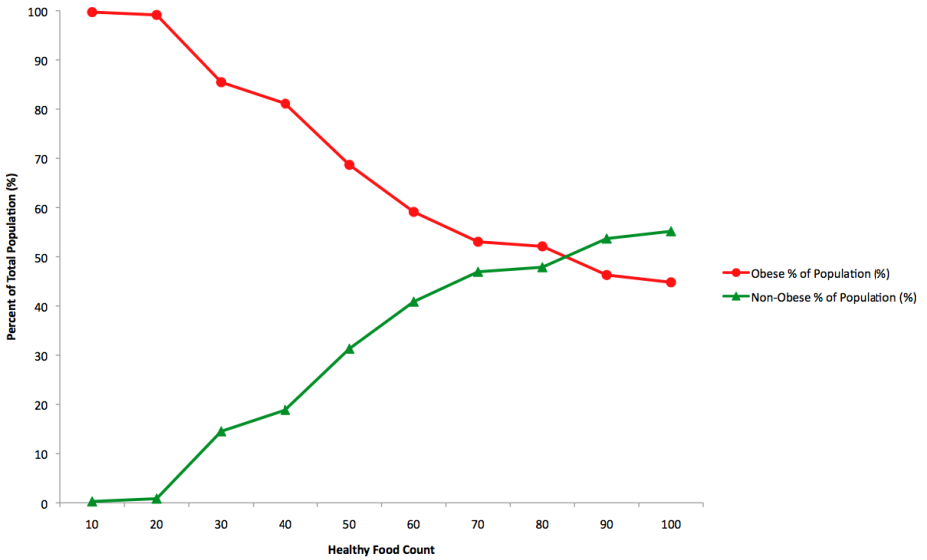


Fig. 2. Results of Experiment 1 depicting percent obese of population along with percent non-obese of population across increasing number of healthy food count (healthy food sources)

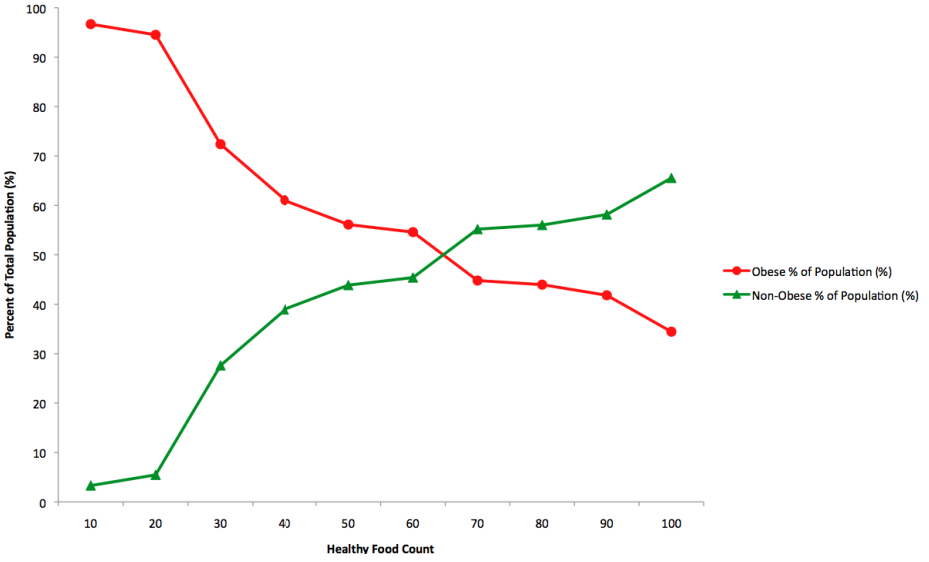


Fig. 3. Results of Experiment 2 depicting percent obese of population along with percent non-obese of population across increasing number of healthy food count (healthy food sources)

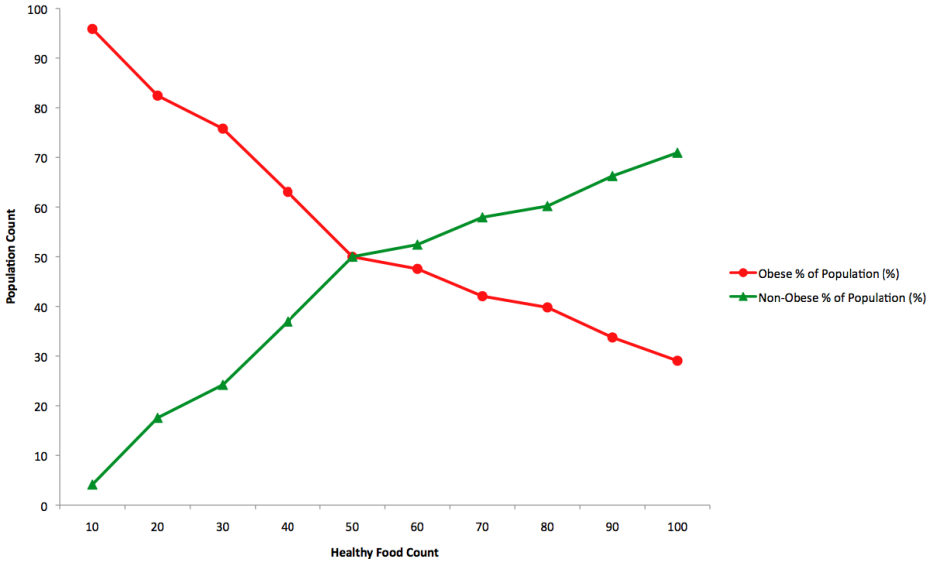


Fig. 4. Results of Experiment 3 depicting percent obese of population along with percent non-obese of population across increasing number of healthy food count (healthy food sources)

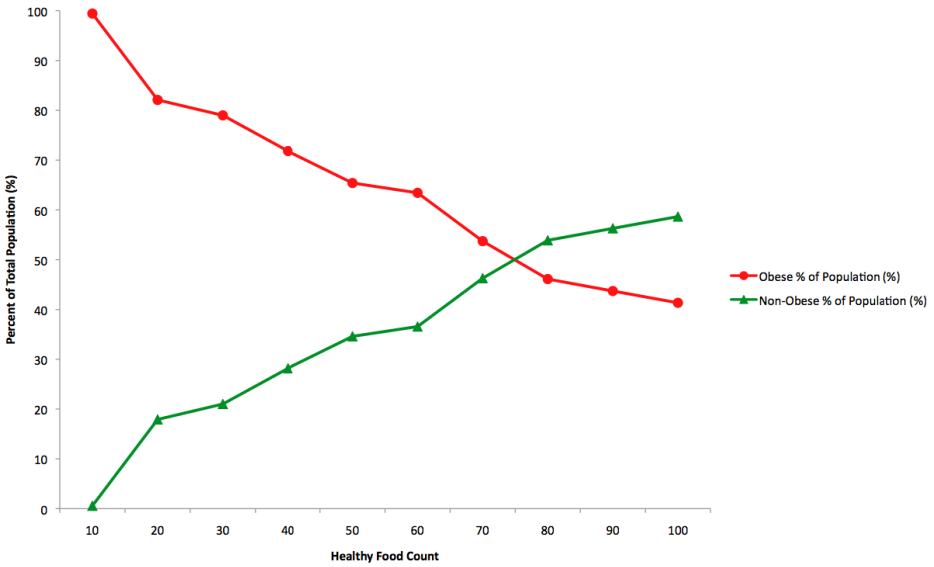


Fig. 5. Results of Experiment 4 depicting percent obese of population along with percent non-obese of population across increasing number of healthy food count (healthy food sources)

In experiments 1-4 (See Table 1) where obese population is equal to non-obese population, the lowest convergence point (number of healthy food count) after which %non-obese became greater than %obese occurred when healthy food prices were

reduced by 25% and non-healthy food prices were increased by 10 percent. This in fact shows that reduction in price of healthy food accompanied by increase in price of non-healthy food is more effective strategy towards reducing obesity rate and prevalence than not making any changes to prices, just reducing prices of healthy food, and even more effective than only reducing price of fee-based exercise facilities. Furthermore, these experiments show that reducing price of healthy food by 25% is more effective (lower convergence point) than reducing price of fee-based exercise facility alone, and increasing the number of healthy food sources (healthy food count) alone respectively.

Experiments 5-8 (Table 1) are setup with an initial obese population of 35.7%. Convergence for Experiment 5 occurs between 80 and 90 healthy food count, which is similar to experiment 1. At -25% change in healthy food price convergence is reduced to occur at 60 healthy food count in experiment 6. In Experiment 7, where -25% change in healthy food price and +10% change in non-healthy food price, the convergence point between obese and non-obese populations occurs between 40 and 50 healthy food count. Reduction by 25% of fee-based price, in experiment 8 allowed the convergence between populations to occur between 40 and 50 healthy food count.

Experiments 5-8 (See Table 1) followed the same procedures as experiments 1-4 but the obese were set at 35.7% to reflect recent US obesity percentage of population [4]. In this regard the convergence points of both experiments 7 and 8 are closer to each other (between 40-50 healthy food count) than when %obese = %non-obese in experiments 3 and 4 . Therefore suggesting that both reducing healthy food price by 25% along with increase in non-healthy food price remains the stronger strategy compared to the other strategies in the respective experiments. With respect to %obese < %non-obese initially, 25% reduction in fee-based exercise facility prices has been shown to be more effective than 25% reduction in healthy food price alone, and increasing healthy food sources alone.

5 Conclusion

In this work we ran 8 experiments to study different variables that have been related to spread of obesity in a population. Based on these experiments it has been shown that reduction in healthy food price accompanied by increase in non-healthy food price is the best strategy towards decreasing obesity in a population. Other effective strategies to reduce obesity include reducing prices of fee-based exercise facilities, and reduction in healthy food prices.

References

1. Wang, Y., Beydoun, M.A.: The obesity epidemic in the United States–gender, age, socioeconomic, racial/ethnic, and geographic characteristics: a systematic review and meta-regression analysis. Epidemiologic Reviews 29, 6–28 (2007)
2. Zhang, Q., Wang, Y.: Trends in the association between obesity and socioeconomic status in U.S. adults: 1971 to 2000. Obesity Research 12, 1622–1632 (2004)

3. Hammond, R.A.: Complex systems modeling for obesity research. Preventing Chronic Disease 6, A97 (2009)
4. Ogden, C.L., Carroll, M.D., Curtin, L.R., McDowell, M.A., Tabak, C.J., Flegal, K.M.: Prevalence of overweight and obesity in the United States, 1999-2004. JAMA: The Journal of the American Medical Association 295, 1549–1555 (2006)
5. Bray, G., Bouchard, C., James, W.: Handbook of Obesity: Clinical Applications. Marcel Dekker, New York (1998)
6. Olshansky, S., Passaro, D., Hershow, R., Layden, J., Carnes, B., Brody, J., Hayflick, L., Butler, R., Allison, D., Ludwig, D.: A Potential Decline in Life Expectancy in the United States in the 21st Century — NEJM. The New England Journal of Medicine 352, 1138–1145 (2005)
7. Bianchini, F., Kaaks, R., Vainio, H.: Overweight, obesity, and cancer risk. The Lancet Oncology 3, 565–574 (2002)
8. Galea, S., Riddle, M., Kaplan, G.A.: Causal thinking and complex system approaches in epidemiology. International Journal of Epidemiology 39, 97–106 (2010)
9. Glass, T.A., McAtee, M.J.: Behavioral science at the crossroads in public health: extending horizons, envisioning the future. Social Science & Medicine 62, 1650–1671 (1982)
10. Finkelstein, E.A., Ruhm, C.J., Kosa, K.M.: Economic causes and consequences of obesity. Annual Review of Public Health 26, 239–257 (2005)
11. Epstein, L.H., Temple, J.L., Neaderhiser, B.J., Salis, R.J., Erbe, R.W., Leddy, J.J.: Food reinforcement, the dopamine D_2 receptor genotype, and energy intake in obese and nonobese humans
12. Stunkard, A., Sorensen, T., Hanis, C., Teasdale, T., Chakraborty, R., Schull, W., Schulsinger, F.: An Adoption Study of Human Obesity. The New England Journal of Medicine 314, 193–198 (1986)
13. DelParigi, A., Chen, K., Salbe, A.D., Hill, J.O., Wing, R.R., Reiman, E.M., Tataranni, P.A.: Successful dieters have increased neural activity in cortical areas involved in the control of behavior. International Journal of Obesity 31, 440–448 (2005)
14. Nederkoorn, C., Smulders, F.T.Y., Havermans, R.C., Roefs, A., Jansen, A.: Impulsivity in obese women. Appetite 47, 253–256 (2006)
15. Lindroos, A.-K., Lissner, L., Mathiassen, M.E., Karlsson, J., Sullivan, M., Bengtsson, C., Sjöström, L.: Dietary Intake in Relation to Restrained Eating, Disinhibition, and Hunger in Obese and Nonobese Swedish Women. Obesity Research 5, 175–182 (1997)
16. Temple, J.L., Legierski, C.M., Giacomelli, A.M., Salvy, S.-J., Epstein, L.H.: Overweight children find food more reinforcing and consume more energy than do nonoverweight children. The American Journal of Clinical Nutrition 87, 1121–1127 (2008)
17. Seckl, J.R., Meaney, M.J.: Glucocorticoid programming. Annals of the New York Academy of Sciences 1032, 63–84 (2004)
18. Cutting, T.M., Fisher, J.O., Grimm-Thomas, K., Birch, L.L.: Like mother, like daughter: familial patterns of overweight are mediated by mothers' dietary disinhibition. The American Journal of Clinical Nutrition 69, 608–613 (1999)
19. De Castro, J.M., Brewer, E.M., Elmore, D.K., Orozco, S.: Social facilitation of the spontaneous meal size of humans occurs regardless of time, place, alcohol or snacks. Appetite 15, 89–101 (1990)
20. Booth, K.M., Pinkston, M.M., Poston, W.S.C.: Obesity and the built environment. Journal of the American Dietetic Association 105, S110–S117 (2005)
21. Papas, M.A., Alberg, A.J., Ewing, R., Helzlsouer, K.J., Gary, T.L., Klassen, A.C.: The built environment and obesity. Epidemiologic Reviews 29, 129–143 (2007)

22. Cutler, D., Glaeser, E., Shapiro, J.: Why Have Americans Become More Obese? Journal of Economic Perspectives 17, 93–118 (2003)
23. Smed, S., Jensen, J.D., Denver, S.: Socio-economic characteristics and the effect of taxation as a health policy instrument. Food Policy 32, 624–639 (2007)
24. Schultz, W.: Getting Formal with Dopamine and Reward. Neuron 36, 241–263 (2002)
25. Wise, R.A.: Brain Reward Circuitry. Neuron 36, 229–240 (2002)
26. Tran, A.H., Tamura, R., Uwano, T., Kobayashi, T., Katsuki, M., Matsumoto, G., Ono, T.: Altered accumbens neural response to prediction of reward associated with place in dopamine D2 receptor knockout mice. Proceedings of the National Academy of Sciences of the United States of America 99, 8986–8991 (2002)
27. Lawson, O.J., Williamson, D.A., Champagne, C.M., DeLany, J.P., Brooks, E.R., Howat, P.M., Wozniak, P.J., Bray, G.A., Ryan, D.H.: The Association of Body Weight, Dietary Intake, and Energy Expenditure with Dietary Restraint and Disinhibition. Obesity Research 3, 153–161 (1995)
28. Leonard, E., Saelens, B.: Behavioral Economics of Obesity: Food Intake and Energy Expenditure. In: Reframing Health Behavior Change with Behavioral Economics, pp. 293–311. Lawrence Erlbaum Associates, New York (2000)
29. Saelens, B.E., Epstein, L.H.: Reinforcing value of food in obese and non-obese women. Appetite 27, 41–50 (1996)
30. DiPietro, L.: Physical activity, body weight, and adiposity: an epidemiologic perspective. Exercise and Sport Sciences Reviews 23, 275–303 (1995)
31. Trost, S.G., Owen, N., Bauman, A.E., Sallis, J.F., Brown, W.: Correlates of adults' participation in physical activity: review and update. Medicine and Science in Sports and Exercise 34, 1996–2001 (2002)
32. Feunekes, G.I., de Graaf, C., Meyboom, S., van Staveren, W.A.: Food choice and fat intake of adolescents and adults: associations of intakes within social networks. Preventive Medicine 27, 645–656
33. Morland, K., Wing, S., Diez Roux, A.: The contextual effect of the local food environment on residents' diets: the atherosclerosis risk in communities study. American Journal of Public Health 92, 1761–1767 (2002)
34. Cariani, P.: Emergence and Artificial Life. In: Langton, C., Taylor, C., Farmer, J.D., Rasmussen, S. (eds.) Artificial Life II, pp. 775–797. Addison-Wesley (1992)
35. Gell-Mann, M.: What is Complexity? Complexity 1 (1995)
36. Fichter, L.S., Pyle, E.J., Whitmeyer, S.J.: Strategies and Rubrics for Teaching Chaos and Complex Systems Theories as Elaborating, Self-Organizing, and Fractionating Evolutionary Systems. Journal of Geoscience Education 58, 65–85 (2010)
37. Fichter, L.S., Pyle, E.J., Whitmeyer, S.J.: Expanding Evolutionary Theory Beyond Darwinism with Elaborating, Self-Organizing, and Fractionating Complex Evolutionary Systems. Journal of Geoscience Education 58, 58–64 (2010)
38. Kastens, K., Manduca, C., Cervato, C., Frodeman, R., Goodwin, C., Liben, L., Mogk, D., Spangler, T., Stillings, N., Titus, S.: How geoscientists think and learn. Eos Trans. AGU 90, 265–266 (2009)
39. Holland, J.H.: Exploring the evolution of complexity in signaling networks. Complexity 7, 34–45 (2001)
40. Holland, J.H., Miller, J.H.: Artificial adaptive agents in economic theory. American Economic Review 81, 365–370 (1991)
41. Parker, D.C., Manson, S.M., Janssen, M.A., Hoffmann, M.J., Deadman, P.: Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review. Annals of the Association of American Geographers 93, 314–337 (2003)

42. Axelrod, R.: The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration. Princeton University Press, Princeton (1997)
43. Wang, J., Gwebu, K., Shanker, M., Troutt, M.D.: An application of agent-based simulation to knowledge sharing. Decision Support Systems 46, 532–541 (2009)
44. Wilensky, U.: Netlogo (1999)
45. Christakis, N.A., Fowler, J.H.: The spread of obesity in a large social network over 32 years. The New England Journal of Medicine 357, 370–379 (2007)
46. Andreyeva, T., Long, M., Brownell, K.: The impact of food prices on consumption: a systematic review of research on the price elasticity of demand for food. Journal Information 100 (2010)
47. Von Tigerstrom, B., Larre, T., Sauder, J.: Using the Tax System to Promote Physical Activity: Critical Analysis of Canadian Initiatives. American Journal of Public Health 101, e10–e16 (2011)
48. Fowler, J.H., Christakis, N.A.: Estimating peer effects on health in social networks: a response to Cohen-Cole and Fletcher; and Trogdon, Nonnemaker, and Pais. Journal of Health Economics 27, 1400–1405 (2008)
49. Veenstraa, G., Luginaahb, A., Wakefield, S., Birchd, S., Eyles, J., Elliott, S.: Who you know, where you live: social capital, neighbourhood and health. Social Science & Medicine 60, 2779–2818 (2005)
50. Kernper, K.A., Sargent, R.G., Drane, J.W., Valois, R.E., Hussey, J.R.: Black and White Females' Perceptions of Ideal Body Size and Social Norms. Obesity Research 2, 117–126 (1994)

An Agent Framework for High Performance Simulations over Multi-core Clusters

Franco Cicirelli and Libero Nigro

Laboratorio di Ingegneria del Software
Dipartimento di Ingegneria Informatica Modellistica Elettronica e Sistemistica
Universitá della Calabria
87036 Rende (CS) - Italy
f.cicirelli@dimes.unical.it, l.nigro@unical.it

Abstract. Agent based modeling and simulation is widely recognized as an effective tool for the analysis of complex systems. This paper proposes a novel approach to modeling and high-performance parallel simulation of scalable agent models based on actors and the THEATRE agency. The approach aims to an exploitation of the computing power of modern clusters of multi-core machines. Key factors of the approach are (i) it allows to take advantage of the lock-free cooperative model of concurrency of actors even in a parallel/multi-threaded scenario, (ii) it avoids serialization of messages exchanged among actors residing on different theatres allocated on a same CPU. Achievable execution performance of the proposed simulation framework is demonstrated through the parallel/distributed simulation of a large-scale multi-agent system.

Keywords: multi-agent systems, actors, modeling, parallel simulation, multi-core architectures, Java.

1 Introduction

There is a broad consensus about the effectiveness of agent based modeling and simulation (ABMS) for problem-solving activities and for explaining, predict and analyze the behavior of complex systems like those relevant to the domains of biology, social science and engineering [1–5]. Several ABMS tools exist, each one characterized by such critical issues as ease of programming, visualization support, integrated development capability, rapid prototyping and so forth [5, 6]. However, being ABMS systems usually high-demanding from a computational point of view [7], a frequent limiting aspect is their execution speed. Most simulators are constrained to a sequential execution environment [8, 9] (e.g. RePast, Swarm, NetLogo), and few tools are currently demonstrated capable of scaling up to more than 10^4 agents with a reasonable runtime [6, 8]. To cope with performance issues, parallel and distributed simulation techniques [7, 10–14] are often used. On the other hand, the actual trend of multi-core processors [15, 16] push forward for the adoption and the development of new algorithms, approaches and software infrastructures especially tailored to an exploitation of the computing

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 49–60, 2013.

potentials of the newly available processor architectures [9, 17–19, 5, 20]. This paper proposes an original approach to modeling and high-performance parallel/distributed simulation of scalable agent models based on a variant of actors [21] and the Theatre agency [11]. The contribution of the approach is threefold: (i) it allows to take advantage of the lock-free cooperative model of actor concurrency even in a parallel/multi-threaded scenario, (ii) it avoids serialization of messages exchanged among actors residing on different theatres allocated on a same CPU, (iii) it enables the achievement of high execution performance in a parallel/distributed context, i.e. a multi-core cluster.

The paper is structured as follows. Section 2 gives an overview of the Theatre architecture. Section 3 details the novel redesign of Theatre on top of multi-core clusters. Section 4 describes a challenging scalable multi-agent model which belongs to the predator/prey pursuit domain. Section 5 reports the carried out simulation experiments on the predator/prey model, and the achieved performance. Finally, conclusions are presented with an indication of on-going and future work.

2 The Theatre Architecture

Features of the Theatre distributed infrastructure [11] belong to two layers: (a) the *execution platform layer*, where *theatres* (i.e. Logical Processes or LPs) are the building blocks "in-the-large" which provide the environmental services supporting actor execution, migration and interactions. Services are made available to actors through a suitable API; (b) the *actor layer*, where actors are the building blocks "in-the-small". They are programmed in Java and capture the application business. Basic components in a theatre platform (see Fig. 1) are (i) an instance of the Java Virtual Machine (JVM), (ii) a Control Machine (CM), (iii) the Transport Layer (TL); (iv) the Local Actor Table (LAT) (v) a Network Class Loader (NCL). CM hosts the runtime executive of the theatre, i.e. it offers basic services of message scheduling/dispatching which regulate local actors. CM organizes all pending (i.e., scheduled) messages in one or multiple message queues. During its basic control loop, a pending message is selected (e.g., the or one of most imminent in time) and dispatched to its destination agent. Message processing is atomic, i.e. it cannot be preempted nor suspended. TL furnishes the services for sending/receiving network messages and migrating agents. Concretizations of TL may refer to Java Sockets, Java RMI, HLA/RTI [11], Terracotta [17] and the services of the Globus Toolkit [22]. LAT contains references to local agents of the theatre. NCL is in charge of getting dynamically and transparently the class of an object (e.g. of a migrated agent) from a network code server.

Actors are reactive objects which encapsulate a data state and communicate to one another by asynchronous message passing. Messages are typed objects. Actors are at rest until a message arrives. Responding to a message causes in general the following reactions: (i) new actors are (possibly) created (ii) some messages are sent to known actors (acquaintances). For proactive behavior, an

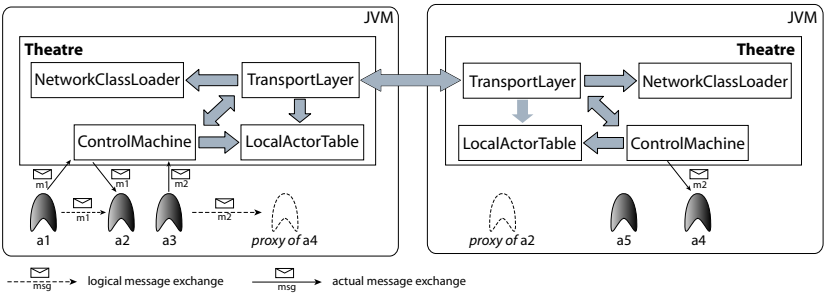


Fig. 1. A THEATRE system

actor can send to itself one or more messages (iii) the actor migrates to a different theatre (iv) current state of the actor is changed. Actors are light-weight: they do not have internal threads. Message handling extends the thread of control of the theatre within which the agent runs. A huge number of application actors can be created. All of this contributes to model scalability and good execution performance [11, 17]. Theatres and actors are assumed to have unique names. At its creation, the Java reference of an actor is stored in the LAT (see Fig. 1). Subsequently, the agent can move to another theatre etc. The Java reference, though, of an agent persists despite migration. After migration, in the LAT of the source theatre the agent reference is kept but now refers to a *proxy* version of the actor, which behaves as a forwarder. Dispatching a message to a proxy actor automatically generates a network message to the destination theatre. An agent can come back to a theatre where a proxy of itself exists. In this case, the proxy is replaced by the normal version of the actor which thus gets its state updated. Agent migration implies the relation proxy/normal of its acquaintances to be revised according to the viewpoint of the destination theatre. Some acquaintances become proxies because the corresponding actors reside in a remote theatre. Other acquaintances can change from proxy to normal in the case the referred actor is local to the reached theatre.

3 Tailoring Theatre to Multi-core Clusters

Current availability of multi-threaded architectures makes it possible to allocate more than one theatre to a single computational node, i.e. to a single CPU. More in particular, one can have, on each CPU, a number of theatres equal to the number of available cores. Obviously, it is not normally possible to bind a theatre to a specific core. The exploitation of cores is ultimately a matter of the underlying operating system. The rationale behind the adaptation design of the THEATRE architecture on top of multi-core clusters relies on three basic principles: (i) avoiding the use of specific linguistic constructs to deal with the parallel/distributed runtime infrastructure, (ii) making as lean as possible the transport layer and the time management service by avoiding any recourse

to a third part, although general purpose, middleware, (iii) maximizing the exploitation of shared memory so as to constrain object serialization to distributed communications only and to agent migration among distributed nodes. Towards a fulfillment of the above mentioned issues, the following remarks are useful. The adopted actor model of computation, by itself, reduces the programming concerns relevant to a parallel/distributed runtime. This is mainly due to the fact that communication among actors is based on asynchronous message passing. Such a communication model, however, becomes problematic in the case more than one thread of control (i.e. multiple control machines) are allocated to a same CPU and a certain number of actors are shared. In this scenario, in fact, multiple messages could be concurrently dispatched to a same actor. As a consequence, some sort of coordination, with an eye to not impairing performance, should be used among control machines. Another issue to take into account refers to the relationship between an agent and its proxy. Each time an actor moves between theatres sharing local memory, turning an agent into a proxy and vice-versa constitutes only a waste of computational resources. A *dynamic* notion of proxy can then be introduced: if an actor is shared among theatres (and thus among control machines) residing on a same CPU, the *real* or *proxy* nature of the actor can be dynamically determined on the basis of the actual control machine which is dispatching messages to it (see also section 3.1). Provided all the above considerations are taken into account, from a linguistic point of view no change would be required in the way a system is modeled. The problem of managing the parallelism may be tackled by acting only at the level of the software infrastructure. In the following, the improvements made to the THEATRE architecture are detailed.

3.1 Specializing the Theatre Architecture

Fig. 2 depicts the THEATRE architecture in the case a cluster of multi-cores is considered. Rounded boxes enclose *tightly coupled* theatres allocated to a same computing node. It is highlighted that in such a case, theatres are instantiated in the same JVM. Two different types of communication links exist. Dashed arrows refer to communication channels based on shared local memory (e.g. shared buffers). All data traversing these channels do not require object serialization. The other kind of arrows refers to communication channels based on sockets. In this case object serialization is mandatory. Issues relevant to managing the two different communication channels are encapsulated and hidden in the Transport Layer. In the case an actor migrates among tightly coupled theatres, the specific actor becomes shared among them, i.e. its Java reference is shared among the various LATs, and the notion of dynamic proxy is adopted. Given a set of tightly coupled theaters, an actor appears as a proxy in all theatres except in the one on which the migration just occurred. A control machine remains in charge of actually dispatching messages only to *real* (i.e. not proxy) actors. All of this implicitly coordinates control machines thus avoiding concurrency problems. A whole actor model is partitioned among theatres. A specific model partitioning may take into account load balancing concerns and/or may group together highly

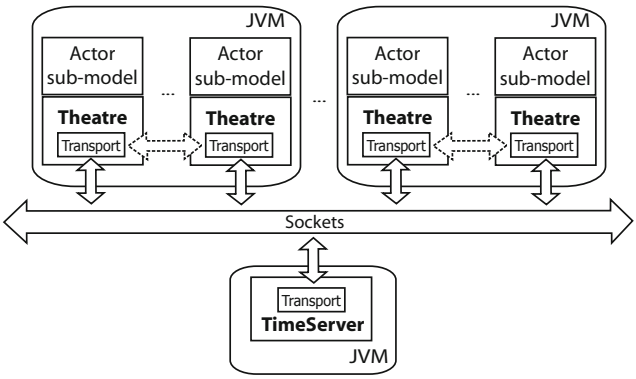


Fig. 2. Snapshot of THEATRE on top of a multi-core cluster

communicating actors. Communications among actors residing in a same theatre do not require the intervention of the transport layer. In Fig. 2 it is also shown another unique component named *TimeServer* which is in charge of managing time synchronization issues. A conservative algorithm [10] was implemented. For generality, in the figure, the time server is supposed to be allocated on its own computing node. For performance reasons, the server can also share the JVM with a running theatre.

3.2 Transport Layer Design

In Fig. 3 are reported some insights about the Transport Layer (TL) component. Each TL is equipped with a *local listener* and a set of *socket listeners*. The former receives messages coming from the other tightly coupled theatres; the latters receive messages originating from theatres allocated on different computing nodes. Other messages can be received from the control machine belonging to the same theatre of the TL. Dashed arrows in Fig. 3 denote communications that do not depend on object serialization. All arriving messages are gathered in a *delivery queue*. After that a message, on the basis of the address of its recipient, is *locally delivered* (i.e. delivered to a local listener of another TL or to the local control machine) or *remotely delivered* via sockets. The same story repeats in the case of actor migration and in the case the TL module is used by the TimeServer. To gain performance, TL is able to lightening the message/actor serialization process. Each message/actor owns a payload/status which can be separated from its owner. Each time a message/actor is sent through the network, only its payload/status is actually sent. Such a provision offers two advantages: (i) the amount of data sent/received is shortened, (ii) in the case an actor remotely migrates to a theatre where a proxy version of itself already exists, the operation which turns the proxy into the normal actor becomes extremely fast. In fact, restoring the actor state reduces to setting only the status object received from the network.

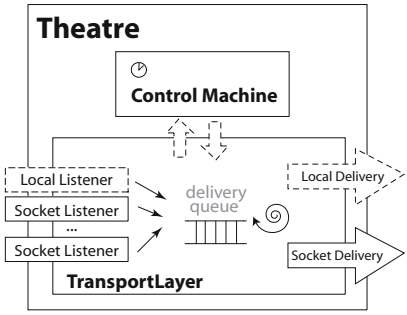


Fig. 3. Details of the Transport Layer component

4 A Predator-Prey Modelling Example

A complex multi-agent system based on the predator/prey pursuit domain [17, 23, 24] was selected for a performance study. Agents move over a territory and have a visibility radius. Predators have a greater visibility radius than preys. The goal of a predator is to attack a prey in order to eat it thus consuming its vital energy. Once a prey has been perceived, a predator has to decide about heading toward it. Information about the number, position and status of both surrounding preys and predators may influence the hunting behavior of a predator. The territory is a bi-dimensional grid without obstacles. Each cell is supposed to be large enough to host a certain number of predators and preys. Information about (part of) the hunting ground is managed by a specific environment actor (`EnvActor`). Predator and prey agents (respectively `PredatorActor` and `PreyActor`) have a behavior which dynamically defines how the entities react to incoming information from the surrounding environment. This information is obtained by exchanging messages with the `EnvActor`. Predators normally explore the territory using a weighted random walk which gives probabilistic preference to the direction chosen in the previous move, until some preys enter the visibility radius. The hunting strategy used by predators to move to a prey is based on a minority game. Once a prey is seen, the minority game is played to establish which predators cooperate to capture the prey, and which ones have to abandon thus possibly engaging for a new hunt. In a similar way, preys make a weighted random walk for the exploration of the territory. They always try to escape directions leading to predators.

4.1 Minority Game Strategy

Minority Game (MG) is an evolutionary game originally proposed in [25]. In its basic formulation there are N (supposed odd) players that make a choice between two options at each turn. Winners are those that have made the choice which is in the minority side, i.e. the one chosen by at most $(N-1)/2$ players. Each player is initially fed with a set of strategies that it may use to calculate its

next choice on the basis only of the past M outcomes of the game recorded in the player memory. Players rank their own strategies based on their respective capability to predict the winner side. Every player associates each strategy with a virtual score, which is incremented every time the strategy, if applied, would have predicted the minority side. At each turn, a player uses the first ranked strategy. The players earn a reward at every step they choose a winner strategy. At the beginning of the game, the content of player memories is randomly initialized and the strategies are equally ranked. A modification of MG, used in this work, consists in (a) blaming bad strategies (and also the players who applied them) by decrementing their virtual score, (b) the minority side is identified by using a cutoff value k different from $(N-1)/2$ [26], (c) an acquaintance relation among players is introduced. Each player owns a local view of the game, i.e. it will win or lose because of the decisions taken by players who are in the acquaintance relationship (Local Minority Game [27]).

4.2 Environment Modeling

For parallel/distributed simulation of the predator/prey model, the mission area must be partitioned among a number of interacting theatres. To each theatre is allocated a portion of the territory (see Fig. 4). The boundary region (two columns in Fig. 4) between two adjacent theatres is supposed to have a local zone handled by the local theatre and a replica of the local zone shared with the neighbor theatre. Due to partitioning the requirement arises for a moving actor (predator or prey) to (possibly) migrate from the local theatre to an adjacent one in the case its next position is located outside the portion managed by the current hosting theatre. Moreover, when a moving actor is located in the nearness of a local boundary, it needs information about the entities inside its visibility radius but located in a different region. Both issues are transparently addressed by environment actors (`EnvActors`), one per theatre. The role of an `EnvActor` is to handle the environmental information that may be of interest to local agents. Besides the local portion of the mission area, an `EnvActor` maintains an updated snapshot of the environment parts that although are not handled locally, fall in the visibility radius of a moving entity. The `EnvActor` exchanges messages with its neighbor peers by sending them updates every time a local event causes a change in a shared zone handled locally, and by receiving updates from partner `EnvActors` notifying a change in a shared zone handled outside. When a `PreyActor` or a `PredatorActor` wants to move, it sends to the `EnvActor` a timed message with next position. The time-stamp of the message mirrors the time needed to reach the destination cell. If the position is under the influence of the `EnvActor`, the latter replies with a message containing information about the area in the visibility radius of the sender. Otherwise the `EnvActor` migrates the sender to an adjacent theatre and asks the relevant `EnvActor` to reply to the migrating actor. It should be noted that the described protocol makes `PreyActors` and `PredatorActors` unaware of distributed simulation concerns, i.e. of model partitioning.

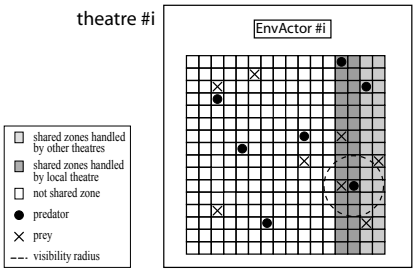


Fig. 4. `EnvActor` and boundary details of a region allocated to a theatre

5 Simulation Experiments

The described predator-prey model was used to evaluate the execution performance of the new THEATRE infrastructure. Three Win7 (64 bits) Intel i7 CPU 960, 4-Cores, 3.20 GHz, 10GB RAM, interconnected by a Gigabit Ethernet were used for the experiments. The Intel Hyper-Threading technology was enabled in order to have 8 virtual cores on each machine. Different simulation settings were considered: (a) a *pure multi-core scenario* (multiple theatres on to one single computational node), (b) a *pure distributed scenario* (one single theatre for each computational node), (c) a *hybrid scenario* (two theatres for each computational node). For load-balancing, in each scenario, the simulation model was equally split among the available theatres. Simulation experiments were carried out by using a territory of 3200x800 cells randomly populated by an equal number of `PreyActor`s and `PredatorActor`s. A variable number of predators ranging from 20 thousand to 140 thousand was admitted. Each simulation run lasts until predators complete their hunting. A hunt is considered completed when the 99% of the preys has been eaten. The wall clock time (WCT) of a simulation run is influenced by the number of time units needed to complete the mission (*huntingTime*). As a consequence, the value of WCT was normalized with respect to the time to hunt (*normalized WCT* = WCT/ huntingTime). All simulation results were obtained as the mean value of three runs.

5.1 Pure Multi-core Scenario

Fig. 5 shows the normalized WCT ($nWCT$) with respect to the number of predators, in the case a variable number of cores is used on a single machine. The nWCT significantly reduces as the number of exploited cores increases, i.e. a growing number of theatres are considered and the model is partitioned among them. In particular, a variable number of theatres, ranging from 1 to 8, was used. The trend of the curves is not strictly linear as the agent population grows. This reflects the fact that, due to the use of local minority game, the number of the exchanged messages is not linear in the number of existing predators.

By monitoring the CPU consumption of the JVM hosting the theatres, a direct correlation emerged between the number of utilized cores and the number of

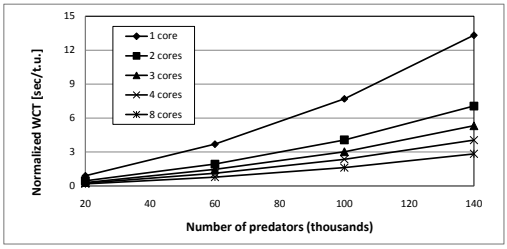


Fig. 5. Normalized WCT vs. number of predators (1 workstation)

Table 1. Observed CPU utilization

Number of theatres	Average CPU utilization
1	14.29% *(about one-eighth of full utilization, i.e. 1 core)*
2	26.18% *(about a quarter of full utilization, i.e. 2 cores)*
4	50.24% *(about half of full utilization, i.e. 4 cores)*
8	95.06% *(about full utilization, i.e. 8 cores)*

instantiated theatres (see Table 1). CPU utilization was evaluated by averaging dumped data collected by using the JConsole tool (available in the standard JDK) in the case a load of 140K predators are considered.

In Fig. 6 it is reported the achieved speedup vs. the number of used cores. Performance is very good especially in the case 2 and 3 cores are used. With 20K predators, i.e. with the smallest considered load factor, the experimental values of speedup are very close to the theoretical ones. With 8 cores, the speedup is about 4.7. It is important to note, though, that such a value was obtained by using the hyper threading technology. By switching off the hyper threading, it was observed that the speedup, in the case of respectively 2, 3 and 4 cores, is the same as that shown in Fig. 6. The speedup reduces as the population grows from 20K to 60K predators. A further growth beyond 60K does not increase the achievable speedup. The reduction in the speedup as population increases seems more related to a modification in the behavior of the simulated model than to a performance degradation of the simulation infrastructure. In fact, the population density affects the behavior of both predators and preys during hunt, thus changing the *intrinsic load factor* of the whole model.

5.2 Pure Distributed Scenario

Fig. 7 shows the achieved speedup vs. the number of predators in the case a distributed simulation based on three networked workstations each one hosting only one theatre, is used. The speedup monotonically increases with the agent population.

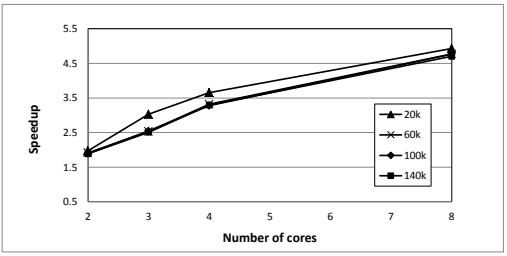


Fig. 6. Achieved speedup vs. number of cores (1 workstation)

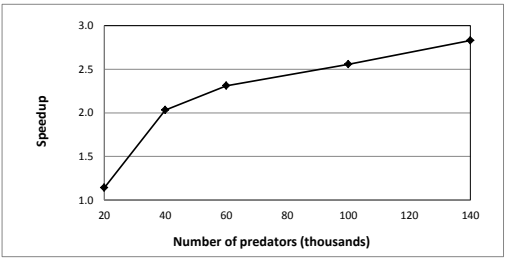


Fig. 7. Achieved speedup vs. number of predators (1 core × 3 workstations)

Differently from the pure multi-core scenario, with a population of 20K predators, the resultant speedup is negligible. This means that, networked communication and distributed coordination among theatres highly impair benefits deriving from the concurrent execution of sub-models allocated to distinct theatres. In order to take advantage of the distributed simulation a higher load factor has to be used. It is worth noting that management of distribution concerns is so predominant in this scenario, that changing the intrinsic load factor of the model is really not observable. A speedup of 2.8 was obtained with 140K predators.

5.3 Hybrid Scenario

This scenario consists of 6 theatres equally split among three networked machines (two involved cores per machine). As witnessed in Fig. 8, also in this case the

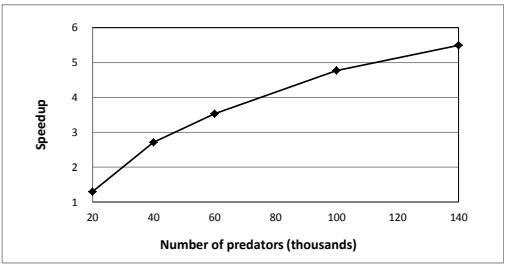


Fig. 8. Achieved speedup vs. number of predators (2 core × 3 workstations)

speedup monotonically increases as the agent population grows. A speedup of about 5.5 results with a population of 140K predators.

6 Conclusion

A novel approach for high performance simulation of agent-based models on top of nowadays multi-core clusters was proposed. The approach rests on the use of the THEATRE agency [11] and on a variant of actors [21]. THEATRE has been proved effective in supporting distributed simulation of large-scale complex systems. Previously authors' design efforts were mainly devoted to permit THEATRE to operate in open and heterogeneous simulation environment/middleware [22, 17, 11]. The work described in this paper, instead, is specifically tailored to improve simulation performance in a multi-threaded and distributed scenario. The achieved experimental results are very encouraging. On-going work is geared at:

- improving and tuning current system implementation;
- experimenting with the infrastructure on real-scale complex systems, e.g. related to crowd simulation, traffic simulation, risk assessment and virtual environments [10];
- extending the approach to explicitly support modeling and simulation of spatial environments [28, 14] into a multi-core infrastructure.

References

1. Wooldridge, M.: An introduction to multi-agent systems. John Wiley & Sons (2002)
2. Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison Wesley Longman (1999)
3. Logan, B.: Evaluating agent architectures using simulation. In: Evaluating Architectures for Intelligence: Papers from the 2007 AAAI Workshop, pp. 40–43. AAAI Press (2007); Technical Report WS–07–04
4. Pawlaszczyk, D., Strassburger, S.: Scalability in distributed simulations of agent-based models. In: Proc. of Winter Simulation Conference, pp. 1189–1200 (2009)
5. Tang, W., Wang, S.: HPABM: A hierarchical parallel simulation framework for spatially-explicit agent-based models. T. GIS 13(3), 315–333 (2009)
6. Frantz, C., Nowostawski, M., Purvis, M.: Multi-agent platforms and asynchronous message passing: Frameworks overview. Information Science Discussion Papers Series 7 (2010)
7. Collier, N., North, M.: Parallel agent-based simulation with Repast for High Performance Computing. In: Simulation, Trans. of SCS (2012), http://dx.doi.org/10.1177/0037549712462620
8. Perumalla, K.S., Aaby, B.G.: Data parallel execution challenges and runtime performance of agent simulations on GPUs. In: Proc. of the Spring Simulation Multi Conference, pp. 116–123 (2008)
9. Richmond, P., Walker, D.C., Coakley, S., Romano, D.M.: High performance cellular level agent-based simulation with FLAME for the GPU. Briefings in Bioinformatics 11(3), 334–347 (2010)
10. Fujimoto, R.M.: Parallel and distributed simulation systems. John Wiley (2000)

11. Cicirelli, F., Furfaro, A., Nigro, L.: An agent infrastructure over HLA for distributed simulation of reconfigurable systems and its application to UAV coordination. Simulation, Trans. of SCS 85(1), 17–32 (2009)
12. Logan, B., Theodoropoulos, G.: The distributed simulation of multiagent systems. Proceedings of the IEEE 89(2), 174–185 (2001)
13. Cicirelli, F., Giordano, A., Furfaro, A., Nigro, L.: HLA_ACTOR_REPAST: An Approach to Distributing RePast Models for High-Performance Simulations. Simulation Modelling Practice and Theory 19(1), 283–300 (2011)
14. Shook, E., Wang, S., Tang, W.: A Communication Aware Framework for Parallel Spatially Explicit Agent Based Models. International Journal of Geographical Information Science (2013), http://dx.doi.org/10.1080/13658816.2013.771740
15. Geer, D.: Industry trends: Chip makers turn to multicore processors. Computer 38(5), 11–13 (2005)
16. Bahulkar, K., Hofmann, N., Jagtap, D., Abu-Ghazaleh, N., Ponomarev, D.: Performance Evaluation of PDES on Multi-core Clusters. In: Proc. of the Intern. Symp. on Distributed Simulation and Real Time Applications, pp. 131–140 (2010)
17. Cicirelli, F., Furfaro, A., Giordano, A., Nigro, L.: Performance of a multi-agent system over a multi-core cluster managed by terracotta. In: Proc. of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, pp. 125–133 (2011)
18. Potuzak, T.: Distributed-parallel road traffic simulator for clusters of multi-core computers. In: Proc. of the IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, pp. 195–201 (2012)
19. Suzumura, T., Kanezashi, H.: Highly scalable X10-based agent simulation platform and its application to large-scale traffic simulation. In: Proc. of the Intern. Symp. on Distributed Simulation and Real Time Applications, pp. 243–250 (2012)
20. Ricci, A., Viroli, M., Piancastelli, G.: simpA: An agent-oriented approach for programming concurrent applications on top of Java. Sci. Comput. Program. 76(1), 37–62 (2011)
21. Agha, G.: Actors: a model of concurrent computation in distributed systems. MIT Press, Cambridge (1986)
22. Cicirelli, F., Furfaro, A., Nigro, L., Pupo, F.: Agents over the grid: An experience using the globus toolkit 4. In: Proc. of the 26th European Conference on Modelling and Simulation, ECMS 2012 (2012)
23. Jim, K.C., Giles, C.L.: Talking helps: evolving communicating agents for the predator-prey pursuit problem. Artif. Life 6(3), 237–254 (2000)
24. Panait, L., Luke, S.: Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-Agent Systems 11(3), 387–434 (2005)
25. Challet, D., Zhang, Y.C.: Emergence of Cooperation and Organization in an Evolutionary Game. Physica A 246(3-4), 407–418 (1997)
26. Johnson, N.F., Hui, P.M., Zheng, D., Tai, C.W.: Minority game with arbitrary cutoff. Physica A 269(2-4), 493–502 (1999)
27. Remondino, M., Cappellini, A.: Minority game with communication of statements and memory analysis: a multi-agent based model. International Journal of Simulation 6(5), 42–53 (2005)
28. Cicirelli, F., Giordano, A., Nigro, L.: Distributed simulation of situated multi-agent systems. In: Proc. of the IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications, pp. 28–35 (2011)

A Behavior Based Crowd Simulation Framework for Riot Controlling in City Environment

Liang Jia-hong[1], Li Meng[1], Fu Yue-wen[1], Yang Mei[1], and Li Shi-lei [2]

[1] College of Information Systems and Management,
National University of Defense Technology, 410073 Changsha, P.R. China
{Liang_jia_hong_prf,mengshuqin1984,fuyuwen,yangmei}@163.com
[2] Department of Information Security, College of Electronic Engineering,
Naval University of Engineering, 430000, Wuhan, P.R. China
leeshilei@yahoo.com.cn

Abstract. Creating complex and realistic crowd behaviors can be a difficult and time consuming task. By drawing the successful experiences of behavior-based AI and multi-agent simulation framework, this paper presents a behavior-based crowd simulation framework for riot controlling in city environment. A well-designed behavior-based prototype system is developed, which takes the advantages of the dynamics of interaction among the basis behaviors repertoires. The basis behaviors repertoires implemented by decision rules are composed of three parts, which correspond to the behaviors of the civilians, riots and soldiers respectively. Then these basis behaviors are combined into more complex behaviors by behavior selection mechanism. Eventually realistic crowd phenomena are created in the scenarios of riot controlling in city environment. We argue that this approach gives better results than conventional methods.

Keywords: Crowd simulation, behavior-based AI, multi-agent system, riot controlling.

1 Introduction

Crowds, ubiquitous in the real world from groups of humans to flocks of insects, are vital features to model in a virtual environment. Various simulation models and architectures have been developed. Crowd management and control means all measures taken in the normal process of facilitating the movement and enjoyment of people, as well as all measures prepared to be taken in the emergent process of people evacuations. Crowd management and control can be successful only when viewed as a combination of management of all the crowds, environment and their relationships [1,2,3].

Behavior-based control employs basis behaviors to collectively achieve the desired system-level behavior. As a set of distributed, interacting modules, behaviors are patterns of the crowd's activity emerging from interactions between the individuals and its environment in order to achieve and maintain a goal. Each behavior receives inputs from perception and/or other behaviors in the system, and provides outputs to individual's actuators or to other behaviors. Unlike reactive control, which utilizes collections of reactive rules, behavior-based system utilizes collections of behaviors, which supports reasoning, planning, and learning [4]. Simultaneously, behavioral

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 61–70, 2013.

based simulations allow for someone to study the result of agents performing a certain behavior, without having to see this behavior in practice.

The idea of composing basis behaviors into complex behaviors is not new. In fact, combining multiple simple basis behaviors into complex crowd behaviors is one of the most natural and general ways to generate realistic behaviors. Arnaud Masciotra, Sam Rodriguez, et al. investigated techniques for generating realistic and complex group behaviors by combining simple behaviors, such as evasion behavior, hiding behavior, waiting behavior, pursuing behavior, searching behavior and following behavior [5]. Research on biology, robotics and animal behaviors has shown that the concept of basis behaviors exists ubiquitously ranging from lower level basic bio-mechanic motor controls to high level ethological basis behaviors [5]. In Reynolds's seminal work, three basis behaviors (avoidance, alignment and coherence) were combined to generate flocking behavior [6]. Also, composing simple behaviors into complex behaviors is proposed in developing behavior based robotics. Dylan A. Shell and Maja J Matari investigated a behavior-based method for modeling and structuring control of one type of special robots which have social characteristics [7]. Xiaoshan Pan, Charles S. Han and Kincho H. Law present a multi-agent simulation framework for the study of human and social behavior during building emergency evacuations. They demonstrated that some emergent human social behaviors, such as competitive, queuing, and herding behaviors can be composed by a series of steering behaviors such as seek, follow, collision avoidance behaviors [8,9]. Sameul Rodriguez and Nancy M. Amato presented a behavior-based evacuation planning method which is inspired by roadmap-based motion planning and behavioral agent-based framework [10,11]. However, all the methods above designed the basis behaviors in single manner. They didn't consider the fact that in riot controlling scenarios, the crowd's composition is more complex and the riot's behaviors have a vast distinction with the civilians.

In this paper, we present a behavior-based multi-agent simulation framework for the study of the special social behaviors of civilian, riots and soldiers who have the responsibility to make the riots in control and to protect the civilians in city environment when crowd incidents happen. Our work is tightly related to ethology, artificial life, virtual reality and graphics, robotics. The methods used in these fields give us inspiration to the studies of this paper. So in this paper, we are focusing on scenarios of military nature, where civilian, riot and soldier agents can move within the environment and soldiers can control the riots and protect the civilian. Different aspects of agent behavior like Protesting, Fleeing, destroying, navigation to a goal or staying in group formation, are implemented by distinct basis behavior modules and the final observed behavior for each agent is an emergent property of the combination of simple behaviors and their interaction with the environment.

More specifically, the crowd investigated in our work is composed of four parts which are civilian individuals, civilian groups, riots groups and soldiers groups. Civilian individuals are the individuals who are neutral to riots and soldiers and can be hurt by the riots. Civilian groups are the groups of civilian in which one leader exist. The riots groups are the groups of riots who try to make some damage to the city and led by one leader. By contrary, the soldier groups are the forces to protect the civilians and facilities, also are led by one leader. By viewing a group within a crowd as an entity, we can identify many significant factors that may contribute to crowd behaviors.

The components of the crowd are represented in Figure 1.

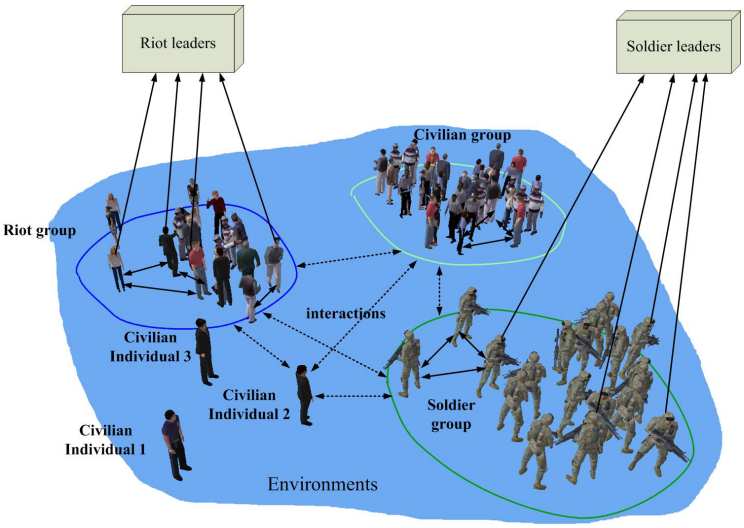


Fig. 1. Components of the crowd

As illustrated in Figure 1, the crowd behaviors are the interactions between these four parts. By designing and combing the basis behaviors of these four parts, realistic and complex behavior phenomena are achieved.

2 Behavior-Based Crowd Simulation Framework and Crowd Simulation Engine

This section presents a behavior-based multi-agent crowd simulation framework that simulates the civilian, riots and soldiers' behaviors in city circumstances. Our framework adopts a behavior-based multi-agent simulation paradigm as a basic scheme to develop the simulation system. Multi-agent simulation has been widely accepted as a promising approach to model complex emergent phenomena. In our framework, each human individual is modeled as an autonomous agent who interacts with the virtual city environment and other agents according to the basis behavior repertories. Each agent attains information of the environment by perception process. Depending on the environment and the behavior repertories of individuals and the groups, the agent could interact and react in a collaborative, neutral or hostile manner.

Our system architecture which is constructed in a distributed and layering manner is schematically shown in Figure 2. The system consists of five basic components: an **Environment Generator**, a **Population Generator**, an **Attributes Renovator**, a

Crowd Behavior Generator, and a **Visualization Engine**. Specifically, the **Crowd Behavior Generator** is composed of **Basis Behavior Repertories**, a **Behavior Filter**, and a **Behavioral Selection Mechanism**.

Environment Generator. The purpose of this module is to produce the geometries representing the physical environment (e.g., a building or a train station of the city, etc.). The environment model built in Flt format in Creator software which is developed by MultiGen-Paradigm is employed in this study. The current environment information is sent to the Crowd Simulation Engine to simulate crowd behaviors.

Population Generator. The Population Generator generates three types of virtual agents (civilian, riots and soldiers) to represent a crowd according to each one's percentage. Also in each type of virtual agents, a distribution of personality, emotion, belief/desire/intention and motivation parameters are considered. For example, we can assume that in city environments, the total number of civilian is twenty times of the number of the riots. And the number of soldiers is approximately equal to the ones of the riot. This module also generates random populations for statistical study of individuals.

Attributes Renovator. The purpose of this module is renewing the attributes of the virtual agent. The attributes are the internal states of the virtual agent including personality, emotion belief/desire/intention and motivation according to current environment information and previous internal states. All the internal states and external information of the agent are updated every simulation step.

Crowd Behavior Generator. The **Crowd Behavior Generator** is the key module of the behavior-based multi-agent simulation system. Based on the type of the agents, each agent is assigned with one basis behavior repertoire. The basis behaviors in the repertoires are sent to behavior filter to derive available behaviors corresponding to agent's current states. Then a behavioral selection mechanism is used to select and integrate the available behaviors. In **Crowd Behavior Generator**, constructing basis behavior repertoires and behavior selection mechanism are two of the key issues in behavior-based system.

Visualization Engine. The visualization engine is primarily used to display the simulated results in an intuitive manner. We employ a human animation package called DI-Guy, which is commercially available from Boston Dynamics Inc. to receive the positions of agents, and then generates and displays 2D/3D visual images in real time. DI-Guy provides an SDK that enables each character's behavior repertoire to be controlled by external user-specified C/C ++ programs. We build **Population Generator**, **Attributes Renovator** and **Crowd Behavior Generator** upon this interface by writes our own control programs.

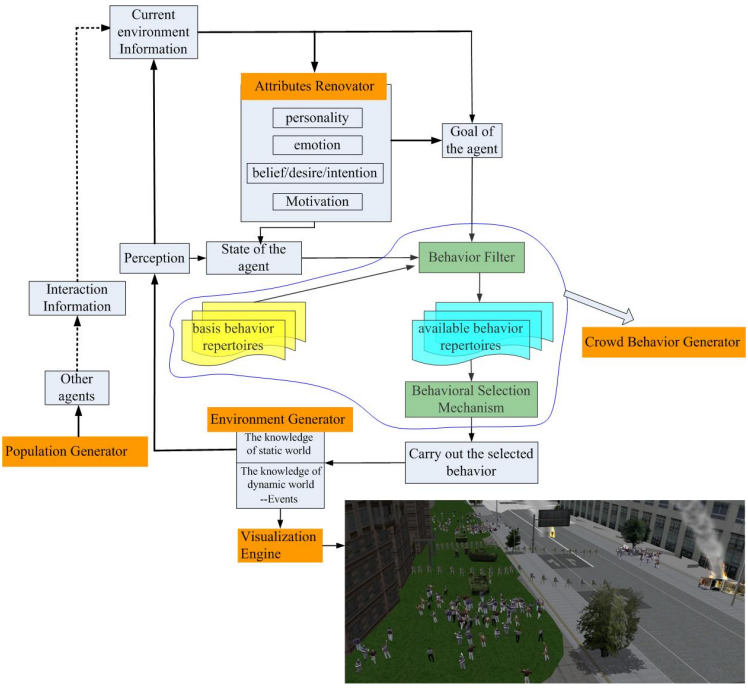


Fig. 2. Behavior-based crowd simulation framework

Main features of our framework for generating crowd behaviors include:

1. Users only need to provide a set of behaviors in the basis repertoire and the system will compose them into complex behaviors by calling the behavior selection mechanism.
2. Users do not need to define the relationships between each behavior, i.e., these simple behaviors can be developed independently without knowing what other behaviors are in the system.
3. The relationships between simple behaviors, and the complex behaviors themselves, can evolve as the agents learn from their past experiences [5]. And the complex behaviors are responsible for dealing the global goals of the crowd by path planning algorithm, e.g., the approach based on probabilistic roadmap method (PRM).

In next section, we will illustrate the two key issues in behavior-based system: constructing basis behavior repertoires and behavior selection mechanism.

3 Constructing Basis Behavior Repertoire and Behavior Selection Mechanism

This section discusses the implementation of the multi-agent system for the simulation of crowd behavior. The discussion is divided mainly into two parts: constructing the basis behavior repertoire, and behavior selection mechanism, which together make up the **Crowd Behavior Generator** with behavior filter.

3.1 Constructing Basis Behavior Repertoire

The behavior that the agent is equipped with will determine how the agent reacts throughout the simulation. These behaviors determine the actions that the agent or groups of agents take. The behaviors we develop need to be dynamic enough so that they can be applied to an agent at any point and the agent will then start performing that behavior. The behaviors we develop need to be general enough to be applicable to a single agent or group of agents. In our behavior-based crowd behavior simulation system, each agent has a list of different behaviors as Table 1 illustrated [12-14].

Table 1. Types of behaviours for the three types of agents in our crowd

Behavior repertoires	**Description**	**Application Agent type**
Stand Around	Just hanging around. Minor motion relative to other agents.	Civilian
Walking Around	Walking around a specific neighborhood. Perhaps driven by a maximum distance from origin.	Civilian individual
Competitive behavior	Competitive behavior is often observed in emergency situations, when human individuals compete for their own chances of fleeing	Civilian individual and group
queuing behavior	queuing behavior emerges spontaneously when a crowd gathers at an exit, permitting the crowd to "stream" out of the exit in an orderly manner.	Civilian individual and group
blind behavior	Blind behavior is often observed during the evacuation of a crowd in a room with two exits – one exit is clogged while the other is not fully utilized	Civilian individual and group
Small Groups Talking	Small congregations with dialogue, then breaking up and reforming.	Civilian , riots group
Flee randomly	Fleeing from local cause for alarm.	Civilian individual and group
Crowding	Any group gathering, density function driven, no specific cause.	Civilian individual and group
Watching / Curiosity	Watching some target with some attention. Curiosity may spread to nearby agents.	Civilian individual and group
Carrying Large Items	Moving carefully with large items, children in arms. Noticeably different movement patterns.	Civilian individual
Flee with intent	Fleeing towards a specific point, either a known safe-haven or a goal	Civilian individual and group
Helping	Seeking for help from other civilian and soldiers.	Civilian individual and group
Inciting	Trying to change behavior status of surrounding agents (e.g. calm them down, make them angry).	riots

Table 1. (*Continued.*)

Cowering	Physically submissive, quiet. Be tamed by soldiers	riots
Protesting	Similar to watching but posture more assertive to aggressive and ambient noise loud.	riots
Subtle Move Towards	Trying to get closer when the riot wants to attack	riots
Making damage	The behavior to make some damage to the city or others	riots
Subtle Move Away	Trying to keep away subtly when the riot wants to play tricks on the soldiers	riots
protecting	Trying to keep between us and it. It might be a building, an individual or a group	soldiers
shepherding	Shepherding is the most important interaction behavior between soldiers and riots, also it is exist between soldiers and civilian. The soldiers trying to keep the riots in control. [15].	Soldiers and riots; Soldiers and civilian
Haggling or Negotiating	Negotiation enables an agent to exchange information and reach agreements with others.	Soldiers and riots
Target following	This behavior allows an agent to follow a moving target.	All the crowd
Directing Behavior	The behavior to lead others, such as Traffic cops. Mothers at children.	All the crowd
Collision avoidance	This behavior gives an agent the ability to maneuver in the virtual environment without running into an obstacle or other agents.	All the crowd
Seek target	A seek acts to steer an agent toward a target point. When a goal point is detected, an agent adjusts its orientation and velocity toward the goal.	All the crowd
Flocking behavior	Flocking behavior is used to model the collective gathering behavior when the members of the group moving to the goal.	Civilian , riots and soldiers group

Because of the flocking behaviour and the shepherding behaviour are the most two complex behaviour in above basis behaviour repertoire, we illustrate them at the end of this section. The other behaviors in Table 1 can be implemented with decision rules.

Shepherding Behavior. Shepherding behaviors, specifically, are one class of flocking behaviors in which one or more external agents (called Shepherds, i.e. soldiers) attempt to control the motion of another group of agents (called a flock, i.e. riots) by exerting repulsive forces from shepherds to the flock. It includes four sub-behaviors: herding, covering, patrolling and colleting behaviors [15]. Herding behavior is responsible for driving the riots from one place to the other place. Covering behavior can be seen as a

series of Herding behaviors. Patrolling behavior is used to place the riots under surveillance and drive off the hostile riots from predefined region. Collecting behavior is used to collect the casualties to some safe place. All these four sub-behaviors are important in riot-controlling. Detailed implements can be founded in [15].

Flocking Behavior. Flocking behavior is used to model the collective gathering behavior when the members of the group moving to the goal. Aiming at the agents group consisting of one leader agents and *N* follower ones, a distributed control method to simulate the flocking behavior is used [16].

The motion of each virtual agent is described by two integrators as:

$$\begin{cases} \dot{\boldsymbol{p}}_i(t) = \boldsymbol{v}_i(t) \\ \dot{\boldsymbol{v}}_i(t) = \boldsymbol{a}_i(t) \end{cases} \quad i = 1, 2, \ldots, N \tag{1}$$

where $\boldsymbol{p}_i$, $\boldsymbol{v}_i$, $\boldsymbol{a}_i$ are respectively the position, velocity and acceleration of agent *i*. Analogously, the virtual leader has the following model of motion.

$$\begin{cases} \dot{\boldsymbol{p}}_\gamma(t) = \boldsymbol{v}_\gamma(t) \\ \dot{\boldsymbol{v}}_\gamma(t) = \boldsymbol{a}_\gamma(t) \end{cases} \tag{2}$$

where $\boldsymbol{p}_\gamma$, $\boldsymbol{v}_\gamma$, $\boldsymbol{a}_\gamma$ are respectively the position, velocity and acceleration of leader agent. the control protocol of flocking behavior is given as:

$$\boldsymbol{a}_i = -\sum_{j \in N_i(t)} \nabla_{\boldsymbol{p}_i} \psi_\alpha\left(\left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|_\sigma\right) + \sum_{j \in N_i(t)} a_{ij}(t)\left(\boldsymbol{v}_j - \boldsymbol{v}_i\right) + \boldsymbol{a}_\gamma^i - c_1\left(\boldsymbol{p}_i - \boldsymbol{p}_\gamma^i\right) - c_2\left(\boldsymbol{v}_i - \boldsymbol{v}_\gamma^i\right) \quad i = 1, \ldots, N \quad c_1, c_2 > 0 \tag{3}$$

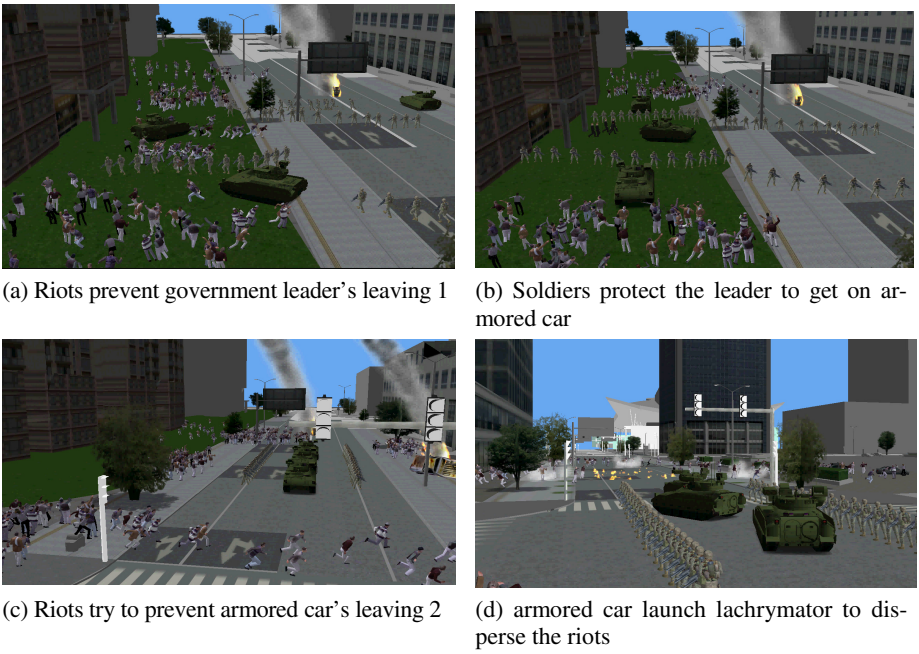
where the $a_{ij}(t)$ is the adjacent matrix, $\psi_\alpha\left(\left\|\bullet\right\|_\sigma\right)$ is a nonnegative smooth pairwise potential function.

3.2 Behavior Selection Mechanism

Behavior selection mechanism is used to arbitrarily select and coordinate the output of multiple single behaviors in the basis behavior repertoire. It can be generally classified as either arbitration or command fusion architectures. Arbitration mechanisms select one behavior from a group of competence modules. Arbitration mechanism for action selection can be divided into fixed priority-based, winner-take-all, and state-based mechanism. Command fusion mechanism is composed of motor schema method, fuzzy fusion method, hybrid behavior selection method and action voting. All these methods give us great convenience to combine the available behaviors in **Crowd Behavior Generator**. In our framework, a feasible scheme is that different mechanisms are used according to different levels of agent's intelligence. When agent's intelligence is low, we use arbitration mechanism, conversely, command fusion mechanisms are used.

4 Experimental Results and Discussion

In our simulation, we employ a human animation software package called DI-Guy, which is commercially available from Boston Dynamics Inc. we control the behaviors of the shepherd and the flock using SDK by C++ programs. The scenario is that: in a city populated with a majority of civilian, a group of riots try to attack the government leaders and impede the soldiers to rescue the leader. The simulation results are illustrated in Figure 3.



(a) Riots prevent government leader's leaving 1

(b) Soldiers protect the leader to get on armored car

(c) Riots try to prevent armored car's leaving 2

(d) armored car launch lachrymator to disperse the riots

Fig. 3. Simulation results for riot controlling

In Figure 3, we can see that our behavior-based crowd simulation framework for riot controlling in city environment is realistic and believable.

5 Summary

Although there have been some research studies on behavior-based crowd simulation for safety engineering purposes, few efforts have been conducted to study the core of crowd safety problem for riots controlling. In this paper, we have presented a computational framework for studying civilian, riots and soldiers' behaviors during emergency events. So, the potential of our framework for studying human and social behaviors is promising.

Acknowledgments. This paper was supported by national natural science foundation (61170160).

References

1. Thalmann, D., Raupp Musse, S.: Crowd Simulation. Springer, London (2007)
2. Pelechano, N., Allbeck, J., Badler, N.I.: Virtual Crowds: Methods, Simulation, and Control. Morgan & Claypool Publishers (2008)
3. Jingjing, S.H.I.: Crowd Management for Large-scale Outdoor Events—Multi-agent Based Modeling and Simulation of Crowd behaviors. Ph.D thesis, The Chinese University of Hong Kong, Hong Kong, China (2006)
4. Bruno, S., Oussama, K.: Springer Handbook of Robotics. Springer, Heidelberg (2008)
5. Masciotra, A., Rodriguez, S., Lien, J.-M., Amato, N.M.: Composable group behaviors. Technical Report, TR05-006, Parasol Laboratory, Department of Computer Science, Texas A&M University (2005)
6. Reynolds, C.W.: Flocks, herds, and schools: A distributed behaviroal model. In: Computer Graphics, pp. 25–34 (1987)
7. Shell Dylan, A., Mataric Maja, J.: Behavior-based methods for modeling and structuring control of social robots. In: Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation. Cambridge university press (2005)
8. Pan, X.: Computational Modeling of Human and Social Behavior for Emergency Egress Analysis, Ph.D. Thesis, Stanford University (2006)
9. Pan, X., Han, C.S., Dauber, K., Law, K.H.: A Multi-Agent Based Framework for the Simulation of Human and Social Behaviors during Emergency Evacuations. AI & Society 22, 113–132 (2007)
10. Rodriguez, S., Amato, N.M.: Behavior-based evacuation planning. In: IEEE International Conference on Robotics and Automation Anchorage Convention District, Anchorage Alaska (2010)
11. Varol, K.: Modeling autonomous agents in military simulations. Ph.D. Thesis, University of Central Florida (2006)
12. McKenzie, F., Kruszewski, M.P.P., Gaskins, R., Nguyen, Q.A., Seevinck, J., Weisel, E.: Integrating crowd-behavior modeling in military simulation using game technology. Simulation & Gaming 39(1) (2008)
13. Wong, Y.H.: Ignoring the Innocent: Noncombatants in Urban Operations and in Military Models and Simulations. Pardee RAND Graduate School (2006)
14. Levesque, J., Cazzolato, F.: Simulating Civilians for Military Training: A Canadian Perspective, `http://www.google.com.hk/gwt/x?gl=CN&hl=zh-Hans-CN&u=http://ftp.rta.nato.int/public//PubFullText/RTO/MP/RTO-MP-HFM-202///MP-HFM-202-05.doc`
15. Lien, J.-M., Rodriguez, S., Malric, J.-P., Amato, N.M.: Shepherding behaviors with multiple shepherds. In: Proc. IEEE Int. Conf. Robot. Autom (ICRA), pp. 3413–3418 (2005)
16. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Transactions on Automatic Control 51(3), 401–420 (2006)

Modified Shuffled Frog Leaping Algorithm for Simulation Capability Scheduling Problem

Yingying Xiao[1,2], Xudong Chai[2], Li Bo Hu[1,2], Chen Yang[1], and Tingyu Lin[1]

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China
[2] Beijing Simulation Center, Beijing 100854, China
xiaoyingying504@126.com

Abstract. Based on the analysis of characteristics of simulation capability scheduling problem in cloud simulation platform, this paper gives its mathematical description and introduces a modified shuffled frog leaping algorithm (MSFL) to solve the above optimization problem with multi-mode constraint. The MFSL introduces GA to code the feasible solution space. During the random execution of coding, decoding and mutation, it increases three layers of coding constraints including simulation capability, task logic and feasible mode, to ensure the randomness of the solving process in the controllable scope. Thus it can reduce the search range of solution space, get rid of the meaningless illegal solution, and ultimately improve the convergence speed of the algorithm and avoid precocity.

Keywords: Modified Shuffled Frog Leaping Algorithm, Cloud Simulation, Simulation Capability Scheduling, Constraint Model.

1 Introduction

With the increasing application of the modeling and simulation (M&S) technology in engineering domain and non-engineering domain, reference [1] proposed a new networked modeling and simulation platform—Cloud Simulation Platform. Based on the idea of cloud computing, Cloud Simulation realizes a new M&S system with the characteristics of distributed, heterogeneous integration, collaboration, interoperability and reuse, provides multiple users with M&S services according to the users' needs via the network. In Cloud Simulation Platform, multi-user acquiring all types of M&S capability on demand is one of the four service patterns [2], while the sharing and providing services on request is one of the core characteristics. By virtualizing the simulation capability and making the simulation capability service-oriented, the platform can provide the users through the network with the simulation capability in form of service [3], and thus realize the circulation and transaction of the simulation capability in large scale.

So far, project management technology has been widely used in the research and development of complex products [4], where the Resource-Constrained Project Scheduling Problem (RCPSP) has already been studied and applied widely. As one of the key technology of the Cloud Simulation Platform, Simulation Capability Project

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 71–81, 2013.

Scheduling Problem (SCPSP) is a special kind of RCPSP, whose scheduling object is the simulation capability with multiple patterns, composited of various heterogeneous resources. Therefore, traditional resource model cannot be used directly to express the simulation capability. Meanwhile, SCPSP emphasizes on the project task's alternative simulation capability constraints, so the scheduled target is also different from the resources demand of traditional decision task, it emphasized that it picks a group of capability combination scheme to meet user goals under the constraints of candidates multi-mode capability set. So researching on the scheduling model and solving method of limited simulation capabilities and developing tools to provide users with project management oriented simulation scheduling plan functions can make users fully enjoy open simulation capability, and at the same time accomplish the unified planning of the centralized resources to improve the overall utilization of resources.

Like RCPSP, SCPSP is also a kind of NP - hard problem [5], traditional methods for solving these problems are accurate solving (dynamic planning, branch definition, etc.), heuristic (based on priority rules, local search techniques, etc.) and intelligent algorithm [6]. Although the traditional methods can find the exact solution of accurate algorithm, but for large projects with large quantity tasks, they can't get the optimal solution within an acceptable time. All kinds of heuristic algorithms are in poor efficiency on common [7], thus are restricted to some extent in engineering application. Along with the development of the computer and algorithm optimization, some intelligent algorithms have been applied gradually in solving RCPSP, and generally have better results [8], such as Simulated Annealing algorithm (SA), Genetic Algorithm (GA), Tabu Search algorithm (TS), Particle Swarm Optimization (PSO) and ant colony algorithm, etc.

This paper tries out a new heuristic algorithm—Shuffled Frog Leaping algorithm (SFL) to solve the problem of SCPSP. It has good global convergence capability, and can further improve the accuracy of the results and practical application value.

2 Simulation Capability Project Scheduling Problem

2.1 Brief Introduction

Simulation capability is the combination of three factors including person/organization, management, and technology, reflecting a kind of the configuration of three elements and integration ability, showing the level of simulation entity to complete a task for the goals of T (time), Q (quality), C (cost), S (service), E (clean environment), K (knowledge), noted as:

Simulation Capability = (Attributes, Organization, Management, Technology), with Attributes = {T,Q,C,S,E,K}.

Therefore, different resources configuration and management structure will make the same simulation capability has different TQCSEK attribute, resulting in the simulation capability become a compound body with a variety of patterns. Meanwhile, in Cloud Simulation mode, centralized management of scattered resources and scattered usage of centralized resources allow users to share the simulation capability in the capability pool, meanwhile restrict them with the limited simulation capability. Simulation Capability Project Scheduling Problem (SCPSP), therefore, is a special kind of

limited resource project scheduling problem, and it restricts and extends the following aspects.

Extension 1. Simulation capability is a kind of flexible renewable resources, that is, a capability can provide service for multiple tasks. At a moment, the supply is limited, but it is not consumed with the progress of the project.

Extension 2. Simulation capability is a special kind of multimode complex. Different way of resource allocation and management mode can make changes of the capability attributes, such as completion time, quality and cost.

Restriction 3. Each task can have multiple candidate capability services, but we only choose one capability assigned to this task, namely the demand of capability for each task is 1.

Restriction 4. In project planning, under constraint of the immediate predecessor logic and simulation capability, the start time of each task should be arranged as soon as possible.

Restriction 5. Before the project planning, candidate capability groups which can satisfy the functional requirements have been automatically chosen by the supply and demand matching engine, each task is restricted by the candidate capability groups.

Restriction 6. When multiple projects grab a capability at the same time, combination of bidding mechanism, with market regulation capability of configuration can be used. This paper does not discuss this problem, and we only study the capability of a single project scheduling.

2.2 Mathematical Model Description of the SCPSP Problem

According to reference [9], this paper uses activity-on-node, as shown in Fig. 1, to describe SCPSP with oriented graph $G = (V, E)$, which has the following characteristics: the project has N tasks with $J = \{J_0, J_1, J_2 \ldots J_j \ldots J_N, J_{N+1}\})$, where $J_0 J_{N+1}$ are virtual task nodes, and those tasks are separated into L stages. Each stage contains Q_l tasks that can be executed in parallel way. The j^{th} task J_j contains n_j candidate capability services obtained by the search of the supply and demand matching engine (noted $MC^j = \{MC_1^j, MC_2^j \ldots MC_{n_j}^j\}$), the start time of the task J_j is noted s_j, and its finish time is c_j, the immediate predecessor tasks set is noted P_j, and the immediate successor tasks set is B_j. The capacity of each capability service is R_i which has m modes (noted $MC_i^j = \{M_1, M_2 \ldots M_m\}$), and the duration of the k^{th} mode is d_k.

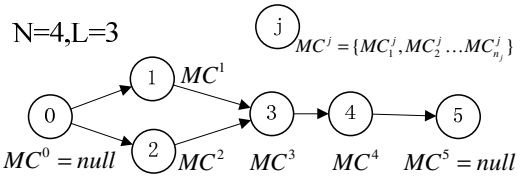


Fig. 1. Activity-on-node AON

The solution to the problem is to find the appropriate schedule and thus to fix a feasible start time(noted $S = \{s_1, s_2 \ldots s_j \ldots\}$) and finish time (noted $C = \{c_1, c_2 \ldots c_j \ldots\}$) for each task, under the restriction of the immediate predecessor and that of simulation capability, in order to satisfy the project planning objectives proposed by the users. We note that at the moment t, the set of tasks in working state is A_t, and the decision variables

$$x_{jnm} = \begin{cases} 1, & task\ \text{j is finished by the m}^{th} \\ & \text{mode of the } n^{th} \text{ capability} \\ 0 & \text{others} \end{cases} \tag{1.1}$$

Then the basic SCPSP can be described as:

$$\text{Min } c_j \tag{1.2}$$

s. t.

$$s_j \geq max_{h \in P_j} c_h \tag{1.3}$$

$$\sum_{j \in A_i} x_{jnm} \leq R_k \tag{1.4}$$

Formula 1.2 is the objective function, representing the minimization of project duration. Formula 1.3 requires that the project schedule plan follows immediate predecessor relationship between tasks, the start time of any task must be greater than or equal to the maximum completion time of all its immediate predecessor tasks. Formula 1.4 requires that project schedule planning meet the simulation capability constraints, at any time, the project's total demand of the any renewable simulation capability cannot surpass its supply.

According to the description above, the project containing L stages has $\prod_{l=1}^{L}(Q_l!)$ feasible execution orders. Meanwhile, a certain feasible execution sequence $\lambda = \{\lambda_1, \lambda_2, \ldots \lambda_j \ldots\}$ has $\prod_{j=1}^{J}(n_j \times m_j)$ kinds of simulation capability allocation modes. Therefore, the SCPSP problem has $\prod_{l=1}^{L}(Q_l!) \times \prod_{j=1}^{J}(n_j \times m_j)$ feasible scheduling solutions.

With increasing number of tasks, extension of the simulation model and raising complexity of business logic, the complexity of solving SCPSP will increase rapidly. The traditional accurate solving methods and heuristic search methods based on rules cannot give the feasible solution in the limited time. So, this paper adopts a new intelligent algorithm named Shuffled Frog Leaping algorithm and its modified form to solve the SCPSP problem.

3 Solution of SCPSP with Modified SFL

SFLA (Shuffled Frog Leaping Algorithm) is a new optimization algorithm proposed by M. Eusutl and E. Lansey in 2003 based on swarm intelligence [10], it combines the good global searching ability of PSO Algorithm and the stronger local search ability of Memtric [11]. It is characterized by simple implementation, high adaptability, good robustness etc. SFLA is an important research and development direction of swarm intelligence optimization technique.

3.1 Basic SFL

The basic principle of SFL algorithm [12] contains a randomly generated swarm with N frogs. This swarm is sorted by the fitness value of each frog's position and is divided into M groups. Then, a local search algorithm is executed n times in each group, and the worst solution (X_w) is updated. Those search and update processes are repeated within the groups, until a fixed number of iterations. After all groups have done a specified number of local searches, the algorithm will mix all frogs together, resort them and redivide them into groups, and then repeat the local search process above, until the termination condition is satisfied.

The update strategy detail of local search is described as follows:

$$D^k = r(X_b^k - X_w^k) \tag{2.1}$$

or

$$D^k = r(X_g^k - X_w^k) \tag{2.2}$$

$$X_w^{'} = X_w^k + D^k \tag{2.3}$$

With D^k is the displacement of the worst frog position in the k^{th} iteration, r is usually a random number between 0 and 1, X_b is the best solution in the group, and X_g is the best solution for the whole swarm. During the update, the worst solution X_w calculates the new position according to X_b with the equation (1) and (3), if the new solution $X_w^{'}$ is better than the old one, and then $X_w^{k+1} = X_w^{'}$ is acceptable. Otherwise, it will calculate the new position according to X_g with the equation (2) and (3), and if the new solution $X_w^{'}$ is better than the old one, then $X_w^{k+1} = X_w^{'}$ is acceptable or a new random location is selected to be the updated position.

According to the above description, the efficiency of SFL algorithm is determined by the generation of new strategy. In order to ensure that a solution which satisfies the convergence precision can be found in finite iterations, the algorithm must expand the randomness of solutions under the premise of gradual convergence so as to avoid the "premature" and to improve the convergence precision.

3.2 Solution of SCPSP with SFL Algorithm

For the discrete optimization problem, each element of the feasible solution is unique and discrete. If we use directly the update formula above, it will produce a large number of infeasible solutions, the algorithm is inefficient.

This paper uses the descriptive idea of generic algorithm on discrete problems for reference, introduces crossover and mutation operator, considers the constraint of simulation capability with different modes and redefines the meaning of frog displacement and position update.

Definition 1. In SFL, the frog position (feasible solution) is encoded by task list, and is noted as sequence $X = [\lambda; mc; mode]$. In sequence X, λ indicates to the task gene which meets the task execution order vector of tight constraints, mc indicates the gene of simulation capability which is a set of ability meeting the simulation capability constraints in vector λ, and mode indicates the mode gene which is mc's optional

set of mode vector. When given a task list, we use the serial schedule generation method to generate ability genes and mode genes, and then encode the feasible solution.

Definition 2. For any two frogs and their corresponding positions X_1 and X_2, the crossover operation is defined as replacing the m^{th} component to n^{th} component of X_1 which generated randomly, to the components of X_2 from m to n, and we note it as $C(m, n|X_1 \rightarrow X_2)$, then the new position after the crossover is defined to be $X'_w = X_w + C(m, n|X_1 \rightarrow X_2)$.

Definition 3. For a frog with position X, the mutation operation is defined as exchanging k_1 and k_2 in λ of X, and regenerating the task gene and mode gene. This operation is noted as $U_{mc,model}(H_\lambda(k_1, k_2))$. Then the new position after the mutation is $X' = X + U_{mc,model}(H_\lambda(k_1, k_2))$.

After redefining, the meaning of encoding and operators are changed. The subtraction between vectors is transformed into extracting the order difference of two frog positions in corresponding sequence, while the corresponding update operation is transformed into replacing the sequence on such different bits. And the operation of random generation is transformed into mutation operation. These transformations make them conform to the description of the discrete problem to solve the SCPSP problem.

4 Solution of SCPSP Based on Modified SFL Algorithm

Basic SFL algorithm can be used to solve the discrete SCPSP after redefinition. The randomly generated crossover-mutation location can maintain group diversity, enhance global searching ability of the algorithm, but has low convergence efficiency. To raise the searching efficiency, this paper uses the local searching techniques to improve the crossover-mutation operation, which makes the algorithm have better searching ability.

4.1 Crossover Operator

Simple genetic algorithm adopts the way of constant crossover probability and random selection of crossover point to make the crossover operation. This operation has some blindness and randomness, which cannot ensure that child generation is better than father generation. For this reason, this paper chooses the crossover locations according to the difference between individuality and the best solution, and uses simulated annealing algorithm to calculate the crossover-mutation probability coefficient of each crossover location to gradually accelerate convergence rate. At the same time, MSPSP is limited by the task logic and the simulation capabilities, which means that the solutions cannot be completely random. Otherwise there will be a large amount of non-feasible solutions, and will reduce the searching efficiency. So the adaptive crossover operator considering the constraint conditions is as follow:

(1) Calculate the individual maximum crossover coefficient.

$$\text{MaxCrossNum} = \text{length}(\text{Diff}(\lambda_b - \lambda_w)) \tag{3.1}$$

This describes the number of different positions of component product λ between the best solution X_b and the worst solution X_w.

(2) Produce the crossover position.

$$\text{ChangeSits} = (m, n) \tag{3.2}$$

$$\text{s.t.}$$

$$r1 = \text{rand} * \text{MaxCrossNum},\ \ r2 = \text{rand} * \text{MaxCrossNum} \tag{3.3}$$

$$m = \min(r1, r2),\ \ n = \max(r1, r2)$$

(3) Calculate the probability coefficient of selected crossover position.

$$\Delta t = \text{fitness}(X_w) - \text{fitness}(X_b) \tag{3.4}$$

This means chooses current position as crossover position with a probability of $\exp(-\Delta t/T)$.

(4) Make crossover operation to the selected crossover position. To reduce the non-feasible solution, firstly do the crossover operation to the component product λ, and ensure that it conforms to the previous constraint conditions. At the same time, make the same replacement for the component product mc and mode. For the new crossover solution, we accept it if the solution can meet the simulation capability constraints. Otherwise, we randomly produce a feasible capability mode according to a new component product λ' to be a new solution.

4.2 Mutation operator

The commonly used mutation operators are inversion mutation, exchange mutation, uniform mutation, non-uniform mutation, Gauss mutation and so on. Because each bit of these mutation operators are generally considered as the same, mutation positions are randomly selected according certain regulations. These operators can enhance the randomness of algorithm, but may influence the convergence speed. The extreme dynamic mechanism in the process of EO evolution has a strong local searching capability. In EO algorithm, the contribution of each bit to the individual quality is different, and each bit's fitness to the individual that it belongs with is calculated according to the contribution to the objective function, where the bit with minimum fitness is defined as the worst bit. In each iteration, EO always chooses the worst bit and its field to make a mutation, and all bits can be coevolution and the individual continues to be improved. So the differential evolution mutation operator is used in this paper, and the steps are shown as following.

(1) Calculate the fitness δ_i of the worst solution X_w of each bit.

$$\begin{aligned}\delta_i = &\left[\left|d_{opt}(P_j) - d_{opt}(j)\right| + \left|d_{opt}(B_j) - d_{opt}(j)\right|\right] \\ &-\left[\left|d(P_j) - d(j)\right| + \left|d(B_j) - d(j)\right|\right]\end{aligned} \tag{3.5}$$

In the formula, $d_{opt}(P_j)$ describes, in the optimal individual, the duration of P_j which represents the predecessor collection of task j in position i on vector λ. $d_{opt}(B_j)$ describes, in the optimal solution, the duration of B_j which represents the successor collection of task j in position i on vector λ. $d(j)$ describes the duration of task j in position i on vector λ of the current individual; $d(P_j)$ describes, in the

current individual, the duration of P_j which predecessor collection of task j of position i on vector λ. $d(B_j)$ describes, in the current individual, the duration of successor collection B_j of task j of position i on vector λ.

The fitness of each bit reflects the duration of task j and the difference between current scheme and optimal scheme of the predecessor collection and successor collection of task j. A big value of fitness means that the current scheme is farther than the optimal objective of minimum duration, otherwise, a small value means the minimum duration scheme is close to the optimal scheme.

(2) Choose the worst bit and its parallel collection elements in mc gene and mode gene to make exchange mutation.

For an individual encoding by task list, random exchange mutation can produce the non-feasible task list. So, in this paper, after choosing the worst bit as exchange bit according to (1), the task which the bit represents is obtained and parallel task collection of this is calculated. Then, another bit is randomly chosen as the exchange bit from the parallel task collection, basing on that the new solution meets the previous constraints. Finally, do the same exchange operating in the same position of mc gene and mode gene, so that all components of the individual can be cooperative coevolution and individual structure can be continuous improved.

4.3 Steps of modified SFL algorithm for SCPSP

Step 1 Initialize the algorithm parameters, including the population size N, the number of groups M, the number of local iterations It, the initial temperature T, the simulation capability constraint matrix of tasks, the duration matrix of simulation capability, and the supply matrix of simulation capability.

Step 2 Randomly generate initial frog group and calculate the fitness of each frog. To avoid the non-feasible solution, first of all, task gene λ is randomly generated according to the constraint of predecessor collection when initialize the individual. Then simulation capability and the corresponding mode of each task in λ are randomly chosen according to the restriction of simulation capability, and the start time and the end time of each task are arranged suitably. So, Individual fitness is the maximum of the end time.

Step 3 Make a descending order according to the fitness and divide all frogs of the population into M groups, where the i frog is divided to the i%M groups.

Step 4 Make n-th iteration meta-evolution in each group of frogs.

Step 4.1 Calculate and obtain the global optimal solution of the population.

Step 4.2 Calculate and obtain the best solution and the worst solution of each group.

Step 4.3 Evolves the worst solution by the crossover operator mentioned in section 4.1.

Step 4.4 If the solution of Step 4.3 is better than the worst solution, we use it to update the worst solution, else evolve the solution of 4.3 by the differential evolution mutation operator mentioned in section 4.2.

Step 4.5 If the solution of Step 4.4 is better than the worst solution, we use it to update the worst solution, else randomly generate a feasible solution.

Step 5 Shuffle all the groups. After a round of meta-evolution to each group, we reorder and redivide all frogs and calculate the global best solution.

Step 6 Test the stop condition. If it meets the convergence condition of the algorithm, we stop the algorithm, else turn to the step 3.

5 Experimental Analysis

There are two types of simulation capability registered in Cloud Simulation Platform, including modeling capabilities and simulation supporting environment capabilities. The case in our experiment chooses the multi-disciplinary collaborative simulation analysis as demo scenario in the background of development of aerospace complex products X-XX. In this case, the virtual prototype system modeling services including pneumatic, power, control and structure disciplinary, and the multi-disciplinary virtual prototype collaborative simulation supporting capability service are needed. So, this scheduling project contains six tasks $\{J_1, J_2, J_3, J_4, J_5, J_6\}$, and the time sequence logical constraints are shown in Fig. 2.

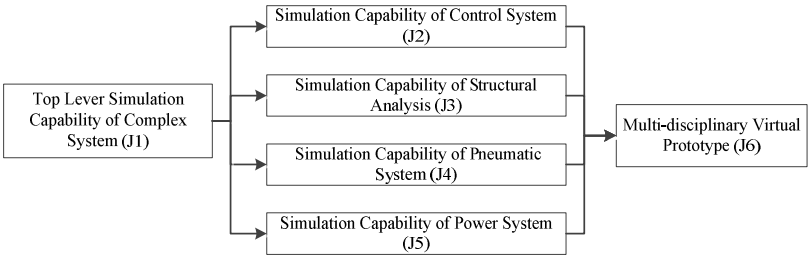


Fig. 2. Design and simulation logic diagram of complex product X-XX

Before the project planning, the candidate capabilities which meet the functional requirements have been chosen automatically by search engine in Cloud Simulation Platform, the task and simulation capability constraints are shown in Table 1.

Table 1. Task and simulation capability constraints

Duration/month	J1	J2	J3	J4	J5	J6
MC1	[4,3]	0	14	0	0	12
MC2	[2,3,4]	0	0	11	0	0
MC3	0	[6,3,4]	0	9	0	11
MC4	0	[5,7]	[20,17]	0	0	0
MC5	0	0	15	0	[1,2,3]	0
MC6	0	0	0	6	0	0
MC7	0	0	0	0	[4,3,6,2]	0
MC8	0	0	16	0	0	0
MC9	0	0	0	14	2	[10,9]
MC10	[5,4,2]	0	0	19	0	0

After 100 iterations, the shortest duration of optimal results is 25 months, and it conforms to the standard solution. A task execution order is [1, 5, 4, 2, 3, 6], and the capability modes and durations in details are shown in Table 2. The convergence speed curves of SFL and modified SFL (MSFL) are shown in Fig.3, and Gantt chart of global optimal planning is shown in Fig. 4.

Table 2. Structure of the optimal and feasible solution

Execution order	J1	J5	J4	J2	J3	J6
Capability number	2	5	3	3	1	9
Mode number	1	2	1	3	1	2
Duration	2	2	9	6	14	9
Start time	0	2	2	2	2	16
Finish time	2	4	11	8	16	25

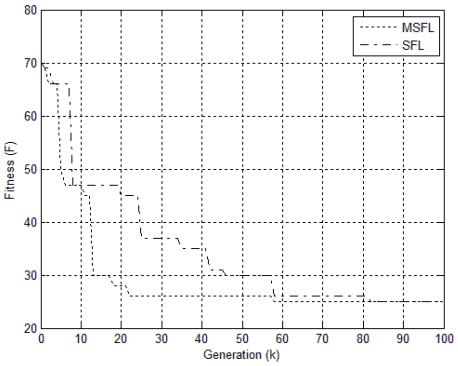


Fig. 3. Convergence speed curves of SFL and MSFL

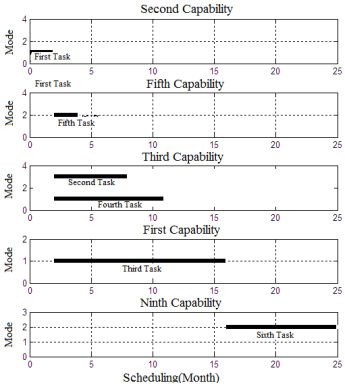


Fig. 4. Gantt chart of task planning

The optimal solution shown in Table 2 is the planning result under the premise of minimum duration of total project. According to the evolution curve in Fig. 3, the MSFL algorithm can converge from the biggest value to 26 which close to the optimal result in 22th generation, and the corresponding simulation capability arrangement for tasks can meet time sequence logical constraint, meanwhile the duration under the optimal capability mode can be minimized. So MSFL proposed in this paper can realize optimization scheduling of multi-mode simulation capability in Cloud Simulation Platform.

In brief, using the random search feature of MSFL, the feasible solution which meets user's objective can be quickly found, and the search speed is accelerated. When user's demand become more and more complex (such as considering project duration and cost balance etc.), we just need to adjust the objective function, which means MSFL has strong commonality, and it is a better method to solve the simulation capability scheduling problem of cloud simulation platform. At the same time, the simulation capability is regarded as a special kind of resources, and its Benchmark is different from PSPLIB which is commonly used by RCPSP. So we need further study on standard problem library of MCPSP.

6 Conclusions

Based on the characteristic analysis of simulation scheduling problem in Cloud Simulation Platform, the mathematical description is given in this paper. Based on this, the principle and the application in the discrete project scheduling problem of shuffled frog leaping (SFL) algorithm has also been analyzed which is an effective intelligent algorithm to solve the NP-hard problem. And a modified SFL (MSFL) is proposed to improve the performance of crossover and mutation operations, which using local search technology, including an improved crossover operator and the differential evolution mutation operator, allowing MSFL have a better search capability. In the end, this paper takes X-XX complex product collaborative design simulation requirements in Cloud Simulation Platform as an example, and the detailed description of the problem is given. The results show that the proposed simulation capability project scheduling model and the solving method based on MSFL have good application value, and can quickly provide users with the appropriate scheduling scheme in engineering projects.

References

1. Li, B., Chai, X., Hou, B., et al.: Networked Modeling & Simulation Platform Based on Concept of Cloud Computing—Cloud Simulation Platform. Journal of System Simulation 21(17), 5292–5299 (2009)
2. Li, B.H., Chai, X., Zhang, L., Lin, T., Yang, C., et al.: New Advances of the Research on Cloud Simulation. In: 2011 Asia Simulation Conference, Seoul, Korea, November 16-18 (2011)
3. Zhang, L., Luo, Y., Tao, F., et al.: Key technologies for the construction of manufacturing cloud. Computer Integrated Manufacturing Systems 16(11), 2510–2520 (2010)
4. Shi, G., Zhu, Y.Q., Li, B.H., et al.: Project Management Technology of Complex Virtual Prototype Engineering. Journal of System Simulation 17(8), 1905–1908 (2005) (in Chinese)
5. Demeulemeester, E., Herroelen, W.: Project Scheduling: A Research Handbook. Kluwer Academic Publishers, New York (2002)
6. Shou, Y.: Model and Method for Resource Constrained Multi-project Scheduling Problem. Zhejiang University Press, Zhe Jiang (2011)
7. Kolisch, R., Hartmann, S.: Experimental investigation of heuristics for resource-constrained project scheduling: An update. European Journal of Operational Research 174(1), 23–27 (2006)
8. Lancaster, J., Ozbayrak, M.: Evolutionary algorithm applied to project scheduling problems: A survey of the state-of-the-art. International Journal of Production Research 45(2), 425–450 (2007)
9. Brucker, P., Drexl, A., Mohring, R., et al.: Resource-constrained project scheduling Notation classification models and methods. European Journal of Operational Research 112, 3–41 (1999)
10. Eusuff, M.M., Lansey, K.: Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resource Plan and Management 129(3), 10–25 (2003)
11. Han, Y., Cai, J., Zhou, G., et al.: Advances in Shuffled Frog Leaping Algorithm. Computer Science 37(7), 16–19 (2010)
12. Emad, E., Tarek, H., Donald, G.: Comparison among five evolutionary-based optimization algorithms. Advanced Engineering Informatics 19, 43–53 (2005)

Improved Shuffled Frog Leaping Algorithm for Solving Multi-aisle Automated Warehouse Scheduling Optimization

Wenqiang Yang, Li Deng, Qun Niu, and Minrui Fei

Shanghai Key Laboratory of Power Station Automation Technology
School of Mechatronics Engineering and Automation
Shanghai University, Shanghai, China
dengli@shu.edu.cn

Abstract. For multi-aisle automated warehouse scheduling optimization problem, a mathematical model with constraints is established, and a new shuffled frog leaping algorithm is proposed. During the process of obtaining optimal solution, to enhance the local search ability, stepsize is adjusted adaptively, and the frog individuals are guided to update. Meanwhile, in order to maintain the diversity of the populations and strengthen the global search ability, heuristic mutation operation is embedded. This not only ensures the global optimization, but also enhances the convergence efficiency. To verify the performance of the proposed algorithm, it is compared with shuffled frog leaping algorithm (SFLA) and genetic algorithm (GA) through simulation combined the industrial real case. Results show that the proposed algorithm achieves good performance in terms of the solution quality and the convergence efficiency.

Keywords: shuffled frog leaping algorithm, adaptive stepsize, diversity of the population, guided update, multi-aisle automated warehouse, scheduling optimization.

1 Introduction

Automated warehouse scheduling optimization problem is a typical NP-Hard problems, which is one of the hardest combinatorial optimization problems[1]. Until now, scholars have made some research: Atoum et al.[2] proposed steady state genetic algorithms to solve the warehouse scheduling problem. In order to enhance order picking process performance, Yu et al.[3] divided warehouse into pick area and reserve area. For a given sequence to retrieve operations, Nishi et al.[4] proposed a beam search method to minimize the number of re-handling operations. Under the aim of the least of the tardiness in the product delivery, Chan[5] proposed a tabu sample-sort simulated annealing (TSSA) algorithm for the warehouse-scheduling problem. However, we find that most warehouse systems from the researches above have characteristics of the input-output at only one end of every aisle. In fact, there exists the case that the input-output at two ends of every aisles. Scheduling

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 82–92, 2013.

optimization problem of such warehouse system is studied in this paper, which increases efficiency of storage to a large extent.

Warehouse scheduling problem with various constraints is a complex combinatorial optimization problem in this paper, in which computational complexity increases exponentially with problem scale. It can lead to a combinatorial explosion. Therefore, accurate algorithms are difficult to solve the optimizations in a limited time[6]. SFLA was proposed in 2003 by Eusuff et al.[7], which simulates the foraging behavior of the frog and is a novel optimization algorithm based on swarm intelligence. It has been already applied to many fields[8-13]. However, only a few papers have reported using SFLA in warehouse scheduling optimization problems. Additionally, random stepsize can not guide the frog to do local rapid search and oneness on the population reduce global search capabilities of SFLA, which directly influence performance of SFLA. Therefore, we present three improvements on SFLA for its shortcomings. Firstly, the direction, in which the worst frog individuals of sub-populations are updated, is selected according to improvement rate. Secondly, stepsize is adjusted automatically based on the times of subpopulation iteration which has happened. Thirdly, heuristic mutation is used to improve the population diversity, which reduces the chance of falling into local optimum. In the end, the simulation results show the feasibility and validity of the proposed algorithm.

2 Problem Description and Modeling

2.1 Problem Description

Automated warehouse system of certain manufacturing enterprise is taken as research objective, which is composed of five attributes: stacker, rails, racks, pipeline and buffers. It deals with some processes of products. Therefore, the performance of scheduling affects the production efficiency. The layout of this system is shown in Fig.1.

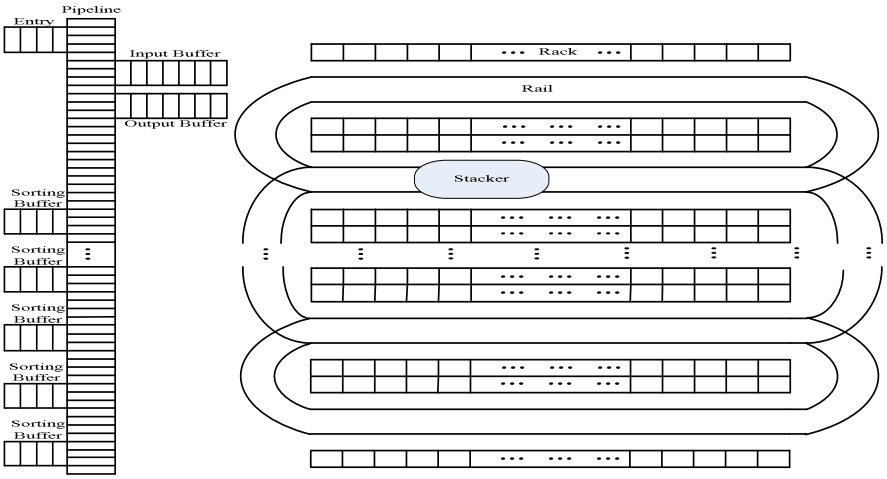


Fig. 1. The layout of the warehouse system

As shown in Fig.1, the stacker can go in and out at both ends of every lane, which is different from the stacker does at only one end of every lanes. In this case, it is worth studying the subject how to improve the input-output efficiency of automated warehouse.

Certain tasks that products are put into storage and retrieved from storage possess the following characteristics.

(I) The number of the storage tasks is m, the assigned locations are represented respectively as ($p_{i1}, p_{i2}, \cdots, p_{im}$);

(II) The number of the retrieval tasks is n, the assigned locations are represented respectively as ($p_{o1}, p_{o2}, \cdots, p_{on}$).

In general, m is not equal to n. Assume that the capacity of the stacker is N. For the tasks which contain m input operations and n output operations, the stacker will operates back and forth $\max\{\lceil m/N \rceil, \lceil n/N \rceil\}$ times from input buffer to output buffer. The objective of automated warehouse scheduling optimization is to select $\max\{\lceil m/N \rceil, \lceil n/N \rceil\}$ optimal routes from all the possible routes.

For the sake of simplicity, startup time and braking time of the stacker are neglected. Horizontal speed and vertical speed, which are mutual independent, are denoted as v_x, v_y respectively. The distance of adjacent lanes is D_l. The number of columns is C for each lane. Meanwhile, L, W and H are corresponding to length, width and height respectively of each storage or retrieval location. Moreover, we suppose that the three-dimensional coordinate of the input or output buffer is $p_o(0, 0, 0)$.

Definition 1. If the sub-route r belongs to one of the routes which the stacker complete the tasks, s_r =1; Otherwise, s_r =0. Similarly, if the location p_i belongs to sub-route r, g_{ir} =1; Otherwise, g_{ir} =0.

Definition 2. For storage and retrieval locations. If the locations $p_i(x_i, y_i, z_i)$ and $p_j(x_j, y_j, z_j)$ are consecutively accessed in the process of the task, e_{ij} =1; Otherwise, e_{ij} =0. The corresponding time t_{ij} that the stacker spends can be expressed as

$$t_{ij} = \begin{cases} \max\{(W \times |x_i - x_j|)/v_x, (H \times |y_i - y_j|)/v_y\} & if \quad z_i = z_j \\ \max\{\min\{(W \times (x_i + x_j) + D_l \times |z_i - z_j|)/v_x, \\ (W \times ((C - x_i) + (C - x_j)) + D_l \times |z_i - z_j|)/v_x\}, (H \times |y_i - y_j|)/v_y\} & if \quad z_i \neq z_j \end{cases} \tag{1}$$

2.2 Mathematical Modeling

A mathematical model is developed in order to formalize the warehouse scheduling problem. Suppose that warehouse capacity is Q and current inventory quantity is

q. Under the objective to minimize the makespan, the mathematical model can be established as

$$\min f(e) = \sum_{r=1}^{\max\{\lceil m/N \rceil, \lceil n/N \rceil\}} \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} t_{ij} \cdot e_{ij} \cdot g_{ir} \cdot g_{jr} \tag{2}$$

s.t.

$$\sum_{r=1}^{\max\{\lceil m/N \rceil, \lceil n/N \rceil\}} s_r = \max\{\lceil m/N \rceil, \lceil n/N \rceil\} \tag{3}$$

$$\sum_{i=1}^{m+n} \sum_{r=1}^{\max\{\lceil m/N \rceil, \lceil n/N \rceil\}} g_{ir} \cdot s_r = m+n \tag{4}$$

$$\sum_{r=1}^{\max\{\lceil m/N \rceil, \lceil n/N \rceil\}} g_{ir} = 1 \quad \forall i \in \{1, 2, \cdots, m+n\} \tag{5}$$

$$\sum_{i=1}^{m+n} g_{ir} \le 2N \quad \forall r \in \{1, 2, \cdots, \max\{\lceil m/N \rceil, \lceil n/N \rceil\}\} \tag{6}$$

$$\sum_{j=1}^{m+n} e_{0j} = \lceil m/N \rceil \tag{7}$$

$$\sum_{i=1}^{m+n} e_{i0} = \lceil n/N \rceil \tag{8}$$

$$p_u \in p_{oj} \quad \forall r(p_i, p_j, \cdots, p_u) \tag{9}$$

$$q + m + n \le Q \tag{10}$$

Eq.(2) is the objective to be optimized. Constraints that are from eq.(3) to eq.(10). Eq.(3) restricts the execution times of the stacker to complete the tasks. Eq.(4) represents the quantity of tasks which stacker only can do. Eq.(5) makes sure that each task is only contained in one route. Eq.(6) indicates that the stacker must not exceed its capacity. Eq.(7) and eq.(8) ensure that starting and ending position are designated. Eq.(9) demands that the input tasks must be fulfilled first. Eq.(10) makes sure that the overload of the warehouse can't happen.

3 Improved Shuffled Frog Leaping Algorithm (ISFLA)

SFLA is a new heuristic computing technology based on swarm intelligence, which has been widely used to optimize continuous systems because of some advantages of strong global search ability and less parameter. However, in this paper, we investigate the problem about optimization of discrete event. The application of SFLA, therefore, is subjected to some limit due to it. In addition, there are also some disadvantages as mentioned before. For these reasons, we make improvements to it, as described next.

3.1 Construct Sub-populations of Frogs

The position of one frog is expressed as $U(i)=(u_1,u_2,\cdots,u_n)$, of which u_i represents storage or retrieval task. The initial population which contains N_t frogs is randomly generated. The N_t frogs are then sorted by fitness $f(U(i))$ in descending order. Namely, $U=\{U(1),U(2),\cdots,U(N_t)\}$. Determine the number of the sub-populations N_m and the number of the frogs which belong to the sub-population N_n, which must meet the condition $N_t=N_m\times N_n$. The affiliation of the frog $U(i)$ and the sub-population Y_i is described as

$$Y_{ij}=U(j+N_n\times(i-1)) \qquad i\in(1,2,\cdots,N_m),\ j\in(1,2,\cdots,N_m) \tag{11}$$

3.2 Strategy of Updating the Frog Individual

In this paper, for two frogs, the positions of which are $U(i)$ and $U(j)$, respectively, if $U(i)\neq U(j)$, for the frog *No.i*, to reach the position $U(j)$, the only way that u_p and u_q ($u_p,u_q\in U(i)\&\&p\neq q$) are swapped among $U(i)$. Here, we denote the operation of swapping u_p and u_q by $s(u_p,u_q)$. In the sequel, the times of the operations of swapping are defined as the distance between two frogs $U(i)$ and $U(j)$, says $D(U(i),U(j))$, and the jump from $U(i)$ to $U(j)$, says $U(i)\rightarrow U(j)$.

$$D(U(i),U(j))=card(\{s(u_p,u_q)\,|\,U(i)\rightarrow U(j),u_p,u_q\in U(i)\}) \tag{12}$$

Hence, we can obtain

$$U(j)=U(i)\oplus D(i,j) \tag{13}$$

Where symbol "$\oplus$" represents the jump operation of the frog.

Notice that there are some different jump ways from $U(i)$ to $U(j)$. Such as $U(1)$ =(1,6,3,2,4,5), $U(2)$ =(3,4,1,5,6,2). If $U(1)=U(2)$ is realized, the frog $U(1)$ will experience six ways of jump alternatively as shown in Fig.2.

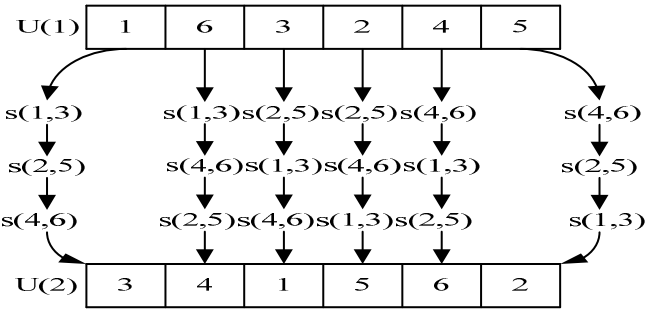


Fig. 2. Six alternative ways of jump from $U(1)$ to $U(2)$ of the frog *No.*1

Fig.2 shows that the frog *No.*1 jumps from $U(1)$ to $U(2)$ in different directions. Moreover, the same step of every direction may have different improvement rates in fitness. The largest improvement rate can guide the frog to a better position. Here, $\overset{k}{U}(i)$, $f(\overset{k}{U}(i))$ stand for the position and the fitness after the k-nd jump of the frog *No.i*, respectively, evaluation of the improvement rate $R(i,k)$ is computed as follows:

$$R(i,k) = ((f\left(\overset{k}{U}(i)\right) - f\left(\overset{k-1}{U}(i)\right)) \Big/ D\left(\overset{k}{U}(i), \overset{k-1}{U}(i)\right)) \times 100\% \tag{14}$$

Finally, when the frog *No.i* jumps from step to step, computes $R(i,k)$ of every direction and selects $s(u_p, u_q)$ which has the largest $R(i,k)$ as the next step of jumping and ensures the new position of the frog *No.i* meet the constraints. Otherwise, select $s(u_p, u_q)$ with a larger $R(i,k)$. Even worse, none of the $s(u_p, u_q)$ exists, and then produces a new position of the frog *No.i* which satisfies the constraints.

We consider that a larger step is beneficial to global optimization in the previous evolution, and a smaller step is good for local optimization in the later evolution. In order to raise the ability of SFLA to search and improve its convergence property, an adaptive adjustment strategy of stepsize is presented. We denote that the number of times local search has been executed by N_l, the worst frog and the best frog of the sub-population by U_w and U_b, and the best frog of the population by U_g. Only U_w is updated, after which it is denoted by U_w^d. The adaptive step $L(\cdot)$ and update strategy is defined as follows:

$$L(N_l) = \left\lfloor (1/N_l)^{\alpha} \cdot D(U_b - U_w) \right\rfloor \tag{15}$$

$$L(N_l) = \left\lfloor (1/N_l)^{\alpha} \cdot D(U_g - U_w) \right\rfloor \tag{16}$$

$$U_w^d = U_w + L(N_l) \tag{17}$$

Where we first update U_w with eq.(15) and eq.(17), if $f(U_w^d) < f(U_w)$, we do with eq.(16) and eq.(17), if not yet, produce a random position satisfying constraints.

3.3 Maintaining of Population Diversity

Taking the relation between diversity of population and optimal solution into account, a heuristic mutation strategy is adopted.

Definition 3. A population $U = \{U(1), U(2), \cdots, U(N_t)\}$ consists of N_t frogs, of which $U(j)$ is the optimal frog, the ratio of the frog's contribution to diversity of population is defined as

$$\rho_i = D(U(i), U(j)) \Big/ \sum_{k \in (1,2,\cdots,N_t) \&\& k \neq j} D(U(k), U(j)) \tag{18}$$

From eq.(18) can also be seen, the smaller ρ_i, the closer $U(i)$ is to $U(j)$. In order to keep population diversity and avoid falling into local optimum, mutation probability of the frog $No.i$ is inversely proportional with ρ_i. At the same time, the strategy of keeping a half of optimal frogs is utilized to ensure the stable convergence. Mutation probability p_i is normalized as

$$p_i = (1/\rho_i) \Big/ \sum_{k=1}^{N_t} (1/\rho_k) \qquad i \in (1, 2, \cdots, N_t/2) \tag{19}$$

3.4 Representation of ISFLA

Suppose that the number of the frogs by N_t, the number of the sub-populations by N_m, the number of the frogs which belong to the sub-population by N_n, the number of iterations of sub-population by N_{si}, the number of iterations of population by N_{ti}.

Step 1. Initialize parameters: N_t, N_m, N_n, N_{si}, N_{ti}, and produce N_t frogs which meet the constraints;

Step 2. Global search: $n = n + 1$;

Step 3. Calculate the fitness of N_t frogs, and the fitness is measured by the reciprocal of eq.(2);

Step 4. Sort N_t frogs by fitness $f(i)$ in descending order;

Step 5. Construct N_m sub-populations of frogs according to section 3.1;

Step 6. Local search: $l = l + 1$;

Step 7. Update the worst frog individual of every sub-population according to section 3.2;

Step 8. If $l \le N_{si}$, then go to step6;

Step 9. If $n \le N_{ti}$, improve population diversity according to section 3.3, and then go to step2; Otherwise, output the optimal solution.

4 Numerical Simulation and Analysis

In order to validate the performance of the algorithm proposed, the examples of automated warehouse scheduling problems of some enterprise are given, where some comparisons are made with SFLA[14] and GA[15]. In addition, simulations are made under the same condition, such as windows XP system, 2.19GHz processor, 1.99GB of RAM and development environment Microsoft Visual C++ 6.0. Population size and the number of iterations of all algorithms are 80 and 600, respectively. Moreover, for SFLA and ISFLA, N_{si} =20. For GA, crossover probability and mutation probability are set 0.9, 0.05, respectively. For warehouse system, L =30cm, W =50cm, H =40cm, D_l =180cm, C =75, N =5, q =497400, Q =507000, v_x =1m/s, v_y =0.5m/s.

The performance of algorithms can be evaluated from three aspects: the quality of the optimal solution, convergence speed and the diversity of population.

Suppose that each task is denoted as (x,y,z,l,f), of which x, y are is the coordinate of storage or retrieval location, z is the lane number, l marks the location being on the left or right, l=1 the left, l=2 the right, f indicates whether the location is for storage or retrieval, f=1 storage, f=2 retrieval. 50 tasks are as follows: (31,1,5,2,1), (21,13,14,1,2), (55,1,10,1,1), (1,6,4,2,2), (31,2,7,1,1), (2,5,4,1,2), (24,20,7,1,1), (5,6,4,1,2), (11,3,5,2,1), (11,6,4,2,1), (18,9,7,1,2), (44,7,5,2,1), (32,15,8,1,2), (10,36,7,1,1), (8,3,1,1,2), (63,14,9,2,2), (16,21,3,1,2), (51,10,8,2,1), (35,21,11,1,2), (23,3,24,2,1), (41,36,3,1,1), (45,6,4,1,2), (11,14,6,2,1), (15,26,2,2,2), (67,3,48,1,2), (41,6,3,1,1), (33,10,52,1,2), (31,61,12,2,2), (17,9,9,2,1), (4,3,2,1,1), (10,7,14,1,2), (23,3,18,1,2), (51,22,31,2,1),(22,44,2,2,2), (21,23,3,1,1), (12,4,25,2,1), (31,1,5,1,2), (57,24,14,2,1), (62,11,30,1,2),(45,6,7,1,1), (29,5,13,2,2), (71,7,10,2,1), (32,8,8,2,1), (33,15,18,1,2), (43,13,8,2,1), (17,1,16,2,1), (3,15,14,1,2), (39,25,18,2,1), (3,35,48,2,2), (13,22,3,1,2).

Considering the influence of the parameters N_m and α on the optimal performance, many experiments are made under different parameter combinations, which are shown in Fig.3. It is clearly observed that the objective value decreases firstly and then increases along with α growing for any given N_m, but the objective value changes irregularly for any given α. In addition, as can be seen that a minimum value is achieved at N_m=8 and α=0.8. Under such conditions, evolution comparison among ISFLA, SFLA and GA is done, as shown in Fig.4. Simultaneously, to offer more intuitionist information of population diversity during the evolutionary, Fig.5 shows initial population distribution and 110th-generation population distributions.

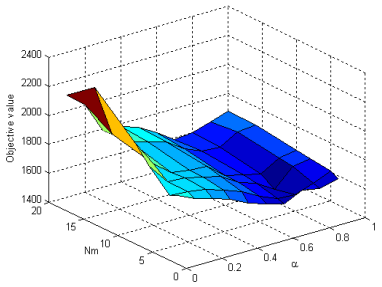


Fig. 3. Effect of parameters on performance

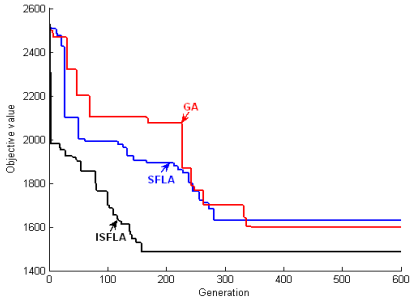
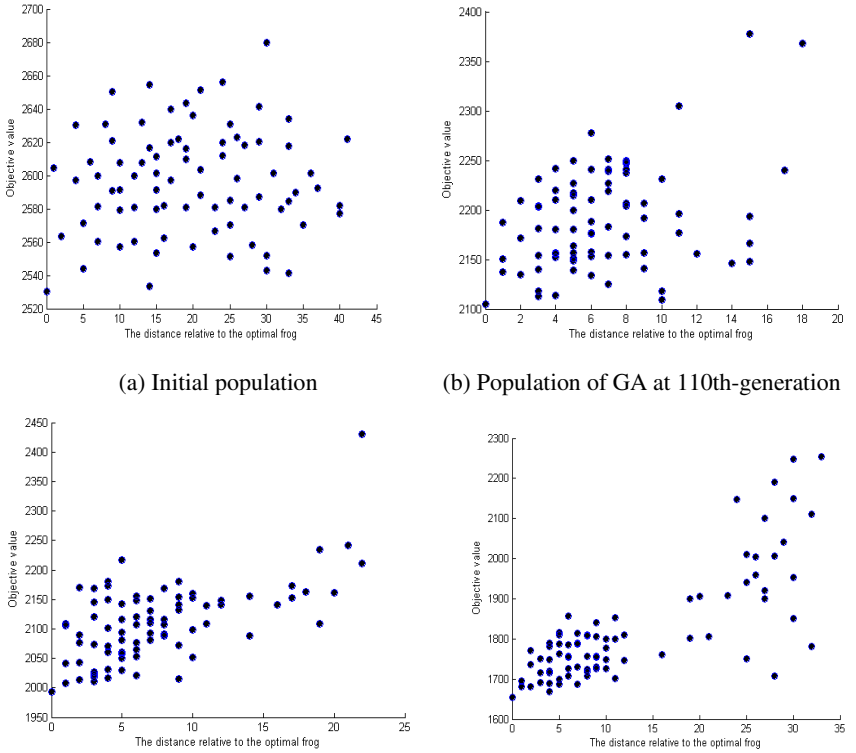


Fig. 4. Evolution comparison between algorithms



(a) Initial population (b) Population of GA at 110th-generation

(c) Population of SFLA at 110th-generation (d) Population of ISFLA at 110th-generation

Fig. 5. The distribution of the population

In order to test the performance of ISFLA which is relative to other two on different sizes of tasks, benchmark instances from TSPLIB are adopted. Each algorithm runs thirty times, results are presented in Table 1, in which Opt_value, Avg_value and Avgcon_time indicate optimal value, average value, and average convergence time respectively.

Table 1. Performance comparison among algorithms

Problem	Algorithm	Opt_value	Avg_value	Avgcon_time
Burma14	GA	30.88	33.96	3.05
	SFLA	30.88	35.62	2.14
	ISFLA	30.88	31.75	1.69
Oliver30	GA	435.30	447.24	8.27
	SFLA	441.18	461.09	7.01
	ISFLA	423.74	425.48	3.13
Eil51	GA	456.39	479.77	18.86
	SFLA	478.69	506.92	14.84
	ISFLA	426.00	430.51	8.37

As shown in Fig.4 and Fig.5, compared with SFLA and GA, ISFLA shows a better performance on optimal solution, convergent speed and population's diversity. Furthermore, SFLA and GA trap into local optima easily. GA, by using fixed mutation, can reduce the chance of falling into local minimum to some extent. Nevertheless, the lack of optimum-oriented destroys population diversity. For ISFLA, improvement strategy of population diversity presented in this paper improves population diversity, providing a larger solution space for optimal searching, which provides advantages for obtaining a globally optimal solution or a sub-optimal solution. Furthermore, guided updating the frog individuals and adaptive stepsize are adopted during local search, which greatly enhance the convergence speed. In addition, as seen in Table 1, the performance difference among algorithms is a little when the size of problems is smaller; however, the performance of ISFLA is much better than that of the other two when the size of problems is larger. It further demonstrates that ISFLA has strong global and local search ability, especially in a larger size.

5 Conclusion

To optimize the scheduling of a multi-aisle automated warehouse, this paper presents an ISFLA, which improves the capability of local search by updating the worst frogs with the guidance and adjusting the stepsize adaptively, ensures the diversity of the population, strengthens the global search ability and restrains the prematurity by applying the heuristic mutation. Compared with the experimental results of SFLA and GA, ISFLA has a better performance, especially for a larger size of tasks. In future, how the population size and sub-population size influence the performance of ISFLA will be studied to further improve the performance of ISFLA. Thus ISFLA can be better applied to the scheduling problems of the related fields such as intelligent manufacture, logistic, airline and vehicle.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant No.61074032 and No.61273040, and the Project of Science and Technology Commission of Shanghai Municipality under Grant No.10JC1405000, and Shanghai Rising-Star Program under Grant No.12QA1401100.

References

1. Gagliardi, J.P., Renaud, J., Ruiz, A.: Models for automated storage and retrieval systems: a literature review. International Journal of Production Research 50(24), 7110–7125 (2012)
2. Atoum, J., Rababaa, M.A.: Multiple Warehouses Scheduling Using Steady State Genetic Algorithms. The International Arab Journal of Information Technology 7(3), 310–316 (2010)
3. Yu, M.F., Koster, R.D.: Enhancing performance in order picking processes by dynamic storage systems. International Journal of Production Research (48), 4785-4806 (2010)

4. Nishi, T., Konishi, M.: An optimisation model and its effective beam search heuristics for floor-storage warehousing systems. International Journal of Production Research (48), 1947-1966 (2010)
5. Chan, F.T.S., Kumar, V.: Hybrid TSSA algorithm-based approach to solve warehouse-scheduling problems. International Journal of Production Research 47(4), 919–940 (2009)
6. De Koster, R.B.M., Le Duc, T., Yugang, Y.: Optimal storage rack design for a 3-dimensional compact AS/RS. International Journal of Production Research 46(6), 1495–1514 (2008)
7. Eusuff, M., Lansey, K.: Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm. Journal of Water Resources Planning and Management 129(3), 210–225 (2003)
8. Gao, H.Y., Cao, J.L.: Membrane-inspired quantum shuffled frog leaping algorithm for spectrum allocation. Journal of Systems Engineering and Electronics 23(5), 679–688 (2012)
9. Bijami, E., Askari, J., Farsangi, M.M.: Design of stabilising signals for power system damping using generalised predictive control optimised by a new hybrid shuffled frog leaping algorithm. IET Generation, Transmission and Distribution 6(10), 1036–1045 (2012)
10. Wang, N., Li, X., Chen, X.H.: Fast three-dimensional Otsu thresholding with shuffled frog-leaping algorithm. Pattern Recognition Letters 31(13), 1809–1815 (2010)
11. Jafari, A., Bijami, E., Bana, H.R., et al.: A design automation system for CMOS analog integrated circuits using New Hybrid Shuffled Frog Leaping Algorithm. Microelectronics Journal 43(11), 908–915 (2012)
12. Park, B., Lee, J.: Optimization of coordinated-actuated traffic signal system: Stochastic optimization method based on shuffled frog-leaping algorithm. Transportation Research Record (2128), 76-85 (2009)
13. Pakravesh, H., Shojaei, A.: Optimization of industrial CSTR for vinyl acetate polymerization using novel shuffled frog leaping based hybrid algorithms and dynamic modeling. Computers and Chemical Engineering 35(11), 2351–2365 (2011)
14. Eusuff, M.M., Lansey, K.E.: Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resources Planning and Management 129(3), 210–225 (2003)
15. Hsu, C.M., Chen, K.Y., Chen, M.C.: Batching orders in warehouses by minimizing travel distance with genetic algorithms. Computers in Industry 56(2), 169–178 (2005)

Solving Battalion Rescheduling Problem Using Multi-objective Genetic Algorithms

Irfan Younas[1], Farzad Kamrani[2], Farshad Moradi[2], Rassul Ayani[1], Johan Schubert[2], and Anne Håkansson[1]

[1] KTH Royal Institute of Technology, Stockholm, Sweden
{irfany,ayani,annehak}@kth.se
[2] Swedish Defence Research Agency, Stockholm, Sweden
kamrani@kth.se, {farshad.moradi,johan.schubert}@foi.se

Abstract. In this paper, we consider the problem of rescheduling human resources in a battalion where new activities are assigned to the battalion by higher headquarters, requiring modification of an existing original schedule. The problem is modeled as a multi-criteria optimization problem with three objectives: (i) maximizing the number of tasks that are performed, (ii) minimizing the number of high-priority tasks that are missed, and (iii) minimizing the differences between the original schedule and the modified one. In order to solve the optimization model, we adopt Non-dominated Sorting Genetic Algorithm-II (NSGA-II). The accuracy of NSGA-II in this context is verified by considering a small-sized problem where it is easy to verify solutions. Furthermore, we consider a realistic problem instance for a battalion with 400 agents and 66 tasks in the initial schedule. We present the computational results of rescheduling when unpredictable activities emerge.

Keywords: Battalion rescheduling, Multi-objective optimization, Genetic algorithms.

1 Introduction

Similar to development in many countries, the Swedish Armed Forces (SAF) has started undergoing a major process of change over the past few years. These changes embrace transformation from an invasion defense to a mission oriented defense. One major consequence of this transformation is that the SAF has become drastically smaller and vulnerable to personnel shortage. The strictly hierarchical structure and closed nature of military organizations do not allow recruitment of new (temporary) personnel with right competencies if an urgent need arises and new tasks are assigned to the organization. Thus, military units are periodically forced to handle an essentially larger number of tasks with the same manpower resources, which requires an efficient utilization of these resources and calls for more flexible and effective planning of personnel, and scheduling of tasks.

A battalion is a military unit with 400 to 1200 soldiers, which is considered to be the smallest unit capable of independent operations. In the SAF, the

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 93–104, 2013.

battalion commander and staff (from hereon referred to as the commander) are responsible for planning and scheduling different activities of the battalion personnel such as education, training, exercises, tasks and missions. All activities are pre-planned and scheduled on a yearly basis by the commander in a way that all tasks and missions are performed and at the same time the personnel attend necessary educational and training sessions and exercises to achieve the required competencies for performing those tasks or missions. Considering that activities require different number of personnel with different types and levels of competencies, scheduling all activities per se is a difficult and complex problem. Consequently, this problem becomes even more complex and challenging since the set of workforce competencies is not static and changes over time as the personnel attend military education, training or exercises. A typical scenario is where personnel perform different types of tasks and later take part in different educational activities to acquire the right competencies needed for more complex tasks and missions scheduled to be performed after the courses. As a result, the schedule for this kind of scenario is sensitive to changes and potentially vulnerable to disturbances.

Nevertheless, in reality unpredictable activities emerge periodically, as a result of unforeseen events and are assigned to the battalion by higher headquarters. The commander then may be forced to do some rescheduling by removing personnel from scheduled tasks, such as educational activities in order to be able to deal with imminent tasks. Assigning personnel to these tasks compromises the commitment of the battalion to perform important and prioritized tasks or missions, since some personnel may miss educational activities and do not have the required competencies for forthcoming tasks and missions.

While doing the rescheduling the commander faces conflicting objectives such as maximizing the total number of performed tasks, prioritizing more critical tasks and missions, and preserving the original schedule to the largest extent. These objectives partly reflect the interests of different stakeholders. For instance, while the commander of the battalion wants to perform as many tasks as possible, and prioritizes the critical tasks, the most important criterion for the personnel is that their schedules are changed as little as possible.

In this paper, we propose a novel model for rescheduling different activities of the battalions personnel. We formulate the rescheduling problem as a multi-objective optimization problem, in which we consider three objectives: (i) maximizing the total number of performed tasks, (ii) minimizing the number of high priority tasks that are missed, and (iii) minimizing the differences between the initial schedule and updated schedule. A multi-objective mathematical model, with three conflicting objectives and a set of constraints is built. In multi-objective optimization problems with conflicting objectives, the goal is to find a set of Pareto-optimal solutions. Multi Objective Evolutionary Algorithms (MOEAs) are well-known metaheuristics for sampling intractably large and highly complex search spaces. MOEAs search more than one solutions in parallel and are suitable for finding the Pareto-optimal set for multi-objective optimization problems. MOEAs maintain population of non-dominated set of

individuals and comparison of two individual solutions is based on Pareto-dominance. A non-dominated solution is one in which it is impossible to improve an objective without worsening at least one other objective. In the last fifteen years, several multi-objective evolutionary algorithms have been proposed. In order to solve our proposed multi-objective problem, we adopt Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [2], which is a well-known evolutionary algorithm. To test the accuracy of NSGA-II for our problem, we consider a small problem. We further conduct experiments on a large scale problem and present computational results.

The rest of the paper is organized as follows: in Section 2, related work is briefly reviewed. In Section 3, a mathematical multi-objective model for battalion rescheduling is presented. In Section 4, the multi-objective algorithm is described. Section 5 presents the experiments, including problem instances, parametrization and discussion of results. Finally in Section 5, the conclusion and summary of the paper are outlined.

2 Related Work

To the best of our knowledge, existing work does not address multi-objective optimization rescheduling problems similar to the battalion rescheduling model discussed in this paper. Even though we find some related work on scheduling and rescheduling problems, it is different in several aspects. Clark and Walker [3] present some models and computational results for nurse scheduling and rescheduling considering nurses' preferences. Due to changes in nursing cover requirements or unavailability of nurses to work their assigned shifts, rescheduling is needed to fill the gaps in the initial schedule. The overall goal is to reschedule nurses while considering their preferences in a manner that disrupts as little as possible the existing schedule. In nurse rescheduling, the aim is to minimize changes to the initial schedule as well as minimizing the total cost.

Moz and Pato have proposed different techniques for solving nurse rescheduling problems with the objective to minimize the changes in the original schedule. Moz and Pato [4] use a genetic algorithm to solve the problem and introduce a second objective into the fitness function. The second objective is to minimize the overtime, which they describe as minimizing the difference between the number of scheduled duties and the number of performed duties.

Maenhout and Vanhoucke [5] propose an evolutionary metaheuristic to solve nurse rerostering problem. Personnel roster determines the line-of-work for each person [5]. Dynamic nature of operating environment can cause unexpected events, which in turn lead to infeasibilities and schedule disruptions. In order to cope with this situation rescheduling is necessary. Computational experiments are performed on a well designed data set and the results of the proposed method are compared with other already existing methods in literature. Chicano et al. [6] focus on the Software Project Scheduling (SPS) [7]. Resources with a set of skills are assigned to the tasks, and the objectives of the problem are to minimize the time and cost of the project. The authors compare performance of different multi-objective evolutionary algorithms in solving the proposed SPS model. Hao and

Lin [8] study evolutionary algorithms to solve multi-objective model of the job shop rescheduling problem. The model considers n jobs to be performed on m machines and the objectives of the problem are to minimize the tardiness and makespan. The rescheduling is needed in case of new job arrivals and machine breakdowns.

The nurse rescheduling problem has some similarity with our problem in one of the objectives, which is to minimize schedule disruptions. However, the battalion rescheduling problem is different in several ways. Firstly, in our model, tasks require different number of personnel with different types and levels of competencies. Secondly, new tasks with different set of requirements can be assigned to the battalion during any time of the year. Thirdly, the set of workforce competencies is not static and changes over time as the personnel attend military education, training or exercises. These characteristics make the battalion rescheduling problem more complex and challenging than the nurse rescheduling problem. Job shop rescheduling problem is also similar to our problem in some aspects but dynamic nature of the set of workforce competencies in our problem differentiates it from the job shop rescheduling problem.

3 Problem Formulation and Model Description

We model the battalion rescheduling problem as a multi-objective optimization problem. Battalion personnel are modeled as the set of agents $A = \{a_i \mid i = 1, \ldots, m\}$, where m is the number of personnel in the battalion (agents). Assuming that q different types of competencies $C = \{c_1, \ldots, c_q\}$ are relevant for the function of the battalion, each agent $a_i \in A$ has a set of numerical attributes $c_i = \{c_{il} \mid l = 1, \ldots, q\}$, where $c_{il} \in \{0, \ldots, 4\}$ specifies to what extent agent a_i is equipped with competency c_l. The competencies of all agents can conveniently be denoted by matrix $C = [c_{il}]_{\{m \times q\}}$.

All activities (tasks, missions, military education, training, exercises) are modeled as the set $T = \{t_j \mid j = 1, \ldots, n\}$, where n is the number of activities. To each task $t_j \in T$, a 4−tuple $(size_j, start_j, duration_j, priority_j)$ is associated, where $size_j$ is the number of agents required to perform the task, $start_j$ is the start date of the task, $duration_j$ is the duration of the task in days, and $priority_j$ is the priority of the task with the binary values *low* and *high*.

Each task t_j consists of a set of positions. We enumerate and index all positions in a sequence from the task with the smallest index to the largest and use the notation $\overline{j} = \sum_{j'=1}^{j} size_{j'}$ for the sum of the number of positions in all tasks t_1, to t_j. Using this notation task t_j consists of positions $\{p_k \mid j = \overline{(j-1)}+1, \ldots, \overline{j}\}$. The set of all position is denoted by $P = \{p_k \mid k = 1, \ldots, \overline{n}\}$. We also define the function $tasknumber : \{1, \ldots, \overline{n}\} \curvearrowright \{1, \ldots, n\}$, which gives the index of task t_j that p_k belongs to (i.e. , $tasknumber(k) = j$).

Each position p_k has competency requirements $w_k = \{w_{lk} \mid l = 1, \ldots, q\}$, where w_{lk}, defines the minimum level of competency type c_l required by position p_k, meaning that agent a_i may be assigned to position p_k only if $\forall l, c_{il} \geq w_{lk}$. The competencies required by all positions can conveniently be denoted by matrix

$W = [w_{lk}]_{\{q \times \overline{n}\}}$. After completing some tasks, one or several competencies of agents may increase one level. These tasks model military education courses, training and exercises. In other words, courses are modeled as ordinary tasks and are distinguishes from them only by being associated with an update function ($update : w_l \curvearrowright w'_l, l = 1, \ldots, q$), which updates competencies of all agents a_i participating in the course. In this study, the update function is defined as $w'_l = w_l + 1$ for some specified values of l, depending on the course.

If all resources (agents) required by a task are available, the task will be assigned to agents, otherwise the task remains unassigned. An agent can be assigned to at most one task at any time.

A schedule is defined by matrix $X = [x_{ik}]_{m \times \overline{n}}$, where $x_{ik} \in \{0, 1\}$ and $x_{ik} = 1$ if position p_k is assigned to agent a_i, and $x_{ik} = 0$ otherwise.

A schedule X is feasible subject to the following constraints:

$$\forall k, \sum_{i=1}^{m} x_{ik} \in \{0, 1\}, \tag{1}$$

$$\forall j, \sum_{k=\overline{j-1}+1}^{\overline{j}} x_{ik}, \in \{0, size_j\} \tag{2}$$

$$\forall k, x_{ik} = 1 \Rightarrow \forall l,\ c_{il} \geq w_{lk}, \tag{3}$$

$$\begin{array}{l} \forall i, x_{ik} = 1 \wedge x_{ik'} = 1 \Rightarrow \\ [start_j, start_j + duration_j) \cap [start_{j'}, start_{j'} + duration_{j'}) = \phi, \\ where\ j = tasknumber(k)\ and\ j' = tasknumber(k'). \end{array} \tag{4}$$

Constraint 1 implies that only one agent is assigned to a position, constraint 2 ensures that tasks are either assigned to all required agents or not performed at all, constraint 3 states that all agents should have all qualifications required by the task and constraint 4 asserts that agents are assigned to only one task at any time.

Given agents A, initial tasks T^0, positions P^0 with requirements W^0 and an initial schedule $X^0 = [x^0_{ik}]_{m \times \overline{n^0}}$, which assigns tasks to agents, a set of new tasks T^1 are arrived at a given time ($rescheduling\_time$). The battalion rescheduling problem is defined as finding a new assigning schedule $X^1 = [x^1_{ik}]_{m \times \overline{n^1}}$ for all tasks, which have not already started such that some optimization criteria are fulfilled, subject to constraints 1 to 4. Tasks that have not started at $rescheduling\_time$ are specified by $T^1 \cup T^0 \setminus \{t_j \mid j = 1, \ldots, n, \wedge start_j < rescheduling\_time\}$.

In this paper, we focus on three criteria, τ, μ and δ, defined as:

$$\tau = |\{t_j \in T^0 \cup T^1 \mid \exists x_{ik} \in X^1\ s.t.\ tasknumber(k) = j \wedge x_{ik} = 1\}| \tag{5}$$

$$\mu = |\{t_j \mid t_j \notin \tau \wedge priority_j = high\}| \tag{6}$$

$$\delta = \sum_{i=1}^{m} \sum_{k=1}^{\overline{n^1}} (x_{ik}^1 - x_{ik}^0) + \sum_{k=1}^{\overline{n^0}} x_{ik}^1. \tag{7}$$

The value τ is the number of tasks performed, μ is the number of high-priority tasks that are not performed, and δ is the difference between the original schedule the new schedule expressed by using the number of agents that have their schedule changed. Maximizing τ and minimizing μ and δ is desirable.

4 Multi-objective Algorithm

The rescheduling problem formulated in the previous section is a combinatorial optimization problem, where the search space becomes intractable even for moderate sized instances. Genetic Algorithms (GAs) [9][10] are widely used metaheuristics for sampling intractably large and highly complex search spaces. The GAs have frequently been used for solving many scheduling and optimization problems [12][11][4][5]. In this paper, we adopt Non-dominated Sorting Genetic Algorithm- II (NSGA-II), which is a well-known genetic algorithm to solve multi-objective optimization problems. We modify the implementation of NSGA-II provided by jMetal framework [2] to solve our problem. In the next subsection, we explain the steps and parametrization of the algorithm for solving the proposed multi-objective optimization problem.

4.1 NSGA-II Algorithm

In order to design a GA for a particular optimization problem, first step is to devise a suitable representation scheme. In this paper, we choose a scheme in which a schedule (chromosome in GAs literature) is represented by a n-dimensional vector of subsets of tasks. In the considered scheme, n-dimensional means that we have n tasks which need to be performed by n groups of agents and each group j consists of $size_j$ agents. An example chromosome shown in Figure 1 consists of 4 tasks requiring 3, 2, 4 and 5 agents respectively. Assume that there are 5 agents available. It means that $m = 5$, $n = 4$, $size_1 = 3$, $size_2 = 2$, $size_3 = 4$, and $size_4 = 5$. We also assume that task 1 and 2 are overlapping, so an agent assigned to one of the task cannot be assigned to another.

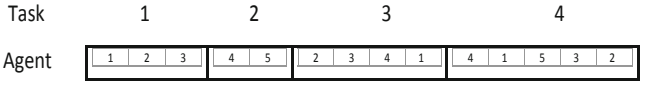


Fig. 1. Chromosome representation of a candidate solution

The steps of NSGA-II are summarized as:

- Generate initial population P_0 which consists of N non-duplicate candidate solutions (schedules). Each task is assigned only to those personnel, which are capable of performing that task.
- Evaluate each candidate solution according to the given fitness functions (5), (6) and (7).
- For $t = 0$ to $M - 1$ (M is the maximum number of generations) do:
  - Generate N non-duplicate offspring (population Q_t) using the following three procedures.
    Selection: Binary tournament selection is used to select parent solutions for reproduction. In binary tournament selection, two individuals are chosen uniformly at random from given pool P_t and the individual with higher fitness is allocated for reproduction trial. Two binary tournaments are conducted to select two parent solutions.
    Crossover: One-point crossover is used to combine two parents to produce an offspring. The crossover point p is chosen uniformly at random such that $0 < p < n$. The first p assignments are taken from parent A and the remaining $n - p$ are copied from parent B such that the constraints are not violated. The duplicate assignments within same task are avoided and no agent is assigned to more than one task at the same time. The corresponding genes where assignments violate any of the constraints are marked with * which need to be repaired. In the repair procedure each of the marked gene is assigned an agent from the set of feasible (capable) agents for that particular gene.
    Mutation: Each gene is selected randomly with probability P_m and it is assigned some other agent from the set of its feasible agents such that it does not violate any of the constraints.
  - Evaluate all generated offspring using the proposed fitness functions (5), (6) and (7).
  - Build the union U_t of parent P_t and offspring Q_t ($U_t = P_t \cup Q_t$).
  - Rank all the solutions in U_t according to Pareto-dominance relation. In the ranking procedure, first step is to find out all non-dominated solutions and rank them as level 1. These solutions form a Pareto-front F_1 with rank 1. Exclude these non-dominated solutions from U_t and find all non-dominated solutions from the rest of the population. That provides a Pareto-front F_2 with rank 2. Keep on doing the same procedure until you have ranked all the solutions into s Fronts.
  - In order to select N candidate solutions for next generation $(t + 1)$, get the sorted fronts ($F = \{F_i | i = 1, \ldots, s\}$) in sequence one by one until the population P_{t+1} is filled with N non-dominated solutions. While processing each front F_i, if the number of candidate solutions is less than number of remaining solutions R of population P_{t+1} ($R = N - C$, where C is current size of P_{t+1}), copy all candidate solutions from F_i to $P_t + 1$. Otherwise, sort the candidate solutions of front F_i in descending order based on crowding distance, iterate in sequence and fill the population P_{t+1} with remaining R solutions.

– Return front F_1, which is an approximated Pareto-front found by the algorithm.

5 Experiments

In this section, we present a set of experiments and their results. The aim of the experiments is to check the accuracy and quality of the results obtained by applying NSGA-II to the proposed rescheduling problem. First, we describe the problem instances, and the parametrization of the algorithm. After that we present and discuss the experimental results of NSGA-II.

We modify the implementation of the presented algorithm which is based on Java based framework jMetal [2]. The algorithm is run on a PC with an Intel Core i5 - 2.60 GHz and 4.0 GB of RAM.

5.1 Problem Instances and Data

In order to discuss the accuracy and quality of the results, we consider two instances of the proposed problem with different sizes as shown in Table 1.

Table 1. Specification of the Problem Instances with Initial Schedule

Prob#	Total Agents	Number of Courses	Number of other Tasks	Groups (# of agents assigned to the tasks)
1	50	1	5	[20, 25, 25, **45**, 40, 40]
2	400	3	63	[100, 50, 50, 40, 30, 45, 40, 35, 70, 45, 40, 100, **50**, 50, 40, 30, 45, 40, 35, 70, 45, 38, 100, 50, 50, 40, 30, 45, 40, 35, 70, 45, 40, 100, **50**, 50, 40, 30, 45, 40, 35, 70, 45, 39, 100, 50, 50, 40, 30, 45, 40, 35, 70, 45, 40, 100, **50**, 50, 40, 30, 45, 40, 35, 70, 45, 40]

Figures in bold are courses.

In all problems, we consider 10 different types of competencies. The data for these competencies is integer values between 0 and 4 inclusive and thus we have 5 levels (0, 1, 2, 3 and 4). We generate different number of profiles of agents for each defined problem. The agents belonging to the same profile have same competencies. For instance, in military troops, one type of profile can be tank drivers. Considering problem number 1 where we have total 50 agents, we generate 10 different types of agent profiles requiring 10, 10, 5, 5, 5, 4, 4, 3, 2 and 2 agents respectively. In problem number 2, we have total 400 agents, we generate 12 different types of agent profiles requiring 10, 10, 20, 20, 30, 30, 40, 40, 50, 50, 50 and 50 agents respectively. For each profile, we select 2, 3, 4 or 5 competencies randomly with a uniform distribution and assign competency levels other than 0 with a given probability distribution. The assumed probabilities for competency levels 1, 2, 3 and 4 are 50%, 25%, 15% and 10% respectively. It means that we have less number of agents, which have higher level of competencies.

The tasks may require a combination of agents from different profiles. Some tasks may even require agents with higher competencies than available. In that case agents may need special courses, which can enhance their competencies.

Initial Schedule data (task start date, duration, priority) for courses and other tasks is generated randomly. In order to verify the working accuracy of the algorithm, we manually create initial schedule for problem number 1 as shown in Table 2.

Table 2. Initial Schedule data for problem number 1

Task#	Day of the year	Task start date	Task duration	Priority	# of agents required	Is course?
1	1	2013-1-1	50	High	20	No
2	2	2013-1-2	25	Low	25	No
3	27	2013-1-27	24	Low	25	No
4	51	2013-2-20	60	Low	45	Yes
5	111	2013-4-21	60	Low	40	No
6	171	2013-6-20	30	Low	40	No

The data for these capabilities and their weights are integer values between 0 and 4 inclusive.

5.2 Parametrization

For experiments, the initial population size is set to 50 and the algorithm terminates after 100,000 evaluations. For problem number 2 given in Table 1, we terminate after 10,000 evaluations. The probability for crossover is 95% and probability of mutation for each gene is 20%.

5.3 Computational Results

In this section, we present the computational results of the proposed rescheduling problem which are obtained using NSGA-II. In order to test the accuracy of the results, we consider a small problem with 1 course, 5 tasks and 50 agents (Problem number 1 in Table 1). Task 1 needs 20 agents, task 2 and 3 need 25 agents each, task 4 which is a course requires 45 agents and task 5 and 6 need to be assigned 40 agents each. The data for the initial schedule (see Table 2) is designed in such a way that at any instance of time at least 80% of the agents are busy in tasks or courses. Furthermore, the requirements for task number 6 are hard and those agents can be assigned to this task which have enhanced their competencies by taking the available course (task number 4). Given the feasible and complete initial schedule, assume that on 10th of January a new high priority task is assigned to the battalion by higher headquarters. This requires rescheduling of the assignments of agents to tasks considering given objectives and constraints. The schedule data for the new task is given in Table 3.

We see that task number 6 and 7 are in parallel to each other, it means the agents, which are assigned to one of task, cannot be assigned to the other. In the

Table 3. Schedule data for a new task introduced on 2013-01-10 for problem # 1

Task#	Day of the year	Task start date	Task duration	Priority	# of agents required	Is course?
7	172	2013-6-21	30	High	10	No

initial schedule, we already have assigned 40 agents to task number 6 and the remaining 10 agents are free. The requirements for the new task (task number 7) are set such that those 10 free agents are not qualified to perform the task. We need to reshuffle the assignments in our initial schedule. The requirements of task number 7 are such that some of the agents need to be reassigned from task number 6 to 7. According to the generated data, the other possible solution can be that we miss the last task. By executing the algorithm, we get the Pareto-optimal front as shown in Table 4. We can see that both types of solutions are found by the algorithm and there can be multiple different Pareto-optimal solutions with same objective values.

Table 4. Pareto-optimal front (for Prob# 1 with 1 new task)

Soln.#	Total number of tasks performed	Assignment Differences from initial schedule	Number of high priority missed tasks
1	6	0	1
2	6	0	1
3	7	7	0
4	7	7	0
5	7	7	0
6	7	7	0
7	7	7	0

Furthermore, to make the problem more difficult, we introduce some new tasks in parallel to the initial set of tasks. The schedule data for 6 new tasks is given below in Table 5. Task number 7 is parallel to task number 6 as before. Moreover, tasks number 8, 9, 10, 11, 12 and 5 are parallel to each other. By executing the algorithm, we get the Pareto-optimal front as shown in Table 6.

Now we consider problem 2, which is a realistic problem with 400 agents, 3 courses and 63 tasks. The requirements and schedule information (task start date, duration, priority) for all the tasks and courses are generated randomly. Given the feasible and complete initial schedule, assume that on 25th of January, 10 new tasks come in, of which 4 of them are high priority tasks. We need to

Table 5. Schedule data for 6 new tasks introduced on 2013-01-10 for problem # 1

Task#	Day of the year	Task start date	Task duration	Priority	# of agents required	Is course?
7	172	2013-6-21	30	High	10	No
8	112	2013-4-22	28	High	10	No
9	115	2013-4-25	56	High	10	No
10	120	2013-4-30	21	High	10	No
11	127	2013-5-07	35	High	10	No
12	129	2013-5-09	14	High	10	No

Table 6. Pareto-optimal front (for Prob# 1 with 6 new tasks)

Soln.#	Total number of tasks performed	Assignment Differences from initial schedule	Number of high priority missed tasks
1	6	0	6
2	7	7	5
3	8	10	4
4	11	47	0
5	11	47	0
6	10	40	1
7	8	10	4
8	11	47	0
9	6	0	6
10	8	10	4
11	10	40	1
12	10	40	1

reschedule the assignments of agents to tasks in such a way that we take care of three given objectives. The schedule data for the new tasks is generated randomly and number of required agents for task number 67 to 76 are 40, 30, 50, 20, 35, 20, 25, 30, 20, and 25 respectively. By executing the algorithm, we get the Pareto-optimal front as shown in Table 7. There can be multiple solutions with same objective values but we have shown only distinct ones. The results show that performing all high priority tasks is in a sharp contrast with preserving the initial schedule and both objectives cannot simultaneously be optimized. It is up to the decision-maker to select the desired solution form the Pareto-optimal front according to her/his preferences.

Table 7. Pareto-optimal front (for Prob# 2 with 10 new tasks)

Soln.#	Total number of tasks performed	Assignment Differences from initial schedule	Number of high priority missed tasks
1	67	0	4
2	68	505	1
3	67	593	0
4	65	467	1
5	67	297	3
6	68	962	0

6 Conclusion and Summary

In this paper, we propose a novel model for rescheduling different activities of the battalions personnel. We formulate the rescheduling problem as a multi-objective optimization problem, in which we consider three objectives: (i) maximizing the total number of performed tasks, (ii) minimizing the number of high priority tasks that are missed, and (iii) minimizing the differences between the initial schedule and updated schedule.

Firstly, a multi-objective mathematical model, with three conflicting objectives and a set of constraints is built. Multi Objective Evolutionary Algorithms (MOEAs) are well-known metaheuristics for sampling intractably large and

highly complex search spaces. In order to solve the optimization model, we adopt a well-known multi-objective genetic algorithm NSGA-II.

We present the computational results of the proposed rescheduling problem, which are obtained using NSGA-II. We verify the accuracy of the algorithm in this context by considering a small problem with easy to verify solutions. Furthermore, we consider a realistic problem instance for a battalion with 400 agents and 66 tasks in the initial schedule. We present the computational results of rescheduling when 10 new tasks come in. The experimental results show that NSGA-II efficiently provides the Pareto-optimal solutions for the proposed rescheduling problem. From the obtained Pareto-optimal front, the decision maker can choose one or more solutions according to her/his preferences.

References

1. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
2. Durillo, J.J., Nebro, A.J.: jMetal: A Java framework for multi-objective optimization. Advances in Engineering Software 42, 760–771 (2011)
3. Clark, A.R., Walker, H.: Nurse rescheduling with shift preferences and minimal disruption. Journal of Applied Operational Research 3(3), 148–162 (2011)
4. Moz, M., Pato, M.V.: A genetic algorithm approach to a nurse rerostering problem. Computers and Operations Research 34(3), 667–691 (2007)
5. Maenhout, B., Vanhoucke, M.: An evolutionary approach for the nurse rerostering problem. Computers and Operations Research 38(10), 1400–1411 (2011)
6. Chicano, F., Luna, F., Nebro, A.J., Alba, E.: Using multi-objective metaheuristics to solve the software project scheduling problem. In: GECCO, pp. 1915–1922. ACM (2011)
7. Alba, A., Chicano, J.F.: Software project management with GAs. Information Sciences 177, 2380–2401 (2007)
8. Hao, X., Lin, L.: Job shop rescheduling by using multi-objective genetic algorithm. In: 40th International Conference on Computers and Industrial Engineering (CIE), pp. 1–6. IEEE (2010)
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Massachusetts (1989)
10. Mitchell, M.: Introduction to genetic algorithms. MIT Press, Massachusetts (1999)
11. Younas, I., Kamrani, F., Schulte, C., Ayani, R.: Optimization of Task Assignment to Collaborating Agents. In: IEEE Symposium on Computational Intelligence in Scheduling, pp. 17–24. IEEE (2011)
12. Gonçalves, J.F., Mendes, J.J.M., Resende, M.G.C.: A genetic algorithm for the resource constrained multi-project scheduling problem. European Journal of Operational Research 189(3), 1171–1190 (2008)

Research on Co-simulation Method in ADAMS and MATLAB for Missile Seeker's Stabilization Platform Design

Xueping Zhu, Zhengchun Liu, and Jun Yang

School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China

Abstract. Missile seeker's stabilization platform, also called coordinator was one of the key parts of seeker. In this study, an ADAMS/Matlab co-simulation environment for the design of seeker's stabilization platform was proposed. First, the mechanical model of coordinator is analyzed, virtual prototype model of the coordinator was built through SolidWorks and ADAMS, and the control model of the coordinator was built through MATLAB and Simulink toolbox. Then co-simulation model was achieved through ADAMS/Control block and Matlab/Simulink port. At last, model accuracy was verified by using coordinator's real data. With all of these, control system of the coordinator was designed in the co-simulation environment, and isolation performance of the coordinator was tested in this co-simulations environment. Simulation results show that the coordinator's design work based on the ADAMS/Matlab co-simulation environment is feasible, effective and practical, and it has advantages such as higher fidelity of controlled objective, easier for modeling and debugging compared with traditional design strategy of mathematical methods, and provided a new design approach for high performance coordinator.

Keywords: ADAMS and MATLAB Co-Simulation Method, Virtual Prototype, Coordinator, Mechanical Control System.

1 Introduction

The virtual prototype technology is a brand new products development strategy, and also a digital design method based on products computer simulation model. The digital model, namely virtual prototype, combines developing models in different engineering fields in order to imitate the real products from their appearance, function and behavior, and at the same time supports for concurrent engineering methodology. The virtual prototype technology is involved with multi-body system kinematics and dynamics modeling theory and its implementation, meanwhile it is an integrated application technology of advanced modeling technology, multi-fields simulation technology, information management technology, interacted user interface technology and virtual reality technology.

The virtual prototype design system can supply integrated virtual prototype environment for missile seeker's design and simulation, and on the basis of full dem-

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 105–113, 2013.

onstration of user requirements, then the designers can focus on developing algorithm, modeling and integrated simulation. This can not only shorten seeker developing cycle, but also reduce design cost and risk.

The coordinator system is one of the vital subsystems in seeker, and its performance has direct effects on seekers' tracking velocity and precision. High performance coordinator is the key for increasing detection probability, recognizing target and tracking target precisely, and it is meaningful to enhance seeker's overall performance. The four functions of coordinator are as follows: First, make optical axis of IS(imaging sensor) stable and unaffected by aircraft attitude motion, and in the end obtain clear images for imaging system; Second, make up IFOV (instantaneous field of view) shortage of imaging system, and make sure enough FOV of search by scanning for imaging system, thereby guarantee no loss of target; Third, realize optical axis preset function, and make seeker optical axis reserved pointing to the requirement direction, thus improve target discovery rate; Forth, achieve target tracking through seeker optical axis tracking target's line-of-sight axis controlled by servomechanism of servo system, then measure the target's line-of-sight angular velocity, after that, form it into guidance signal and deliver to autopilot.

In recent years, multi-body dynamics has become the most active modeling theory in the engineer fields, the virtual prototype technology becomes mature, meanwhile simulation and test based on all kinds of virtual prototype model has become an important approach of testing and theoretical analysis. Much simulation software, such as ADAMS, MATLAB, ANSYS, etc., has their own advantages and disadvantages, and therefore it is always dissatisfactory with the simulation results got by single software. However, the co-simulation breaks their own defects, connects two or more than two simulation software through bidirectional data port, and make data real-time exchangeable during simulation, bring every simulation software's superiority into full play simultaneously, this makes virtual model get much closer to real model, and thus the most ideal simulation results can be obtained.

This study adopts ADAMS and MATLAB software, and build bidirectional communication link through their own communication port and a common agent, thus associate mechanical model of the coordinator on ADAMS with control system on MATLAB/Simulink to conduct co-simulation test, therefore, the controlled object modeling can be simpler and more accurate and provides a more precise controlled object for control system design. The ADAMS/Matlab co-simulation method of the coordinator affords a brand new thought for high performance coordinator system.

2 Three Degree of Freedom Coordinator Model

The ADAMS has advantage on kinetics simulation but lack of modeling function; however ADAMS has perfect model import ports for almost all of the popular CAD model software. This paper uses SolidWorks to construct coordinator's virtual prototype model firstly, and then exports model in the form of Parasolid, finally import the Parasolid model to ADAMS through ADAMS transform port. In the end, this paper has got available of this co-simulation model by combining virtual prototype with control calculation program via related ports in MATLAB, ADAMS/Control and MATLAB/Simulink.

This coordinator's mechanical model used in this study consists of outer, middle and inner frame, and there is a driver motor on one side of every frame and a resolver installed on the other side. In order to isolate the effect of body disturbance on seeker direct, a rate gyroscope is installed on inner frame and outer, which is used for measuring pitch, yaw and rotate angular velocity in inertial system. Figure 1 shows the mechanical structure of three degree of freedom coordinator.

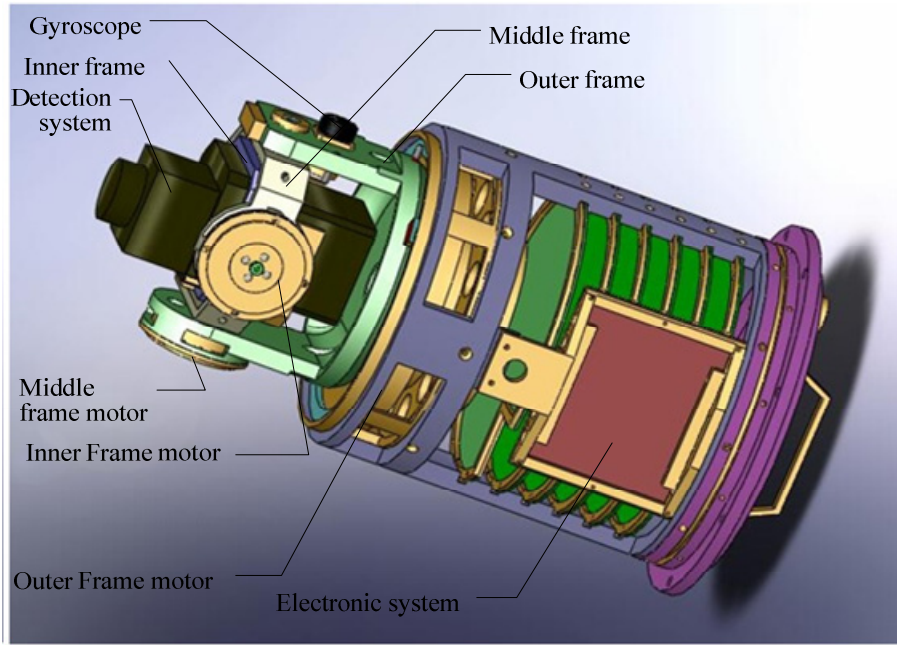


Fig. 1. Three degree of freedom coordinator mechanical model

When models were imported directly into ADAMS, there is not any quality information, restraint or driver, so there it in impossible to launch dynamic simulation, proper restraint should be added first, meanwhile designate material density and poisson's ratio on related components. After this, the ADAMS can calculate components' quality, rotational inertia and automatically establish center of mass coordinate system at centroid. The restraint added to all components which are called kinematic pair or hinge is for the sake of regular motion in the mechanical model. On the purpose of imitating system real motion, the kinematic pair must be abstracted from real model and defined between components. At the same time, driver and load should be added into kinematic pair. In accordance with construction of testing system, driver moment has added into three motor shafts, and the size of moment is exported by the form of state variable and imported from outside. In the meantime, build mark point in the center of gyroscope to imitate export angular velocity of real rate gyroscope.

Satisfy the integrity of simulation motion, and simplify model reasonably at the same time, then the components and details that have no effects on relative motion can be ignored, such as some transition, chamfer, rounding in screw, nut, guide

block and components can effectively enhance efficiency of simulation calculation and reduce difficulty of model transformation. The way to set components real value by hand can eliminate the related components physical change as rotational inertia and quality which are caused by ignored components and geometry details, and thus make model the has same physical property as physical prototype. In conclusion, the simulation results are equivalent in physical prototype test.

3 Three Degree of Freedom Coordinator Control System Model

Figure 2 shows the structure of single channel electromechanical model on coordinator.

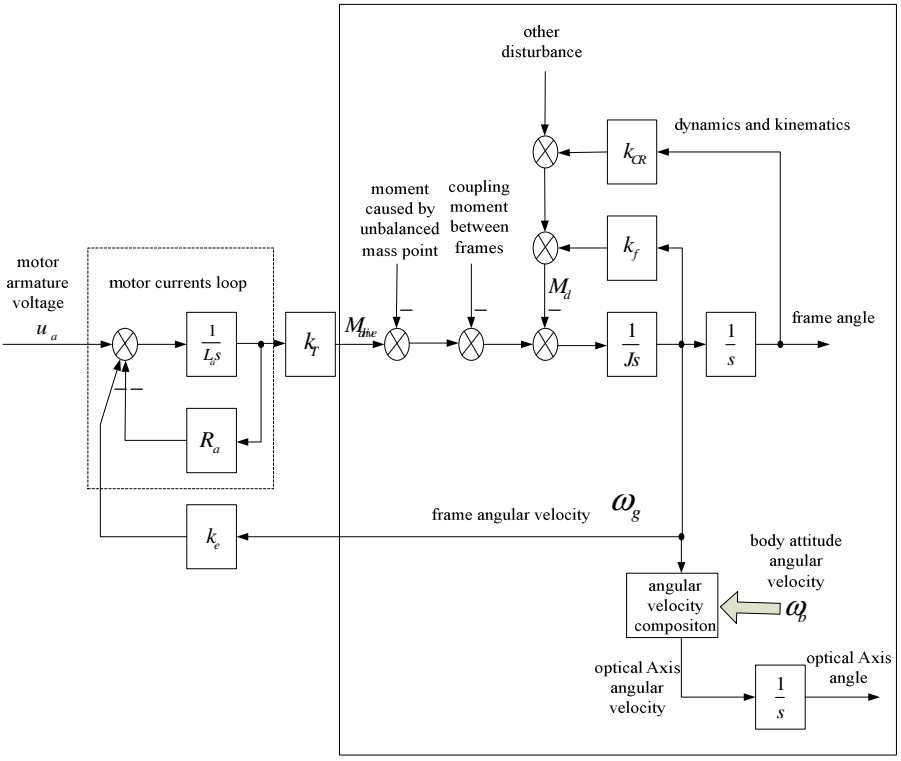


Fig. 2. Structure of single channel electromechanical model on stable platform

In Figure 2, u_a denotes motor armature voltage; L_a denotes motor armature inductance; R_a denotes motor armature resistance; K_T denotes moment coefficient; k_e denotes back EMF(Electromotive Force) coefficient; M_{drive} denotes Motor drive moment; M_d denotes disturbance torque; J denotes rotational inertia of load; k_f

denotes friction coefficient; k_{CR} denotes lead resistance moment coefficient; ω_g denotes frame angular velocity; ω_b denotes body attitude angular velocity.

In order to construct simplify mathematic model of the coordinator, we ignored unbalanced moment of frame command, coupling moment between frames, etc., so the Torque motor model is as follows:

$$M_{drive} = \frac{k_T}{L_a s + R_a}\left(u_a - k_e \omega_g\right) \tag{1}$$

According to the newton's second law, the torque equilibrium equation is as follows:

$$M_{drive} = k_f \omega_g + J\dot{\omega}_g \tag{2}$$

Where f is regarded as friction coefficient.

Then derives the transfer function of stable platform is as follows:

$$\frac{\omega_g(s)}{u_a(s)} = \frac{k_T}{\left(L_a s + R_a\right)\left(Js + k_f\right) + k_T k_e} \tag{3}$$

4 Construction of co-simulation Model and Verification

4.1 Model Construction

There is a port connected to Matlab in the ADAMS, and that can transfer the nonlinear mechanical model in ADAMS to s_function in Simulink, then add control model

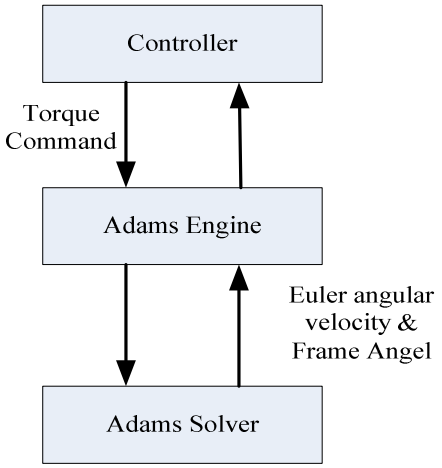


Fig. 3. Connection ways of ADAMS and MATLAB

to s_function for the sake of using many kinds of tools in Simulink to conduct the Matlab and ADAMS co-simulation. According to the controller design theory as shown before, figure 3 shows the construction of co-simulation model in Matlab/Simulink.

4.2 Model Verification

To verify the model and the interactive of ADAMS and Matlab/Simulink, this paper enter a cosine form of torque command to an assigned channel and put in missile body's perturbation angular velocity simultaneously, then observe the output of gyroscope and frame angle to achieve model verification, the figure 4 shows the verification results.

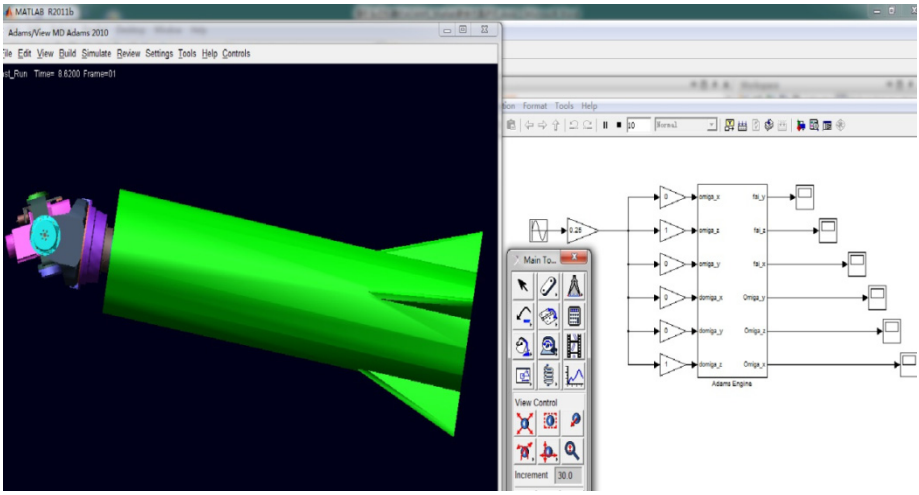


Fig. 4. Verification results

In the real test, the body perturbation angular velocity is given as 0.25cos(t), and angular acceleration of frame angle(moment divide rotational inertia) command is also 0.25cos(t), the simulation results show that the value measured by gyroscope is the superposition of body perturbation angular velocity and frame angular velocity, that is $\int_0^t 0.25\cos(t)dt + 0.25\cos(t)$, and the frame angle is $\int_0^t (\int_0^t 0.25\cos(t)dt)dt$, thus confirm the validity of model.

5 Co-simulation Test and Analysis

Using the seeker stable loop to eliminate the influence of body perturbation on seeker optic axis direction, and the stable loop must have enough bandwidth. Figure 5 shows the rate gyroscope coordinator stable program.

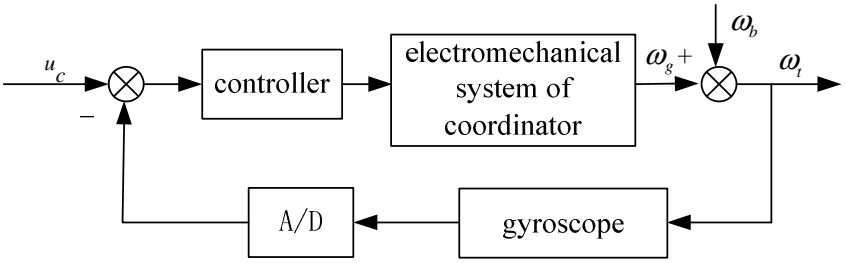


Fig. 5. Velocity gyroscope coordinator stable program structure

In order to realize rate gyroscope stable platform control, the paper introduce PI control to loop, and the controller is as follows:

$$G_A(s) = K_p(1 + \frac{1}{T_i s}) \tag{4}$$

The motor armature inductance is 5mH; the motor armature resistance is 8Ω; the moment coefficient is 0.2Nm/A; the back EMF coefficient is 0.2971V.S/rad; the static friction torque is 0.03Nm; the rotational inertia of load is 0.0077 $kg \bullet m^2$; the friction coefficient is 0.01Nm/(rad/s).

In the ADAMS/Matlab co-simulation environment, the PI controller transfer function is as follows:

$$G_A(s) = 0.95 + \frac{90}{s} \tag{5}$$

When body perturbation amplitude is 40(°)/s, the perturbation frequency is 5Hz, figure 6 shows the simulation results of inside casing stable loop decoupling performance based on ADAMS/Matlab co-simulation. The simulation results show that the decoupling coefficient of inside casing stable loop on 5Hz body perturbation is 5.9% which meets the design demand.

Utilize linear mathematic model of inside casing stable loop to analyze control system stability. Figure 7 shows open-loop frequency domain characteristics of inside casing stable loop, that is the open loop cut-off frequency is 203rad/s, the magnitude margin is 7.74dB, the phase margin is 38.4°, which can meet the design index requirements, and then verify the rationality of design results.

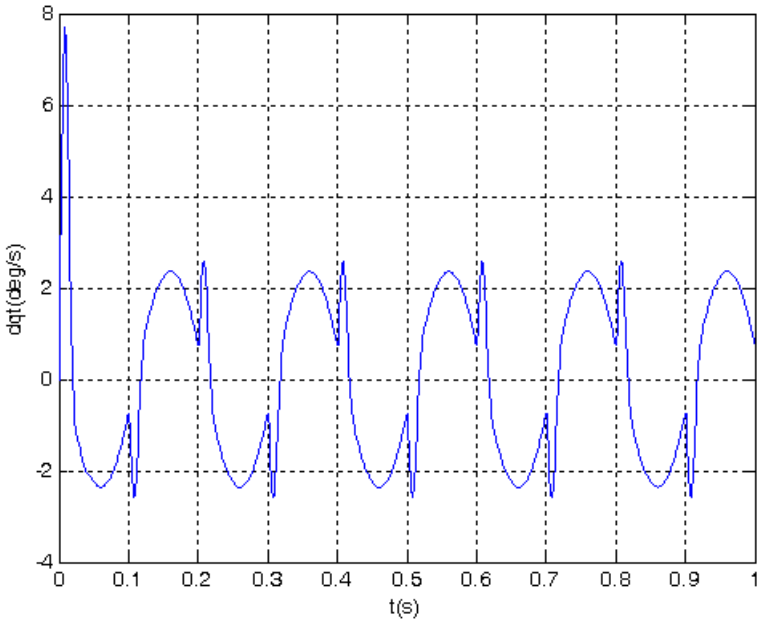


Fig. 6. Simulated results curve on decoupling characteristics of inside casing stable loop

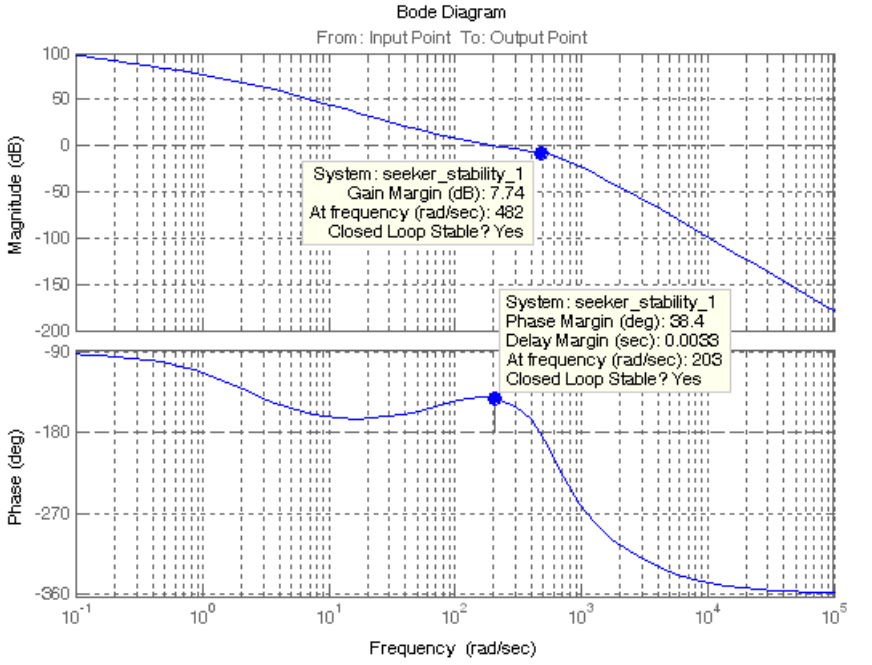


Fig. 7. Open-loop frequency domain characteristics of inside casing stable loop

6 Conclusion

This paper presented a strategy to realize ADAMS/Matlab co-simulation, based on this strategy we can realize both the mechanical and control system of the missile seeker's coordinator on its virtual prototype. Using ADMAS based mechanical model as controlled plant, with the controller designed by Matlab/Simulation toolbox to carry out ADAMS/Matlab co-simulation can get good design result, and avoid design and analysis error due to inaccurate expression of Matlab electromechanical system through mathematic model, and make the coordinator virtual prototype much closer to real prototype to bring more reliable on design and simulation results.

This research idea and methodology in this study can be extended into more considerable forms of coordinators, meanwhile provide reference to the high performance coordinator's design and intense research.

References

[1] Dong, L., Zhaoxiang, D., Jun, H., Hengyuan, W.: Unite simulation and analysis for vehicle suspension based on ADAMS /Matlab. Modern Manufacturing Engineering 8, 71–75 (2012)

[2] Li, B.-M., Qian, Z.-B., Cheng, H.-J., Liu, Z.-H.: Co-simulation of Engine for AUV in ADAMS and MATLAB. Journal of System Simulation 22(7), 1668–1673 (2010)

[3] Xiao, P., Xie, Z.: Co-simulation of Rotor System Supported by Magnetic Bearings Based on ADAMS and MATLAB. System SimulationTechnolog 7(1) (2011)

[4] Zhang, J., Hu, Y., Wu, H.: Co-simulation of magnetic suspended rotor system: research and application. In: International Conference on Mechatronics and Automation, pp. 1711–1715. IEEE Press, Changchun (2009)

[5] Mou, W., Xie, Z.: Variation parameters control of active magnetic bearing flexible rotor system, pp. 55–62. Wuhan Technology Press, Wuhan (2009)

[6] Zhang, X., Sun, B., Sun, Q.: ChenNan: Vehicle and terrain interaction based on Adams-Matlab co-simulation. Journal of Southeast University 25(3), 335–339 (2009)

[7] Lin, L., Corina, S.: On the impact of cargo weight vehicle parameters, and terrain characteristics on the prediction of traction for off-road vehicles. Journal of Terramechanics 44(3), 221–238 (2007)

[8] Carsten, H., Bjoern, L., Roland, J.: A new tire-soil interaction model for vehicle simulation on deformable ground. Vehicle System Dynamics 43(sup), 384–394 (2005)

An Automatic Registration Method Based on Fiducial Marker for Image Guided Neurosurgery System

Minjie Yin, Xukun Shen, Yong Hu, and Xiaorui Fang

State Key Laboratory of Virtual Reality Technology and Systems,
Beihang University, Beijing 100191, China

Abstract. Patient-to-image registration is a fundamental step of Image Guided Neurosurgery System. In this paper, we propose an automatic technique to register the patient space with the preoperative images based on fiducial markers. Our technique includes three parts. First, we identify the markers in the image space based on multi-scale features and then cluster all this features to find marker centers. And in patient space, we combine Trajkovic's and Harris's algorithm to detect the corners in the center of the markers and then reconstruct 3D coordinates based on binocular stereo vision. At last, we register these two sets of centers using RANSAC. Experiments show that the marker centers can be localized precisely and the two space can be perfectly registered.

Keywords: multi-scale feature, Image Guided Neurosurgery System, registration, fiducial marker detection, corner detection.

1 Introduction

Recently, Image Guided Neurosurgery System (IGNS) are playing a more and more important role in the field of minimally invasive surgery. By combining highly accurate localization of lesion position and guiding the surgical instruments efficiently, it makes the surgery more effective and less invasive, and thus reduces the risk of surgery [1].

During the surgery, the preoperative CT/MRI images need to be precisely registered with the patient, in order to reflect the relationship between the instruments and lesion position [2],[3]. Registration, defined as the alignment of the different coordinate systems, is an essential step in IGNS. It directly affects the accuracy of the whole system [4].

The technique currently used for patient-to-image registration in IGNS is mostly based on point-pair matching. By selecting at least three pairs of corresponding points from both coordinates, a spatial transform can be computed to register the patient space to the image space. At present, most IGNS need the user to select the points in images manually, in this way the resulting error depends to a large degree on the user's personal experience. In addition, the accurate selection of points is time-consuming. So if the points could be

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 114–125, 2013.

automatically and precisely localized, the accuracy of the guide system will be enhanced and the preparation time will be cut down.

Two main kinds of points can be used for this point-based registration in IGNS: anatomical landmarks and fiducial markers. Woerdeman pointed out that the registration based on fiducial markers is more reliable [5]. In addition, fiducial markers are more easily to use than anatomical landmarks, thus registration based on fiducial marker is used most widely in clinical application.

There have been some researches done for automatic localization of the center of fiducial markers. Maurer [6] proposed a hybrid registration technique that used a weighted combination of points and surfaces. But the position of each marker was acquired by the probe, this need the user's intervention. Wang et al. [7] proposed an approach based on shape index and curvedness, experiments on clinical CT data showed that the center of the fiducial markers could be automatically localized with high precision. But the results on MRI data were not as good as that with CT data. In 2009, they designed another method to automatically identify the centers of markers, their method compared the marker model and the surface patch of the patient's head segmented from 3D CT or MRI images [8]. If their Project Height Images(PHI) are alike, this means the center voxel of the surface patch was a candidate for the marker center. This algorithm performed well on both CT and MRI data, but needed a little longer execution time, especially for MRI data. Zheng et al. [9] described a semi-automated solution for precise detection of fiducial screws. They regarded the detection of the fiducial screws as a pose estimation problem and solved it by iteratively matching a computer aided design (CAD) model of the fiducial screw to features extracted from the CT/DVT data. But their solution called for a user-supplied initialization and the screws meant invasive to the patient.

In patient space, many fiducial markers take corner as interesting points, since corner can be robustly detected and do not change greatly under different light conditions. The earliest corner detection algorithm was proposed by Morevec [10], which was based on autocorrelation measurement response in four directions. Morevec operator is easy to calculate but it is anisotropic and noise-sensitive. Harris [11] proposed a new algorithm based on Morevec's mind. This algorithm used the autocorrelation matrix of brightness function to realize corner detection. But it is computationally intensive and time-consuming, difficult to meet real-time requirement. MIC method proposed by Trajkovic [12] made a progress in speed, but it had similar problems with Morevec's. Guo [13] combined Harris's with Trajkovic's algorithm to cut down the computation time. However this approach could not filter out false corners, so it did not apply to be used in surgical navigation system directly.

In this paper, we propose an automatic approach to localize the center of fiducial markers in image space and patient space. In image space, we use the multiple scale features to recognize the field of the markers and then cluster all this features to find the centers. In patient space, we use a binocular camera to detect the corner in the center of the marker, combining Trajkovic's and Harris's algorithm to increase the speed while maintain the accuracy. The flowchart of

the whole system is showed in section 2. We will describe the methods in detail in section 3 and section 4, and give the experiment results on simulation data in section 5. Finally we'll conclude this paper by conclusion and outlook.

2 Overview

There are two kinds of fiducial markers in use in current IGNS: bone implanted and skin affixed [4]. The former one is not only time-consuming but also invasive to the patient. Now most fiducial markers are adhered to the patient's skin before scanning. In this paper, we make our fiducial marker as a cylinder which is shown in Fig.1. The thickness of the marker is 1.7mm, the diameter is 20mm.

Fig. 1. Markers in our IGNS

The core idea of point-based registration involves determining the coordinates of corresponding points in image and in physical space (namely patient space) and computing the geometrical transformation that best aligns these points (generally in a least-squares sense) [6]. In image space, we detect the marker centers from the preoperative images like CT/MRI. And in patient space, we reconstruct the spatial information based on binocular stereo vision. The binocular camera we use is Bumblebee made by Point Grey Research, Inc.

Fig.2 shows the flowchart of the patient-to-image registration. During the surgery, the patient lies on the operation table with the stereo-vision cameras facing down. In image space, the mesh of the head is extracted from the preoperative CT/MRI images through some surface reconstruction technique. Then we detect the features around the markers on the mesh basing on the multiple-scale space. After this, we cluster all feature points to find the centers of the markers. In patient space, we obtain the pictures in real time using the calibrated stereo-vision cameras. And then combine Trajkovic's and Harris's algorithm to detect the corners in the pictures. Because there may be some misidentified points (what we need is the corner in the center of the marker), we adopt a method based on gray transition to filter out the wrong corners. Next, reconstruct the three-dimensional coordinates of the centers. Finally, after getting two sets of the center points, we calculate the transformation matrix to finish the registration.

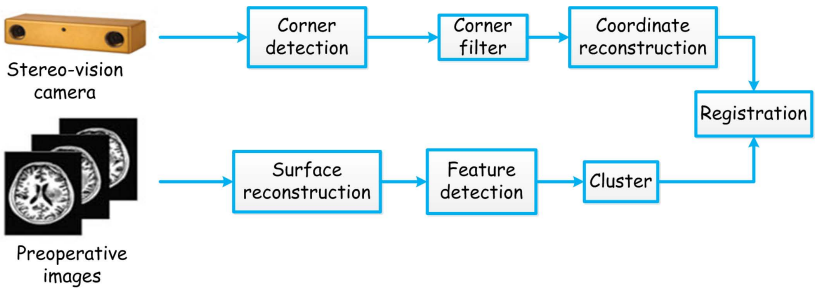


Fig. 2. Flowchart of our registration method

3 Detection of Fiducial Marker Center Based on Multi-scale Feature

In the field of computer vision, Lowe [14] proposed the SIFT algorithm to find the key points in two-dimensional images. The basic idea of his algorithm can be concluded that smoothing the images with Gaussian function to construct a scale space, then calculating the difference between these smoothed images, the pixel which has a big difference makes a key point. We extend this idea to the spatial space. First, create a scale space of the mesh data. Then find the extreme value as the feature points. Considering the points in the neighbor of the markers have relatively large curvatures, many points around the markers will be detected by this method. Then by the means of clustering, we can find the centers of the markers.

3.1 Multi-scale Representation of Triangular Mesh

For a two-dimensional image $I(x, y)$, the representation in different scale space $L(x, y, \sigma)$ can be gotten by convolution with Gaussian function $G(\sigma)$. σ is called scale space factor. The smaller the value is, the less the image is smoothed, and vice versa. In order to represent triangular mesh in different scale space, we need to smooth the mesh under different scale space factor σ. In terms of the two-dimensional image, the gray value of each pixel can represent the intensity of the signal. For the mesh in the three-dimensional space, what the data contain is only the vertex location information. Therefore we need to redefine a new signal to indicate the intensity for every vertex in different scale space.

If the original grid image is M and the scale factor is σ, the result is M_σ after smoothed by the Gaussian filter. Let p be a vertex of M, p_σ is the vertex in M_σ corresponding to p, then in the scale space with the factor σ, We define the signal strength of p is $l(p, \sigma)$, $l(p, \sigma) =< n_\sigma(p), p - p_\sigma >$, where $n_\sigma(p)$ is the normal of p_σ in M_σ. In general, there will be an angle between $n_\sigma(p)$ and $p-p_\sigma$. Namely the movement of vertex p includes both normal and tangential components in the process of smoothing. The signal strength of p is the displacement between p and

p_σ in the normal direction of p_σ. In order to reduce the influence of tangential component, p should only move along the normal direction of p_σ.

Jones [15] smoothed the grid by local first-order predictor. However, he did not limit the vertex movement in the normal direction. It would lead to vertex drift. We borrowed the idea of the predictor. For the vertex p, $N(p)$ are the vertices in the neighbor of p. $\forall q \in N(p)$, we define the predictor of q as $\prod_q$ by the surface S_q decided by q and the normal $n_\sigma(p)$ of p_σ. So, the predictive value $\prod_q(p)$ under the predictor $\prod_q$ is the projection of p on the surface S_q. Fig.3 shows the theory. But before smoothing, $n_\sigma(p)$ is unknown, so we need to estimate n_σ first.

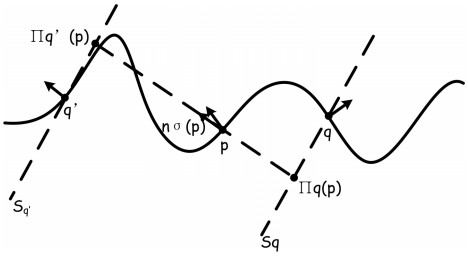


Fig. 3. The theory of vertex predictor

The algorithm is as follows:

(1) Detect all boundary points of M. In the smoothing process, boundary points remain the same, in order to avoid the contraction of the borders.

(2) $\forall p \in M$, calculate the normal $n_\sigma(p)$ of the smoothed vertex p_σ corresponding to p in the scale space with σ as the scale factor. Eq.1 is the Gaussian function.

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2}{2\sigma^2}} \tag{1}$$

Let $N(p)$ be the points in the neighbor field of p, then $\forall q \in N(p)$, define $\|q-p\|$ as the weight of spatial distance. Taking into account that the grid may be non-uniform sampled, we introduce the Voronoi area of q as the weights A_q. Thus, we use Eq.2 to estimate $n_\sigma(p)$.

$$n_\sigma(p) = \frac{\sum_{q\in N(p)} n_q A_q G_\sigma(\|q-p\|)}{\sum_{q\in N(p)} A_q G_\sigma(\|q-p\|)} \tag{2}$$

(3) $\forall p \in M$, calculate the corresponding vertex p_σ in the scale space with σ as the scale factor. We use the weighted predicted value $\prod_q(p)$ under the predictor $\prod_q$ to calculate the p_σ. As shown in Eq.3.

$$p_\sigma = \frac{\sum_{q\in N(p)} \prod_q(p) A_q G_\sigma(\|q-p\|)}{\sum_{q\in N(p)} A_q G_\sigma(\|q-p\|)} \tag{3}$$

Through our smoothing algorithm we can get that, $\forall p \in M$, the displacement between p and $p_\sigma \in M_\sigma$ only exists in the $n_\sigma(p)$ direction. This displacement is the signal strength $l(p, \sigma)$ in the scale space with scale factor σ. Thus, we get the $M^{'}s$ multi-scale representation $L(M, \sigma)$. By choosing different scale factors: $\sigma_0, \ldots, \sigma_n$, where $\sigma_{i+1} = k\sigma_i, i = 0, 1, \ldots, n-1$, we can get the representation $L(M, \sigma_i)$ of the grid M in different scale space. Fig.4 shows the representation of the model in different scale space and its signal strength.

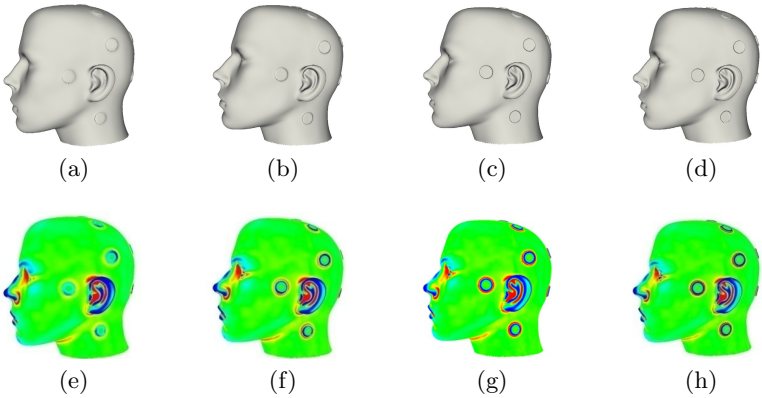


Fig. 4. The first line is the smoothed head model under different scale factors, and the second line is the signal strength of corresponding model. In this figure, $k = 0.6$, σ_0 is the average distance between adjacent vertices in the mesh.

3.2 Feature Point Detection

In order to detect stable points in multi-scale space, Lowe [14] used the extremum of the Difference of Gaussian function (DOG), and Lindeberg [16] pointed out that Difference of Gaussian function is the approximation of scale-normalized Laplacian of Gaussian $\sigma^2\nabla^2 G$. Mikolajczyk's experiments [17] showed that the extremum of $\sigma^2\nabla^2 G$ have the stable features. The value $D(M, \sigma_i)$ of Difference of Gaussian function can be calculated by the Eq.4 where $i = 1, 2, \ldots, n$.

$$D(M, \sigma_i) = L(M, \sigma_{i+1}) - L(M, \sigma_i) \tag{4}$$

After calculating the value of Difference of Gaussian function, we detect the extremum in scale space as the feature point. $\forall p \in M$, compare its value with its one-ring neighbors of the same scale space and the adjacent scale spaces. As shown in Fig.5, the red vertex means the point p and all blue vertices means its neighbors. If the DOG value of the red vertex is largest or smallest, then we define p as an extreme point.

For localizing the center of the markers, all feature points are clustered. In our method, a procedure similar to the cluster algorithm in [7] is used to pick

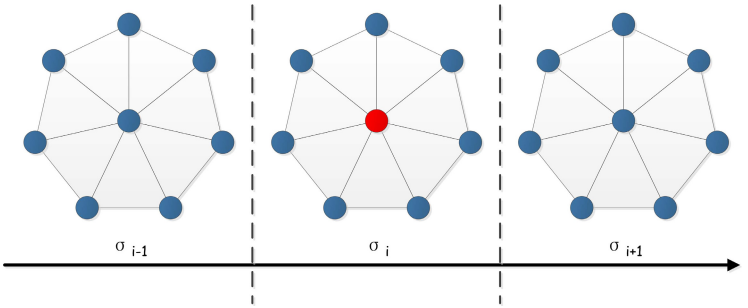


Fig. 5. Extreme point in scale space

out the groups that correspond to the markers according the distribution of the feature points. Next, compute the centers of the markers based on the geometry information. The clustering is done as follows and the result is shown in Fig.6:

(1) Initialization. In the first step, we pick k feature points as the initial centers of the groups using Kmeans++ algorithm. Then we scan all feature points. For each point, calculate its distance to every center of the groups, add the point to the group which is closest to it.

(2) Group processing. In case that cluster divides the feature points belong to the same group into two, we merge two groups if the distance between their centers is less than a threshold (maybe the diameter of the marker, as the distance of the markers is often larger than the diameter). After merging, we delete the groups with few points, due to the group corresponds to the marker usually has many feature points.

(3) Cluster again. For localizing the centers, what we need is only the points that are around the bottom of markers. So if the distance between the feature point and its group center is within a certain range (eg 7-15mm), the point is reserved. Otherwise, the point will be deleted. After scanning all these points , repeat (1)-(3) steps.

(4) Calculate the centers. Influenced by the density of feature points, simply treating the centers of the groups as the centers of the markers will produce large error. So we make use of the geometry feature of the marker that the marker is a cylinder. And after the last step, all remaining feature points are around the markers at the bottom. What we next to do is to find the center of the circle. According to the principle that the intersection of three perpendicular bisectors of a triangle is the center of a triangle circumcircle, we take three feature points randomly from one group, find the center O of the circumcircle and also the radius R. Calculate the distance between all feature points to the center O, if the distance is approximately equal to R, the feature point is considered as qualified. Count the ratio of the qualified points. The center O is acceptable if the ratio is more than 2/3.

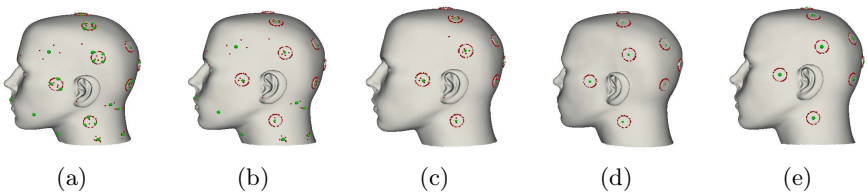


Fig. 6. The procedure of Clustering the feature points, the red points mean feature points and the green one means the center of the group. (a) result after the first clustering; (b) merge the adjacent group; (c) delete the small group; (d) result after the second clustering; (e) calculate the centers according to the geometry feature.

4 The 3D Reconstruction Algorithm for Specific Corners in Patient Space

In patient space, we need to detect the central corners of the markers to register with the centers detected in image space. The flowchart of the algorithm is shown in Fig.7. First we acquire two images from each len of the binocular camera, then extract corners from these images. Considering the special design of the markers, the corner points on the circumference can also be detected, these points should be filtered out. After this step, recovery the spatial coordinates according to the theory of binocular stereo vision.

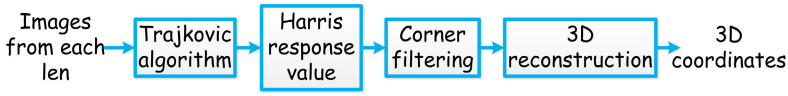


Fig. 7. Flowchart of marker detection algorithm in patient space

In the corner detection field, Harris corner detector [11] is one of the most widely used currently. It is based on the autocorrelation matrix of the image gradient. Since this algorithm needs to calculate the autocorrelation matrix for every pixel in the original image, it is quite time-consuming. Though it can detect the corners with acceptable accuracy but it can not be used in real-time IGNS.

The MIC corner detector proposed by Trajkovic [12] using multi-grid algorithm. At first, calculate the minimum intensity change in the horizontal and vertical direction under a low-resolution version of the original image, obtaining the possible corner position. Then only processes the pixels in original image corresponding to those points.

We combine these two detectors, through MIC algorithm to identify those potential corner points, then calculate their Harris corner response value to choose true corners. In this way, we can not only speed up the computation time but also guarantee the accuracy.

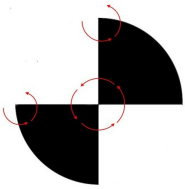


Fig. 8. The schematic of gray transitions

After extracting the corners, we need to screen out those corners on the circumference of the markers. The method we adopt is based on the gray transitions which is shown in Fig.8.

In Fig.8, what we want is the cross-type center of the marker. The corner on the circumference is defined as false corner. After the binarization of original image, draw circle around the corners achieved by the last step, then count the times of the gray transitions. A gray transition means the gray changed from white to black or black to white. We can see that the false corner has only two times of gray transitions, but the center point has four times. Thereby we can eliminate the false corner.

In order to test the algorithm, we have designed an experiment. The resolution of the binocular camera is 640×480, and CCD pixel size is $3.75um \times 3.75um$. Arbitrarily selected 5 points, the distance between these five points are given in advance as a ground-truth reference. Reconstruct these 5 points using the API of camera system, then calculate mutual distance between these points. The mean error between measurement distance and the ground-truth distance is about 0.5mm. And the time is shorter about 3.3 times than the Harris's algorithm.

5 Registration

The registration of the INGS can be abstracted into the following mathematical problem: Given two sets of points in different space $P = \{p_1, p_2, \ldots, p_m\}$ where $p_i \in R^3$ $(i = 1, 2, \ldots, m)$ and $Q = \{q_1, q_2, \ldots, q_n\}$, where $q_j \in R^3$ $(j = 1, 2, \ldots, n)$. The objective of the registration is to find a rigid transformation matrix M which makes the error metric e minimal, as shown in Eq.5

$$e = min \sum \|Mp_i - q_j\|^2 \tag{5}$$

We adopt RANSAC method to find the corresponding point pairs. The detail is described as follows:

(1) Select three points from P and Q respectively. P represents the centers of markers in patient space, and Q in image space.

(2) Calculate the rigid transformation matrix M.

(3) After P transformed by M, if the distance between p_i and its nearest point q_j in Q is less than a given threshold, then the point-pair (p_i, q_j) can be treated as qualified, otherwise not. The qualified point-pair is marked as inlier.

(4) If the number of the inliers reaches a certain number (or the ratio to the total point-pairs reaches a predetermined threshold), stop the iteration. Otherwise do again the last three steps. After identifying the set of correct corresponding points, we use Horn's method to get the rigid transformation matrix between the two spaces.

6 Result

We use simulated data to validate the proposed approach. In the surface of a head model, we paste eight markers randomly. After surface reconstruction, we get the mesh of the head model which is consisted of approximately 470,000 facets. Then the automatic approach proposed in Section 3 is used to identify the centers automatically. In our experiment, all 8 markers are correctly identified. The result is shown in Fig.9.

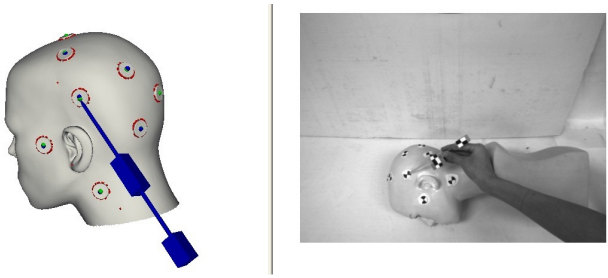


Fig. 9. Result of the registration. The left image shows the head model, and the right image is captured in patient space. The green points mean the centers detected in image space. The blue ones represent the center corners detected in patient space which is transformed into model space by the registration matrix.

On a PC of which the CPU is Intel Core i7 with 4GB memory, multi-scale spatial representation and feature point detection cost about 40 seconds. But this step can be done before the surgery, and save the results in a file. The clustering of the feature points is performed within 0.01 seconds. The registration of two space takes about 0.03 seconds.

In our experiment, we use cross-validation method to verify the algorithm. Among the points identified by the binocular camera, we choose a part of the points to do the registration, and the left points to verify. The error is evaluated by the target registration error (TRE), which is defined as follows:

$$TRE = \|Mp_i - q_j\| \tag{6}$$

In Eq.6, M is the transform obtained by the registration algorithm, p_i stands for the target selected from patient space, and q_j represent the same target from image space. Through 105 combinations of the data, the mean error is about 0.68 mm, the detail of the result is shown in Fig.10.

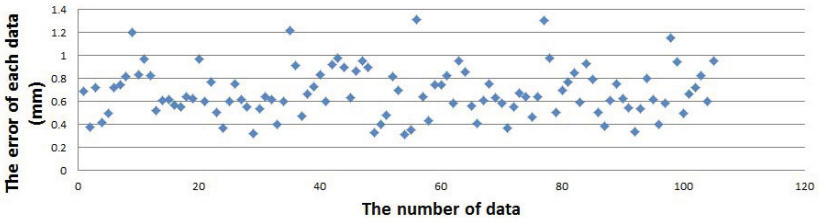


Fig. 10. The error distribution of 105 groups of data

7 Conclusion and Future Work

In this paper, we have proposed an automatic approach to localize the center of fiducial markers in image space and patient space respectively. In image space, we identify the markers based on multiple scale features and then cluster all this features to find the centers. While in patient space, we reconstructed the central corners based on binocular stereo vision. Experiments on simulated data showed that these approaches can automatically localize the centers of the fiducial markers with high precision in both image and patient space. The automatic methods lead to high precise in the patient-to-image registration and also saving the preparation time.

The future work will be extending our current methods to more diverse scenes and clinical data.

Acknowledgment. This research work is supported by National High Technology Research and Development Program of China under Grant No.2013AA013803.

References

1. Cleary, K., Peters, T.M.: Image-guided interventions: Technology review and clinical applications. Annual Review of Biomedical Engineering 12(1), 119–142 (2010)
2. Grimson, W.E.L., Ettinger, G.J., White, S.J., Lozano-Perez, T., Wells, W.M., Kikinis, R.: An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. IEEE Transactions on Medical Imaging 15(2), 129–140 (1996)
3. Lee, J.D., Huang, C.H., Wang, S.T., Lin, C.W., Lee, S.T.: Fast-micp for frameless image-guided surgery. Medical Physics 37(9), 4551–4559 (2010)
4. Fitzpatrick, J.M.: The role of registration in accurate surgical guidance. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine 224(5), 607–622 (2010)

5. Woerdeman, P.A., Willems, P.W., Noordmans, H.J., Tulleken, C.A., van der Sprenkel, J.W.: Application accuracy in frameless image-guided neurosurgery: a comparison study of three patient-to-image registration methods. Journal of Neurosurgery 106(6), 1012–1016 (2007)
6. Maurer Jr, C.R., Maciunas, R.J., Fitzpatrick, J.M.: Registration of head ct images to physical space using a weighted combination of points and surfaces. IEEE Transactions on Medical Imaging 17(5), 753–761 (1998)
7. Wang, M., Song, Z.: Automatic detection of fiducial marker center based on shape index and curvedness. In: Dohi, T., Sakuma, I., Liao, H. (eds.) MIAR 2008. LNCS, vol. 5128, pp. 81–88. Springer, Heidelberg (2008)
8. Wang, M., Song, Z.: Automatic localization of the center of fiducial markers in 3d ct/mri images for image-guided neurosurgery. Pattern Recogn. Lett. 30(4), 414–420 (2009)
9. Zheng, G., Gerber, N., Widmer, D., Stieger, C., Caversaccio, M., Nolte, L.P., Weber, S.: Automated detection of fiducial screws from ct/dvt volume data for image-guided ent surgery. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2325–2328 (2010)
10. Morevec, H.P.: Towards automatic visual obstacle avoidance. In: Proceedings of the 5th International Joint Conference on Artificial Intelligence, IJCAI 1977, vol. 2, pp. 584–584 (1977)
11. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of Fourth Alvey Vision Conference, pp. 147–151 (1988)
12. Trajkovic, M., Hedley, M.: Fast corner detection. Image and Vision Computing 16, 75–87 (1988)
13. Guo, Y., Yu, M., Sun, Y.: Study on advanced rapid corner detection method. Computer Engineering and Applications 47(12), 159–161 (2011)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
15. Jones, T.R., Durand, F., Desbrun, M.: Non-iterative, feature-preserving mesh smoothing. ACM Trans. Graph. 22(3), 943–949 (2003)
16. Lindeberg, T.: Scale-Space Theory in Computer Vision. Kluwer Academic Publishers, Norwell (1994)
17. Mikolajczyk, K.: Detection of local features invariant to affines transformations (July 2002)

A Method for Simulation Model Validation Based on Theil's Inequality Coefficient and Principal Component Analysis

Jiao Song, Li Wei, and Yang Ming

Control and Simulation Center
Harbin Institute of Technology, Harbin 150080, China
Jiaosong1985@163.com

Abstract. The creditability of simulation model is validated by the classical method of Theil's inequality coefficient though analyzing the consistency between the simulation output and reference output. The reference output is not treated as the benchmark for comparison in the classical method and the difference of trend between the simulation output and reference output is not considered. For solving the problems, the algorithm of Theil's inequality coefficient was improved, the models for describing the coincident degrees of position and trend between the simulation output and reference output were given and the simulation model validation method based on principal component analysis was proposed. The rationality and efficiency of the method were validated in the application.

Keywords: simulation, simulation model validation, Theil's Inequality Coefficient, Principal Component Analysis.

1 Introduction

Simulation is an integrated technology for scientific research with the model of actual system or imaginary system, which is based on similarity theory, control theory, computer technology, information technology and professional knowledge of its applications field [1] and [2]. Because the simulation models are the research objects in the simulation activity, the users of model pay more and more attention to the credibility of simulation models [3]. Analyzing the degree of consistency between the simulation output and the reference output is a common method for the simulation model validation. The classical methods include Theil's inequality coefficient (TIC) method [4-6], error analysis [7], gray relational analysis [8] and [9], similarity coefficient analysis [10] and frequency domain analysis method [11] and [12].

TIC method does not have any restrictions on the simulation output and reference output. Its principle is simple and it is easy to be used with very small amount of calculation. In 1978, TIC method was used to validate the credibility of the simulation model of a missile by Kheir and Holmes for the first time [4], which is widely used for the simulation model validation until now [5] and [6]. But, on the one hand the

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 126–135, 2013.

reference output is not considered as the benchmark in the classic method of TIC, on the other hand only the difference of position between the simulation output and the reference output is considered, without considering the differences of trend between them. In References 8 and 9, although taking into account the two differences of position and trend, the method does not have obvious geometric meaning and relies on the engineering experiences to synthesize the two differences, which cause the fact that the method is not suitable in some special cases.

This paper analyzes the flaws of the classic simulation model validation method based on TIC, thereby proposes an improving TIC model and gives the models for depicting the differences of position and trend between simulation output and reference output. Because there is a correlation between the two differences, the principal component analysis is used for synthesizing them to obtain validation results. In the application, the example proves that the method is rationality and efficiency.

2 Problem Description and Analysis

Suppose $X_r =< x_r(1), x_r(2), \cdots x_r(n) >$ and $X_s =< x_s(1), x_s(2), \cdots x_s(n) >$ are reference and simulation output respectively. The classic simulation model validation method based on TIC is given by

$$C(X_r, X_s) = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_r(i) - x_s(i))^2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_r(i)^2} + \sqrt{\frac{1}{n}\sum_{i=1}^{n}x_s(i)^2}} = \frac{\sqrt{\sum_{i=1}^{n}(x_r(i) - x_s(i))^2}}{\sqrt{\sum_{i=1}^{n}x_r(i)^2} + \sqrt{\sum_{i=1}^{n}x_s(i)^2}} \tag{1}$$

Where, $C(X_r, X_s)$ is the coefficient of TIC.

By (1), $C(X_r, X_s) \in [0,1]$ .When $C(X_r, X_s)$ is more close to 0, it indicates that X_r and X_s has the better degree of consistency, and the simulation model is more credible. When $C(X_r, X_s)$ is more close to 1, it indicates that the consistency between X_r and X_s is worse, and the simulation model is more incredible. TIC coefficient is a relative error and easy to understand and use, but there are two flaws when TIC coefficient is applied to the simulation model validation.

2.1 Benchmark Problem

In Fig.1 (a), $x_r(t) \equiv c_r$, $x_s(t) \equiv c_s$, $t = 1,2,\cdots n$, $c_r \le 0$, $c_s > 0$ and they are all constants. By the equation (1), $C(X_r, X_s) \equiv 1$ which indicates that the degree of consistency between X_r and X_s is the worst, which is obviously unreasonable. In fig 1(b), $x_r(t) = f(t)$, $x_{s1}(t) = f(t) + c$, $x_{s2}(t) = f(t) - c$, $t = 1,2,\cdots n$, $c > 0$ and they are all constants. $C(X_r, X_{s1}) = C(X_r, X_{s2})$ is obtained by intuition

while $C(X_r, X_{s1}) > C(X_r, X_{s2})$ is obtained by the equation (1). After analysis, we know that the above situation occurs because the equation (1) is used to calculate the relative error between X_r and X_s with the benchmark $\sqrt{\sum_{i=1}^{n} x_r(i)^2} + \sqrt{\sum_{i=1}^{n} x_s(i)^2}$, not the reference output directly.

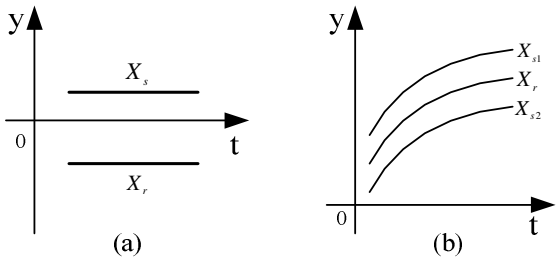


Fig. 1. Reference output and simulation output diagram

2.2 Trend Problem

In Fig.2 (a) and (b), X_{r1} and X_{r2} represent the reference outputs, X_{s1} and X_{s2} represent the simulation outputs. Using equation (1), we can get the conclusion, " the degree of consistency between X_{r2} and X_{s2} is better than that between X_{r1} and X_{s1} ".

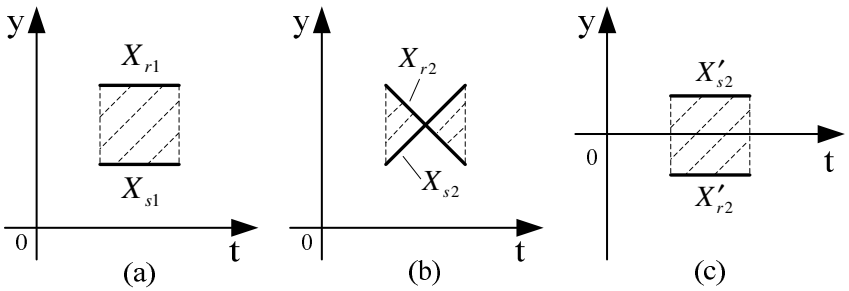


Fig. 2. Reference output and simulation output diagram

From Fig.2 (b), we know that the trends of X_{r2} and X_{s2} are increment and decrement respectively. It shows that completely different system characteristics are given by X_{r2} and X_{s2} respectively, which indicates that there is a serious error in the simulation model, so it is reluctant to consider the simulation model is credible. In Fig.2 (a), although there is a deviation between the X_{r1} and X_{s1}, the trends of

them are exactly the same, which indicates that between the simulation system and the reference system there are some tolerable deviations and the simulation output is considered as effective. Thus, the opposite conclusion to the equation (1) is got.

In Fig.2(c), the derivative functions of the outputs in Fig.2 (b) are given, which reflect the trends of X_r and X_s. From Fig.2 (c), there is a large difference between X'_{r2} and X'_{s2}, while the difference is not considered in the equation (1). So the conclusion is unreasonable. From the above, in order to obtain the consistency degree between X_r and X_s, we not only need to consider the difference of position between them, but also need to consider the difference of trend.

3 Simulation Model Validation Method

3.1 The Models for Depicting the Conformities of Position and Trend

For solving the benchmark problem, the following improved model is given based on the equation (1).

$$C(X_r, X_s) = \begin{cases} \dfrac{\sqrt{\sum_{i=1}^{n}(x_r(i)-x_s(i))^2}}{\sqrt{\sum_{i=1}^{n}x_r(i)^2}} & ,\sqrt{\sum_{i=1}^{n}x_r(i)^2} \neq 0 \\ \sqrt{\sum_{i=1}^{n}x_s(i)^2} & ,\sqrt{\sum_{i=1}^{n}x_r(i)^2} = 0 \end{cases} \tag{2}$$

By the equation (2), $C(X_r, X_s) \in [0, +\infty)$. For making the range of $C(X_r, X_s)$ be mapped to $(0,1]$, the model is given as follows:

$$D(X_r, X_s) = e^{-\xi C(X_r, X_s)} \tag{3}$$

Where, $D(X_r, X_s)$ represents the position conformity of X_r and X_s; $\xi > 0$ and it is the model parameter whose value is set by domain expert according to the purpose of simulation application.

By the equation (3), $D(X_r, X_s) \in (0,1]$. $D(X_r, X_s)$ is closer to 1 indicates that X_s is closer to X_r. $D(X_r, X_s)$ tends to 0 indicates that X_s is far from X_r.

The trends of X_r and X_s could be approximately depicted by X'_r and X'_s respectively, which is defined as follows:

$$X'_r = < x'_r(2), x'_r(2), \cdots x'_r(n-1) > = < \frac{x_r(2)-x_r(1)}{\Delta t}, \frac{x_r(3)-x_r(2)}{\Delta t}, \cdots \frac{x_r(n)-x_r(n-1)}{\Delta t} > \tag{4}$$

$$X'_s = < x'_s(2), x'_s(2), \cdots x'_s(n-1) > = < \frac{x_s(2)-x_s(1)}{\Delta t}, \frac{x_s(3)-x_s(2)}{\Delta t}, \cdots \frac{x_s(n)-x_s(n-1)}{\Delta t} > \tag{5}$$

It is the same with the position conformity of X_r and X_s, the trend conformity is described as follows:

$$T(X_r, X_s) = e^{-\lambda\delta(X_r', X_s')} \tag{6}$$

$$\delta(X_r', X_s') = \begin{cases} \dfrac{\sqrt{\sum_{i=1}^{n-1}(x_r'(i) - x_s'(i))^2}}{\sqrt{\sum_{i=1}^{n-1} x_r'(i)^2}} & , \sqrt{\sum_{i=1}^{n-1} x_r'(i)^2} \neq 0 \\ \sqrt{\sum_{i=1}^{n-1} x_s'(i)^2} & , \sqrt{\sum_{i=1}^{n-1} x_r'(i)^2} = 0 \end{cases} \tag{7}$$

Where, $T(X_r, X_s)$ represents the trend conformity of X_r and X_s; $\lambda > 0$ and it is the model parameter whose value is set by domain expert according to the purpose of simulation application.

Further, inserting the equations (4) and (5) into $\delta(X_r', X_s')$ gives

$$\delta(X_r', X_s') = \begin{cases} \dfrac{\sqrt{\sum_{i=1}^{n-1}((x_r(i+1) - x_r(i)) - (x_s(i+1) - x_s(i)))^2}}{\sqrt{\sum_{i=1}^{n-1}(x_r(i+1) - x_r(i))^2}} & , \sqrt{\sum_{i=1}^{n-1} x_r'(i)^2} \neq 0 \\ \sqrt{\sum_{i=1}^{n-1}(x_s(i+1) - x_s(i))^2} & , \sqrt{\sum_{i=1}^{n-1} x_r'(i)^2} = 0 \end{cases} \tag{8}$$

3.2 The Integration of Position Conformity and Trend Conformity

Suppose $O_s = \{X_{s1}, X_{s2}, \cdots X_{sp}\}$ is the output set of simulation models to be selected. For selecting the most credible model from the p simulation models, the degrees of consistency between all the simulation output and the reference output should be ranked. To solve this problem, $D(X_r, X_s)$ and $T(X_r, X_s)$ need be considered at the same time. Considering the correlation between $D(X_r, X_s)$ and $T(X_r, X_s)$, in order to extract the comprehensive information of both for validating the simulation models, in this paper, the simulation model verification method based on principal component analysis is proposed [13]. According to equations (3) and (6), the matrix can be obtained as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{p1} & a_{p2} \end{bmatrix} \tag{9}$$

Where $a_{i1} = D(X_r, X_{si})$, $a_{i2} = T(X_r, X_{si})$ and $i = 1, 2, \cdots p$.

a_{i1} and a_{i2} denote the position conformity and trend conformity respectively. Because they have different meanings, before integration a standardized processing of both should be done as follows [14]:

$$s_{ij} = \frac{a_{ij}}{\overline{a}_{\bullet j}}, i = 1, 2, \cdots p, \quad j = 1, 2 \tag{10}$$

Where s_{ij} is the sample data after standardized processing and $\overline{a}_{\bullet j} = \sum_{k=1}^{p} a_{kj} \Big/ p$.

Then the matrix below is obtained:

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \\ \vdots & \vdots \\ s_{p1} & s_{p2} \end{bmatrix} \tag{11}$$

The two columns of data in S are considered as p samples of the random variables s_1 and s_2, a combination is defined as $y = k_1 s_1 + k_2 s_2$, and $k_1^2 + k_2^2 = 1$. For making the differences of s_1 and s_2 among the p samples be expressed comprehensively through y, $s = [s_1, s_2]^T$ is defined and its covariance is as follows:

$$V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \tag{12}$$

Where $v_{ij} = Cov(s_i, s_j)$, $i = 1, 2$ and $j = 1, 2$.

Let $k = [k_1, k_2]^T$, $k^T k = 1$, and $y = k^T s$. Thereby the variance of y is:

$$Var(y) = Var(k_1 s_1 + k_2 s_2) = k^T V k \tag{13}$$

To make the variance of y be as large as possible, a function is constructed as follows:

$$h(k) = k^T V k - 2\lambda(k^T k - 1) \tag{14}$$

To get the maximum value of h, let its partial derivative be 0 and the following equation is obtained.

$$\frac{\partial h(k)}{\partial k} = 2Vk - 2\lambda k = 0 \tag{15}$$

Thus the equation can be obtained as follows:

$$Vk = \lambda k \tag{16}$$

This indicates that k is the eigenvector of V. Inserting equation (15) into equation (12) gives

$$Var(y) = k^T Vk = k^T \lambda k = \lambda \tag{17}$$

By equation (17), the largest characteristic root of V is the maximum variance of y, and its corresponding unit eigenvector is k. So, according to the sample of s, the value of y corresponding to each simulation output can be obtained. Finally, the degrees of consistency between all the simulation output and the reference output can be ranked according to y, and the most credible model can be judged.

4 Application

The simulation output set which contains nine simulation outputs is expressed as $O_s = \{X_{s1}, X_{s2}, X_{s3}, X_{s4}, X_{s5}, X_{s6}, X_{s7}, X_{s8}, X_{s9}\}$. As shown in Fig.3, the simulation output is denoted by star-shaped dots in each figure, and X_r is the reference output denoted by round dots. The sampling period is one second, and the concerned time interval is from 0 to 20 seconds.

If only the position difference between the simulation output and the reference output is consider, with using the equation (1) for validating the simulation models, the validation results are obtained as shown in Table 1.

Table 1. Validation results

sequence	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
$D(X_r, X_s)$	0.138	0.138	0.172	0.107	0.089	0.149	0.191	0.377	0.109
result	4	4	7	2	1	6	8	9	3

As known from Fig.3, X_{s2} has two very distinct peaks, the consistency degree between it and X_r should be less than that between X_{s1} and X_r, but from table1 both X_{s2} and X_{s1} have the same consistency degree with X_r. X_{s9} has a greater volatility, the consistency degree between it and X_r should also be less than that between X_{s1} and X_r.However, from the results in table 1, X_{s9} has better consistency degree with X_r than X_{s1}. In addition, X_{s3} have the same change trend as X_r with only a certain gap in the spatial position, but X_{s3} has relatively poor consistency degree with X_r comparing with X_{s2} and X_{s9}.The reason for the results is that the classic simulation model validation method based on TIC only considers the position difference between the simulation output and the reference output but the trend difference between them.

Considering the both factors of position and trend at the same time, the model parameters in both equations (3) and (6) are set 1, and after obtaining $D(X_r, X_s)$ and $T(X_r, X_s)$ the results are shown in tables 2 and 3 with standardized processing via using equation (10).

Table 2. Position conformity between simulation output and reference output

sequence	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
$D(X_r, X_s)$	0.729	0.733	0.663	0.813	0.840	0.724	0.690	0.555	0.800
s_1	1.002	1.008	0.911	1.118	1.155	0.995	0.948	0.763	1.100

Table 3. Trend conformity between simulation output and reference output

sequence	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
$T(X_r, X_s)$	1.000	0.050	1.000	0.517	0.730	0.652	0.368	0.185	0.192
s_2	1.917	0.096	1.917	0.991	1.400	1.250	0.706	0.355	0.368

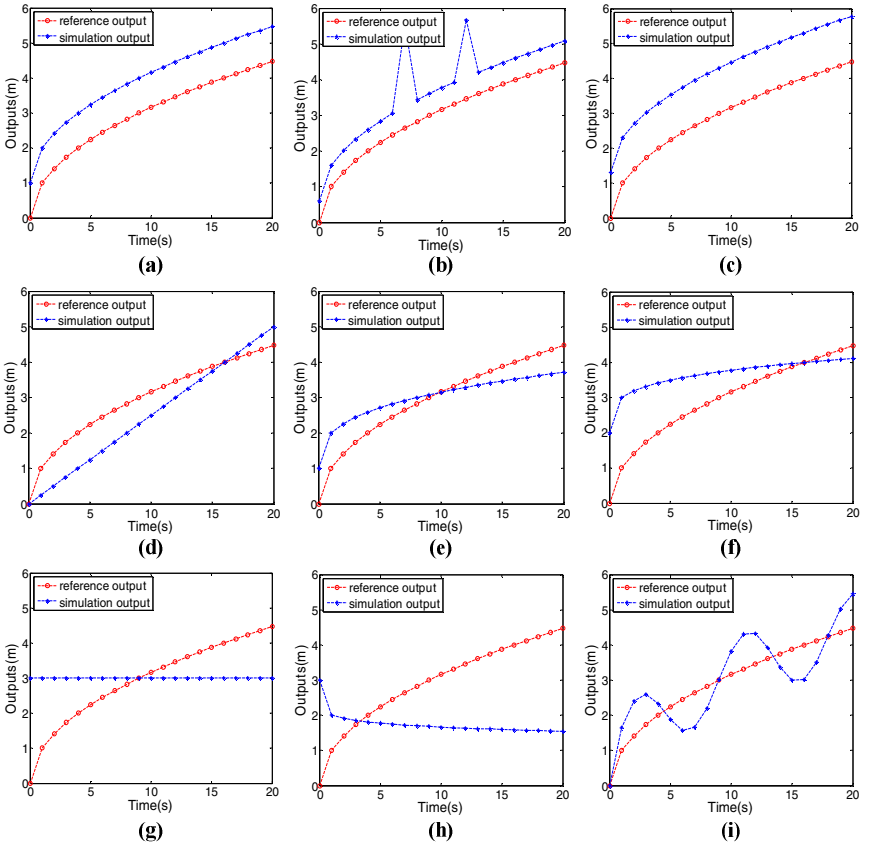


Fig. 3. Nine groups of simulation output and the reference output

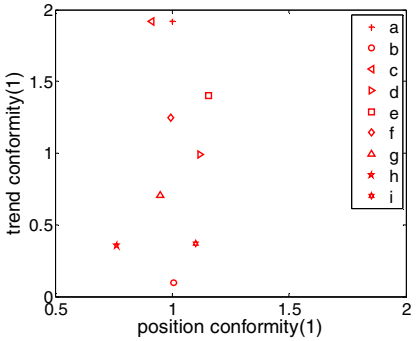


Fig. 4. Nine groups of location conformity and trend conformity

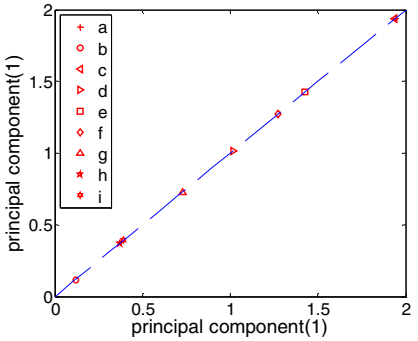


Fig. 5. Nine groups of principal components

As shown in Fig.4, the difference among the position conformities of nine simulation outputs is not significant. So the differences among the consistency degree of nine simulation outputs should be determined primarily by the trend conformities of them. Equation (18) is obtained according to equations (11) and (15) as follows:

$$y = 0.0223s_1 + 0.9998s_2 \tag{18}$$

The validation results shown in table 4 are obtained according to equation (18). From table 4, X_{s1} has the consistency conformity degree with the reference output in all simulation outputs. Because X_{s2} has peaks, X_{s9} has a greater volatility, and there exists large differences between X_{s8} and X_r both in the position and trend, so X_{s2}, X_{s9} and X_{s8} be the worst three simulation outputs. Although $D(X_r, X_{s1})$ and $D(X_r, X_{s3})$ is smaller comparing with $D(X_r, X_{s4})$, $D(X_r, X_{s5})$, $D(X_r, X_{s6})$ and $D(X_r, X_{s7})$, the trends of them are consistent with the reference output. So X_{s1} and X_{s3} become the best two simulation outputs.

Table 4. Validation results with considering the differences of both position and trend

sequence	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
y	1.940	0.118	1.937	1.016	1.425	1.272	0.727	0.372	0.393
result	1	9	2	5	3	4	6	8	7

5 Conclusion

This paper focused on the study of the validation of simulation model based on TIC. Our goal is to improve the classic validation method based on TIC which does not make the reference output as the benchmark and does not consider the difference of trend between the simulation output and the reference output. In this paper, the TIC model is improved, the measurement models of position and trend conformities

between the simulation output and the reference output are given, and principal component analysis is used to integrate the position and trend conformities for validating the simulation models. In the application, the most credible model in the nine provided models is selected. Future research will concentrate on how to integrate with the expert knowledge to validate simulation model.

References

1. White, K.P., Ingalls, R.G.: Introduction to Simulation. In: Proceedings of the 2009 Winter Simulation Conference, pp. 12–23 (2009)
2. Sanchez, S.M.: Better Than A Petaflop: The Power of Efficient Experimental Design. In: Proceedings of the 2008 Winter Simulation Conference, pp. 73–84 (2008)
3. Min, F.Y., Yang, M., Wang, Z.C.: Knowledge-based method for the validation of complex simulation models. Simulation Modelling Practice and Theory 18, 500–515 (2010)
4. Kheir, N.A., Holmes, W.M.: On Validating Simulation Models of Missile Systems. Simulation 30(4), 117–128 (1978)
5. Hvala, N., Strmcnik, S., Sel, D., et al.: Influence of model validation on proper selection of process models—an industrial case study. Computers and Chemical Engineering 7(29), 1507–1522 (2005)
6. Huilinir, C., Romero, R., Munoz, C., et al.: Dynamic modeling of partial nitrification in a rotating disk biofilm reactor: Calibration, validation and simulation. Biochemical Engineering Journal 1(52), 7–18 (2010)
7. Damborg, M.J.: An example of error analysis in dynamic model validation. Simulation 44(6), 301–305 (1985)
8. Sun, Y.C., Zhou, X.Z., Li, G.F., et al.: Validation of Simulation Models Based on Grey Relational Analysis and Improvement. Journal of System Simulation 17(3), 522–524 (2005)
9. Wu, J., Wu, X.Y., Chen, Y.X., et al.: Validation of simulation models based on improved grey relational analysis. Systems Engineering and Electronics 32(8), 1677–1679 (2010)
10. Liu, Z.Z.: Model and Simulation Validation Based on the Data of the Aero Experimentation. Journal of System Simulation 14(3), 281–284 (2002)
11. Li, P.B., Gao, X.: Application of MESA on Validating Missile Simulation Model. Journal of National University of Defense Technology 21(2), 9–12 (1999)
12. Montgomery, D.C., Conard, R.G.: Comparison of simulation and flight-test data for missile systems. Simulation 34, 63–72 (1980)
13. Pan, C.G., Chen, Y.W., Wang, H.: Principal Component Analysis' Application to the Software Metrics-based for Risk Assessment. Operations Research and Management Science 14(5), 80–84 (2005)
14. Xu, Y.J., Wang, Y.Z.: The Improvement of the Application Method of Principle Component Analysis. Mathematics in Practice and Theory 36(6), 68–75 (2006)

Compression for Large-Scale Time-Varying Volume Data Using Spatio-temporal Features

Kun Zhao[1], Naohisa Sakamoto[2], and Koji Koyamada[2]

[1] Graduate School of Engineering, Kyoto University, Japan
[2] Institute for the Promotion of Excellence in Higher Education, Kyoto University, Japan
{zhao.kun,naohisas}@viz.media.kyoto-u.ac.jp,
koyamada.koji.3w@kyoto-u.ac.jp

Abstract. Data compression is always needed in large-scale time-varying volume visualization. In some recent application cases, the compression method is also required to provide a low-cost decompression process. In the present paper, we propose a compression scheme for large-scale time-varying volume data using the spatio-temporal features. With this compression scheme, we are able to provide a proper compression ratio to satisfy many system environments (even a low-spec environment) by setting proper compression parameters. After the compression, we can also provide a low-cost and fast decompression process for the compressed data. Furthermore, we implement a specialized particle-based volume rendering (PBVR) [2] to achieve an accelerated rendering process for the decompressed data. As a result, we confirm the effectiveness of our compression scheme by applying it to the large-scale time-varying turbulent combustion data.

Keywords: compression, time-varying, volume data, visualization.

1 Introduction

Time-dependent simulations can be found in many scientific fields (e.g. computational fluid dynamics, electromagnetic field simulation or ocean prediction). Good visualization for these time-dependent simulations is very important to clearly show changes and variations over time. Since the time-varying volume data from these simulations always have a very large data size, compression for the volume data is always needed to achieve an efficient visualization.

On the other hand, in some cases a high compression ratio is of course needed, low-cost and fast decompression for the compressed data is also required. For example, as an application of the visualization for ocean prediction dataset, “smart fishing" [7] is one of the cases that require a high compression ratio and also a low-cost and fast decompression. “Smart fishing” means efficient fishing (e.g. rapid search for fishery grounds) by using ocean prediction dataset. This dataset is the large-scale time-varying volume data, and needs to be transferred to fisherman to provide a detailed and interactive analysis for fishery grounds. Since the wireless communication speed on the ocean (mainly through the satellite) is slow, the

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 136–148, 2013.

large-scale time-varying ocean data need to be compressed into a small data size to transfer it to the fisherman. Furthermore, the decompression process should be very low-cost and fast to satisfy low-spec hardware such as a tablet, which the fisherman may hold.

In the previous researches, there are many volume compression methods proposed [3, 4, 5, 6, 8, 9] to achieve the time-varying volume compression. However, these compression approaches always need a costly decompression process or would generate a large decompression size, which would not fit for the application requiring low-cost decompression (e.g. “smart fishing”).

To solve this problem, in the present paper we propose a volume compression scheme based on the spatio-temporal features of the time-varying volume data. In our proposed compression scheme, there are two compression processes performed to separately utilize the spatial features and temporal features of the large-scale time-varying data. In our previous work [1], we have proposed a compression scheme to utilize spatial features. In the present work, we add temporal domain compression to the previous method to achieve higher compression ratio. In detail, to utilize the temporal feature, we first calculate the temporal coherence between every two consecutive time steps, and then delete the vertices sharing coherence with the previous time step to compress the volume into a smaller data size. The compressed volume data is stored with a vertex table, which records the coherence information for each vertex, and a compressed values array. This simple structure could provide a fast decompression process (see details in section 3.2). Additionally, we provide two parameters: block size and tolerance rate, to control the precise of the compression precise and compression ratio. As a result, our compression scheme could provide a proper compression ratio and a low-cost decompression process. Moreover, we also implement a special method into the particle-based volume rendering (PBVR) [2] to render the decompressed volume data. As a result, the temporal features could not only be used to compress the volume data but also to accelerate the rendering process of PBVR.

To verify the efficiency of our proposed compression scheme, we apply our system to the large-scale time-varying turbulent combustion volume data. The experimental results show the efficiency of our proposed method.

2 Related Work

The large-scale time-varying volume data makes visualization a challenging problem. Due to the large data size, many volume compression techniques have been proposed to decrease the data size so that the rendering process could be done more easily.

Jens Schneider et al. [4] proposed a vector-quantization-based compression scheme for both static and time-varying volume data. With vector-quantization, this compression scheme could achieve a high compression ratio. However, it requires a costly decompression process, which could lead to a low frame-rate rendering result. Chaoli Wang et al. [9] utilized a wavelet-based time-space partitioning (WTSP) tree to compress the time-varying volume data in spatio-temporal domain to achieve an

efficient rendering. Wavelet is a very powerful compression technique. However, the decompression algorithm would cost much time to perform the inverse wavelet transform. Moreover, the preprocess for compression of wavelet is also costly. Sohn et al. [8] described a compression scheme for encoding time-varying volumetric features to support isosurface rendering. It is also based on a wavelet transform with temporal encoding so that the decompression process would need a costly inverse wavelet transform. Weiler et al. [6] proposed a hierarchical wavelet functional representation approach for interactive volume rendering, which also requires a inverse wavelet transform for decompression. Jorg Mensmann et al. [5] proposed a GPU-supported compression scheme for time-varying volume data. It needs to convert simulation data from 32-bit float into 16-bit integration as a preprocess to make sure the data size could be stored in GPU memory. However, the convert may lead to a large data loss because the the precise is decreased from 32-bit to 16-bit. Moreover, they use Lempel-Ziv-Oberhumer (LZO) as the compression algorithm, which could provide a fast decompression speed. But the decompressed data will have a same size with the original data. As for some time-varying data with a very high spatial resolution, the GPU memory still runs a risk of insufficiency especially for a low-spec hardware. The same problem could be found in the work of Yun Jang et al. [3], who used a functional representations to visualize the time-varying data. The decoded frame would need a 3D texture size depends on the original input data. This also would lead to an insufficiency of graphics memory for a low-spec hardware.

In conclusion, even though these compression approaches could achieve a high compression ratio for the time-varying volume data, they always need a costly decompression process or a large decompression size or a costly rendering process, which are not suitable for low-spec hardware.

3 Spatio-temporal Compression

In this paper, we propose a volume compression scheme based on the spatio-temporal features of the time-varying volume data. This compression scheme contains two compression processes: spatial domain compression and temporal domain compression, which to utilize the spatial features and temporal features. Here, the spatial features represent the properties in the volume structure for each independent time step. The temporal features mean the properties the volume data shares by consecutive time steps. These features could be utilized to decrease the volume size.

3.1 Spatial Domain Compression

For the first process, it is the spatial domain compression. We have proposed this compression process last year [1]. Here we make a brief review on it.

With the structured volume data, we use the two-level division to reduce the vertex number firstly, and fast cubic b-spline to reconstruct the divided volume (Figure 1). The two-level division first divides the volume into many blocks with the same cell number on each side (block size), and then subdivides every block into 24 tetrahedra.

After that, the value of newly generated vertex (black dots in Figure 1) is evaluated by fast cubic b-spline. We hereinafter call this structure as "24-tetrahedra" mesh.

After the compression, we implement a special compression data structured to store the "24-tetrahedra" mesh. The compressed data does not need the coordinate and connection information for the compressed tetrahedral mesh. Generally, the tetrahedral volume data needs the coordinate and connection data for the tetrahedral structure. However, in our system, since the division is done uniformly for the whole volume, a proper order of the values array is enough for the "24-tetrahedra" mesh. As a result, the compressed data contains: the block size used in the block division, the original volume resolution and the ordered values array for the divided mesh. Since the vertex number is decreased in the division process, the compressed data can have a very smaller size than the original data. Note that, different block size can generate different vertex number. Hence, we can provide a different compression ratio with a different block size.

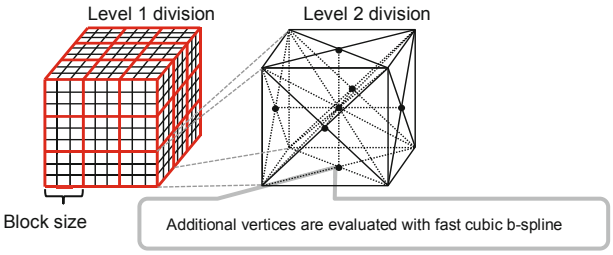


Fig. 1. Spatial domain compression with block size = 3. The left figure shows the first-level division and the right figure shows the second-level division. The black knot means the newly generated vertices by the second-level division.

3.2 Temporal Domain Compression

In many cases, time-varying volume data always keep an invariant mesh. We call such a kind of data as the mesh-invariant time-varying volume data. Since this kind of data often contains much temporal coherence, some parts of the volume may have a strong similarity for the consecutive time steps. Hence, we can achieve volume compression by deleting these similarity parts.

In our proposed technique, we explore the temporal coherence by calculating the difference of the vertex value between every two consecutive time steps. With the temporal coherence, we could compress the volume data by deleting the coherency vertices. At the meanwhile, we keep a very simple structure - vertex table (VT) to manage the coherency property for each vertex, which is referred to reconstruct the volume during the decompression. This simple structure is able to provide a low-cost decompression process. In general, a tree structure such as time-space partitioning (TSP) tree [10] is usually used to manage the time-varying volume to explore the temporal coherence [8, 9]. However, the tree traverse process of the TSP tree during the decompression process is very costly.

(A) Algorithm

Assume two consecutive time steps as the previous time step and current time step; we calculate the value of vertex table for the *i*th vertex as:

$$VT_i = \begin{cases} 1, & \left|V_i^{prev} - V_i^{curr}\right| < tolerance \\ 0, & else \end{cases} \quad (1)$$

Where V_i^{prev} represents the *i*th vertex value of the previous step, and V_i^{curr} represents the *i*th vertex value of the current step. Obviously, "1" means the vertex is coherency, and "0" means the vertex is not coherency. User can set a tolerance rate ϵ, and our system will calculate the tolerance as:

$$tolerance = \epsilon \times (V_{max}^{prev} - V_{min}^{prev}) \quad (2)$$

Here, V_{max}^{prev} and V_{min}^{prev} respectively means the maximum value and minimum value of the previous step. With the user assigned tolerance rate ϵ, the compression algorithm is shown as follows:

1. Calculate VT_i for the *i*th vertex with equation 4.2.
2. If VT_i = 1, delete this vertex from the current time step. If VT_i = 0, store this vertex into the compressed data. Move to the $i + 1$th vertex.
3. f i < N, move to step 1. If not, set i = 0 and perform step 1, 2, 3 to the next consecutive time steps.

(B) Compressed file

With the above compression algorithm, the coherency vertices are deleted. At the meanwhile, the vertex table is also stored with the compressed volume. We use a special structure to store the compressed volume (Figure 2), which contains vertices number, vertex table and compressed values array.

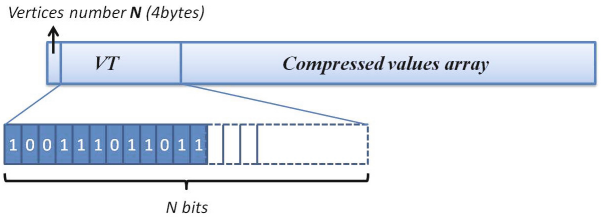


Fig. 2. The structure of temporal compressed volume file. *N* is the vertex number of the spatially compressed volume.

Obviously, if the time-varying volume data contains much temporal coherence, many coherence vertices could be deleted and a high compression ratio can be provided. We also allow user to change the tolerance rate ϵ to control the compression precise. In general, a high tolerance rate can provide a high compression ratio. On the contrary, a low tolerance rate would provide a relatively low compression ratio while keep a high compression precise.

4 Decompression and Visualization

4.1 Decompression

With compressed data, the decompression process only needs to copy the coherency vertices from the previous time step. Note that our decompression is not to reconstruct the compressed data into the original mesh but to reconstruct it into the "24-tetrahedra" mesh. That is because the original mesh has a large data size, which is very hard to handle. If we decompress the compressed data into the original size, the compression would be meaningless.

In detail, for each time step, we check the VT to determine whether to copy the vertex from the previous time step or not. VT is a simple array so that we do not need the costly tree traverse process, which are often used in other compression method to manage the compressed data structure [8, 9]. For each vertex, if the VT value for the vertex is "1", then copy the vertex from the previous time step; if not, use the vertex stored in the compressed data for this time step. This process is very fast since we only need to conduct the simple vertex copying process.

4.2 Visualization with Specialized PBVR

After the decompression process, we use PBVR to render the decompressed "24-tetrahedra" mesh. PBVR [2] is a stochastic rendering method, which first generates particles for the volume data and then projects particles to get the rendering image. The particle generation is performed in cell-by-cell manner and is done multiple times (repetition level) by using different random number. Generally, a higher repetition level can provide a better image quality. Because PBVR is very suitable for the large-scale irregular volume data [2], we use it to render the "24-tetrahedra" mesh. However, the particle generation always cost much time if the mesh contains many cells. In fact, since the time-varying volume data often contains much temporal coherence, these coherence is also be able to accelerate the particle generation process.

(A) Calculation for CT

Since PBVR generate particles in cell-by-cell manner, we utilize a temporal coherency cells table (CT) to record the coherence property for every cell. This temporal coherency CT could be simply calculated from VT. Basically, we use the following criterion to calculate the *i*th cells table value CT_i:

- CT_i = 1, if and only if the vertex values belonging to this cell are all calculated as "1" in VT.
- CT_i = 0, for any other case.

Figure 3 shows an example to calculate the cells table for the decompressed volume data. Since the decompressed volume data have a tetrahedral mesh, we need to check four vertex values, which belong to some cell, to calculate the CT value for this cell. As a result, the coherency cells are marked with "1" and none-coherency cells are marked with "0".

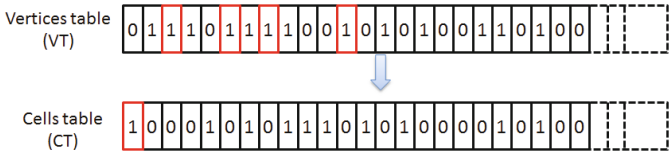


Fig. 3. The process of calculating the cells table. Since a tetrahedral cell has four vertices, every value in the CT needs to check four values on VT, which belongs to this cell.

(B) Generate Particles with CT

After the calculation of CT, the next step is to generate particles with CT. It is obvious that, we only need to generate particles for the none-coherency cell to save the generation time. Assume the cell number of a certain time step as N_c, the generation process is listed as following:

1. For the ith cell, check the value of CT_i.
2. If $CT_i = 0$, generate particles for this cell, and store the generated particles number into a particles table (PT). If $CT_i = 1$ skip this cell.
3. If $i \leq N_c + 1$, do $i = i + 1$ back to step 1. If not, move to the next time step, back to the step 1.

Figure 4 shows an example of generating particles with CT. Since the first time step does not have a CT, our system would first generate particles for all cells of this time step. For the following steps, our system only generates particles for the cell with the CT value equal to 0 and skips the cell with the CT value equal to 1. Moreover, to manage the particles, we also keep a particles table (PT) that stores the number of particles for each cell. During the rendering process, PT would be referred to copy the particles from the previous time step for the coherency cell.

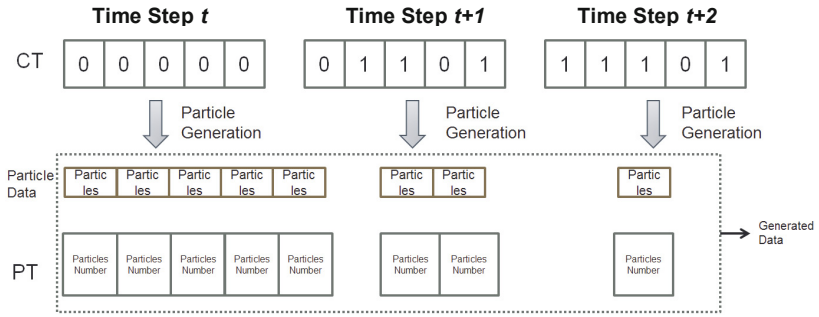


Fig. 4. The particle generation process with CT

(C) Rendering with CT and PT

With the partly generated particle data, we need to reconstruct it into the complete data before we render it into image plane. Here, the reconstruction process of the particle data mainly contains the copy process for the coherency cell. This copy process is performed with referring CT and PT simultaneously (Figure 5). In detail, the reconstruction process for the particle data is shown as follows:

1. For the *i*th cell, check the value of CT_i.
2. If $CT_i = 0$, do $PT_i^{curr} = PT_j^{part\_curr}$. And copy the particles from the partly generated particle data by referring the value of PT_i^{curr} to get the particle index of the particles needed to be copied. Then, perform $i = i+1$; $j = j+1$, move to step 4.
3. If $CTi = 1$, do $PT_i^{curr} = PT_i^{prev}$. And copy the particles from the reconstructed particle data of the previous time step by referring the value of PT_i^{curr} to get the particle index of the particles needed to be copied. Then, perform $i = i + 1$, move to step 4.
4. If $i \leq N_c + 1$, back to step 1. If not, move to next time step, back to step 1.

Here, PT_i^{curr} represents the PT value for the current time step of cell i. $PT_j^{part\_curr}$ represents the PT value for the partly generated particle of the current time step of the cell i (PT_0 in Figure 5). PT_i^{prev} represents the PT value for the reconstructed particle data of the previous time step of the cell i (Reconstructed PT in Figure 5). Figure 5 shows an example for the particle reconstruction process. After the reconstruction process, the particle data is projected to the image plane so that we can obtain the visualization result for the time-varying volume data.

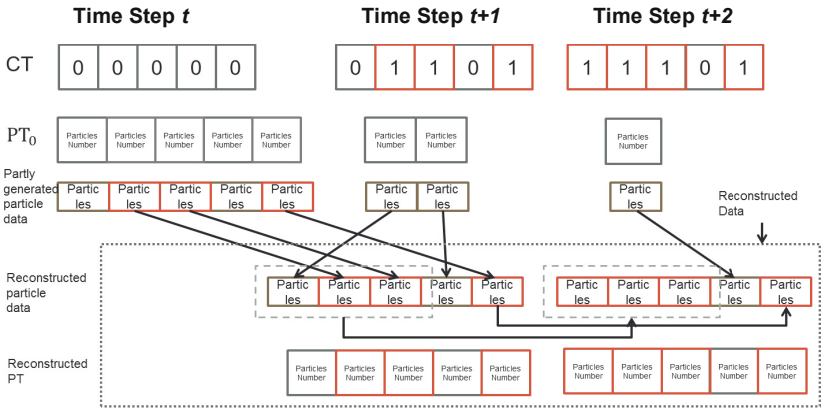


Fig. 5. The particle reconstruction process with CT

5 Experiments and Results

In this paper, we apply our method to the large-scale time-varying dataset obtained from a turbulent combustion simulation. The grid structure is Cartesian with uniform spacing and it is mesh-invariant time-varying volume data. There are 480×720×120 voxels composed in float type, and a total of 122 time steps. At each time step, there are 5 variables and about 158.2MB for each variable. In our experiment, we use the Y modality data and it reaches 18.9GB for all time steps. The experiment is conducted with an Intel Core i7-2820QM CPU (2.3 GHz), an NVidia GeForce GTX580M 2GB GPU, and 16GB system memory. The operation system is Ubuntu 12.04 LTS.

5.1 Compression Ratio

In our experiment, the compression ratio has been calculated as original data size over compressed data size. The compression ratios with different block size and tolerance rates are shown in Table 1. From this table we can see, a larger block size and a higher tolerance rate can provide a higher compression ratio. Of course, a smaller block size could provide a finer mesh and a lower tolerance rate could also generate a lower compression error.

Table 1. Compression ratio with different block size and tolerance rate

Tolerance Rate \ Block Size	2	3	4	5	6	7
10^{-7}	3.95:1	13.40:1	31.65:1	61.35:1	105.26:1	166.67:1
10^{-6}	4.14:1	14.06:1	33.11:1	64.52:1	111.11:1	175.44:1
10^{-5}	4.39:1	14.88:1	35.09:1	68.03:1	117.65:1	185.19:1
10^{-4}	4.92:1	16.67:1	39.37:1	76.34:1	131.58:1	208.33:1
10^{-3}	6.15:1	20.83:1	49.26:1	95.24:1	163.93:1	256.41:1

5.2 Compression Error

Except for compression ratio, a quantitative metric to evaluate the error is also needed. We use the following expression to evaluate the compression error (*CE*):

$$CE = \frac{\sum_{i=1}^{N} \left| V_{ori}^{i} - V_{com}^{i} \right|}{N} \tag{6}$$

Here N is the vertex number in the original volume. V_{ori}^{i} is the normalized value for vertex i of the original volume. Since the compressed volume is in "24-tetrahedra" structure, we linearly interpolate this mesh into the uniform mesh that is same with the original volume. V_{com}^{i} is the normalized value for vertex i of the interpolated volume. Table 2 shows compression error generated for the time step 35.

Table 2. Compression error with different block size and tolerance rate (time step 35)

Tolerance Rate \ Block Size	2	3	4	5	6	7
10^{-7}~10^{-3}	0.24%	0.46%	0.65%	0.99%	1.24%	1.65%

In fact, in our experiment we found that the compression error is mainly affected by spatial domain compression process. For example, Table 3 shows the compression error generated by the temporal domain compression separately. As we can see this compression error is much smaller than the results shown in Table 2. That is why we have the same total compression error for different tolerance rate (Table 2).

Table 3. Separate compression error for temporal domain compression (block size = 2, time step 35)

Tolerance Rate	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}
Compression Error	8.11×10^{-10}	8.03×10^{-9}	8.03×10^{-9}	8.03×10^{-9}	8.03×10^{-9}

5.3 Decompression Speed

With our proposed compression algorithm, the decompression time are measured with different block sizes and different tolerance rates. Here, the decompression time for all the time steps is shown in Table 4. From this table we could see that, since we do not need any value extraction process during the decompression, it is performed as fast as about 6 second or 7 seconds for all time steps.

Table 4. Decompression time with different block size for all time steps

Tolerance Rate \ Block Size	2	3	4	5	6	7
10^{-7}	6.63s	6.07s	5.92s	5.86s	5.84s	5.83s
10^{-6}	6.78s	6.17s	6.00s	5.96s	5.95s	5.92s
10^{-5}	6.88s	6.16s	6.01s	5.97s	5.93s	5.87s
10^{-4}	7.02s	6.37s	6.24s	6.18s	6.16s	6.13s
10^{-3}	6.92s	6.30s	6.19s	6.09s	6.07s	6.06s

5.4 Rendering with PBVR

After the decompression for the volume data, we render the decompressed data with our specialized PBVR process. Table 5 shows the speed of the particle generation with the decompressed volume data. Here, the block size is 2 and repetition level is 144. Since different tolerance rate would also generate different temporal coherence for the compressed volume data, we show the particle generation time for different tolerance rate.

Table 5. The result of particle generation for all time steps (block size)

Tolerance Rate	N/A	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}
Particle Generation Time	3,947.9s	3,256.8s	3,214.5s	3,149.1s	3,149.1s	2,786.1s

To make a comparison, the particle generation time without using the coherence cells table is shown in "N/A" row. This result shows our specialized PBVR can save much particle generation time (especially for a tolerance rate of 10^{-3}), which verifies the utilizing of temporal coherence into the PBVR process is very efficient.

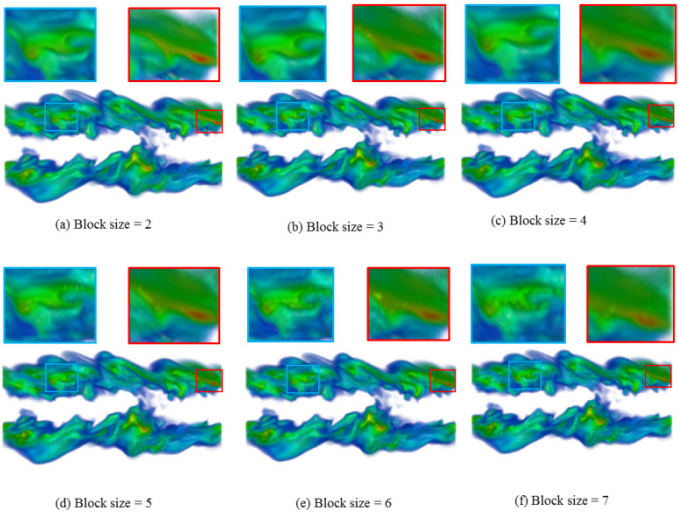


Fig. 6. Rendering result for the spatio-temporally compressed volume data with different block size and a tolerance rate of 10^{-3} (PBVR with phong shading and repetition level of 144). The part in the frame shows the detailed change with different block sizes.

Moreover, the rendering result for different block size with a tolerance rate of 10^{-3} is also shown in Figure 6. Here, PBVR is done with phong shading and a repetition level of 144. As mentioned in section 5.2, compression error is mainly affected by the block size, so that the image quality is also mainly affected by block size. It could be observed that finer details of the data are kept better when we have a low block size (see the blue frame of Figure 6). But we could also see that, even a larger block size such as 7, the important features (such as the red part in the red frame of Figure 6) could still be kept.

6 Discussion

From the experimental results we could see, our compression scheme could compress the time-varying data into different size with different compression parameters - block size and tolerance rate, which are used to control the calculation precise for spatial features and temporal features. Thus, we could provide a proper compression ratio to satisfy many system environments (even a low-spec environment) by setting proper compression parameters. This is a main feature of our compression scheme. Moreover, another main feature is that a fast decompression speed could be provided for the spatio-temporally compressed data. This could be observed in Table 4, where the decompression process cost only about 6 seconds or 7 seconds for all 122 time steps of the time-varying combustion data. After decompression, our specialized PBVR to utilize the temporal coherence also shows efficiency that the particle generation time could be accelerated remarkably (see Table 5). This accelerated rendering process is another feature of our compression scheme.

On the other hand, we also made a study for compression error. The block size and tolerance rate also affect the compression error. Obviously, a smaller block size with a lower tolerance rate could provide a compression result with a higher precise. In the experiment for the combustion data, we could see that when the tolerance rate is set smaller or equal to 10^{-3} the compression error is mainly affected by block size. But generally, we could also see that, even a larger block size such as 7, the important features (see Figure 6) could still be kept. This result shows that our compression scheme could provide a high compression ratio while keep a good visual quality.

However, a shortcoming of our system is that the particle generation time is still too long. As we can see from Table 5, even though we set the tolerance rate as 10^{-3}, it could still cost 2,786.1s for all time steps. This is because we have set the repetition level as 144, which means for each time step we need to generate particles with different random number for 144 times. By doing this we can get a good image quality for the time-varying data. Of course, we can get a much faster particle generation speed by setting a lower repetition level, but the image quality would also be affected (see details in T. Kawamura et al.'s work in [2]).

7 Conclusion and Future Work

In this paper, we have proposed a volume compression scheme for the large-scale time-varying volume data using spatio-temporal features. With this compression scheme, we could provide a proper compression ratio to satisfy many system environments (even a low-spec environment) by setting proper compression parameters. Moreover, since our compression scheme does not need special process such as inverse transfer during the decompression, we could provide a low-cost and fast decompression. Furthermore, we also implemented a special method into PBVR to accelerate the particle generation process when we render the spatio-temporally compressed volume. These features are verified in an experiment for the time-varying combustion data. As our future work, we plan to implement other rendering methods into our system to achieve a faster rendering while keep a good visual quality. Furthermore, since our system is suitable for the low-spec hardware for the low-cost decompression and rendering, we also plan to make an application for "smart fishing", to help fishers find rich fishery grounds.

Acknowledgement. This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Grant-in-Aid for Research Program on Climate Change Adaptation (RECCA), and by Japan Science and Technology Agency (JST), A-STEP project ("The research and development of fusion visualization technology", AS2415031H).

References

1. Zhao, K., Sakamoto, N., Koyamada, K.: A Volume Compression Scheme Based on Block Division with Fast Cubic B-spline Evaluation. In: Xiao, T., Zhang, L., Fei, M. (eds.) AsiaSim 2012, Part III. CCIS, vol. 325, pp. 373–387. Springer, Heidelberg (2012)

2. Kawamura, T., Sakamoto, N., Koyamada, K.: A Level-of-Detail Rendering of a Large-Scale Irregular Volume Dataset Using Particles. Journal of Computer Science and Technology 25(5), 905–915 (2010)
3. Jang, Y., Ebert, D.S., Gaither, K.: Time-Varying Data Visualization Using Functional Representations. IEEE Transactions on Visualization and Computer Graphics 18(3), 421–433 (2012)
4. Schnerder, J., Westermann, R.: Compression domain volume rendering. Proceedings of IEEE Visualization, 293–300 (2003)
5. Mensmann, J., Ropinski, T., Hinrichs, K.: A GPU-Supported Loss-less Compression Scheme for Rendering Time-Varying Volume Data. Volume Graphics Eurographics Association, pp. 109–116 (2010)
6. Weiler, M., Botchen, R.P., Stegmeier, S., Ertl, T., Huang, J., Jang, Y., Ebert, D.S., Gaither, K.P.: Hardware-assisted feature analysis of procedurally encoded multifield volumetric data. Computer Graphics and Applications 25(5), 72–81 (2005)
7. Aps, R., Fetissov, M., Lassen, H.: Smart management of the Baltic Sea fishery system: Myth or reality? In: Baltic International Symposium (BALTIC) 2010 IEEE/OES US/EU, (2010), pp. 1-9.
8. Sohn, B.-S., Bajaj, C., Siddavanahalli, V.: Volumetric video compression for interactive playback. Computer Vision and Image Understanding 96(3), 435–452 (2004)
9. Wang, C., Gao, J., Li, L., Shen, H.-W.: A Multiresolution Volume Rendering Framework for Large-Scale Time-Varying Data Visualization. In: Proceedings of the International Workshop on Volume Graphics, pp. 11–223 (2005)
10. Shen, H.W., Chiang, L.J., Ma, K.L.: A Fast Volume Rendering Algorithm for Time-Varying Fields Using a Time-Space Partitioning (TSP) Tree. In: IEEE Visualization 1999, pp. 371–377 (1999)

The Division Method in Visualization of High Frequency Electromagnetic Wave Propagation for Distributed Computing on CAVE System

Hua Xie[1] and Mitsunori Makino[2]

[1] Graduate of Information and System Engineering, Chuo University, Japan
[2] Kasuga 1-13-27, Bunkyo-ku, Tokyo 112-8551, Japan
bakerwhy@gmail.com
makino@m.ieice.org

Abstract. In order to let administrators know their QOS of wireless LAN both in convenience and safety of service range, an interactive and immersive visualization of high frequency electromagnetic wave propagation on the CAVE has been studied with authors, which represents the propagation as the volume data defined by the beam tracing technique with sets of three rays. In the previous study, we used distributed computing on the tracing and successive algorithm to generate the volume data. Through this way, the authors reduced the waiting time largely. However, the accuracy of generated beam need to be improved. Especially the method of division of beam which is from one zone enter the another zone must be improved. In the previous system, we just stack the intensity of electromagnetic wave which enter into the new zone, but have not considered about the direction of the ray. Therefore in this paper, we aim at making the method of division to get better.

Keywords: Division, Visualization, OpenGL, CAVE, Multi-Thread, Ray-Tracing, Beam, Distributed Computing, DCOM.

1 Introduction

Visualization technology consists of creating images, diagrams, and animations, for users to understand all the data either abstract or concrete, information and ideas. Many technologies on computer graphics (CG) and virtual reality (VR) contribute towards recent visualization application.

One of the important roles of the visualization is to make others see the unseen. Radio propagation is typically unseen situation and it is an important service for us in the ICT society, Namely, wireless LAN service has been rapidly spreading anywhere. We get the benefit from connecting with people and society almost at anytime anywhere. However, the fading problem causes less quality of communication service. Furthermore, propagation to unplanned area might cause invasion of privacy. Therefore, an interactive visualization tool for administrators, including non-experts, should be provided.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 149–159, 2013.

The so-called ray launching method [1] is often used to know electromagnetic wave propagation, which traces rays from the given antennas (base station). On the other hand, the so-called beam tracing [2] is proposed as a CG rendering method which generates images of polygonal objects with anti-aliasing effect. It is one of the derivatives of ray tracing method [3]. The method can eliminate uncertainty of propagation in the gap between rays.

In the previous study, we used distributed computing on the tracing and successive algorithm to generate the volume data. Through this way, the authors reduced the waiting time largely. Using the method, users can view and operate the propagation more rapidly and efficiently through the interactive visualization on CAVE system. However, the accuracy of generated beam need to be improved. Especially the method of division of beam which is from one zone enter the aother zone must be improved. Therefore in this paper, we aim at making the method of division to get better.

2 The Visualization Method of Electromagnetic Wave Propagation

In the past studies, we have proposed a beam based tracing method with adaptive subdivision technique [4], [5], and visualize interactively the propagation in the virtual real environment [6]. In our method, tracing beams is replaced by make up sets of rays [7], which represent the reflection/refraction from roads, buildings and other urban constructions.

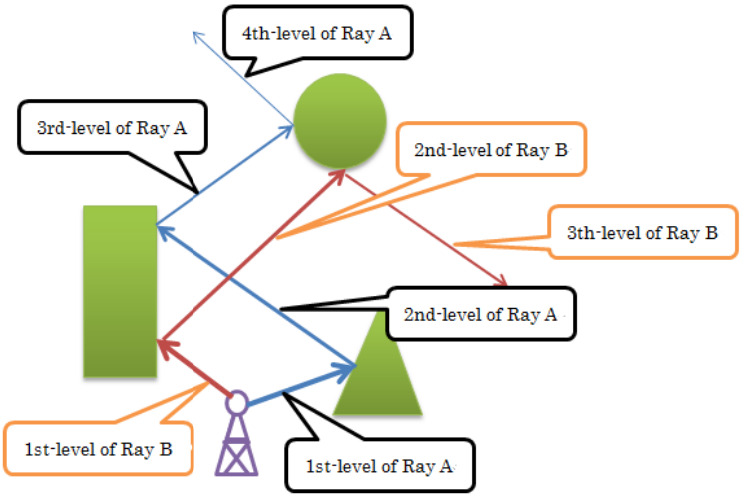
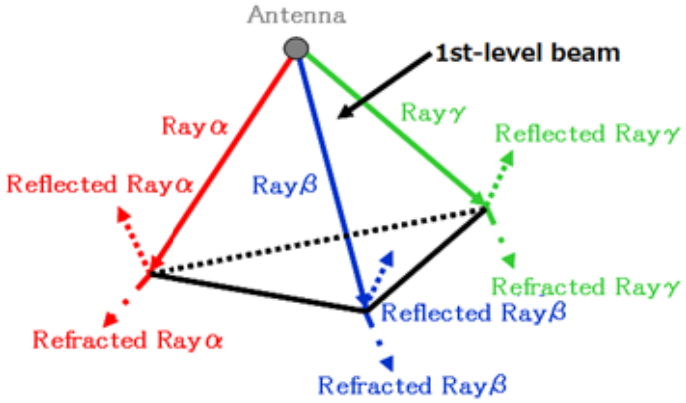


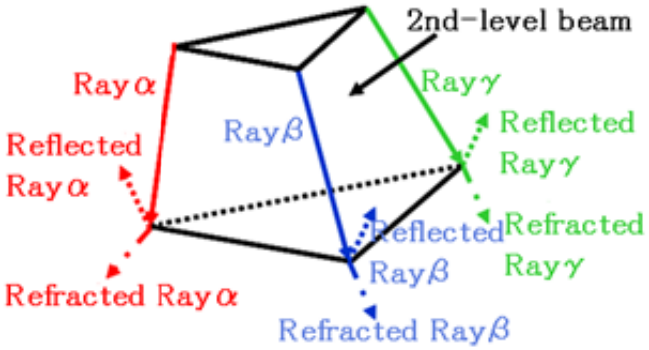
Fig. 1. Level of Tracing Ray [8]

Like the Figure 1, high frequency electromagnetic wave propagation is emitted from an antenna (base station) radially. We defined the rays that are emitted from antenna directed as the 1st level rays. When they intersect an object, new reflecting/refracting rays are generated which we defined them as the 2nd level ones. New Tracing rays are happening repeatedly, we define them as N-level (N>2).

As you will see in Figure 2(a). Three 1st level rays intersect an object or are elongated into infinity, a triangular pyramid (beam) is constructed by the adjacent three rays as edge lines, while the antenna's position and cross points as vertical lines. When the three 2nd level and N-level rays intersect at the same object, they form a frustum of triangular pyramid (beam) looks like figure 2(b).



(a) the First-level Beam



(b)the Second-level Beam and behind

Fig. 2. Construction of Beam [8]

When it matches the one of the following conditions [8],

1. Three rays don't intersect at the same surface (Figure 3(a))
2. Accumulated Propagation distances at endpoints of three adjacent rays are not in defined threshold (Figure 3(b))
3. The Positions of endpoints of three adjacent rays are not within the defined threshold (Figure 3 (c))

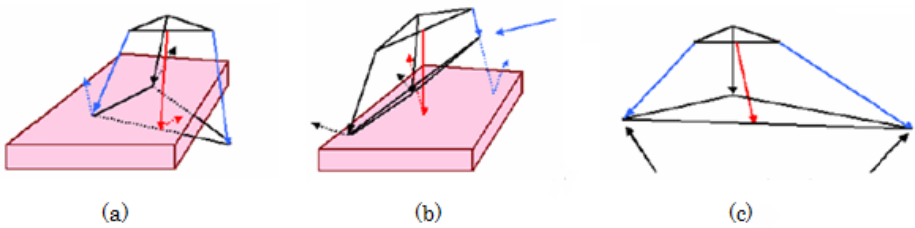


Fig. 3. Condition of Divided Beam [9]

We need to divide the beams into two parts as figure 4, 5.

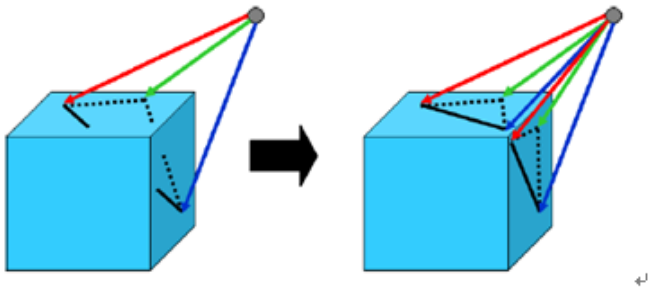


Fig. 4. The Division of The 1st-Level beam [9]

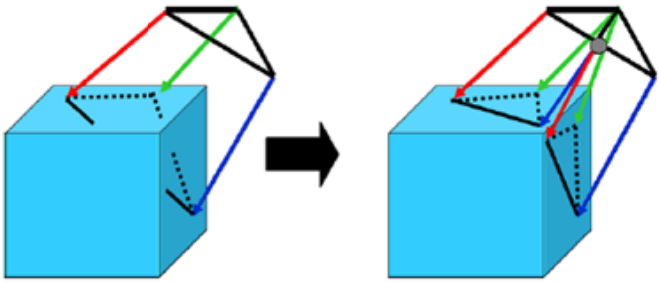


Fig. 5. A division of The 2nd-Level and Subsequent Beams [9]

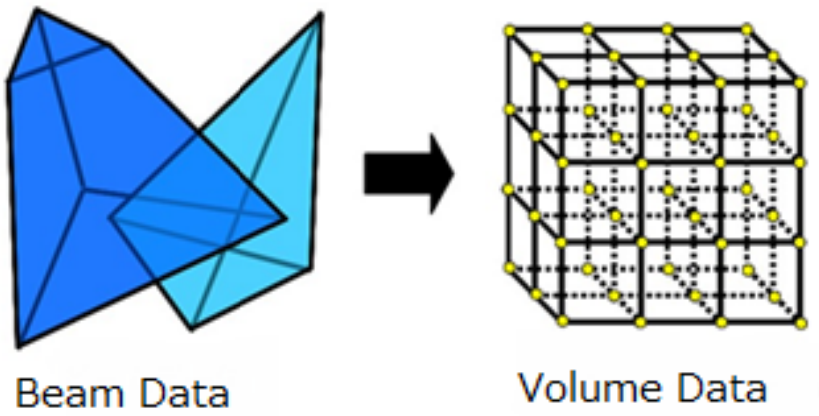


Fig. 6. Generation of Volume Data From Beam Data [8]

The generated beams have shown the electromagnetic wave propagation, and overlaid beams has showed possibility of the interference of the propagation. In order to visualize the propagation with interference, we transform the energy intensity of beams into grid-type volume data [8]. (see Figure 6)

3 Distributed Computing Method and Algorithm

However, the system should be more dynamic and interactive manipulation with less waiting time. The total amount of calculation of the tracing causes the most part of the time loss. So, in order to reduce the time of making volume data, we have introduced a successive method of making volume data with priority, and multi-thread programming [8]. But the result isn`t enough as we have expected. After that, we proposed to use the distribute computing to reduce the user's waiting time but not reduce the quality of making volume data.

In order to implement distributed computing, we need to divide the space into four zones of origin of antenna (see figure 7). This method consists of rendering part "CAVE App", synchronize part "Sync Service", and tracing part "Tracing Service". CAVE App is used to deal with interaction and rendering as the main application. At first, it must be run, and Then, Sync Service will be launched by it. Sync Service is used to launch Tracing Service and synchronize Tracing Service and CAVE App. Tracing Service is installed in different 4 PCs. Each of the Tracing Services takes one zone of cyberspace and trace the rays in this zone.

Sync Service is used to receive data from each of the Tracing Service and send it to the CAVE App. As we have said, we used a successive method for making volume data (see figure 1). In the CAVE App renders the new data on screen when each of

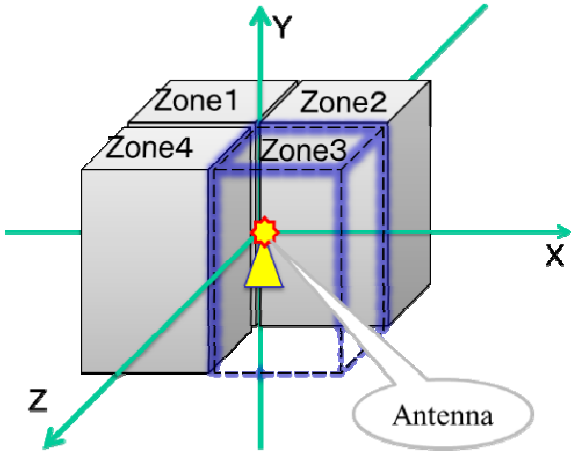


Fig. 7. Structure of Four Zones

the level's ray has been traced. But now, we used 4 PCs to compute tracing of each level's ray and each PC was used in a different zone. Obviously, each PC has different operations, the time of computing is different too. So, we try to let Sync Service to control CAVE App. Each Tracing Service needs to notify Sync Service when its computing of tracing is completed. At the same time, each Tracing Service will stop tracing and wait for the next level's tracing command. Sync Service will send tracing data to CAVE App and notify it to draw the data on screen.

In the Tracing Service, we have used a successive method. We have defined a series of levels when the ray intersects an object (I) or critical surface between two zones (II). As soon as (II) happened, Tracing Service can record the coordinate of cross point in critical surface and energy of electromagnetic waves. Then, Tracing Service will transfer them to the new zone's Tracing Service. The coordinate will be used in the next level's tracing computing of new zone as the start point.

We have used DCOM (Distributed Component Object Model) to ensure the high-efficiency. DCOM is a proprietary Microsoft technology for communication between software components distributed across networked computers. It equips high-efficiency remote procedure call and callback notification by In-Process technology.

Our CAVE system consists of a PC for master controller and six PCs for rendering three stereo shuttered screens. In this system, the CAVE App as the major part of the proposed system is installed to six rendering PCs and run with CAVELib. Both of the Sync Service and Tracing Service are on the windows DCOM service. The Sync Service is installed on one PC, while the Tracing Service is installed to 4 PCs. Figure 8 shows the construction of the proposed system.

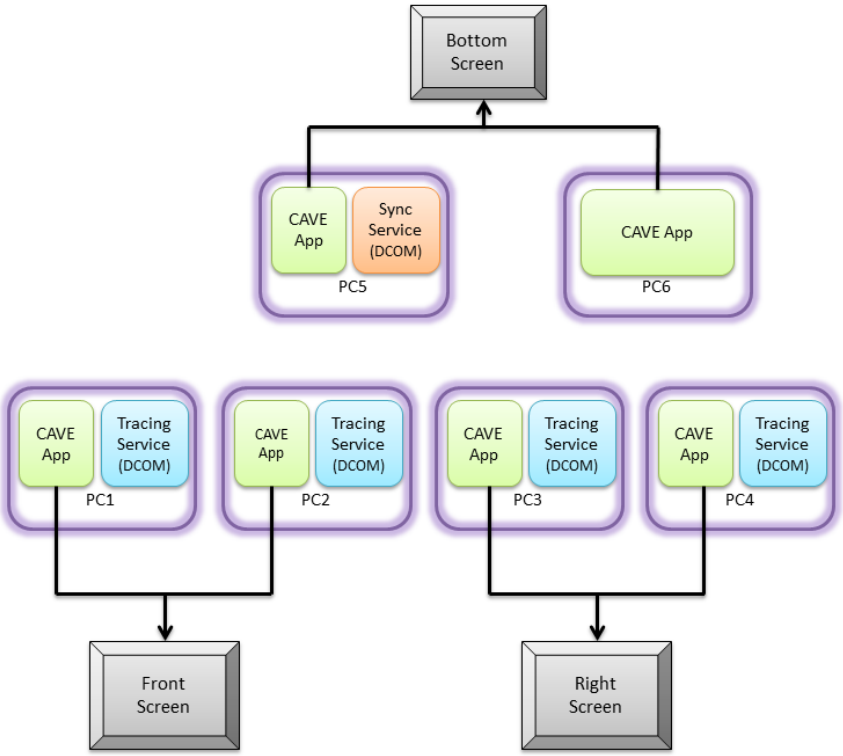


Fig. 8. Construction of Proposed System

4 Proposed Improvement Method

In the description we did before, the divided zone is just an abstract zone, which is not existed in the real world. So, there is something we have to do.

When the ray is reflected by the object, it might enter into the other zone. At this time, we need to re-trace the ray that enters a new zone.

In the previous system, we just stack the intensity of electromagnetic wave which enter into the new zone, but have not considered about the direction of the ray. As the result, the intensity of electromagnetic wave has increased in the spot of the new zone, but the direction is changed. Obviously, the simulated result has some error.

In this paper, we changed the solution of ray's management and beam's division. There have two situations probably, the one is enter completely , the other is partly. We need to divide the beam when it enters a new zone. The information of beam we get after divided, includes of ray's direction, enter point's intensity of electromagnetic wave and so on, will be sent to Sycn Service. Then, Sync Service will manage this information and pass it to the new zone's Tracing Service, as a starting data for the next level's tracing computing. The Sync Service must send it to Tracing Service as attached information when it sends the start command to Tracing Service.

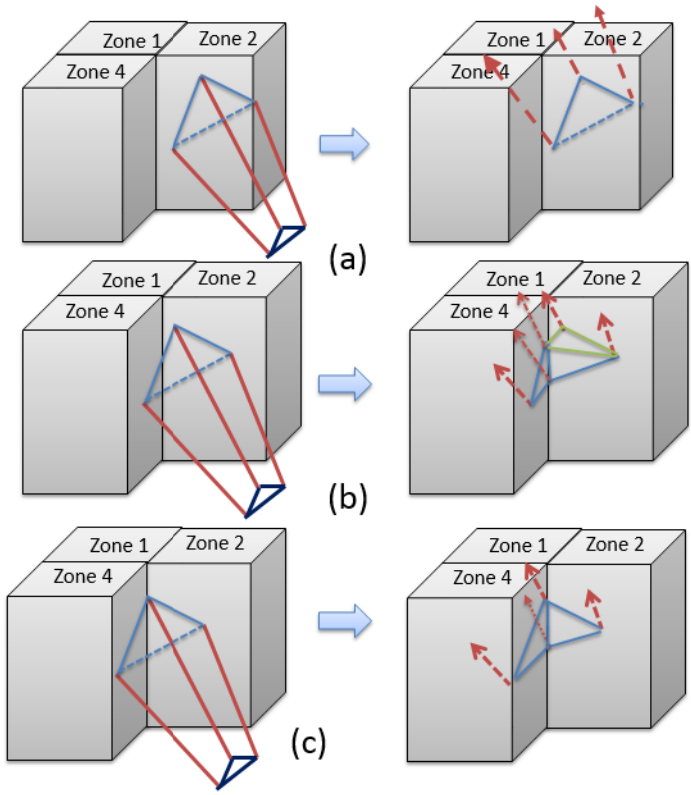


Fig. 9. Method of Divide Beam When Ray Enter Unassigned Zone

While the Tracing Service begins the new tracing of next level, it is necessary to divide the beam and trace the ray of it.

Concrete proposal of division has three-condition as you see in figure 9 (a) to (c). The (a) is the situation when the beam enters another one zone completely, the (b) is the beam enters another two zone but need to divide into 3 new beams which two is in one new zone, the (c) is the beam enters two zone.

Finally, the algorithm of the proposed improvement method is as follows:

- Step 1: At the beginning, CAVE App launches at the first. Then, CAVE App will load 3D map data, and users can set antenna interactively at any position in 3D space.
- Step 2: CAVE App launches Sync Service and sends basic information to it, which includes a connection pointer be used for the callback function. At the same time, the information about the antenna's position be set by the users and the 3D map is sent to Sync Service.
- Step 3: Sync Service launches four Tracing Services and passes the information data that is used to trace calculate with them.

- Step 4: Each Tracing Service begins to generate beam data by tracing rays in its assigned area. Immediately after it completes the task on the current level of rays, the result will be sent to CAVE App. At the same time, "I have completed tracing task at the current level of rays" is notified of Sync Service.
- Step 5: In the tracing process, if some beams might enter a new zone, the tracing of this beam will be stopped, and divided it, then Tracing Service need to record the information on this beam's ray include which zone it will entered, direction vectors and intensity of electromagnetic wave. The Tracing service will send this information to Sync Service when it completed all beam's tracing of current level.
- Step 6: When all of 4 zone's tracing is completed, and the numeric of next level is less than the numeric of total level we set or have some beam entered new zone, the Sync Service need to send the start command of the next level's tracing and information of new entered beam's ray to each Tracing Service. Tracing Service will trace the new beam's ray together.
- Step 7: After all Tracing Services completes the task on the current level of rays, Sync Service notifies CAVE App to the generation. CAVE App visualized the result of the current level, and let Tracing Service to begin the next level's tracing task. While the ray enters an unassigned area after reflections, a new ray is defined at the surface of the area.

5 Implementation

The proposed system is installed on the ChuoCAVE system (see figure 10), which is one of CAVE type VR environment, with active stereoscopic projection on three faces of 80 inch screen. Projected images are generated by a PC cluster with ORAD-DVG. The PC cluster consists of a master PC and 6 rendering PCs. So, we can use the 6 PCs to implement distributed computing.

Fig. 10. ChouCAVE System

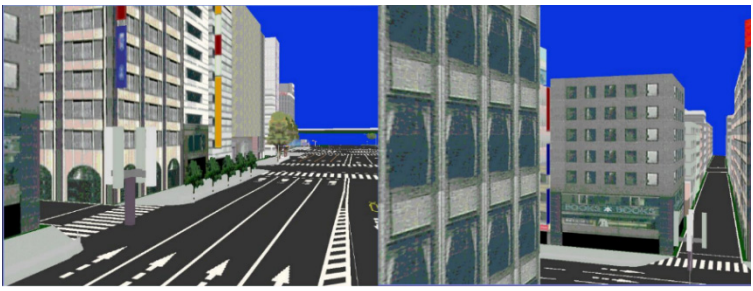


Fig. 11. Set up an Antenna

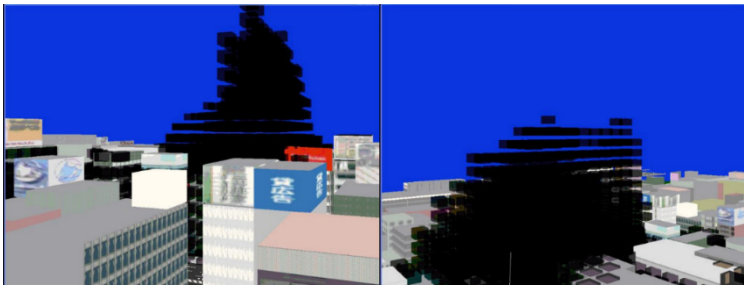


Fig. 12. Visualization of Proposed System

Figure 11 shows the status while user set up an antenna, it is the capture of front-side screen from different perspective in the cave system.

Figure 12 shows the different of the proposed system and previous system. You can see the result of the proposed system (left side) can visualize more electromagnetic wave than the previous system (right side).

6 Conclusion

In this paper, we improved the accuracy of visualization of electromagnetic wave propagation by dividing the beam which enters a different zone in distributed computing on the CAVE system. As a result, we can greatly reduce the computation time of ray tracing without giving up the accuracy. Using the method, users can could see the visualization result which is more accurate on screen quickly after setting up an antenna in the system. In future studies, improvement of tracing algorithm in the efficiency and increase interactivity manipulation will be discussed.

Acknowledgement. This work was partially supported by MEXT/JSPS KAKENHI Grant Number 24500129.

References

1. Durgin, G.D., Patwari, N., Rappaport, T.S.: An advanced 3D ray launching method for wireless propagation prediction. In: IEEE 47th Veh. Technol. Conf., Phoenix, AZ (May 1997)
2. Heckbert, P.S., Hanrahan, P.: Beam Tracing Polygonal Objects. Computer Graphics 18(3), 119–127 (1984)
3. Whitted, T.: An improved illumination model for shaded display. In: Proc. of the 6th Annual Conference on Computer Graphics and Interactive Techniques (1979)
4. Ohsaki, A., Makino, M., Shirai, H., Shinoda, S.: A Visual Simulation for High Frequency Electromagnetic Wave Propagation. In: Proc. of ITC-CSCC 1999, vol. I, pp. 229–232 (1999)
5. Makino, M., Ohsaki, A., Shirai, H., Shinoda, S.: A Visual Simulation of Ray Propagation in the Use of Adaptive Beam Tracing for Wireless Communications Systems. In: Proc. of MDMC 2001, pp. 125–132 (2001)
6. Ikeda, M., Makino, M.: An Interactively Stereoscopic Visualization of High Frequency Electromagnetic Wave Propagation on Digital Map. In: Proc. of NICOGRAPH International 2008 (2008)
7. Nakano, H., Cao, X., Makino, M.: An Adaptive Subdivision of Beams for Polygonal Objects in Visual Simulation of High Frequency Electromagnetic Wave Propagation. In: Proc. of International Workshop on Advanced Image Technology 2007 (IWAIT 2007), pp. 159–164 (2007)
8. Xie, H., Makino, M.: Successive visualization of high frequency electromagnetic wave propagation using multi-thread on CAVE system. In: Xiao, T., Zhang, L., Fei, M. (eds.) AsiaSim 2012, Part III. CCIS, vol. 325, pp. 188–196. Springer, Heidelberg (2012)
9. Xie, H., Makino, M.: High-Speed, Dynamic, Interactive Visualization of High Frequency Electromagnetic Wave Propagation. In: Proc. of 2011 International Conference on Modeling and Simulation Technology (JSST 2011), October 22-23. CD-ROM (2011)

3D Fused Visualization Based on Particles-Based Rendering with Opacity Using Volume Texture

Kyoko Hasegawa, Kozaburo Hachimura, and Satoshi Tanaka

College of Information Science and Engineering, Ritsumeikan University,
1-1-1 Noji-higashi,Kusatsu, Shiga, Japan
{hasegawa,hachimura,stanaka}@media.ritsumei.ac.jp

Abstract. In medical, scientific and other fields, transparent surface visualization is useful to investigate inner 3D structures. Such visualization usually uses the polygon graphics, where the polygons must be sorted along the line of sight. The sorting, however, requires long computation time for large-scale data. Besides order of polygons in the sorting often becomes indefinite especially for intersecting surfaces. On the other hand, the particle-based volume rendering was proposed as a transparent-rendering method which does not require sorting. In addition, the important feature of this method is to generate the particles first only once. The purpose of this paper is to execute the surface rendering with the opacity by using the volume texture. Furthermore, we show the 3D fused image, such as volume-surface.

Keywords: Particle-based surface rendering, Volume texture, Fused Visualization.

1 Introduction

In medical, scientific, and other fields, transparent surface visualization is used to investigate interior 3D structures. Usually, this type of visualization uses polygon graphics, with the polygons sorted along the line of sight. The sorting, however, takes a long computation time for large-scale data. In addition, the order of polygons in the sorting often becomes difficult to interpret especially for intersecting surfaces.

Recently, Koyamada et al. proposed particle-based volume rendering (PBVR) [1], which uses tiny particles as rendering primitives. This method does not require any sorting and is applicable to large-scale data. Moreover, it enables natural volume fusion [2]. We extended PBVR to make it applicable to surfaces as well as volumes [4]. We term this method "particle-based rendering (PBR)." An important advantage of this extension is that 3D-fused visualization of different volume/surface/slice objects becomes possible simply by merging particles prepared for each element to be fused. In PBR, surface is visualized at a constant opacity by applying a uniform-sampling method.

On the other hand, the technique for 3D modeling are developed [3] that the texture mapping applies the 3D texture instead of the 2D texture. Using 3D texture,

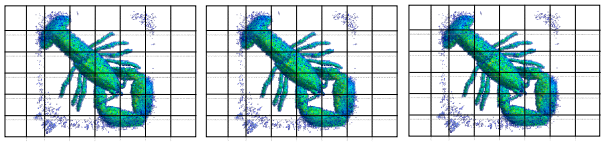
G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 160–165, 2013.

we create the 3D model to waste no memory space. In this research, we propose the technique that the surface rendered with color and opacity by using the volume data as the 3D volume texture. Furthermore, we show the 3D fused image, such as volume-surface.

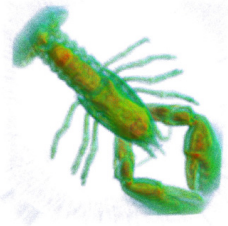
2 Particle-Based Rendering with Color and Opacity

PBR renders 3D scalar fields as particle clouds and incorporates both the emission and absorption effects. Particle density depends on the transfer function and is used to evaluate the number of particles to be generated in the volume data. Because the particles can be considered opaque, no visibility sorting processing is required. PBR has three processes: particle generation, particle projection onto the image, and ensemble averaging of particle luminosities. The first process stochastically generates particles according to the form of their transfer function. We repeat the particle distribution until L_R statistically independent particle sets are prepared [See Fig. 1(a)]. Below we call L_R the "repeat level." Such particle projection with the occlusion effect is executed for each particle set prepared in the first process. As the result, L_R similar images are created [See Fig. 1(b)]. The second process projects particles onto an image plane. The third process calculates the ensemble average of the L_R images created in the second process [See Fig. 1(c)], which realizes transparent images.

(a) L_R independent particles sets



(b) L_R intermediate images



(c) Final transparent image

Fig. 1. Schematic view of the PBR

To visualize a surface, we stochastically generate particles on the surface[5,6]. The number of particles is defined by the following formula:

$$n = \frac{\ln(1-\alpha)}{\ln(1-s_p/s_A)} L_R, \tag{1}$$

where α, s_p, and s_a are the opacity, area of the particle, and surface, respectively[4]. The slice plane is drawn same as that in the above technique.

2.1 Creating Particles on the Surface

PBR may have applicability to defined drawing surface. In the case of a polygonized surface defined with uniform opacity, the total number of particles can be calculated by the replacement of the counting sphere with the total area size. In this sub-section, we describe the method of creating particles in a polygonized surface defined with non-uniform opacity.

First, we explain how to execute the uniform sampling of a polygonized surface, i.e., a polygon mesh in case with uniform opacity. For a polygonized surface, it is easy to calculate the total areas of the constituent polygons, which we use as area size S in Eq. (1). Correspondingly, we regard the total number of particles generated on all the constituent polygons as n_{all}.

The sampling is made such that the generated particles form the 2D square grid with inter-particle distance d in each polygon (see Fig.3). Then the particle density becomes $1/d^2$, i.e., one particle per one square. On the other hand, the particle density should be n_{all}/S. Therefore an equation $1/d^2 = n_{all}/S$ holds, and d is determined as:

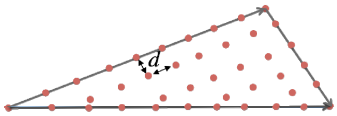
$$d = \sqrt{\frac{s_A - s_p}{\ln(1-\alpha)L_R}}. \tag{2}$$

The sampling consists of three steps:

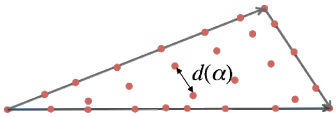
Step 1. sampling polygon edges,
Step 2. sampling inside of polygons,
Step 3. particle shuffling.

By executing step 1 first, we can avoid duplicative sampling of edges shared by neighboring polygons. In step 1, we simply put particles with distance d on each edge of the whole polygon mesh (see Fig.2 (a)). In step 2, for each polygon, we put particles with distance d on scan lines, which are placed parallelly to an edge at intervals of d. In step 3, we shuffle the generated particles that are stored in an array. Step 3 ensures randomness that is required to apply the probabilistic theory of opacity developed in the PBSR. Note that computation time for the particle shuffling is much shorter than the particle sorting. The former is proportional to n_{all}, while the latter is proportional to $n_{all} \log n_{all}$. After completion of the above three steps, we obtain a particle set that is uniformly distributed and has good a statistical property.

In this research, distance d are defined with function of the opacity α (see Fig.2 (b)), in order to achieve a surface with non-uniform opacity defined as the transfer function. The opacity on the surface is obtained by projecting onto the volume texture, and the distance is defined with Eq. (2).



(a) Creating particles in case of uniform opacity



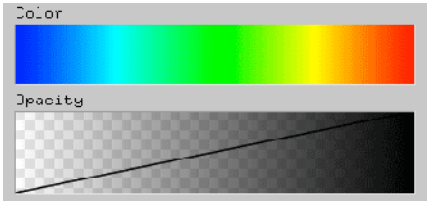
(b) Creating particles in case of non-uniform opacity

Fig. 2. Creating particles on the polygon

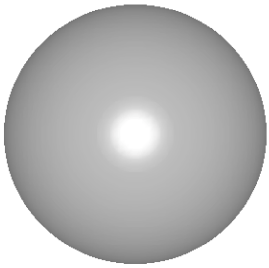
3 Experiments

Through this section our using data sets are shown in Fig.3. In this research, the volume data, the grid size is $64\times64\times64$, is used as the volume texture (see Fig. 3 (a)), and the transfer function, that is color and opacity map, is shown in Fig. 3 (b). Fig.3 (c) is the surface data we use. In order to project onto the surface easily, the center of the volume data is equal to the surface.

(a) Volume data

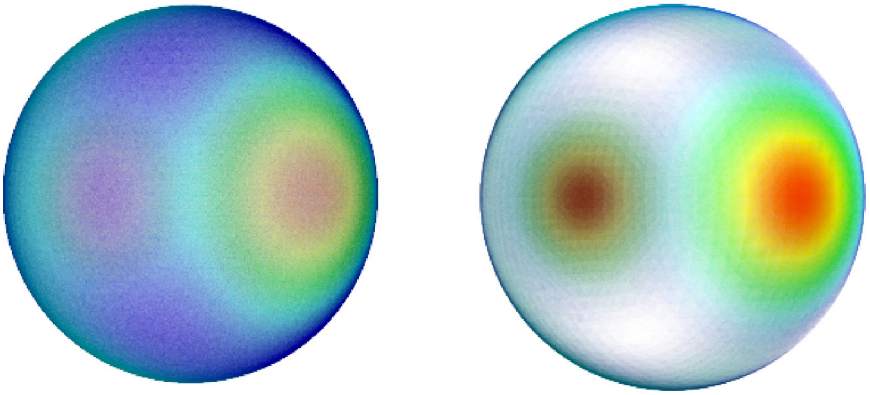


(b) Upper part: color map, lower part: opacity map



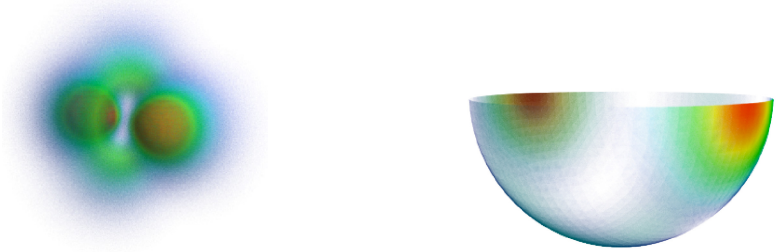
(c) Surface data

Fig. 3. Visualization of volume texture and polygon surface model. Here, (a) and (b) are volume date and transfer function for using volume texture, (c) is a polygon model.

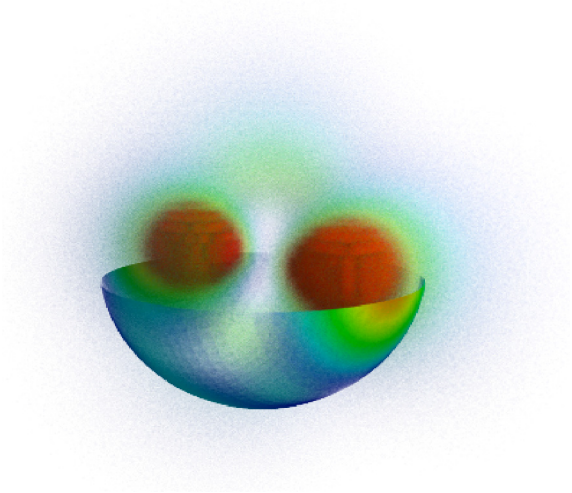


(a) Surface with the color map (b) Surface with color and opacity

Fig. 4. Result of the surface rendering with the color map (a) and the color and opacity map (b)



(a) Volume rendering (b) surface rendering with opacity



(c) Fused visualization

Fig. 5. Fused visualization from surface, cross-section surface and contour line

Fig. 4 (a) is shown in the result of the surface with only color map, the opacity has the constant value a = 0.3. On the other hand, Fig. 4 (b) is shown in the result of the surface applied with the color and opacity map. It was shown that our technique is available for creating particles not only the color map but also the opacity map. In these figure, the number of particles of (a) and (b) are 34 million and 26 million, respectively, and these figures are rendered with $L_R = 500$.

We show the fused visualization from different rendering type. Fig. 5 is an example of the fused visualization consisted of the volume and surface, these are used the same transfer function Fig. 3. To fuse these images in three dimensions, only the created particles have to be combined. Fig. 5 is the result of fused visualization, where the number of particles used is about 63 million (including 50 million particles in the volume, 13 million particles on the surface). We can confirm that natural fusion is realized by simply merging the particle sets from the different rendering types.

4 Conclusion

In this paper, we described a method to create fusion images using particle-based rendering. The particle-based rendering creates transparent images with right depth feel without necessity of particle sorting. Our probabilistic opacity theory can be controlled on the surface with non-uniform opacity. In addition, we showed that 3D fusion images, volume-surface fusion, with right depth could be created only by combining particles prepared for each element to be fused.

Acknowledgments. The authors wish to thank Prof. Koji Koyamada, Prof. Hiromi T. Tanaka, Dr. Susumu Nakata, and Dr. Naohisa Sakommoto for their valuable suggestions.

References

1. Koyamada, K., Sakamoto, N., Tanaka, S.: A Particle Modeling for Rendering Irregular Volumes. In: Proceedings of the International Conference on Computer Modeling and Simulation (UKSIM 2008), Cambridge, England, pp. 372–377 (2008)
2. Sakamoto, N., Kawamura, T., Koyamada, K.: Improvement of particle-based volume rendering for visualizing irregular volume data sets. Computers & Graphics 34(1), 34–42 (2010)
3. Takayama, K., Okabe, M., Ijiri, T., Igarashi, T.: Lapped solid textures: filling a model with anisotropic textures. ACM Transactions on Graphics (TOG) 27(3) (2008)
4. Tanaka, S., Hasegawa, K., Shimokubo, Y., Kaneko, T., Kawamura, T., Nakata, S., Ojima, S., Sakamoto, N., Tanaka, H.T., Koyamada, K.: Particle-Based Transparent Rendering of Implicit Surfaces and its Application to Fused Visualization. In: EuroVis 2012, Vienna, Austria, June 5-8 (2012)
5. Satoshi, T., Akio, M., Satoru, N., Yasushi, F., Hiroaki, Y.: Sampling Implicit Surfaces Based on Stochastic Differential Equations with Converging Constraint. Computers & Graphics 24(3), 419–431 (2000)
6. Hasegawa, K., Ojima, S., Shimokubo, Y., Nakata, S., Hachimura, K., Tanaka, S.: Particle-Based Transparent Fused Visualization Applied to Medical Volume Data. IJMSSC (in Press)

Error Modeling of Hardware-in-the-Loop Simulation System and Its Influence Analysis for a Certain Missile

Xiao-fei Chang, Shi-zheng Wan, Wen-xing Fu, and Jie Yan

School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China

Abstract. Hardware-in-the-loop simulation is necessary for the development of missile's flight control system. The influence of the simulation equipment errors to simulation results cannot be ignored. This paper introduces both the composition of hardware-in-the-loop simulation system and the clock synchronization strategy in particular for a certain missile. Major errors, which mostly affect the precision of simulation, are studied. Through theoretical derivation and experimental verification, the error models of communication time delay, three-axis rotary table, load simulator and data acquisition are established. Finally, based on mathematical simulation the influence of errors on simulation results is confirmed, and the validity of the error models is verified by contrast with hardware-in-the-loop simulation test results. The experimental results show that each error model can meet the demand of the simulation accuracy analysis. It can provide the theory basis for analysis, measurement and error compensation for hardware-in-the-loop simulation system.

Keywords: Hardware-in-the-loop simulation, mathematical model, error analysis, simulation precision.

1 Introduction

Hardware-in-the-loop simulation brings in real key components to computer simulation loop in place of the corresponding parts of the mathematical model. It can overcome both the imprecision of the mathematical model and the influence of interference, more truly reflecting the actual situation of the system, thus to get high simulation confidence level [1]. Due to its good controllability, non-destructive, replicability and economy, it is widely used in weapon developments, aerospace and industrial control fields.

At present, the hardware-in-the-loop simulation system is a large-scale test system mainly adopts the distributed design and consists of a variety of complex simulation equipment. Simulation equipment is used to provide working environment for simulation subject parts, but there are no these parts in original system, so the join of simulation subject parts directly affects the accuracy and credibility of simulation results.

Numerous researchers carried out a lot of research on the error influence to simulation precision. Guan Ruxun and others analyzed the requirements of simulation equipment based on a hardware-in-the-loop simulation, presented the error analysis of

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 166–175, 2013.

the basic method and the model checking method [2]. Lu Hao and others analyzed the effects of the seeker installation error, and researched the error compensation method[3]. Zhang Hongxi analyzed the installation error of the seeker, and researched the error compensation method[4]. Waldemar and Lais applied the hardware-in-the-loop simulation to the simulation of satellite attitude control system simulation, and compared the simulation results with mathematical simulation[5]. Nikos Andrianos and Christian Diez analyzed and measured the errors of five-axis rotary table including axis vibration, axis orthogonal, and axis cross, provided the reference to reduce the error caused by five-axis rotary table[6]. However, the composition and performance index of different simulation system equipment are distinct, so the system error influence on simulation precision needs a detailed analysis of specific objects.

As a test system used for particular purpose, there are a variety of error factors in hardware-in-the-loop simulation system. In-depth researching and analyzing of the various error factors in system, establishing the error model of simulation and the error analysis of simulation results, are important link in the process of hardware-in-the-loop simulation, and it is of great significance for improving the precision of simulation system. This paper takes a certain missile hardware-in-the-loop simulation system as the research object, determines the main error factors that affect the accuracy of system simulation by analyzing the system structure and working mechanism, establishes the error model by theoretical and experimental analysis, confirms the influence level by a lot of digital simulation, and verifies the correctness of models by comparing the simulation results.

2 System Overall Scheme

2.1 System Structure

A certain missile use the SPC(Stored Program Control) guidance scheme[7]. Its flight control system includes the flight control computer, gyro and accelerometer inertial measurement component, and four-channel steering gear actuators. According to task demand and components, the overall scheme of hardware-in-the-loop simulation system design is shown in figure 1:

In simulation system, the integrated console is responsible for the scheduling of the system and the generation of synchronous clock; the three-axis rotary table completes the simulation of missile space position change; the four-channel load simulator loads the hinge moment on the shafts of the steering gears to simulate the real flight state, and acquire the turning angles of the rudders; The IMU(Inertial Measurement Unit) simulator inputs the line acceleration signal into the flight control computer; according to the angles of the rudders in the simulation process the real-time simulation computer completes the real-time solution of simulation model to realize the close-loop simulation of missile.

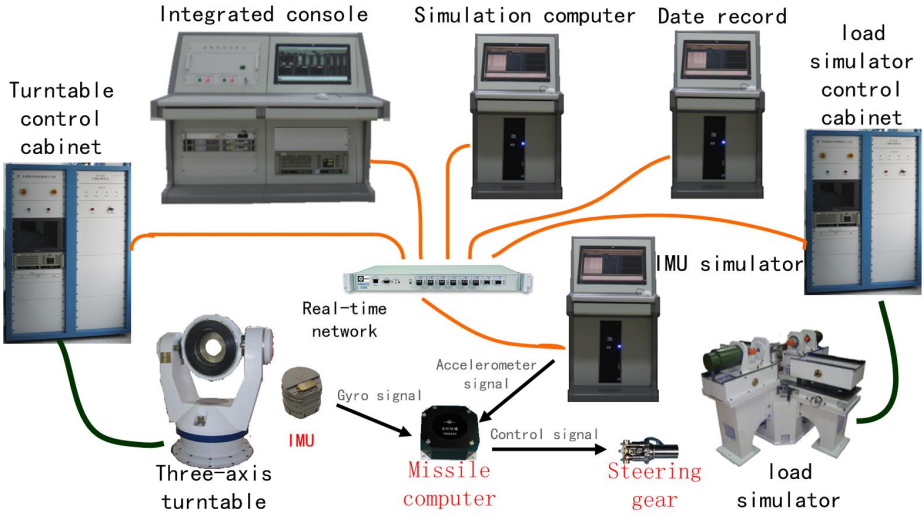


Fig. 1. The overall scheme of hardware-in-the-loop simulation system design

2.2 Clock Synchronization Design

From the overall scheme, the hardware-in-the-loop simulation system is a distributed computer networks composed of various computers that implement different functions. Precise timing of the system, data transmission between each node and clock synchronization are key problems of system design[8].

This simulation system uses the VMIC board, produced by GE, for data transmission between each node to reduce the data transmission error in the maximum degree. High precision simulation clock is generated by the RTX software in integrated console. In each clock cycle, data transmission in the form of radio by VMIC network ensures the consistency of the entire system simulation clock. Every simulation node completes the step advance by responding time interrupt message, and completes the data reading, task execution and data upload in each interrupt response. The interrupt response should be set to the highest level in order to ensure that the simulation task will not be interrupted.

Using the clock synchronization and data transmission mechanism mentioned above can guarantee the accuracy and uniformity of the simulation system clock, effectively eliminate the phenomenon of time ambiguity and minimize data transmission delay.

3 Error Analysis and Modeling

In the process of the hardware-in-the-loop simulation, there are a variety of error factors affecting the accuracy of the simulation, such as design error in scheme design, equipment error in hardware processing and software coding, environment error

caused by the change of system working environment condition , human introduced operating error and error caused by the method of data processing, etc.

This paper pays attention to the effects of various equipment errors on simulation precision for the missile hardware-in-the-loop simulation system. Through the analysis of missile flight control system and system overall scheme design, the error of the simulation equipment mainly comes from the communication time delay, rotary table error, load simulator error and signal acquisition error.

3.1 The Error of Network Communication Time Delay

For hardware-in-the-loop simulation system, the real-time property of system is the basis of simulation accuracy. This system adopts the communication mode of RTX precise timing and VMIC network transmission. It can reduce the timing error and data transmission error in the maximum degree. But the network communication process is affected by many factors, including the timer precision, network topology, network communication protocols, network load, network transmission rate, packet size and so on. In addition, other simulation nodes run under the Windows operating environment, therefore, network communication time delay error still exists.

In order to establish the communication time delay error of the model, it's required to test network communication time. In the system, integrated console sends the interrupt messages to other nodes in each timer interrupt of RTX. After child nodes receiving the interrupt, the CPU clock is queried in the interrupt response function to get the time of nodes. The adjacent interrupted interval is the simulation step size of this node, and the difference from the set step size is the delay error, which includes simulation clock error, data transmission delay, node task execution delay and error caused by the interrupted interruption.

Test results are shown in the figure below:

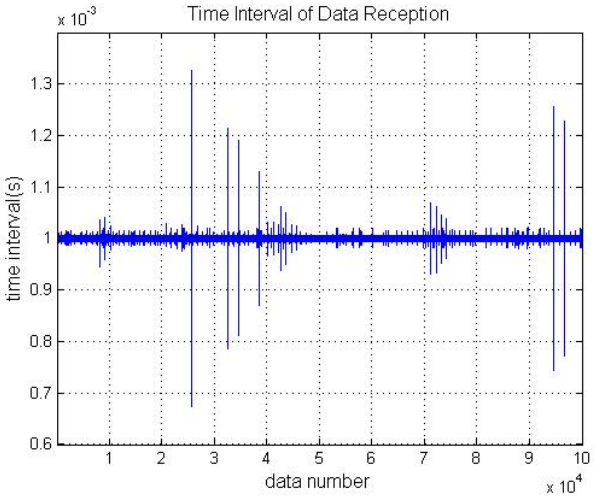


Fig. 2. Time Interval of Data Reception

Through the statistics of a large number of test data, the average simulation step size is 0.99997 ms. It shows the system time synchronization method is feasible, and using the step advance mechanism can significantly reduce the accumulative error between simulation time and real time, improving the accuracy of time similarity of simulation system effectively.

In addition, the probability of step size above 1.05ms existing in the simulation is less than 0.01%, and the maximum step size is 1.3 ms, the reason is that the influence of the other processes of windows causes the interrupt response timeout. In this case, the timeout nodes do not update the data in this simulation step.

Through the analysis of the timing synchronization strategies and test results, the communication time delay model is established. The data transmission between each node contains a simulation step delay, and also includes a 0.01% chance of random timeout phenomenon.

3.2 The Error of the Three-Axis Rotary Table

By rotation of its body, the three-axis rotary table is used to achieve the attitude control of missile therefore simulating the missile rotation around the centroid in the air. In this way it provides the simulated environment for the simulation subject parts. Because the rotary table and the hardware-in-the-loop simulation system is series connected, the influence of system error directly affects the amplitude attenuation and phase lag of the guidance control system[9].

Rotary table error mainly includes various static errors which are inevitable during the manufacture and installation, and the dynamic error caused by bandwidth constraints and control performance. In this simulation system, the input of flight control system is the missile's attitude angular velocity, therefore, the dynamic performance of rotary table has large influence on the simulation results.

Concerning engineering practice, the mathematical model of the three-axis rotary table can be expressed by a second order oscillation system, whose transfer function is described as follows:

$$G(s) = \frac{1}{T^2 s^2 + 2\xi T s + 1} = \frac{\omega_n{}^2}{s^2 + 2\xi\omega_n s + \omega_n{}^2} \tag{1}$$

Through theoretical derivation and actual test, the modeling process now can be accomplished based on the special error principle called double-ten guideline, namely the amplitude error for three-axis rotary table is less than 10%, and the phase delay is not greater than 10°.

Firstly, the amplitude function of this second order model can be obtained from above transfer function according to the logarithmic amplitude-frequency characteristics:

$$L(\omega) = 20\lg\left|G(j\omega)\right| = -20\lg\sqrt{\left(1 - T^2\omega^2\right)^2 + \left(2\xi T\omega\right)^2} \tag{2}$$

Applying the double-ten guideline, the parameters of system's dynamic characteristics can be identified:

$$\begin{cases} L(\omega)_{\omega=2\pi f} = -0.1 \\ L(\omega)_{\omega=T} = 0 \end{cases} \tag{3}$$

Then, given particular frequency design index for each frame, for example, 15Hz for inner frame, 12Hz for middle frame and 10Hz for outer frame, the second order damping ratio and natural frequency can be solved by substituting frequency design index to equation(3). Last but not least, modify and check the parameters according to the results of frequency scanning test of rotary table. Thereby, mathematical model guideline of the rotary table is given as follows:

Table 1. The parameters of second-order model of the three-axis rotary table

Index	Inner frame	Middle frame	Outer frame
Damping ratio ξ	0.707	0.707	0.707
Natural frequency ω_n	245.2	193.4	162.3

Considering the position control precision of rotary table system, sensing accuracy and minimum stable index speed, the random white noise whose amplitude is 0.01° is added to the control output of rotary table.

3.3 Error of the Load Simulator

The load simulator is used to simulate the dynamic load change and check the performance of the steering gears in the flight by load the hinge moment. Because the hinge moment has a great influence for the dynamic performance of missile steering gears, the load precision directly affects the load simulator working state of the steering gears. Then it affects hardware-in-the-loop simulation test result.

The error of the load simulator is mainly caused by installation error, load error control performance and the extraneous force of control.

The mathematical model of load simulator can also be simplified as the second order link. In this system, the frequency index of the four-channel electric load simulator is 12 Hz. Its error modeling methods and procedures are similar to rotary table.

Through theoretical derivation and the test checking, mathematical model of four-channel load simulator is:

Table 2. The parameters of second-order model of load simulator

Index	Channel 1	Channel 2	Channel 3	Channel 4
Damping ratio ξ	0.707	0.707	0.707	0.707
Natural frequency ω_n	191.5	188.3	194.2	189.4

Likewise, considering control accuracy and minimum moment of load force of the load simulator, the noise model whose maximum amplitude is 1% of input and minimum is 0.05 N*m is added to the output moment of load simulator.

3.4 The Error of Signal Acquisition

In the simulation system, the angle of rudder is measured by rotary transformer installed on the load simulator to complete the transformation from mechanical angle in degrees to analog signal. Its accuracy is about 3', namely 0.05 °.

In order to pass the rudder angle to simulation model, simulation system completes analog signal acquisition and conversion through data acquisition. Because the simulation system includes rotary table, load simulator and many other high voltage devices, in order to reduce the effects of electrical noise, the signal is insulated, amplified, and filtered before signal acquisition. After test, considering each error and precision, analog signal acquisition error is about 10 mv, 0.03°.

4 Error Influence of Simulation Equipment

In order to research and analyze the influence of simulation equipment to simulation results, lots of mathematical simulations and statistics are needed. Based on the digital simulation of missiles, the error model of equipment is brought in. Use Monte Carlo method for simulation calculation, and compare the error effect on the simulation results.

The position of the error models is shown below:

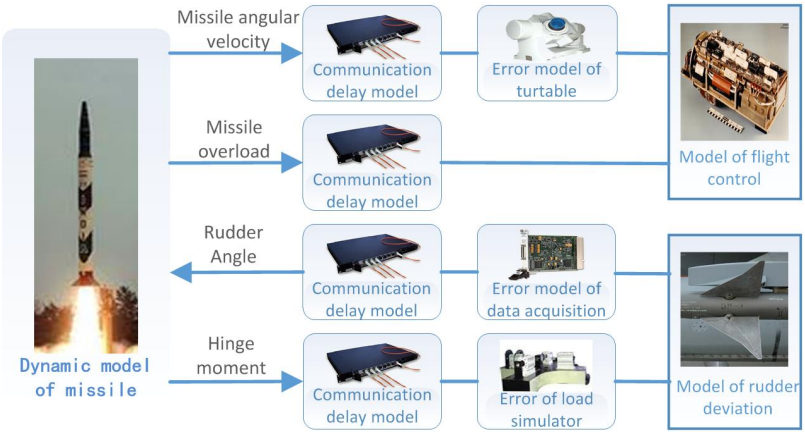


Fig. 3. The position of the error models

For this missile, its flight distance is the main concern in this simulation. The influence of different equipment on simulation results is shown below. The error model of each group simulates 1000 times. The property of influence is obtained by statistical calculation. In order to understand the influence of error, the given results are dimensionless. Namely the given data are the results relative to ideal mathematical simulation model.

Table 3. Influence of equipment errors for flight distance

Error	Mathematical simulation	Mean	Maximum	Mean square deviation
Time delay	1.000000	0.9993590	0.9995375	1.749721e-05
ADE error	1.000000	1.0000308	1.0002262	7.447698e-05
Rotary table error	1.000000	0.9976275	0.9978569	7.185935e-05
Load simulator error	1.000000	1.0000154	1.0007995	3.544055e-04
Time delay + ADE error	1.000000	0.9993712	0.9996503	8.394793e-05
Time delay + Rotary table Error	1.000000	0.9969648	0.9973169	1.142859e-04
Time delay + Load simulator error	1.000000	0.9993443	1.0007016	6.221941e-04
Total system error	1.000000	0.9963991	0.9986031	1.08089e-03

Because the missile control system is a closed-loop system, there exist interactions among different parameters in flight. As a consequence, the error influence of equipment for hardware-in-the-loop simulation cannot be simply superimposed. In this system, the equipment with the biggest error influence is rotary table error.

Furthermore, because the sensitivity of different simulation objects and control system is not identical, the error influence of different mathematical models of the missile is not the same.

5 Hardware-in-the-Loop Simulation Test

In order to check the effectiveness of all the error models, the mathematical simulation including all the equipment error is compared with the hardware-in-the-loop simulation under the same model condition and initial state. Real objects including flight control computer, IMU and steering gears are used in hardware-in-the-loop simulation. Contrast curves of typical parameters among mathematical simulation, mathematical simulation with error model and hardware-in-the-loop simulation are shown below:

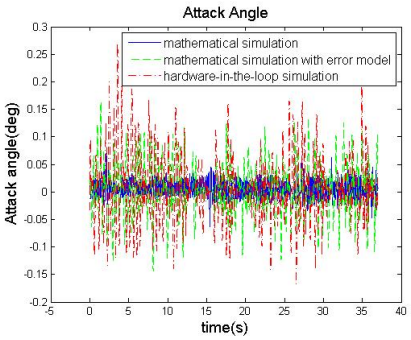


Fig. 4. Attack Angle contrast curves

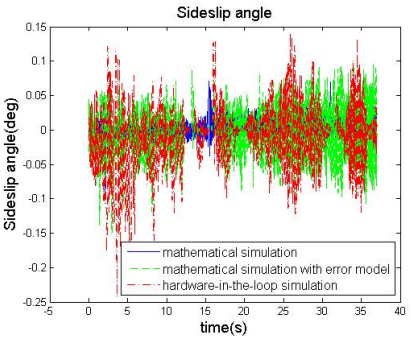


Fig. 5. Sideslip Angle contrast curves

As can be seen from the simulation results contrast curve, owning to the join of the hardware-in-the-loop equipment, there is some difference between hardware-in-the-loop simulation and mathematical simulation. After appending the error model to the mathematical model, the results are consistent with the hardware-in-the-loop simulation results. The simulation contrast results prove the effectiveness and correctness of the error models built in this paper.

6 Conclusion

Hardware-in-the-loop simulation is a necessary link in the design process of missile control system. Because the hardware-in-the-loop simulation system is a large-scale test system which consists of a variety of complex simulation equipment, the influence of system for the simulation results cannot be ignored. Therefore, it is necessary to research and analyze the influence of system error for test results.

Aiming at the missile hardware-in-the-loop simulation system, based on the simulation system overall scheme and working mechanism, this paper illustrates the main factors which influent the system precision. The factors include communication delay, dynamic property of the three-axis rotary table, load simulator and the error of signal acquisition. Through theory deduction and experiment verification, the error of each link is established, and the influence is analyzed on the basis of the technical indicators of the missile. The results show that the influence of the error model of different link for simulation results is not linear superposition. At last, through the comparison with the results of the hardware-in-the-loop simulation to validate the effectiveness of error models, it shows that the error models are correct and can be used for simulation precision check and analysis. The deducing and modeling of error models for hardware-in-the-loop simulation system has reference meaning to the development and precision analysis of similar simulation systems.

References

1. Fu, W.-X., Yu, Y.-F., Huang, Y., Wei, J.-l.: Precision Guided Missile Control System Simulation. Northwestern Polytechnical University Press, Xi'an (2010)
2. Guan, R., Tao, Y.: Analysis of Accuracy and Credibility of Simulation Experiment. Measurement & Control Technology 5, 7–9 (1995)
3. Hao, L., Zhuo, L.: Analysis of static errors of infrared optical guidance hardware-in-the-loop system. In: The Ninth Session of the National Photovoltaic Technology Academic Communication, pp. 260–263 (2010)
4. Zhang, H., Cui, L.: Analysis of Influence Induced by Installation Error of Seeker in Hardware-in-the-loop Simulation. Computer Simulation 27(12), 31–34 (2010)
5. Leite Filho, W.C., Mallaco, L.M.R., Carrijo, D.S.: Hardware in the Loop Simulation of an Attitude Control System. AIAA-99-4323,1999: 423-426 (1999)

6. Andrianos, N., Diez, C., DeMore, L.A.: Flight table orientation error transparency for hardware-in-the-loop facilities. In: Murrer Jr., R.L. (ed.) Technologies for Synthetic Environments: Hardware-in-the-Loop Testing VI. Proceedings of SPIE, vol. 4366, pp. 269–282 (2001)
7. Chang, X., Yang, T., Yan, J., Wang, M.: Design and integration of hardware-in-the-loop simulation system for certain missile. In: Xiao, T., Zhang, L., Ma, S. (eds.) ICSC 2012, Part II. CCIS, vol. 327, pp. 229–237. Springer, Heidelberg (2012)
8. Li, Q., Wang, M.: Research on Time and Data Synchronization in the Hardware-in-the-loop Simulation for Missile. Computer Measurement & Control 18(1), 110–115 (2010)
9. DeMore, L.A.: Design Study for a High-Accuracy Three-Axis Test Table. J. Guidance 10(1), 104–114

An Agent-Based Modeling and Evolutionary Optimization Approach for Vulnerability Analysis of Critical Infrastructure Networks

Akhila Kizhakkedath[1], Kang Tai[1], Mong Soon Sim[2], Robert Lee Kong Tiong[3], and Jiaying Lin[1]

[1] School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore
[2] Information Division, DSO National Laboratories, Singapore
[3] School of Civil and Environmental Engineering, Nanyang Technological University, Singapore
{akhila1,LINJ0046}@e.ntu.edu.sg, {mktai,clktiong}@ntu.edu.sg, smongsoo@dso.org.sg

Abstract. Critical infrastructure networks include the highly complex and interconnected systems that are so vital to a city or state that any sudden disruption can result in debilitating impacts on human life, the economy and the society as a whole. Some of the interdependencies among infrastructure components are perhaps unforeseen and methods for vulnerability analysis of infrastructure networks should therefore incorporate the possibility of potential unforeseen interdependencies in such networks. This paper proposes using an optimization approach to iteratively search for potential unforeseen interdependencies and failures that can maximize connectivity loss in infrastructure networks due to cascading failures. In order to illustrate the proposed approach, an agent based model of an infrastructure network and its known interdependencies has been presented, with a genetic algorithm applied to search for potential unforeseen interdependencies as well as node failures that can result in the maximum loss of infrastructure network connectivity.

Keywords: critical infrastructure network, interdependencies, network analysis, evolutionary optimization.

1 Introduction

Critical infrastructure refers to the basic physical and operational assets and systems such as the power grids, transportation services, telecommunication systems, water supply systems, banking and financial systems, emergency services, oil and gas supply systems, food and industrial production, security and health services, etc. that are very essential to the functioning of the economy, the government and the society as a whole [1-2]. Due to the vitality of the services provided by these systems, a sudden unavailability of any of them or part thereof may result in a crippling effect on public health, safety as well as economy of a society [3].

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 176–187, 2013.

Owing to technological advancements, the previously independent components of infrastructures are now highly interconnected or interdependent on each other. For example, power grids are made up of components like regional power generation plants, regional substations for transmission and interregional distribution stations [4]. If any component of the power grid such as a generating plant or a substation fails or is incapacitated, the failure effects will cascade as a result of the interdependencies among power grid components. In addition to the fact that each of the critical infrastructures is highly complex, interconnected and also geographically dispersed, modern infrastructures also rely heavily on the services of other infrastructures. Such an interdependency within an infrastructure sector (e.g. power grid) as well as between infrastructure sectors (e.g. power grid and water supply) has resulted in a global system of systems that is highly vulnerable to cascading failures initiated by technical errors, deliberate attacks, climate changes, natural disasters and so on [5]. As an example, the Northeast Blackout of 2003 was initiated by a number of software failures at the First Energy control center, Ohio, followed by the tripping of two high voltage transmission lines which were in contact with untrimmed trees [6]. The subsequent cascading failures resulted in a Blackout that left 50 million people without power in the Northeastern and Midwestern United States and in Ontario, Canada. In addition to the cascading failures within the power grid, the Blackout also initiated failures in other infrastructure sectors. For example, many water pumps were taken out of service resulting in contamination of water, the trains running to and from New York were cancelled, some of the airports were also closed due to the lack of a screening facility for passengers, many oil refinery plants were shut down and petrol prices went up, telecommunication systems were affected and landlines became jammed. Also affected was the production sector due to the unavailability of raw materials resulting from the traffic interruption at the borders since the electronic checking systems were affected [7]. The example shows that identifying the risks and vulnerabilities in infrastructure systems is extremely important and must take into account the effects of failure propagation on the affected sector as well as on other dependent sectors.

The massive disruptions in critical infrastructures due to various failure events over the years like the Terrorist attacks on the World Trade Centre on September 11, 2001 [8], the Northeast Blackout of August 14, 2003 [6-7], the Sumatra earthquake and Tsunami on December 26, 2004 [9] and the recent Tohoku earthquake and Tsunami in Japan on March 11, 2011 [10] have motivated much research over the last decade in critical infrastructure protection. Most of the works focuses on prioritization of critical infrastructure protection, i.e. identifying the vulnerabilities and critical components and on recovery planning and mitigation strategies. In most of the works, critical infrastructures are characterized as complex networks in which the various infrastructure components like generating plants, bus stops, petrol pumps, airports, etc. constitute the nodes and the interconnectedness or interdependencies between the various components are the links. Interdependencies can take different forms like physical, cyber, geographical, logical or policy related, and they are described as follows [11-12].

• Physical - A physical or topological reliance between infrastructure components like material flow between them.
• Cyber - Reliance between infrastructure components on transfer of information, e.g. dependence of power grids on control systems like SCADA.
• Geographic/Geospatial - A reliance that exists between the infrastructure components due to proximity.
• Policy/Procedural - Reliance between infrastructure components in which the state of one component depends on the state of other due to policies or higher level decisions.
• Societal/Logical - A relationship that exists between infrastructure components due to societal factors like public opinion, fear, cultural issues, etc.

Since critical infrastructures are mostly characterized as complex networks, various metrics and centrality indices from network theory have been used to find the central/critical nodes as well as to determine the overall vulnerability of critical infrastructure networks [13-16]. Optimization techniques have also been combined with network theory to identify the combinations of nodes which when failed would result in the maximum decrease in network performance characteristics like connectivity loss, loss of system flow, etc. [17]. Due to the increasing complexity and size of infrastructures, various modeling and simulation methods like agent-based modeling [18-19], input-output inoperability modeling [20] and system dynamics [21] are also popular in critical infrastructure vulnerability analysis and protection.

Although numerous methods and metrics have been developed to model critical infrastructures and to analyze their vulnerability to failures, one of the common features of all these works is that they assume that the information regarding the interdependencies among infrastructure components is completely known. However, many of the interdependencies may be unforeseen due to the presence of several relations and complex feedback paths that exists among infrastructure components. Critical infrastructures often undergo many subsequent upgrades and hence the information regarding critical infrastructures may also be incomplete [5]. It has also been reported in [5] that infrastructure failures often cascade along indirect links that are mostly originated by proximity which emerge at the time of crisis. Some of these unknown or unforeseen interdependencies may not be critical because they may not result in any additional loss of infrastructure network connectivity upon failures. However, some of the unforeseen interdependencies may be crucial because they may result in the cascading failure of many other components resulting in a larger loss of infrastructure network connectivity. Unraveling where these potential unforeseen interdependencies can be, that may cause the greatest loss in connectivity upon some failure in the infrastructure network is therefore an interesting and novel research problem. This paper therefore proposes a method to incorporate such unforeseen interdependencies in the vulnerability analysis of infrastructure networks. The method involves identifying all the relevant infrastructure components from different sectors and their known interdependencies, and then applying an optimization method to iteratively add additional links or unforeseen interdependencies to the network until the connectivity loss upon some failure is maximized. In order to test whether the

proposed approach is feasible, this paper presents an agent based model of a complex infrastructure network and its known interdependencies, with a genetic algorithm applied to search for potential unforeseen interdependencies as well as node failures that can result in the maximum loss of network connectivity.

The paper is organized as follows: Section 2 provides a description of the proposed method and its application to a case study, Section 3 presents the initial results from the proposed study and the paper concludes with Section 4 that also discusses on future work of this study.

2 Methodology Applied to a Case Study

2.1 Constructing the Network Model

The key building blocks for the network model comprise the infrastructure components which are represented as nodes and their interdependencies which are represented as links. All the relevant components/nodes of the different infrastructure sectors, their characteristics as well as the different types of known interdependencies (physical, cyber, etc.) between them have to be therefore identified for modeling. Agent-based modeling has been used to model the complex infrastructure network [18]. Agent-based modeling is very effective as a computer modeling tool for studies on complex systems. In a complex system, the nonlinear interactions among its constituents give rise to emergent behaviors which cannot be broken down into the properties or behaviors of individual constituents. Agent-based models are well suited for studying such emergent behaviors [19] which appear in complex infrastructure networks in the form of cascading failures, traffic congestions, etc. In agent-based

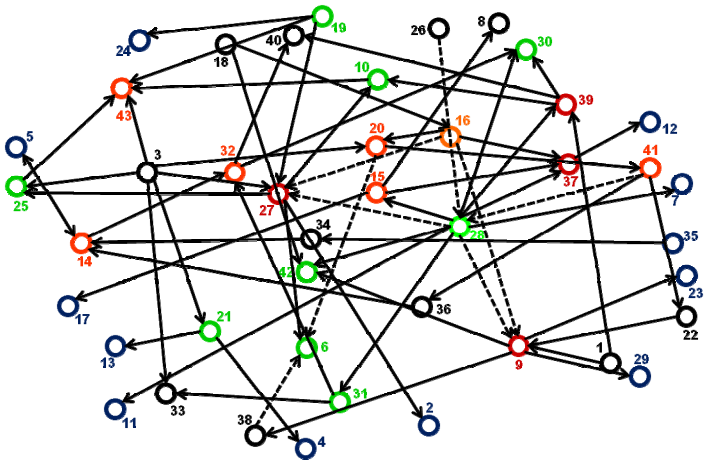


Fig. 1. The infrastructure network used for modeling; nodes are represented as circles and are labeled with their node numbers, primary interdependences are shown by solid lines and redundant interdependencies are represented by dashed lines [22]

modeling, an agent can be anything from a software agent without much intelligent behavior to a smart agent with intelligent behavior. In the proposed agent-based model of a complex infrastructure network, each agent is a software agent that represents a network component like a node and the behavior of the agents are also specified by simple "if-then" rules. The interdependencies or interactions between the agents can be represented as links.

A graphical representation of a critical infrastructure network which is used as a case study is shown in Fig.1. The network is obtained from the work presented in [22]. An open source agent-based modeling platform NetLogo [23] was used to model the studied network where nodes are represented as agents and interdependencies are represented as links. The detailed explanation on the infrastructure network as well as its characteristics can be found in [22] and a short summary has been given below.

Node Characteristics Identified for Modeling

Buffer – When an infrastructure component or node has buffer, it will be able to sustain its operation/function for a short time period even after the inputs from all other infrastructure components or nodes have become unavailable. The time period for which the node is able to sustain its operation by itself is called buffering time.

Recovery – When an infrastructure component or node has recovery property, it will be able to recover from failures and restore its operations in a reasonable period of time. The time taken for the recovery process to be completed is called recovery time.

The buffering and recovery times are randomly assigned to the different nodes for illustration.

Interdependency Characteristics Identified for Modeling

Primary Interdependencies – Primary interdependencies represent the various relationships/interconnectedness between infrastructure components such as the supply of commodities, information flow, geographic proximities, etc.

Redundant Interdependencies – They are the interdependencies that come into operation only when a node is unable to get input for operation from its related primary interdependencies and its entire buffer has been consumed. They act as redundancies to the node which when activated ensure that the node remains functional.

The primary and redundant interdependencies together constitute the basic or known set of interdependencies.

2.2 Analysis of Network Failures

The current study focuses on node failures and on the loss of connectivity of infrastructure networks after those failures. A variety of metrics like diameter [13],

number of spanning trees [14], tenacity [15], giant component size [16], etc. have been used by various researchers in order to quantify connectivity loss in infrastructure networks. In this paper, connectivity loss has been quantified by measuring the giant component size of infrastructure network after node failures. When a node fails, the failure propagates through the entire network as a result of which additional nodes may fail and the network may get disintegrated into separate components. A component is a group of nodes that are all connected to each other either directly or indirectly and the largest of all the components formed is called the giant component. According to Percolation theory, the nodes in the giant component of the network remaining after failure will be still functional and the other nodes will be non-functional [16]. Therefore, the size or the number of nodes in the giant component (giant component size) is used as an indicator of network connectivity loss after node failures. Since a smaller giant component size indicates that only a small number of nodes are still connected to each other, maximization of connectivity loss requires a minimization of giant component size. The giant component size of networks can be computed using Depth First Search algorithm from network theory [24].

Implementing Failure Propagation/Cascade

Nodes can fail due to various reasons like technical errors, deliberate attacks, natural disasters and so on. When a particular node failure has to be studied, the node is fully destroyed and all the other nodes which are completely dependent on the failed node for input will be affected. However, all these affected nodes do not fail immediately because they possess various characteristics like buffer, redundancy and recovery. The affected nodes whose buffering times are greater than zero would enter buffering states and when buffering time reduces to zero (buffer has been consumed), the presence of redundant interdependencies to these nodes will be checked. The nodes that have redundant interdependencies would return to normal state and the ones without redundancies would enter recovery state. If the recovery times are greater than zero (which indicates the ability to recover), the nodes would return to normal state at the end of recovery process, else they would fail. Once an additional node (other than the studied node) has failed, a similar check is performed and the process continues until all the nodes in the network reach their steady states (normal operating state or failed state). Once all the nodes have reached their steady states, the size of the giant component of the network remaining after failures is quantified as a measure of connectivity loss. Further details on implementing failures as well as on the failure propagation algorithm has been given in [22].

2.3 The Optimization Process Using Genetic Algorithms

The proposed method uses an optimization technique to iteratively add unforeseen interdependencies to the constructed infrastructure network and select nodes to be failed until the giant component size is minimized. The discrete nature of the problem enables the use of evolutionary optimization techniques like genetic algorithms (GA)

for performing the optimization. Genetic algorithms are search heuristics that mimics Darwin's theory of evolution [25]. An individual in GA is characterized by its chromosome that encodes the decision variables of the optimization problem. The fitness of each individual is calculated as the value of the objective function for that individual. The algorithm begins with a population of individuals that are generated randomly and proceeds over generations. In every generation, the fitness or objective function value of individuals will be quantified and multiple individuals will be selected based on their fitness values, recombined and mutated to populate the individuals of new generation. GA usually stops when either a predefined generation number has been reached or an acceptable level of fitness/objective function value for the population has been achieved [26].

In order to investigate where the potential unforeseen interdependencies can be in the infrastructure network and also to identify the crucial node failure in the network that can result in the maximum reduction of giant component size, three different optimization experiments were performed using a simple GA. The objective of all the three experiments involves a minimization of giant component size, but the decision variables vary depending on the number of potential unforeseen interdependencies to be searched for. All the experiments use a population size of 50 and each run terminates when the maximum number of generations reaches 50. Crossover rate was allowed to vary in the range 0.10-1.00 and mutation rate was allowed to vary between 0.01 and 0.10.

GA Experiment 1: The decision variable of this experiment includes only one node failure and there are no unforeseen interdependencies involved. For this experiment, an individual in GA is represented as an integer that represents the node number of the node to be failed. During optimization, single nodes will be failed iteratively in the network model until the node failure that results in the smallest giant component size of the infrastructure network has been found.

GA Experiment 2: The decision variables of this experiment include one unforeseen interdependency and one node failure and are encoded as integers. The optimization is performed by iteratively adding an unforeseen interdependency to the network model and failing a node until the giant component size of the infrastructure network remaining after cascading failures is minimized.

GA Experiment 3: The decision variables of this experiment include two unforeseen interdependencies and one node failure. During optimization, two unforeseen interdependencies are added iteratively to the network model and one node is failed until giant component size of the network after failure propagation is minimized.

Several runs of each of the GA experiments were performed by varying the parameters of the genetic algorithm (crossover and mutation probabilities) and the smallest giant component sizes of each run were noted. The unforeseen interdependencies as well as the node failures that resulted in the smallest giant component sizes were also recorded.

3 Simulation Results

The network studied consists of 43 nodes and therefore the number of node failures to be studied is 43. For a 43 node network, a directed interdependency/link can be added to the network in $P\,(43, 2)$ ways, i.e. the permutation of 43 objects taking 2 objects at a time as shown by Eq.1.

$$P\,(43, 2) = 43 * 42 = 1806 \tag{1}$$

Since the network already has 64 known interdependencies (56 primary interdependencies and 8 redundant interdependencies), the number of ways in which one unforeseen interdependency can be added to the network becomes $1806 - 64 = 1742$. Therefore, the number of ways in which two unforeseen interdependencies can be added to the network is given by $C\,(1742, 2)$, i.e. combination of 1742 objects taking two at a time as shown by Eq.2.

$$C\,(1742, 2) = \frac{1742*1741}{2!} = 1516411 \tag{2}$$

Hence, the total number of scenarios to be investigated by GA becomes 43, $(43 * 1742)$ and $(43 * 1516411)$ respectively for GA experiments 1, 2 and 3 on the assumption that unforeseen interdependencies can be added between any two nodes in the network, provided there are no existent interdependencies between them. This shows that the number of scenarios to be tested rises exponentially with the number of potential unforeseen interdependencies to be investigated in the infrastructure network. The optimization experiments using GA in this study could effectively find the optimal solutions from large scenario spaces with reasonable running times. The plots showing the improvement of fitness (decrease in giant component size) over successive generations in one of the runs of GA Experiments 2 and 3 are illustrated in Fig. 2. The plots show that the optimal solution was found by the 17th and 40th generations respectively for experiments 2 and 3 for this particular run.

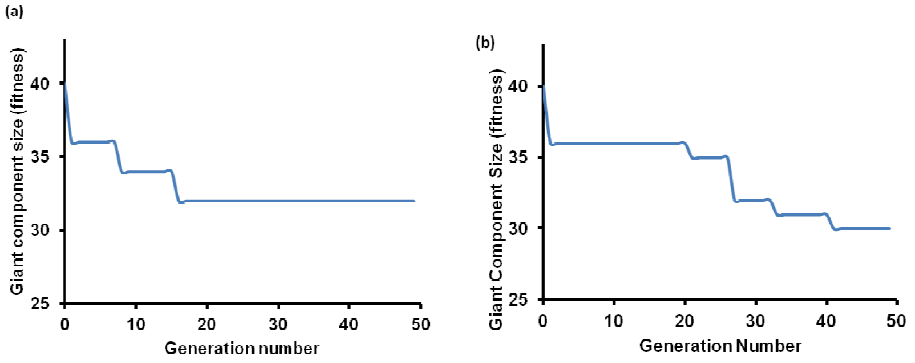


Fig. 2. Plot showing the improvement of fitness (giant component size) over successive generations in (a) Experiment 1 and (b) Experiment 2

3.1 Identifying Critical Node Failure

All the GA experiments showed that the node failure that results in the smallest giant component size or the largest loss of infrastructure network connectivity is the failure of node 28. This is not surprising since a large number of nodes or infrastructure components have interdependencies to node 28 and rely on it for their operation. An illustration of how the network disintegrates upon failure of node 28 is shown in Fig.3. The values of the smallest giant component sizes obtained in all the experiments have been listed in Table 1.

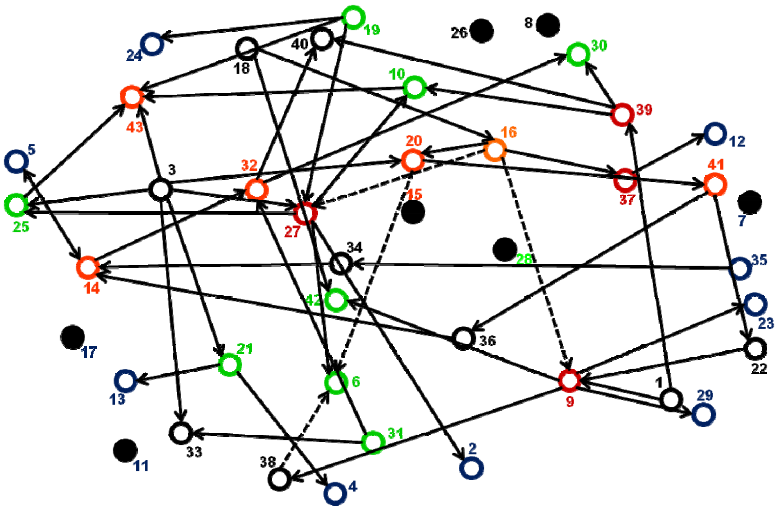


Fig. 3. An illustration of how the network disintegrates upon failure of node 28 in experiment 1; the nodes in black get separated from the giant component of the network reducing the giant component size to 36

3.2 Identifying Potential Unforeseen Interdependencies

Once the node failure with the smallest giant component size has been found, the next step is to identify the potential unforeseen interdependencies that can result in the smallest giant component sizes. The effect of the number of unforeseen interdependencies on the giant component sizes obtained after the failure of node 28 has been illustrated in Table 1 which shows that the addition of more interdependencies to an infrastructure network can result in an increase in connectivity loss. This means that failure consequences can increase with the presence of potential unforeseen interdependencies. Table 1 also summarizes the crucial unforeseen interdependencies obtained from the optimal solutions of different runs of all the three experiments. In the table, a directed interdependency is represented as an arrow, e.g. 7→3 represents an interdependency from node 7 to node 3. The results from Table 2 show that the presence of an unforeseen interdependency to node 3 is the most crucial single unforeseen interdependency and the

interdependencies to nodes 3 and 19 are the most crucial combination of two unforeseen interdependencies. Both nodes 3 and 19 are supply nodes that provide input to many other nodes in the network. Hence, the failure of some other nodes in the network results in the failure of these supply nodes due to the additional interdependencies added to them which further results in the failure of all the nodes that are dependent on these supply nodes for input. This cascade of node failures results in the giant component size reducing to 30 when interdependencies to both nodes 3 and 19 are added, indicating a 30.23% reduction of network connectivity.

Table 1. The effect of adding potential unforeseen interdependencies on giant component size after the failure of node 28

Experiment Number	No. of unforeseen interdependencies added	Unforeseen Interdependency added by GA in the optimal solutions	Worst Giant component size
1	0	Nil	36
2	1	One of (7→3, 8→3, 11→3, 15→3, 17→3, 28→3, 31→3)	32
3	2	One of (7→3, 8→3, 11→3, 15→3, 17→3, 28→3, 31→3) and one of (3→19, 4→19, 13→19, 15→19, 21→19, 28→19)	30

4 Conclusions

The modeling and simulation of infrastructure networks and identifying the vulnerabilities in them is challenging, but is extremely important for disruption prevention and recovery planning. All the previous works on critical infrastructure protection were based on the assumption that the knowledge about the interdependent infrastructure network is complete. This research assumes that the knowledge about the interdependencies may not be completely known and hence an optimization algorithm has been used to search for potential unforeseen interdependencies and failures that can maximize cascading failures in the network resulting in the maximum network connectivity loss. The study also shows how a model combining agent-based modelling, genetic algorithms, and network analysis can be built in a computational platform in order to simulate infrastructure network failures.

Although the interdependent infrastructure network considered in this paper is real, the network properties were more general and hence, the approach has to be further validated on real interdependent infrastructure networks. Furthermore, in some networks, for e.g. the power grid, giant component size may not be able to fully

reflect the failure consequences because the components other than the giant component may still be functional for serving their local customers. Hence, measures other than giant component size need to be used to measure the failure consequences. In addition, the method will be extended to a multiobjective optimization problem by incorporating the failure probabilities in addition to the failure consequences, thereby formulating a risk analysis framework for infrastructure failures.

References

1. Clinton, W.J.: Executive order 13010 - Critical infrastructure protection. Federal Register 61, 37347–37350 (1996)
2. Lewis, T.G.: Critical infrastructure Protection Homeland Security. John Wiley & Sons, New Jersey (2006)
3. Rinaldi, S.M.: Modeling and Simulating Critical Infrastructures and Their Interdependencies. In: International Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004), Hawaii, USA, January 5-8, pp. 20054–20062 (2004)
4. Rocco, C.M., Ramirez-Marquez, J.E., Salazar, D.E., Hernandez, I.: Implementation of Multi-objective Optimization for Vulnerability Analysis of Complex Networks. In: Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, pp. 87–95 (2010)
5. Setola, R., De Porcellinis, S.: Complex networks and critical infrastructures. In: Chiuso, A., Fortuna, L., Frasca, M., Rizzo, A., Schenato, L., Zampieri, S. (eds.) Modelling, Estimation and Control of Networked Complex Systems. Understanding Complex Systems, vol. 2, pp. 91–106. Springer, Heidelberg (2009)
6. Pourbeik, P., Kundur, P.S., Taylor, C.W.: The anatomy of a power grid blackout-Root causes and dynamics of recent major blackouts. Power and Energy Magazine IEEE 4, 22–29 (2006)
7. Chai, C.L., Liu, X., Zhang, W.J., Baber, Z.: Application of social network theory to prioritizing Oil & Gas industries protection in a networked critical infrastructure system. Journal of Loss Prevention in the Process Industries 24, 688–694 (2011)
8. O'Rourke, T.D.: Critical infrastructure, interdependencies, and resilience. The Bridge, Quarterly publication by the National Academy of Engineering. Washington 37, 22 (2007)
9. Levy, J.K., Gopalakrishnan, C.: Promoting disaster-resilient communities: the Great Sumatra–Andaman earthquake of 26 December 2004 and the resulting Indian Ocean tsunami. Water Resources Development 21, 543–559 (2005)
10. Lekkas, E., Andreadakis, E., Alexoudi, V., Kapourani, E., Kostaki, I.: The Mw= 9.0 Tohoku Japan earthquake (March 11, 2011) tsunami impact on structures and infrastructure. In: Environmental Geosciences and Engineering Survey for Territory Protection and Population Safety (EngeoPro) International Conference, Moscow, pp. 97–103 (2011)
11. Rinaldi, S.M., Peerenboom, J.P., Kelly, T.K.: Identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Systems 21, 11–25 (2001)
12. Pederson, P., Dudenhoeffer, D., Hartley, S., Permann, M.: Critical infrastructure interdependency modeling: a survey of US and international research. Idaho National Laboratory, Idaho Falls, Idaho 83415 (2006)
13. Gorman, S.P., Schintler, L., Kulkarni, R., Stough, R.: The revenge of distance: Vulnerability analysis of critical information infrastructure. Journal of Contingencies and Crisis Management 12, 48–63 (2004)

14. Li, F.-y., Cao, Y.-z., Li, G.: A node importance assessment method of complex networks based on reliability measure. In: Zeng, D. (ed.) Future Intelligent Information Systems. LNEE, vol. 86, pp. 271–278. Springer, Heidelberg (2011)
15. Wang, Y., Cai, W.D., Duan, Q.: Computing Vulnerability of Network Based on Genetic Algorithm. Journal of System Simulation 21, 1628–1632 (2009)
16. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Catastrophic cascade of failures in interdependent networks. Nature 464, 1025–1028 (2010)
17. Murray, A.T.: An overview of network vulnerability modeling approaches. Geo Journal 78, 209–221 (2011)
18. Dudenhoeffer, D.D., Permann, M.R., Manic, M.: CIMS: A framework for infrastructure interdependency modeling and analysis. In: Proceedings of the 38th Conference on Winter Simulation (WSC 2006), pp. 478–485 (2006)
19. Helbing, D.: Agent-Based Modeling. In: Social Self-Organization, pp. 25–70. Springer, Heidelberg (2012)
20. Haimes, Y.Y., Horowitz, B.M., Lambert, J.H., Santos, J., Crowther, K., Lian, C.: Inoperability input-output model for interdependent infrastructure sectors. II: Case studies. Journal of Infrastructure Systems 11, 80–92 (2005)
21. Conrad, S.H., LeClaire, R.J., O'Reilly, G.P., Uzunalioglu, H.: Critical national infrastructure reliability modeling and analysis. Bell Labs Technical Journal 11, 57–71 (2006)
22. Lam, C.Y., Lin, J., Sim, M.S., Tai, K.: Identifying Vulnerabilities in Critical Infrastructures by Network Analysis. International Journal of Critical Infrastructures 9, 190–210 (2013)
23. Wilensky, U.: NetLogo. Centre for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, http://ccl.northwestern.edu/netlogo
24. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM Journal on Computing 1, 146–160 (1972)
25. Holland, J.H.: Genetic algorithms. Scientific American 267, 66–72 (1992)
26. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)

Choosing Models of Appropriate Resolutions for Simulation: A MRM Approach

Huachao Mao, Gongzhuang Peng, and Heming Zhang

CIMS, Department of Automation, Tsinghua University, China
maomaer@gmail.com, hmz@mail.tsinghua.edu.cn

Abstract. Multi-resolution modeling (MRM) is widely used in manufacture industry, environment science (climate, geometry, map), science (material, biology) and so on. Dozens of theories and methods are proposed to MRM. However, most of these MRMs are not designed for simulation, which leads to MRM failures in terms of information loss, consistency maintenance and resolution changes. To solve these failures, this paper introduces **C**onnector-**o**riented **Re**solution **S**tate Chart-based **S**ystem (CORES): a novel MRM approach with emphasis on choosing appropriate resolutions for simulation. In CORES, Resolution State chart, a UML state chart, is modeled to specify the resolution changes. And Connector, the connection of different resolutions, is proposed as a standby part to fulfill the four requirements on relationships between models of different resolutions. Finally, a dimension-variable linear system is modeled to demonstrate CORES approach, and the numerical results verify our approach.

Keywords: Multi-resolution modeling (MRM), Resolution State chart (ReS chart), Connector, resolution control, information difference.

1 Introduction

Multi-resolution modeling (MRM) are widely used, like, manufacture industry [1], combat simulation [2], environment science (climate[3], map [4]), natural science (material [3], biology [5]) and so on. Multi-resolution modeling attempts to capture the hierarchical and multilevel property of a complex system. Many efforts have been made on the multi-resolution modeling fundamental theories, approaches, implementations and applications.

These MRM approaches focus on different aspects of multi-resolution modeling, which leads to the different capacity and ability of multi-resolution M&S. Aggregation / disaggregation (A/D) [6] and CAVE [7] solve the problems of how different resolution models generate each other coordinately, but lose information during A/D. IHVR [8] and Mixed resolution modeling (MRMAide) [9] focus on how to build a different resolution model quickly, in which abstraction and generalization are often used, but narrow in specific fields. Distributed Component Substitution (DCS) [10] and Hooking (VRM) [11] face the difficulties of changing resolutions within simulation, without considering the concurrent interaction between different resolutions.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 188–199, 2013.

Selective View (SV) [6, 8] and Multiple Representation Entity (MRE) [12, 13], provide a unified entity to solve the interaction and consistency, at the cost of computational resource. BOM-based MRM [14] and Optimization-based MRM [15] pay attention to interface uniform from outside. DEVS-based [16] MRM attempts to formalize the MRM, which is limited to complexity of theory application and indirect description of resolution changes.

In the literature of MRM, to our best knowledge, there are few approaches synthetically considering controlling resolution changes, maintaining consistency, resolving interaction, and handling information loss. And most of them focus on consistency modeling and aggregation for MRM, but less emphasis has been put on the resolution changes and information loss within simulation environments.

To compel the attention to resolution changes and information loss, this paper design and implement a novel multi-resolution modeling framework, which fully accounts for the adaptive resolution change, dynamic structure, and information loss and consistency maintenance during simulation. Some key technologies of our work are discussed in Section 2. And then our multi-resolution modeling approach, CORES, is presented in Section 3. A case study is available and some numerical experiments are conducted in Section 4. Finally, we conclude this paper in Section 5.

2 Resolution Control and Consistency Maintenance

Before our approach is proposed, two key technologies should be addressed firstly. Resolution control and consistency maintenance are core components of our framework. Resolution state chart is presented to control resolution changes, and Connector is designed to solve the problem of consistency maintenance, information loss, and cross interaction.

2.1 Resolution State Chart

The root why most MRM theories have not fully revealed the power of MRM lies in their de-emphasis on controlling the resolution changes. The essential benefit of MRM is that one can choose the appropriate resolutions to adapt different simulation situations. From this view, an effective MRM method should provide the mechanism to specify the change of resolutions, and allow resolution to vary during the simulation. Although dozens of previous approaches have been put forward, few of them have the power to describe and control resolution changes. A/D merely addresses the function to realize the changes of resolution, but does not give us more information about when to disaggregate or aggregate. SV and MRE primarily do not allow the changes of resolution, let alone the control of resolution. DEVS based MRM attempts to specify the resolution change by the hierarchical and dynamic property of DEVS. However, the challenge of these kinds of MRM is the complexity and indirection of DEVS's specification of resolution changes.

In order to clearly, simply, and meaningfully solve the specification and controlling of resolution change, a novel method based on UML state charts is proposed, connoted as Resolution State Chart (ReS chart). Before we construct our Resolution State Chart, some definitions should be claimed first:

Definition of Key Items

– Resolution State

A binary notation describing status of all the resolution of a certain entity, and higher bit of a binary notation represents higher resolution. For example, S011 is a resolution state indicating that the high and middle resolution are activating and the low resolution is not.

– Resolution State Transaction type

According to the type of triggering source, resolution state transaction types are divided into three types: cross resolution interaction (CRI), human-machine interaction (HMI) and computational resource optimization (CRO). CRI is a most common triggering source to change resolution state, which describe the interaction between two resolutions. HMI captures the interest and focus of human. CRO is another limitation and condition to trigger the move of resolution state, which means running high resolution models with abundant computational resource and low resolution models with limited computational resource.

An Example of ReS Chart

Built on these definitions, an example of ReS chart is constructed as Fig. 1:

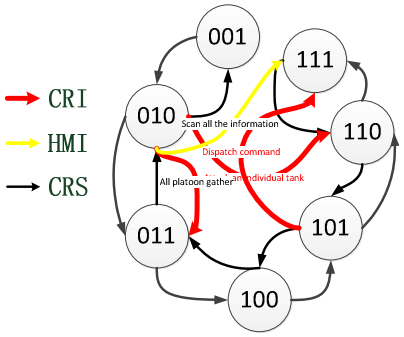


Fig. 1. Resolution state chart

Formalism of ReS Chart

Resolution state chart (ReS chart) is a UML state chart that can specify the resolution states of a certain multi-resolution object and describe the changes of resolution state. A ReS chart can be represented as a six-tuple [17, 18]:

$$R = (\Sigma, \Gamma, S, s_0, \delta, \omega) \tag{1}$$

These six tuples reveal six different facets of our ReS chart:

Tuple	Facet of ReS chart
Σ	Σ is a set of conditions and input events. All these conditions will be formed within our MRM framework. To be specific, conditions and input events are the set of CRI, HMI, and CRS, i.e. $\Sigma = \{CRI, HMI, CRS\}$
Γ	Γ is the output of ReS chart, which indicates the current resolution state of modeled object. It can be used in other higher ReS chart as an input, a foundation for a hierarchical nested ReS chart.
S	S is a set of all the possible resolution states in ReS chart. The number of S is finite, and is a function of the number of independent resolutions, connoted as N. Thus the possible number of resolution states is: $\lvert S \rvert = 2^N$
s_0	s_0 is always set as 0. Usually, in initial resolution state, no resolution model is activated, thus the initial state is 0
δ	δ is the state transaction function, which enable ReS chart to control resolution of modeled object. Here, this function is modeled to have two parts: 1) move the resolution state, 2) call the aggregation and disaggregation function, which is the connection between ReS chart and models.
ω	ω is the output function,which adopts mealy state machine: $\omega: S \to \Gamma$

2.2 Connector

In multi-resolution modeling, consistency is universally accepted as the fundamental requirement and obstacle to MRM. Natrajan [12] proposed an associative method, called multi-representation entity (MRE), unifying both high resolution entity (HRE) and low resolution entity (LRE) into a single Entity. Due to co-exist of HRE and LRE, MRE approach maintains consistency by calling the functions in its unified entity. However, this coupling between model entity and consistency maintain functions introduces many problems, like, unable to run at an individual resolution, hard to manage the resolution from outside and so on. Considering these drawbacks of MRE, we extract the consistency maintenance part as a standby proxy: Connector.

Definition of Key Items

- Information difference (ID)

  Information difference refers to the equivalent difference of information richness between different resolutions entities, which describes the information loss in A/D. For example, there are three tanks in a tank platoon. The positions of HRE have more information than the position of tank platoon. The outweighed information is the platoon formation. And such formation is our information difference.

- Resolution related variables (RRV)

  Like MRE, we define the variables related to different resolution as Resolution Related Variables. The position of tank is a case in point. The access and assignment to RRV enable A/D.

Requirement to Model the Relationship between Resolutions

The modeling of connection between two resolutions is the biggest difference between multi-resolution modeling and normal multi-model modeling. This connection has various forms, like, abstraction, aggregation, synthesis, rescaling, and their counterparts. No matter what kind of connection is, we propose that the connection should account for the following four requirements:

R1: Map functions: Provide a mechanism the map the variables of different resolutions.

R2: Maintain consistency: Inspect the inconsistency between resolutions, maintain consistency and transfer the variables of different resolution.

R3: Detect the conditions triggering the change of resolution state chart: Monitor the resolution related variables and trigger CRI, HMI.

R4: Balance the information difference between resolutions: Handle the information difference, extract and save some information difference, which can help to save information loss when resolution state changes. Those systems with Connector are retentive.

Structure of Connector

Plenty of theories and methods are proposed previously to model these property of connections, like, A/D, SV, MRE, IHVR, CAVE and so on. But most of them address only part of these requirements. MRE is relatively a complete one, and introduces an integrated method to put the connection into an individual entity, which leads to its complexity and static resolution state. By contrast, we design a separate and standby part to model the connection between resolutions, named Connector. To satisfy the four requirements, Connector absorbs information storage and CRI sensor part. As shown in Fig. 2, Connector has all the ingredients necessary to model the relationship between two resolutions.

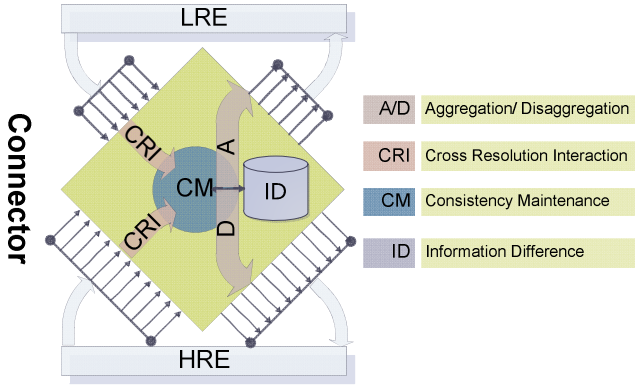


Fig. 2. Structure of Connector

- CRI sensor

  From left of Connector, CRI sensor is designed to detect the RRVs, to meet the monitoring requirement R3. CRI sensor can detect the resolution of a certain interaction, and dispatch this interaction to the right resolution entity.

- A/D

  The Aggregation/Disaggregation is a guarantee to realize resolution change. When ReS chart detects a change of resolution state, the A/D of Connector will be called to calculate the variables of the new resolution models. The essence of A/D is two functions, which map the RRVs between high resolution and low resolution. These A/D of Connector are designed specially, named A/D with information difference (IDAD), intending to cover the information loss as much as possible. IDAD is presented as follows:

$$\begin{aligned} [s, ID] &= \alpha(S) \\ S &= \delta(s, ID) \end{aligned} \tag{2}$$

  Where, α is aggregation, δ is disaggregation, S represents the RRV of high resolution, s represents RRV of low resolution, and ID is the information difference.

- Consistency maintenance

  Consistency Maintenance is to maintain the consistency between HRE and LRE at a certain schedule. Consistency Maintenance will call A/D to calculate the value of corresponding resolution related variables, and evaluate the information difference.

- Information difference storage

  Information Difference Storage is to store the Information difference, which will keep the system retentive, and balance the high fidelity and lower cost. The Information Difference is generated in two ways: A/D at resolution changes and A/D at consistency maintenance.

3 CORES: A MRM Approach

To address the two overlooked problems in MRM: controlling resolution changes and handling information loss, and the two prevailing problems of MRM: maintaining consistency and resolving interaction, this section will present our MRM approach based on above background technologies: ReS chart and Connector.

3.1 CORES Approach

Based on the above technologies, we design a Connector-Oriented Resolution State Chart-based System (CORES) to conduct MRM. Fig. 3 shows the process of CORES.

Apparently, there are four major steps of CORES:

1. Identify the resolution structure of object: Identify the resolution structure of object, using the SES method in [19].
2. Build single resolution models: The designer follows normal modeling process and builds the single resolution models, and then recognizes the RRV in a single resolution models. The individual models should also provide interface for Connectors to access and assign resolution related variables, which enables aggregation/ disaggregation and consistency maintenance.
3. Construct the Connectors: Section 2.2. The Connections between different resolution models are built by extracting the resolution related variables from individual models, and modeling the aggregation and disaggregation function and consistency maintenance policy, and then specifying the information difference of difference resolutions.
4. Draw the ReS chart: Section 2.1. Following the process of UML state chart construction, CORES draw ReS chart by determining resolution state first, and then identifying the CRO, CRI and HMI, finally constructing the resolution state transactions.

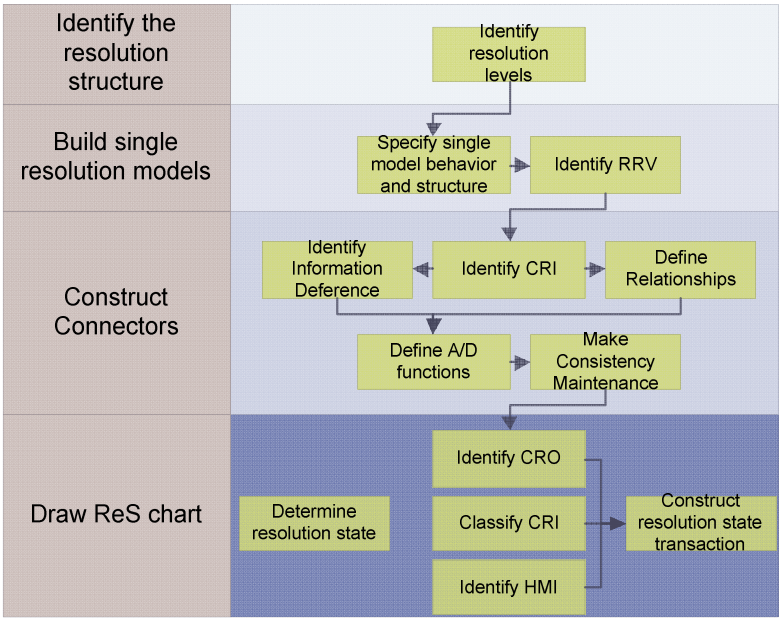


Fig. 3. Process of CORES

Advantages of CORES
The separate Connector is of great importance to a real multi-resolution modeling and simulation and brings much more benefit: easy to model, easy to update and de-coupled to single resolution models.

The central ReS chart plays the role of controller and balances the high fidelity with limited computational resource, which guarantees optimistic simulation in the way of multi-resolution simulation.

4 Case Study

4.1 Case Description: Dynamic Dimension of Large System

Large system consists of a great number of components, and it is relatively hard to build a precise model for large system, due to the complex relationship between these components and numbers of inputs and outputs. In the field of controlling and designing, low dimension's systems are fully studied. Thus, it is highly recommended that the large system can be projected into lower dimension's system [20], maintaining the main character of the primary system. Meanwhile, in the stage of simulation, all the different resolutions of large system can be activated to adapt the computational resource limitation and results analysis requirement.

4.2 Applying CORES Approach

Model the Individual Models
Please note that our approach is not focusing on the models of individual resolution models, but the connection and dynamic changes of different resolution.

- High resolution model

In our approach, the individual models should be modeled first. Most large system can be described as a set of differential equations, with states, time, and outputs. In order to demonstrate our approach directly, a three dimension MIMO linear system is studied:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + D \end{cases} \tag{3}$$

Where,

$$A = \begin{bmatrix} -0.1 & 0.2 & 0.01 \\ 0.01 & -0.2 & 0.01 \\ 0.001 & 0.002 & -0.5 \end{bmatrix}, B = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- Low resolution model

As mentioned in [20] there are lots of methods to get a lower resolution models. Here, we adapt the concept of dimension abstract, in which we evaluate the singularity values of the structure matrix A, and eliminate the smallest one for the minor influence of the smallest singularity value [20]. Therefore, by ignoring the smallest value, the third one, a low resolution model is obtained:

$$R_1 : \begin{cases} \dot{x}_1 = Ax_1 + Bu_1 \\ y_1 = Cx_1 + D \end{cases} \tag{4}$$

Where,

$$A = \begin{bmatrix} -0.1 & 0.2 \\ 0.01 & -0.2 \end{bmatrix}, B = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Construct Connector

The Connector has four basic parts: variable sensor, information difference, aggregation/disaggregation, and consistency maintenance.

Variable sensor should obtain x_1, x_2, y_1, y_2 from low resolution model, and $x_1, x_2, x_3, y_1, y_2, y_3$ from high resolution models. The information difference should be $x_3, A_{11}, A_{12}, A_{22}$. The aggregation and disaggregation should be

$$\begin{aligned} A &\equiv \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, x \equiv \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ u &\equiv \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, y \equiv \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \end{aligned} \tag{5}$$

And the consistency maintenance period is the same as the step size of simulation.

Hereto, we obtain the MRM of large system. The simulation will be conduct in next part.

Draw ReS Chart

There are two resolutions in large system, low resolution and high resolution. So the initial resolutions state is S00, and the number of resolution state is $|S| = 2^2 = 4$.

CRI is modeled as:

$$CRI = \{x_3 < 0.3, |\dot{u}| < 1\} \tag{6}$$

CRI captures the possible interactions between resolutions.

Finally, ReS chart is designed as Fig. 4:

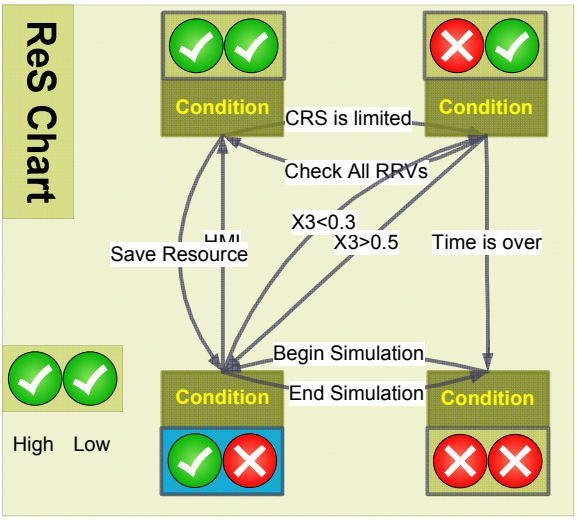


Fig. 4. ReS chart of large system

4.3 Numerical Results

The above multi-resolution model is simulated in Matlab. We simulated three modes: run with HRE only, run with HRE only and run with HRE and LRE.

Fig. 5 shows the results of different modes, and the error between modes.

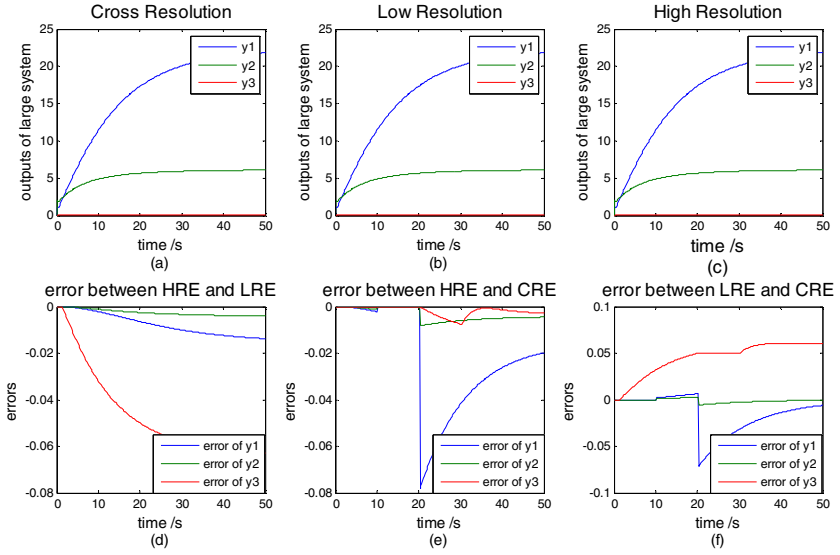


Fig. 5. Results of different modes

(a), (b) and (c) in Fig. 5 depict the outputs of Cross Resolution mode, Low Resolution mode, and High Resolution mode, respectively. The second row's figures reveal the accurate error of each mode. The results clearly show that Cross Resolution's accuracy is in the middle of HRE mode and LRE mode, which balances the computational cost and accuracy.

Fig. 6 is the sequential changes of Resolution State of Cross Resolution simulation mode. This figure verifies how CORES works: the RRV can be monitored, RS changes can be triggered by CRI, and the A/D can be called by ReS chart.

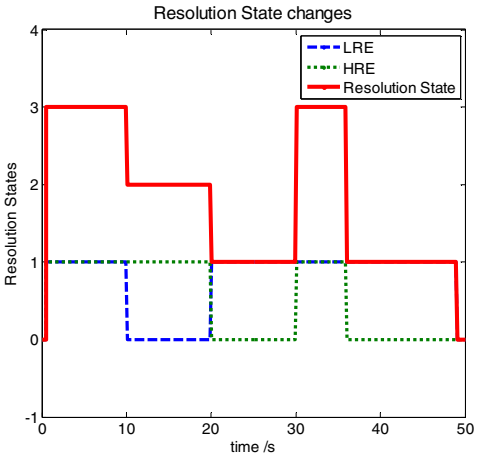


Fig. 6. Resolution State changes of large system

5 Conclusion

To address the problem of consistency maintenance, information loss, and resolution control in previous MRM approaches, this paper develops a novel MRM approach CORES. The ReS chart is a formalized specification of resolution changes during the simulation, which identifies CORES easier to model, more directly to control resolution than previous ones. Moreover, ReS chart is application of UML state chart. Thus the formalism can ensure the validated fundamental theory of CORES. Connector, a complete specification of connections between resolutions, is another characteristic of our approach. Connector owns the power to satisfy the four requirements for a connection part of resolutions: 1) realize aggregation and disaggregation with A/D part; 2) maintain consistency by CM; 3) monitor the CRI, by access to resolution related variables; 4) balance the information loss and high computational resource by Information Difference.

CORES is applied to model the MRM of large system. The modeling procedure demonstrates the advantages of our approach, and the results of Matlab simulation verify the validation of CORES.

Acknowledgement. This work is supported by The GF Basic Research Foundation of China (Grant No. A0420110104).

References

1. Werther, J., Heinrich, S., Dosta, M., Hartge, E.: The ultimate goal of modeling—simulation of system and plant performance. Particuology 9, 320–329 (2011)
2. Petty, M.D., Franceschini, R.W., Panagos, J.: Multi-Resolution Combat Modeling. Engineering Principles of Combat Modeling and Distributed Simulation, 607–640 (2012)
3. Horstemeyer, M.F.: Multiscale modeling: a review.: Practical aspects of computational chemistry, pp. 87–135. Springer (2010)
4. Stoter, J., Visser, T., van Oosterom, P., Quak, W., Bakker, N.: A semantic-rich multi-scale information model for topography. Int. J. Geogr. Inf. Sci. 25, 739–763 (2011)
5. Dada, J.O., Mendes, P.: Multi-scale modelling and simulation in systems biology. Integrative Biology 3, 86–96 (2011)
6. Davis, P.K.: An introduction to variable-resolution modeling. Naval Research Logistics (NRL) 42, 151–181 (1995)
7. Powell, D.R.: Control of entity interactions in a hierarchical variable resolution simulation. Los Alamos National Lab., NM, United States (1997)
8. Davis, P.K., Bigelow, J.H.: Experiments in MRM. RAND MR-100-DARPA (1998)
9. Cubert, R.M., Goktekin, T., Fishwick, P.A.: MOOSE: architecture of an object-oriented multimodeling simulation system, 78–88 (1997)
10. Rao, D.M., Wilsey, P.A.: Multi-resolution network simulations using dynamic component substitution, 142–149 (2001)
11. Li, V.S., Ka, M., Me, V.N., Hart, P., V S Afa V R I K, J.: Simulating Details on Demand Using Variable-Resolution Modeling., Vol. 98. 39-46 (1998)
12. Natrajan, A.: Consistency maintenance in concurrent representations. University of Virginia (2000)
13. Natrajan, A., Reynolds Jr, P.F.: Concurrent Representations for Jointly-executing Models.. University of Virginia Technical Report No. CS-2001-20 (2001)
14. Chase, T., Gustavson, P., Eifert, L., RDECOM, U.A.: The Application of Base Object Models (BOMs) for Enabling Multi-Resolution Modeling (2004)
15. Drewry, D.T., Reynolds Jr., P.F., Emanuel, W.R.: Optimization as a tool for consistency maintenance in multi-resolution simulation. DTIC Document (2006)
16. Liu, B.H., Huang, K.D.: A formal description specification for multi-resolution modeling (MRM) based on DEVS formalism. In: Kim, T.G. (ed.) AIS 2004. LNCS (LNAI), vol. 3397, pp. 285–294. Springer, Heidelberg (2005)
17. Harel, D.: Statecharts: A visual formalism for complex systems. Sci. Comput. Program. 8, 231–274 (1987)
18. UML state machine - Wikipedia, the free encyclopedia, `http://en.wikipedia.org/wiki/UML_state_machine#Hierarchically_nested_states`
19. Kim, T.G., Lee, C., Christensen, E.R., Zeigler, B.P.: System entity structuring and model base management. IEEE Transactions on Systems, Man and Cybernetics 20, 1013–1024 (1990)
20. Kokotovic, P.V., O'Malley Jr., R.E., Sannuti, P.: Singular perturbations and order reduction in control theory — An overview. Automatica 12, 123–132 (1976)

Effects of Random Number Generators on V2X Communication Simulation

Robert Protzmann[1], Björn Schünemann[2], and Ilja Radusch[2]

[1] Fraunhofer FOKUS, Berlin
Automotive Services and Communication Technologies
robert.protzmann@fokus.fraunhofer.de
[2] Technische Universität Berlin
OKS / Daimler Center for Automotive IT Innovations

Abstract. Detailed simulation models have to incorporate random effects. Since the generation of randomness is subject to several shortcomings, this needs to be considered for the setup and evaluation of simulations. On the basis of well-known metrics for the domain of V2X communication we will evaluate the influences of differently generated random sequences on the simulation. We will show that it is important to pay attention to avoid skewed results caused by random number generation and ensure the statistical relevance of the simulation series. It can be stated that well established random number generators are suitable. Meaningful simulation results rely rather on a sufficient number of simulation runs which in turn will depend on the applied models.

Keywords: RNG, V2X simulation verification, VSimRTI.

1 Introduction

Vehicle-to-X communication is an up-and-coming technology for improvement of road safety, traffic efficiency, and infotainment applications [1]. At the present time, already field operational tests are performed to investigate the behavior of V2X communication based applications in reality. However, a good portion of research is done with the help of simulations.

For credible statements about the application behavior, it is important to ensure simulations that are able to produce qualified results. Much effort is spent for the development of preferably detailed models that include the according realworld properties. E.g. in [2] a V2X simulation environment was extended with advanced communication models. Since many processes in nature can only be modeled as random, the computational models need to rely on generators for randomness. In the V2X simulation context, examples for models using Random Number Generators (RNG) are e.g. the vehicle movement models in a traffic simulator or the radio propagation models in a communication simulator. The expected value of these models can be described to follow a known probability distribution. During the simulation run, a RNG then delivers a certain representation of the distribution or more concrete a sequence of individual samples as a

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 200–211, 2013.

subset of the whole sample space. At this point a first important issue arises. A sufficient coverage of the sample space and finally a result close to the expected value needs multiple samples to be generated which means in turn that multiple simulation runs need to be performed. Depending on the law of large numbers, the number of simulation runs tends to infinite for a perfect result. However, due to practical reasons a reduction of simulation runs is always desirable to save computing costs and time. A second issue is to avoid skewed results. This can be ensured for individual simulation models, as the results can be cross compared with the according probability distributions. When finally the application behavior is evaluated, all aspects of the simulated models, protocols and application algorithms play together. Hence, it is not trivial to trace the aggregated result back to a certain influence or even to be sure that the whole spectrum of the sample space is represented. This may lead to wrong statements or wrong optimizations of the application algorithms, merely due to an insufficient setup of the simulation series.

In this paper we present an approach to ensure correct results in simulations based on randomness. The central question is how strong the used RNGs do affect the results. If the choice of the RNG itself or the parameterization of the RNG would have a substantial impact on the results, it would mean that great care is advisable when setting up simulations with random components. It is undesirable that RNGs affect the functioning of the simulated processes even more than certain details in the models themselves. Furthermore, we show a direction how the results converge with different numbers of simulation runs.

The paper is structured as follows. In the next Section 2 we give a short overview of important properties of RNGs and introduce three examples which are used in this work. In Section 3 we present our simulation setup and give additional details on the randomness-based models and the metrics to measure the impact of randomness. The results are given in Section 4. The findings are resumed and the paper is concluded in Section 5. Finally, we state the future work on the topic in the outlook in Section 6.

2 Random Number Generators

For simulation purposes two features of RNGs are demanded, 1) the generation of truly random and statistically credible results with a limited number of simulation runs and 2) the reproducibilty of results, which is important for debugging and comparison [3]. Due to the latter requirement only deterministic RNGs can be used. These RNGs are based on deterministic (software) algorithms and deliver, when the starting situations are equal, also equal results. As the generated sequences are not truly random anymore, they are called pseudo-random. The dimension of incorporated errors compared to true randomness depends on the quality of the used generator and its according initialization. The quality of a RNG can be described first by the distribution of the random samples which should be uniform and the individual samples should be statistically independent [4]. The statistical independence of the samples is demanded so that the

aggregated result converges to the expected value according to the law of large numbers [5]. Several simulation model implementations are based on probability distributions which are not necessarily uniform. With the method of the inverse transform sampling it is possible to generate any distribution given its cumulative distribution function (cdf) out of the uniform distribution. If the distribution is not uniform, certain number patterns are generated more often. This effect is known as falling in the planes [6]. The second characteristic to measure suitability of RNGs is the period of the random sequence, before it starts again. This limit is a result from the fact that all deterministic RNGs are finite state machines. However, a long period not necessarily ensures a high quality, but a short period is often a problem.

Actually there exist a quite high number of suitable RNGs, but in our work we have concentrated on the Linear Congruential Generator (LCG) [7], the Mersenne-Twister [8], and the Blum-Blum-Shub Generator (BBS) [9].

The importance of the **Linear Congruential Generator** stems from the fact that it is very well established for a long time. It is part of the runtime libraries of the GNU Compiler Collection and the Java SDK and consequently it is directly available for the simulation program code. The LCG is based on the linear modulo operation, described with the following Equation 1.

$$x_i = (a * x_{i-1} + b) \bmod m \tag{1}$$

A typical problem of the LCG is the occurrence of the falling-in-the-planes effect which means that the generated numbers are not distributed perfectly uniform. Furthermore, depending on the platform, the period as one important measure for the quality of RNGs, may be quite short, starting at an order of 2^{32} (or respectively 10^9 to 10^{10}) for 32 bit systems. The LCG in Java.util possesses a period of 2^{48} (ca. 10^{14}).

The **Mersenne-Twister** is a newer generator which delivers random sequences of very high quality [8]. Hence, it is very well suited for simulation purposes and already incorporated in various simulators as SUMO, OMNeT++ and JiST/SWANS. The components of the Mersenne-Twister are given in the following recurrence Equation 2. Here, the | operation is a just a concatenation of the upper and lower part. The multiplication with the matrix A performs the twist transformation. Finally, $\oplus$ is the bitwise XOR which is actually an addition with modulo two.

$$x_{k+n} = x_{k+m} \oplus (x_k^u | x_{k+1}^l) A \tag{2}$$

The MT19937 is a specific implementation of the Mersenne-Twister that features an outstanding period of 2^{19937} (10^{6001}). Moreover, it generates an almost uniformly distributed sequence and passes most of the mathematical randomness tests. Due to these advantages it is commonly used for large scale simulations, e.g. according to the Monte-Carlo approach.

The **Blum-Blum-Shub** was designed for cryptographic systems. For this purpose it needs to satisfy slightly different requirements, e.g. that the random sequence cannot be predicted. The unpredictability arises from the quadratic

residue in the modulo-operation as outlined in Equation 3.

$$x_{i+1} = x_i^2 \bmod n \tag{3}$$

Whereas n (and in turn the period of the BBS) is defined as the product of two prime numbers p, q of the form $4k + 3$. For a sufficient period these primes need to be very high which in turn slows down the simulation performance. However, this fact allows different configurations for a comparison of a period that should be still sufficient for our simulations with a bad configuration.

3 Simulation Study

In our investigation we place the emphasis on the dependency of the simulated communication behavior on different RNGs. We simulate a single-access scenario and a multi-access scenario where we apply the already introduced RNGs LCG, Mersenne-Twister, and BBS with an appropriate and inappropriate configuration. We analyze three communication metrics to compare the influences of the different RNGs. To avoid side-effects we have configured the models for traffic simulation in our setup to simulate exactly the same deterministic behavior in each V2X scenario. For all V2X simulations we use the simulation environment VSimRTI [10]. VSimRTI couples discrete event simulators from different fields. In our concrete case we use JiST/SWANS for the simulation of the communication stack. Furthermore, we use the application simulator VSimRTI_app which usually embeds the logic of V2X applications, but can also be used to generate only a certain data load for the communication simulator. At last we use the traffic simulator SUMO for the vehicle movements.

3.1 Analyzed Communication Models with Randomness

In our setup, the **IEEE 802.11p MAC layer** is the first module which incorporates a random component. When the Distributed Coordination Function (DCF) of the sender senses the radio channel as occupied upon a transmission attempt, it needs to listen to the channel to defer the transmission until the channel is free again. But when during an ongoing transmission more than one sender are waiting for the moment when the channel is free again and would directly start their own transmission, a packet collision would occur. Thus, the DCF provides the backoff procedure to minimize the probability of a concurrent transmission. For the backoff procedure one random integer value is selected out of the interval of the contention window. This means e.g. for the case of broadcasting where the initial contention window is 31 slots, every node selects a random value between 0 and 30. According to the back backoff procedure, the node has to sense the channel as free for the duration of the slot time and then decrement the previously selected backoff timer. When the backoff timer reaches the value of 0, the node can start its transmission. For a fair prioritization of the different senders, the used random variable needs to be distributed uniformly.

Furthermore, the **IEEE 802.11p PHY layer** model uses a random component to simulate a packet error rate (PER). In contrast to the random component in the MAC layer it is not a feature of the protocol according to the IEEE 802.11p standard specification, but it is more a characteristic of the simulation model itself. A packet error exists in the situation when PHY layer was able to sense at least a received signal, but the signal, or more concrete the SNIR (Signal to Noise and Interference Ratio), is to weak that the signal processing chain in the PHY is not able to decode the packet correctly. As a signal processing chain would be computationally too time-consuming, an efficient abstraction is implemented. The model calculates the PER out of the SNIR, the symbol modulation and the packet length. The PER can have a value between 0 and 1. To determine if the packet is eventually received or not, a comparison is performed with a random value which is sampled uniformly between 0 and 1. If the PER is lower than this random value, the packet is regarded as successfully received.

At last, more realistic models for the **Radio Channel** also incorporate fading effects. We use the well-established Rayleigh and Rice Fading Models for our investigation. These fading models already incorporate probability distributions which are not uniform. As the Rayleigh Fading is mathematically described by the integral cdf of the Rayleigh distribution, the inversion sampling method can be applied for the concrete implementation of the Rayleigh model. Hence, the implementation can be reduced to one random sample of the uniform distribution. The Rice Fading is mathematically based on a Bessel function which needs to be approximated numerically in the implementation and uses two random samples for one calculation.

3.2 Analyzed Metrics for Simulation

The **Backoff Timer** is linearly dependent on the calculated random value from the according call in the MAC module. Thus, the quality of the applied RNG should be directly assessable on the basis of the distribution of the backoff timer. In the single-access case, the sending node is always able to directly transmit its packet and no further backoff mechanism is needed. Hence, this metric is only evaluated in the multi-access case.

The **Channel Access Delay** is the time from the moment where a packet arrives in the MAC layer to the moment where it can be send to the channel. It depends in the single-access case merely on a possible simulated processing delay for the layer, yet our model considers no processing time. In the multi-access case it also depends on the selected backoff timer and furthermore on the backoff timers of the other simulated nodes. E.g. when two nodes wait for a free channel and then select their backoff timer, the node with the longer backoff timer needs to additionally defer its transmission until the node with the shorter backoff timer has finished its own transmission. The channel access delay is only evaluated in the multi-access case.

The **Packet Delivery Rate** shows the influences of the random numbers on the simulation of the radio channel fading. The PDR for a single packet can be either 0 or 1. So it needs to be evaluated in a statistical context, whereas

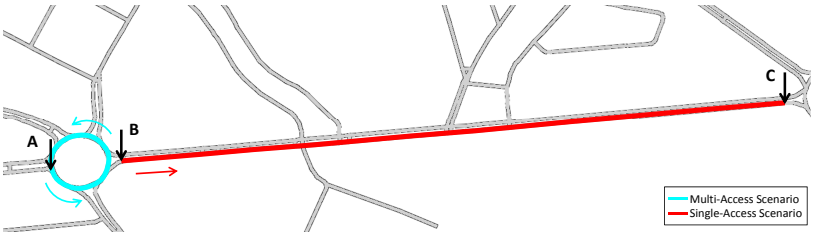


Fig. 1. Simulated scenarios multi-access (circle), single-access (line)

the arithmetic-mean aggregation of all packet sending attempts in the simulated time frame delivers the most significant result. Especially in the single-access case the PDR depends directly on the applied fading models which affect only the signal term of the SNIR for the packet error model in the PHY layer. When no other sender exists, the noise and interference term is constant and the medium access function works transparently. In the multi-access case it is not trivial to trace the result back to one single influence. Hence, this metric is only evaluated in the single-access scenario.

3.3 Simulated Scenarios

The two simulation scenarios for the evaluation are located in the Strasse des 17. Juni in Berlin (Figure 1). In the multi-access scenario especially the backoff RNG is considered. 20 vehicles communicate concurrently via periodic broadcasts with a frequency of 10 Hz. The nodes have to sense the medium and apply their backoff mechanisms to avoid possible packet collisions. However, with the given number of senders and the configured sending frequency the radio channel is still free enough that every packet can be successfully transmitted. The hidden terminal problem and consequently a malfunction of the backoff procedure does not exist in this scenario, as all vehicles are always in communication range in the cyan colored traffic circle, which has a diameter of 150m. The vehicles are introduced in the simulation with a time delay at position A and move around the circle again to point A (Fig. 1). This round is repeated until the simulation ends.

The single-access scenario is intended to be used primarily for the evaluation of fading RNGs. It consists of a stationary node, located at point B which is the receiver and a sender which starts also at point B near the receiver and then moves away on the red route with a constant speed of 10 m/s towards point C (Figure 1). The final point C of the sender is nearly 1.9 km away from receiver. Consequently the route is long enough up to the maximum distance where the connection between both nodes is finally lost. The sender emits messages with a constant period of 10 Hz. In this way communication measurements with a resolution of 1 m can be captured.

The following calculation should clarify the maximum number of random samples needed for the simulation. For every sent packet one backoff timer needs to be generated. For every packet a fading channel needs to be simulated (in the

case of Rice Fading by the use of 2 random samples). Finally, when the signal energy at receiver side is at least high enough to sense the packet, the PHY layer PER is calculated. For a traffic-efficiency scenario where $n = 400$ vehicles with a sending frequency of $f = 10Hz$ are investigated over one hour of simulation time ($t = 3600s$), the result would be the following (Eq. 7), whereas N_{BT} is the number of random samples for the backoff timer, N_{FC} for the fading channel, and N_{PER} for the packet-error check.

$$N_{BT} = n \cdot f \cdot t = 14400000 \tag{4}$$

$$N_{FC} = 2 \cdot n(n-1) \cdot f \cdot t = 11491200000 \tag{5}$$

$$N_{PER} = n(n-1) \cdot f \cdot t = 5745600000 \tag{6}$$

$$N = N_{BT} + N_{FC} + N_{PER} = 1.72512 \cdot 10^{10} \tag{7}$$

When the period of a RNG is considered as quality measure, it can be stated that it should be in the order of 10^{11} or higher to be appropriate for the scenario. The LCG and certainly the Mersenne Twister should be appropriate. The BBS in the good configuration should also exhibit a period higher than N.

4 Simulation Results

The results for three metrics are presented in the following. The backoff timer is selected, because it is known that the samples need to be distributed uniformly. Furthermore, the channel access delay is selected as it already includes the protocol behavior of the IEEE 802.11p MAC layer. Finally, the PDR is presented as a metric where the required distribution is also known in advance, but it is not uniform. Moreover, the applied models for Rayleigh and Rice Fading rely on a different number of random samples for one calculation.

4.1 Backoff-Timer

Two comparisons are carried out for the backoff timer. First, the result of each individual RNG is compared with the uniform distribution, which is important to facilitate a fair prioritization of all senders in the system. Second, the BBS is compared with the good and the bad parameterization to get an overview, if it is possible at all to set up the RNG in an insufficient way. In Figure 2 one histogram for each RNG is displayed. The histograms directly allow a visual examination of the probability distributions.

Generally, the LCG, the Mersenne-Twister, and the BBS (good) produce a fairly uniform distribution. In comparison with the first two generators, it can be noticed that distribution for the BBS (good) is slightly more uneven. The second comparison of the Blum-Blum-Shub shows that the BBS (bad) is not able to generate a uniform distribution at all, although the period of this RNG should be able to deliver more than the required number of different samples. This may be traced back to the fact that the random integer number (in the interval

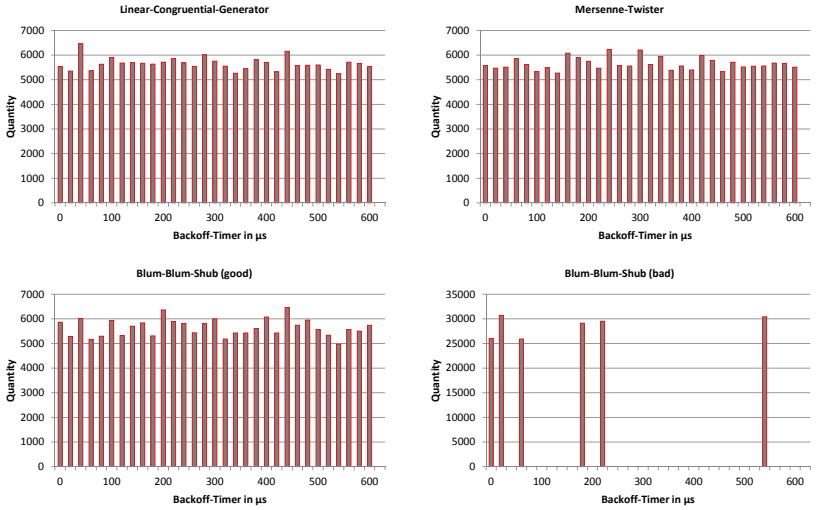


Fig. 2. Distribution of the backoff timer, different RNGs, 30 simulation runs

between 0 and 30) is generated by the use of a real number sample (between 0 and 1) and a subsequent real to integer conversion in the implementation. This conversion may intensify the effect that not the whole sample space is covered.

Additionally, we have applied the Kolmogorov-Smirnov test to measure the goodness of fit [11]. The following table 1 contains the result for the K-S test D-statistic, which indicates the maximum deviation of the empirical distribution generated by the RNGs and the hypothetical (uniform) distribution. A perfect fit would result in the D-value of 0. This test supports the previous finding that the LCG, Mersenne-Twister and BBS (good) are very close to the uniform distribution, while the BBS (bad) involved insufficient results.

Table 1. K-S test D-statistic for the evaluated RNGs

RNG	**K-S test D-statistic**
LCG	0.006465
Mersenne-Twister	0.006071
Blum-Blum-Shub (good)	0.006791
Blum-Blum-Shub (bad)	0.435661

The metric of the backoff timer especially demands a uniform distribution of random numbers. Generally, a collision occurs when two nodes select the same backoff timer and start their transmission at the same time. When certain random values are generated more frequently due to the RNG, this issue is over amplified in the simulation and leads to skewed results. Thus, a wrong selection and parameterization of the RNG builds a wrong basis of the whole simulation.

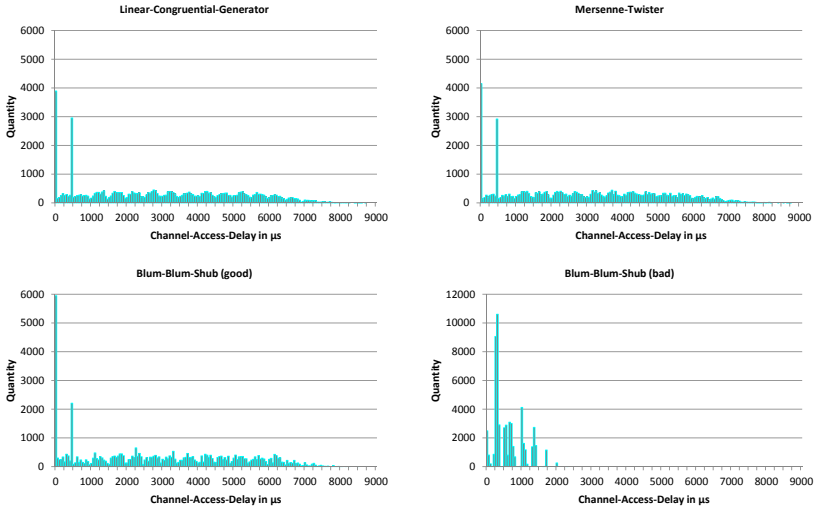


Fig. 3. Distribution of the channel access delays, different RNGs, 30 simulation runs

4.2 Channel Access Delay

The channel access delay already incorporates the protocol characteristics of the IEEE 802.11p MAC layer in the situation when multiple nodes compete for the radio channel. The diagrams (Figure 3) show that it is not trivial to compare the results with a known distribution.

For the Mersenne-Twister, the LCG and the BBS (good) there exists one high peak for the lowest delay which represents the case when one sender has selected the lowest backoff timer of all competing nodes and can be the first sender. The second highest peak may indicate nodes which have selected a higher backoff timer have to defer their transmission to be the second sender, but it also includes nodes which are the first sender and have selected a higher backoff timer and no other sender is competing for the access. The BBS (bad) again produces a different result compared to the other configurations.

The results show that errors from the backoff timer emerge to have an effect on the channel access delay. Already for this metric it is difficult to acknowledge the correctness. This means in turn that for key performance indicators on application layer it would be more complicated to prove the result. Hence, it is recommended to record metrics as the backoff timer for a direct comparison to ensure the correct simulation setup.

4.3 Packet Delivery Rate

The PDR evaluation first aims to compare how models with different number of random samples influence the result. Second, a comparison is drawn where only few simulation runs are evaluated, because few simulation runs are desired of practical reasons to save time.

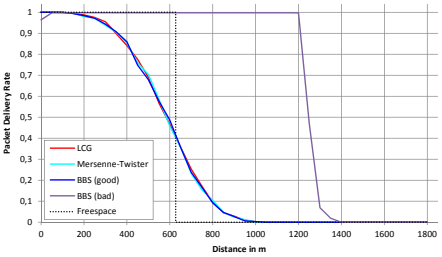


Fig. 4. PDR for Rayleigh Fading, different RNGs, all 30 simulation runs

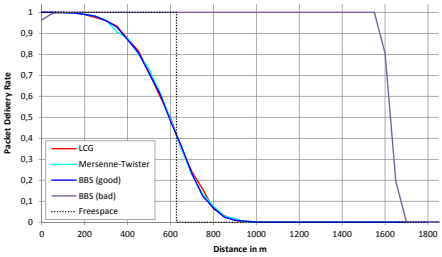


Fig. 5. PDR for Rice Fading, different RNGs, all 30 simulation runs

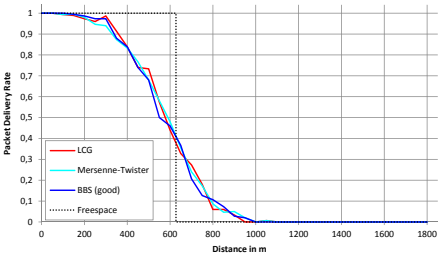


Fig. 6. PDR for Rayleigh Fading, different RNGs, only 3 simulation runs

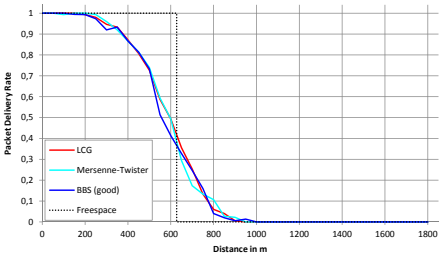


Fig. 7. PDR for Rice Fading, different RNGs, only 3 simulation runs

The results are presented for the arithmetic mean of the PDR for 30 simulations with Rayleigh (Fig. 4) and Rice Fading (Fig. 5). Additionally, the PDR graph for the Freespace Model is included as a reference. In both figures it is observable that the results for the LCG, the Mersenne-Twister and the Blum-Blum-Shub (good) are almost identical. Furthermore, a comparison with the Freespace reference shows that the graphs of the PDRs have the correct trend and start to decline at the right distance. The Blum-Blum-Shub with the bad configuration exhibits a different and certainly wrong behavior. This behavior is even worse for the evaluation of Rice Fading and can probably be traced back to the fact that Rayleigh Fading uses only one random call for one calculation, while Rice Fading uses two random realizations.

In a second evaluation again Rayleigh (Fig. 6) and Rice Fading (Fig. 7) was analyzed but only 3 simulations where selected for the PDR aggregation. Assuming that one RNG possesses a lower quality regarding the random number distribution, it would show a less smooth run. As the Blum-Blum-Shub (bad) already showed a wrong result in the first evaluation, it was omitted in the following diagrams. The graphs display that no RNG produces an especially uneven result. Moreover a comparison of Rice and Rayleigh Fading indicates that the application of two random realizations in the Rice model leads to an already smoother result. This means that the number of required simulation runs for a

satisfying coverage of the sample space depends more on the applied simulation models than on the applied RNG.

To sum up the results, we see that the LCG and the Mersenne-Twister exhibit nearly identical results for all evaluated metrics. The sufficiently configured Blum-Blum-Shub shows similar results for the PDR. The distribution of the backoff timer seems to be not completely uniform, but one can assume that for the medium scale scenario the quality of the randomness is sufficient. Yet, there also exist RNGs which lead to a completely wrong outcome, as the BBS with the configuration of insufficient p and q values. In this case the period was although selected to meet at least the requirement for number of random values, but the sequence was not uniformly distributed. The channel access delay is presented as a metric where the comparison between the different RNGs was still possible, while it is not straightforward to assess the correctness as it incorporated already several influences.

Aside from this, it is important to use known random seeds. With equal random seeds the simulation behavior can be reproduced exactly for debugging purposes. With different seeds in the multiple simulation runs the result converges to the expected value according to the law of large numbers. Whereas the number of simulation runs for a statistical significance depends more on the used models than on the used RNGs.

Finally, the approach to record and evaluate metrics with an expected result is deemed to be successful to ensure a correct simulation setup. In our case the backoff timer and the PDR are suitable, because it is known that they are demanded to follow the uniform or respectively the Rayleigh and Rice distribution. In this way it is also possible to check the statistical relevance.

5 Conclusion

In this paper we have evaluated the influences of RNGs on the simulation results of V2X communication scenarios. We have set up two scenarios for a single-access and a multi-access case. The focus of our investigations was put especially on the communication metrics of the packet delivery rate, the distribution of the backoff timer, and the channel access delay. In the simulations we have applied different RNGs, 1) the LCG which is the standard RNG in many compilers for famous programming languages as C/C++ and Java, 2) the Mersenne-Twister which is known to deliver especially high quality random sequences and for comparison the Blum-Blum-Shub in a 3) good as well as in a 4) bad configuration.

We have seen that it is straightforward to assess the behavior affected by the RNG for certain metrics where a dedicated result is expected in advance. This was the case for the backoff timer and the PDR. The metric of the channel access time was not trivial to assess as it incorporates already several aggregations. Key performance indicators for applications depend even on more influences.

For the actual comparison of the RNGs we have seen, that the LCG and the Mersenne-Twister both delivered very good results and the BBS in the good configuration is still sufficient. In contrast, the BBS in the bad configuration

always delivered completely wrong results. This means that indeed care needs to be taken for the correct choice of the RNG and its parametrization.

A comparison of the number of simulation runs to achieve a statistical significant result lead to the finding that it depends more on the simulation model which relies on the random samples than on the RNGs themselves.

Finally, as specific lessons learned it can be stated that known metrics should be recorded and evaluated to ensure the correctness and the sufficient number of simulation runs to cover the whole sample space for a statistically significant result.

6 Outlook

As already mentioned, an evaluation of known metrics along the way is important to ensure meaningful results for the actually investigated key performance indicators. Hence, in future simulation studies we want to introduce further statistical tests for the verification to our workflow. The presented K-S test was applied to check the correct distribution of the analyzed metrics. In addition, there exist other suitable tests as the runs-test according to Wald-Wolfowitz for the check of the statistical independence of the random samples.

References

1. Hartenstein, H., Laberteaux, K.: VANET: vehicular applications and internetworking technologies, vol. 1. Wiley Online Library (2010)
2. Protzmann, R., Mahler, K., Oltmann, K., Radusch, I.: Extending the v2x simulation environment vsimrti with advanced communication models. In: 2012 12th International Conference on ITS Telecommunications (ITST), pp. 683–688. IEEE (2012)
3. Law, A.M.: Simulation Modeling and Analysis. 4rd edn. McGraw-Hill Higher Education (2007)
4. Matsumoto, M., Saito, M., Haramoto, H., Nishimura, T.: Pseudorandom number generation: Impossibility and compromise. J. UCS 12(6), 672–690 (2006)
5. Fishman, G.S.: Monte Carlo: concepts, algorithms, and applications. Springer series in operations research. Springer (1996)
6. Marsaglia, G.: Random numbers fall mainly in the planes. Proceedings of the National Academy of Sciences 61(1), 25–28 (1968)
7. Entacher, K.: A collection of classical pseudorandom number generators with linear structures: advanced version (2000)
8. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation (TOMACS) 8(1), 3–30 (1998)
9. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM Journal on Computing 15(2), 364–383 (1986)
10. Schünemann, B.: V2x simulation runtime infrastructure vsimrti: An assessment tool to design smart traffic management systems. Computer Networks (2011)
11. Massey Jr, F.J.: The kolmogorov-smirnov test for goodness of fit. Journal of the American statistical Association 46(253), 68–78 (1951)

Generating the Confidence Interval of Time Averaged Estimator Using Threshold Bootstrap

Jinsoo Park[1], Yun Bae Kim[2,*], Haneul Lee[2], Gisun Jung[2], Sangeun Kim[2], Jeong Seok Kang[3], Jay Jou[3], and Yeon Jae Jung[3]

[1] Dept. of Management Information Systems, Yong In University, Republic of Korea
jsf001@yongin.ac.kr
[2] Dept. of Systems Management Engineering, Sungkyunkwan University, Republic of Korea
{kimyb,roar5334,sunmoon00,yuan19}@skku.edu
[3] Dept. of Management of Technology, Sungkyunkwan University, Republic of Korea
{jskang0213,socalboi,jyjs2004}@skku.edu

Abstract. Threshold bootstrap is a modified bootstrap method that resamples data from the autocorrelated simulation outputs. The threshold bootstrap calculate the ensemble average of sample as an estimator for population mean as do other bootstrap methods. Sometimes, however, an estimator of simulation output is generated by the concept of time average such as mean queue size in queueing system. In this situation, to analyze the simulation output more efficiently, we introduce a method of generating the confidence intervals for time averaged estimators using the threshold bootstrap. Numerical examples are provided to verify the confidence interval produced by our method.

Keywords: Simulation Output Analysis, Confidence Interval, Time Averaged Estimator, Threshold Bootstrap.

1 Introduction

We generally use the ensemble average method to estimate parameters or statistic from simulation outputs. The central limit theorem (CLT) guarantees that the ensemble average of independent and identically distributed (IID) sample approximately follows the normal distribution. Accordingly, we can infer the confidence intervals and perform the hypothesis tests simply. In simulation output analysis, however, we sometimes should hire the concept of time average to infer an estimator such as the mean queue size in queueing system. Since it is not sure that the time averaged statistic holds the CLT, we have difficulties to calculate the confidence interval of interested statistic. If there are autocorrelation structures in the simulation output, the difficulties become worse. In this situation, we should compute the percentile confidence interval using the IID data streams from the independent simulation replications. As these replications may produce huge cost or long time, it is not efficient to apply in real world. However, employing bootstrap method is one of the applicable methods that reduce time and cost [1-2].

* Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 212–218, 2013.

Bootstrap method was originally introduced by Efron to resample the IID data [1-2]. There are also several modified methods which are applicable to data that have the autocorrelation structures [4-11]. Among those methods, moving block bootstrap (MBB) [6], [9-10], stationary bootstrap (SB) [5], [11] and threshold bootstrap (TB) [3], [7-8] are commonly and widely used for autocorrelated data in various fields. The bootstraps are nonparametric methods which generate the percentile confidence interval using the resampled data from the only single output of the simulation [1]. Especially, the TB shows the competitive performances to infer the parameters or the statistic [7]. Accordingly, we apply the threshold bootstrap to generate the confidence interval of time averaged estimators in this paper.

This paper is organized as follows. Section 2 introduces the brief concept of ensemble average and time average and discusses the difficulties of calculating the confidence intervals of the time averaged estimators. Section 3 describes an algorithm for generating the percentile confidence interval of time averaged estimator using the threshold bootstrap. Section 4 provides the numerical examples to guarantee the relevance of our proposed method and section 5 concludes our research and suggests further studies.

2 The Time Average

2.1 Ensemble Average vs. Time Average

The concept of ensemble average observes the state on discrete time space. Denoting P_k is an ensemble averaged probability that outside observer sees the system in state k, P_k is defined as follows.

$$P_k = \lim_{n\to\infty} \frac{the\ number\ of\ state\ k\ observed}{N} \tag{1}$$

N is total number of observations or replications. The sample mean $\bar{S}$ is also defined as following equation for sample $\{S_1, S_2, \dots, S_n\}$.

$$\bar{S} = \frac{1}{n}\sum_{i=1}^{n} S_i \tag{2}$$

On the concept of time average, since a state occupies the continuous random time, we should observe both the state and its occupying time. The time averaged probability is defined by the rate of occupying time in state k for the total observing time:

$$P_k = \lim_{T\to\infty} \frac{T(k)}{T} \tag{3}$$

T is total operation time of system and $T(k)$ is occupying time in state k. On the concept of time average, sample mean is not computable with one dimensional sample. The sample defined by the concept of time average consists of the states and their occupying times. Accordingly, the sample becomes two dimensional. Denoting S_i be the state of i^{th} observation and T_i be the occupying time of S_i, the time averaged

sample is defined by two dimensional data $\{(S_i, T_i),\ i = 1, \ldots, n\}$. This notation produces the sample mean as follows.

$$\bar{S} = \frac{\sum_{i=1}^{n} S_i T_i}{\sum_{i=1}^{n} T_i} \tag{4}$$

2.2 Confidence Interval of Time Averaged Estimator

If the arrival of a queueing system follows Poisson process and total operation time is sufficient to reach the steady state, both the ensemble average and the time average become identical in probability. In addition, either the ensemble average or the time average produces the identical mean if the sample is generated by a renewal or regenerative process even though the system does not reach the steady state [3], [12]. In this case, we simply construct the confidence interval using the ensemble average concept based on the CLT. In real system simulation, however, most of the arrival processes may not be Poisson or regenerative. The outputs from the real system simulation may have autocorrelation structure. In this situation, the sample mean defined by equation (4) may not satisfy the condition of CLT. An appropriate method to estimate the confidence interval is generating the percentile interval through the independent replications. To use the independent replication method, however, long time or huge cost is required to get the samples from steady state in the simulation runs.

On the other hand, in the simulation output analysis, the bootstrap method is developed for reducing the time and cost [1-2]. This method only uses a sample which is generated by the single simulation run. Using the bootstrap, the percentile confidence interval can be achieved by nonparametric method. The traditional IID bootstrap method, however, does not describe the autocorrelation structure in the original sample. To overcome this limitation, the modified methods such as MBB, SB, and TB are introduced [4-11]. In this paper, we introduce a bootstrap algorithm that applies the TB showing superior performance compared to the other methods [7].

3 Generating Confidence Interval of Time Averaged Estimator

TB is applicable to the only one dimensional sample but the time averaged sample is defined by two dimensional data as described in equation (4). In this section, we introduce an algorithm to apply the TB to the two dimensional sample. We also provide the percentile method to generate the confidence interval.

3.1 Typical Threshold Bootstrap

In general, TB is performed as follows [4], [7-8]. Let $\{X_1, \ldots, X_n\}$ be a random sample of size *n* that is a sequence of autocorrelated data. The next step is selecting *threshold* value such as sample mean and dividing the series by *runs* that are either above or below the threshold (high and low runs). Two successive runs compose a *cycle* and a fixed number of chained cycles generate a *chunk*. Fig. 1 illustrates a sample of threshold, runs, cycles, and chunks. Fig. 1 also shows the mechanism for

generating chunk as an example; two chained cycles compose a chunk. Note that the remaining runs or cycles should be treated as a chunk. We create a bootstrap replication by resampling the chunks at random with replacement. If their total length exceeds n, we just truncate the concatenated runs. Finally, we can get a statistic by repeating B times.

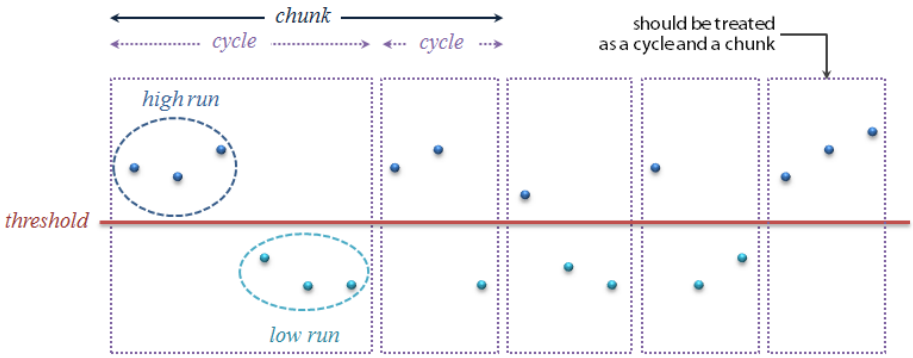


Fig. 1. Illustration of threshold, runs, cycles, and chunks in TB

3.2 Time Average and Confidence Interval

Finding correct confidence interval is difficult if the estimator is calculated by time average concept. The autocorrelated data make it worse to construct the confidence interval. Expanding the concept of TB to two dimensional data, we can perform resampling from the time averaged sample. Using vector notation where $\boldsymbol{X}_i = (S_i, T_i)$, let $\{\mathbf{X}_1, \dots, \boldsymbol{X}_n\}$ denote the two dimensional data sample. We generates the threshold, runs, cycles, and chunks with only an element of the vector; i.e., we use either S_i's or T_i's. We suggest that the elements should be a series having the autocorrelation structure. When we create a bootstrap replication, we do not resample the elements but the vectors. The rest of bootstrap procedure is identical to the typical TB.

The i^{th} repetition of TB generates a bootstrap sample $\{\boldsymbol{X}^*_{i1}, \dots, \boldsymbol{X}^*_{in}\}$ and the time averaged sample mean $\overline{\boldsymbol{X}^*_i}$. As the bootstrap replication is repeated B times as mentioned previously, a set of bootstrapped sample mean $\{\overline{\boldsymbol{X}^*_1}, \dots, \overline{\boldsymbol{X}^*_B}\}$ will be created. Using these bootstrapped sample means, we calculate the percentile confidence interval of random variable (or sample) $\boldsymbol{X}$ defined by time average. Denoting $\{\overline{\boldsymbol{X}^*_{(1)}}, \dots, \overline{\boldsymbol{X}^*_{(B)}}\}$ as order statistic of bootstrapped sample means, the percentile confidence interval with significant level α is defined as follows.

$$\boldsymbol{X} \in \left[\overline{\boldsymbol{X}^*}_{\left(\left\lfloor \frac{B\alpha}{2} \right\rfloor\right)}, \overline{\boldsymbol{X}^*}_{\left(\left\lfloor \frac{B(1-\alpha)}{2} \right\rfloor\right)}\right] \tag{5}$$

4 Numerical Examples

In this section, we run some simulation experiments and provide the percentile confidence intervals for time averaged estimators applying our proposed method. In addition, to confirm that our confidence interval is permissible, we perform the coverage tests using results of independent replications.

4.1 Design of Experiments

To generate the time averaged samples which have autocorrelation structure, we compose two intentional simulation models. A random process $N(t)$ changes the state as time elapsed as follows.

$N(t)$: state of system at time t

In convenience, if we assume that $N(t)$ has only discrete states, $N(t)$ generates a sample that has two dimensional data. Fig. 2 shows an example of two dimensional sample produced from the simulation of $N(t)$. In Fig. 2, $N(t)$ produces two dimensional vectorized data set $\{(S_i, T_i), i = 1, 2, \ldots n\}$. S_i implies a state at time interval $[t_i, t_{i+1})$ and T_i does its occupying time $(t_{i+1} - t_i)$. For example, we observe that $N(t)$ has the state S_1 for time interval $[t_1, t_2)$.

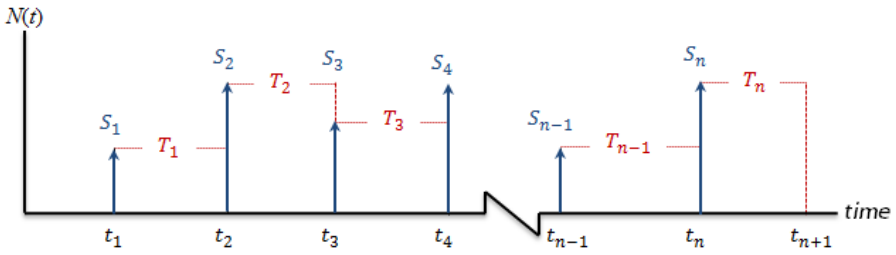


Fig. 2. Illustration of two dimensional sample produced by simulating $N(t)$

We reversely simulate S_i and T_i to produce a sample of $N(t)$ which has the autocorrelation structure. We construct two cases of autocorrelation for each T_i and S_i. The case of autocorrelated T_i is modified by the first order autoregressive (AR) model as follows.

$$Z_t = \varphi Z_{t-1} + \varepsilon_t \,,\ \varepsilon_t \sim N(0, 1)$$
$$T_i = | Z_t + \mu |$$

We assume that ε_t's are IID and S_i's are also IID by geometric with the success probability p. We simulate the geometric random variates as the number of trials until the first failure occurs. In the simulation run, we set the value of φ as 0.9, μ as 2, and p as 0.5. For the case of S_i, we also modify the first order AR model as follows.

$Z_t = \varphi Z_{t-1} + \varepsilon_t$, $\varepsilon_t \sim N(0, 1)$
$S_i = \lfloor \nu Z_t + \mu \rfloor$

In this model, T_i's are IID by exponential with the rate parameter λ and we set the value of φ as 0.9, ν and μ as 2, and λ as 1. For a simulated sample of size n, $\{(S_1, T_1), \ldots, (S_n, T_n)\}$, the sample mean is defined by equation (4). Accordingly, the CLT is not applicable to construct the confidence interval because random variates S_i's or T_i's are autocorrelated.

In our experiments, we simulate samples of size n = 10000 for each model. For these samples, we repeat the bootstrap replications B = 300 times. To determine the optimal chunk size, we use the method that minimizes the mean squared error (MSE). In addition, using 100 independent simulation outputs, we estimate the percentile confidence intervals and compare it to the results of our proposed method. Finally, we perform the coverage test for the confidence intervals produced by our method.

4.2 Experiment Result and Analysis

Table 1 summarizes the results of our experiments. In Table 1, C.I. means the percentile confidence interval produced by independent replication (IR) and our proposed method. While the theoretical population mean of the first case can be found, that of the second case is not simple to find. In the second case, therefore, we use the grand mean of 100 IR means as the population mean. As shown in Table 1, the C.I.'s of both methods include the true values and we confirm that they have similar half widths for each model. Note that the IR performs 100 simulation runs but the TB uses only one sample from single run.

Table 1. Comparison of confidence intervals produced by IR and TB

Model	True value	IR	TB	
		90% C.I.	90% C.I.	Coverage (%)
Autocorrelated T_i	2.000	1.998±0.029	2.016±0.026	88±6.4
Autocorrelated S_i	1.863	1.863±0.321	1.898±0.306	90±5.9

At next, we perform coverage test in order to verify the confidence intervals inferred by our proposed method, TB. To test the coverage of each C.I., we perform the bootstrap repetition for all samples from IR. Using those results, we calculated binomial confidence interval defined as follows.

$$\left(\hat{p} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right) \tag{6}$$

In equation (6), $\hat{p}$ is a success probability of Bernoulli trial that is defined by the number of containing the true value for 100 independent replications. The rightmost column in Table 1 explains that the C.I.'s from our proposed method is permissible. In other words, about 90% of confidence intervals include the true values.

5 Conclusion

At first, in this paper, an application of bootstrap method was introduced to estimate the confidence interval of time averaged statistic. And then, described the modified algorithm that resamples the two dimensional data which generate the time average. After that, the efficiency and performance of our proposed method was verified through experiments. We created simulation samples of size 10000 and found the percentile confidence intervals with 300 repetitions of bootstrap replications. Finally, the coverage of the estimated confidence intervals was tested. These results imply that the bootstrap methods are useful for generating the confidence intervals of time averaged estimators.

We performed experiments for the case that the only either the state or occupying time has the autocorrelation structure. If both of the state and its occupying time are autocorrelated, our method is not efficient sometimes. Moreover, there might be correlation between the state and its occupying time. In this situation, we need another efficient method to resample the two dimensional data. Our ongoing study focuses on this topic.

References

1. Efron, B.: Bootstrap Methods: Another Look at the jackknife. The Annals of Statistics 7, 1–26 (1979)
2. Efron, B., Tibshirani, R.: Bootstrap Methods for Standard Errors, Confidence Intervals and Other Measures of Statistical Accuracy. Statistical Science 1, 54–75 (1986)
3. Melamed, B., Whitt, W.: On Arrivals that See Time Averages: A martingale Approach. Journal of Applied Probability 27, 376–384 (1990)
4. Kim, Y.B., Whillemain, T.R., Haddock, J., Runger, G.C.: The threshold bootstrap: a new approach to simulation output analysis. In: Proceedings of the 1993 Winter Simulation Conference, pp. 498–502 (1993)
5. Kùnsch, H.R.: The jackknife and the bootstrap for general stationary observations. Annals of Statistics 17, 1217–1241 (1989)
6. Liu, R., Singh, K.: Moving blacks jackknife and bootstrap capture weak dependence. In: Lepage, R., Billard, L. (eds.) Exploring the Limits of Bootstrap, pp. 225–248. Willey, New York (1992)
7. Park Dae, S., Kim, Y.B., Shin, K.I., Willemain, T.R.: Simulation output analysis using the threshold bootstrap. European Journal of Operational Research 134, 17–28 (2001)
8. Daesu, P., Willemain, T.R.: The threshold bootstrap and threshold jackknife. Computational Statistics & Data Analysis 31, 187–202 (1999)
9. Hall, P., Horowitz, J.L., Jing, B.-Y.: On blocking rules for the bootstrap with dependent data. Biometrika 82(3), 561–574 (1995)
10. Hall, P., Jing, B.: On Sample Reuse Methods for Dependent Data. Journal of the Royal Statistical Society 58(4), 727–737 (1996)
11. Politis, D.N., Romano, J.P.: The Stationary Bootstrap. Journal of American Statistics Association 89, 1303–1313 (1994)
12. Wolff, R.W.: Poisson Arrivals See Time Averages. Operations Research 30, 223–231 (1982)

Reliability Analysis Method Using Dynamic Reliability Block Diagram Based on DEVS Formalism

Sol Ha[1,*] Namkug Ku[1], Myung-Il Roh[2], and Ju-Hwan Cha[3]

[1] Engineering Research Institute,
Seoul National University, Seoul, South Korea
{hasol81,knk80}@snu.ac.kr
[2] Department of Naval Architecture and Ocean Engineering,
Seoul National University, Seoul, South Korea
miroh@snu.ac.kr
[3] Department of Ocean Engineering, Mokpo National University,
Muan-gun, Jeollanam-do, South Korea
jhcha@mokpo.ac.kr

Abstract. This paper adopted the system configuration to assess the reliability of a target system instead of making a fault tree (FT), which is a traditional method to analyze reliability of a certain system; this is the reliability block diagram (RBD) method. The RBD method is a graphical presentation of a system diagram connecting the subsystems of components according to their functions or reliability relationships. The equipment model for the reliability simulation is modeled based on the discrete event system specification (DEVS) formalism.

Keywords: Reliability analysis, dynamic reliability block, diagram, DEVS(Discrete Event System Specification).

1 Introduction

A system is a collection of components, subsystems and assemblies that are arranged according to a specific design to achieve acceptable performance and reliability levels (Distefano and Puliafito, 2009). The main goal of system reliability/availability evaluation is the construction of a model (life distribution) that represents the time to the failure of the entire system based on the life distributions and maintenance policies for the components, subassemblies and/or assemblies (black boxes) with which the system was composed.

There are several ways to represent and analyze system reliability. A fault tree (FT) is a compact graphical method of analyzing system reliability (Vesely et al., 1981). FTs use Boolean gates to show how unit failures combine to produce system failure. They do not have any elements or capabilities to model the reliability interactions among the components or subsystems — aspects that are conventionally qualified as dynamic. Common examples of dynamic reliability behavior are standby redundancy, interferences, dependencies and common-cause failures. These arguments awakened

* Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 219–230, 2013.

the scientific community to the need for new formalisms such as dynamic fault trees (DFTs) (Boyd, 1991). Markov models are widely used to calculate system reliability with DFTs (Stamatelatos, 2002). Markov models do not directly use the structure of the DFTs, however, and they need too many states when the DFTs are more complex. Because of these disadvantages of Markov models, Bayesian networks are used to calculate system reliability (Bobbio et al., 2001). Bayesian networks can use the structure of DFTs without any changes, so they are more compact and fast than Markov models.

All the aforementioned system reliability analysis methods are powerful though not always user-friendly, because it is sometimes very hard to obtain a model directly from the specifications of the system, especially of complex systems. To make the DFTs of the system, an expert's experience and high-cost complex systems are needed. In addition, the results of the analysis differ according to the selection of the logical gate, so they depend mostly on the expert who makes the DFTs.

This fact prompted the definition of specific reliability modeling formalisms as RBDs. An RBD is a graphical presentation of a system diagram that connects subsystems of components according to their function or reliability relationships. It is user-friendly and makes it easy to obtain a model directly from the specifications. Furthermore, from its concept, dynamic RBDs (DRBDs) were proposed in this thesis based on an extension of the existing RBD formalism (Distefano and Puliafito, 2007). The main advantage of a DRBD is its capability to model dependencies among subsystems or components concerning their reliability interactions. A DRBD can be quantitatively analyzed using existing methods like Markov chains and Monte Carlo simulation.

The causes of system failure are trigger events, so the DEVS formalism, which can represent a model for event-based simulation, was used in this thesis. All the used reliability simulation models were based on the DEVS formalism. The reliability of the system was calculated using Monte Carlo simulation.

2 Reliability Analysis Method

2.1 Fault Tree Analysis

There are several approaches to representing and analyzing system reliability. In tradition, fault tree (FT) analysis is a compact graphical method for analyzing system reliability (Vesely et al., 1981). Fault tree analysis (FTA) can be simply described as an analytical technique whereby an undesired state of the system is specified (usually a state that is critical from a safety or reliability standpoint), and the system is then analyzed in the context of its environment and operation to find all the realistic ways in which the undesired event (top event) can occur. The FT itself is a graphical model of the various parallel and sequential combinations of faults that will result in the occurrence of the predefined undesired event. The faults can be events associated with component hardware failures, human errors, software errors, or any other pertinent events that can lead to the undesired event. A FT thus depicts the logical interrelationships of basic events that lead to the undesired event, the top event of the FT.

Intrinsic to a FT is the concept that an outcome is a binary event (i.e., to either success or failure). A FT is composed of a number of entities known as "gates" that serve to permit or inhibit the passage of fault logic up the tree. The gates show the relationships of the events needed for the occurrence of a "higher" event. The "higher" event is the output of the gate; the "lower" events are the "inputs" to the gate. The gate symbol denotes the type of relationship of the input events required for the output event.

Fig. 1 shows a simple FT for a simple serial system with two valves. The system in Fig. 1 has two valves connected serially. If a certain valve is broken for some reason, the flow cannot follow the cycle. Thus, the system failure can be represented by using the "OR" gate, as shown in Fig. 1.

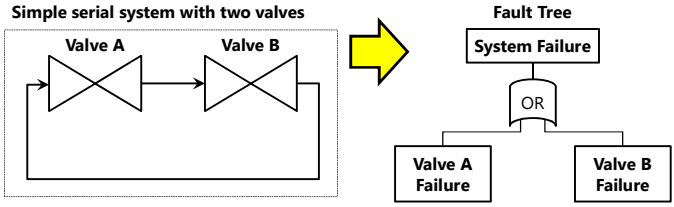


Fig. 1. Simple FT for a simple serial system with two valves

As shown in Fig. 1, the system failure is determined from the combination of the failure events of the equipment in the system. The combination of the failure events is based on the logical operator, such as "and" or "or," and this is called gate. There are two basic types of FT gates: the OR gate and the AND gate. All the other gates in the FT (normally the static FT) are special cases of these two basic types. As mentioned in the previous paragraph, the gates in the FT are based on the OR gate and the AND gate. Thus, the failure probability of the system can be determined by logical operations.

FTs do not provide any element or capability to model the reliability interactions among components or subsystems, aspects that are conventionally qualified as dynamic. Common examples of dynamic reliability behaviors are standby redundancy, interferences, dependencies, and common-cause failures. These arguments awakened the scientific community to the need for new formalisms, such as dynamic fault trees (DFTs) (Boyd, 1991).

Fig. 2(a) shows a simple redundant system with two valves and a switch. In this system, the order of the equipment failure affects the system failure. Suppose that the failure mode of the switch is such that when it fails, it is unable to switch between two valves. The failure of the switch only matters if switching from valve A to valve B is of interest. As shown in Fig. 2(a), if the switch fails after valve A fails (and thus, valve B is already in use), then the system can continue to operate.

If the switch fails before valve A fails, however, as shown in Fig. 2(b) then valve B cannot be switched into active operation, and the system fails when valve A fails. The order in which the primary and switch fail determines if the system will continue to operate.

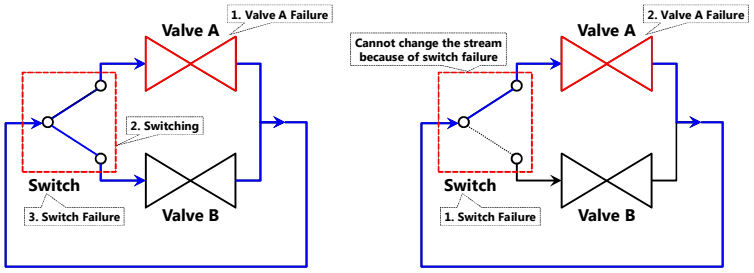


Fig. 2. (a) simple redundant system with two valves and a switch: switch failure after valve A failure, (b) simple redundant system with two valves and a switch: valve A failure after switch failure

This example shows that the order of the equipment failure influences the failure of the whole system. As in the static FT, there is no gate to express these dynamic reliability behaviors, gates are added to consider the dynamic behaviors, such as standby redundancy, interferences, dependencies, and common-cause failures.

Fig. 3 shows the DFT for the simple redundant system in Fig. 2, where the PAND gate captures the sequence-dependent failure of valve A and the switch. The order of the equipment failure is considered by using the PAND gate on the "switch fails before valve A fails" node.

In the DFTs, the sequence of the failure events affects the failure of the whole system. Thus, the failure probability cannot be calculated by using a logical operation. In the DFTs, the Markov chain or the Bayesian network is used to calculate the failure probability of the system.

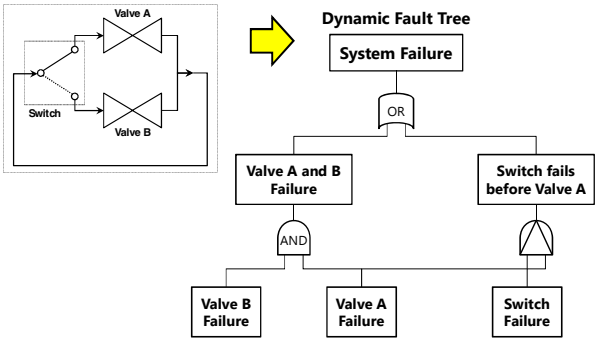


Fig. 3. DFT for the simple redundant system with two valves and a switch

Although many systems with sequence dependencies can be analyzed using traditional FTA, the use of DFT constructs offers an interesting alternative. The DFT, in addition to supporting an exact solution of the PAND gate, facilitates the analysis of other sequence dependencies.

There are several methods of calculating the failure probability using FTs. In the static FT, the simplest method for the calculation involves the use of binary decision diagrams (BDDs), which is based on logical operations. The failure probability of the

DFT, however, cannot be calculated by this operation because of the sequence dependencies. The DFTs are solved by conversion to the equivalent Markov chain. Monte Carlo simulation can also be used to evaluate the DFTs without conversion to a Markov chain. In an attempt to address the limitations of the solutions using the Markov chain, a solution method using conversion to the equivalent Bayesian networks was recently proposed.

2.2 Reliability Block Diagram

In the previous section, three methods of calculating the system failure probability were introduced: BDDs, the Markov chain, and the Bayesian network. BDD can be used only for the static FT, but the Markov chain can calculate the failure probability of the DFT. As the Markov chain has the state space explosion problem as the number of basic events increases, the Bayesian network is proposed for probability calculation.

All these methods need FTs for calculating the failure probability of the system. An expert is needed to make the FT of a certain system. It also entails a high cost and has the risk of producing different results depending on the selection of the logical gate of the FT, which is dependent on the experience of the expert.

Thus, the key idea in this thesis came from the use of the system configuration to assess the reliability instead of making a FT; this is the reliability block diagram (RBD) method. The RBD method is a graphical presentation of a system diagram connecting the subsystems of components according to their functions or reliability relationships. It is a user-friendly method, and it is easy to obtain a model directly from the specifications. From this concept, a new dynamic modeling method involving the use of dynamic reliability block diagrams (DRBDs) was proposed based on an extension of the existing RBD formalism.

The main advantage of DRBD is the capability to model dependencies among subsystems or components concerning their reliability interactions. A quantitative analysis of DRBD can be conducted using the existing methods, such as the Markov chain and Monte Carlo simulation. In this paper, Monte Carlo simulation is used to calculate the failure probability.

3 Reliability Analysis Based on DEVS Formalism

In this paper, the RBD method is used to assess the reliability of the target system. All conceptual block model in the RBD is configured to determine the failure of the equipment, and based on the DEVS formalism.

3.1 Procedure of Reliability Analysis

Fig. 4 shows an example of the conversion of the configuration of the target system to the equivalent RBD to assess the reliability of a refrigerator (simple liquefaction process system). As shown in the right side of Fig. 4, the equipment and interconnections of the equipment are converted to the RBD as they are.

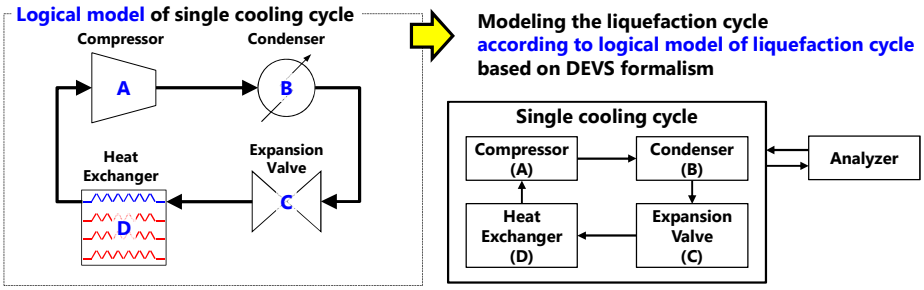


Fig. 4. Conversion of the system configuration to the equivalent RBD to assess the reliability of a refrigerator

Fig. 5 shows the procedure of the probability calculation using the RBD. Suppose that the operating time of the liquefaction cycle is 20 hours and that the failure rate of the equipment is given as shown in Fig. 5.

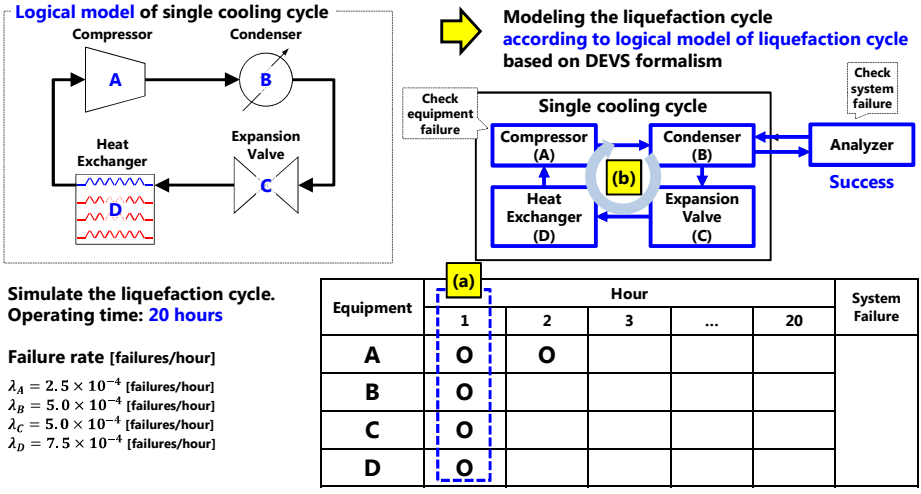
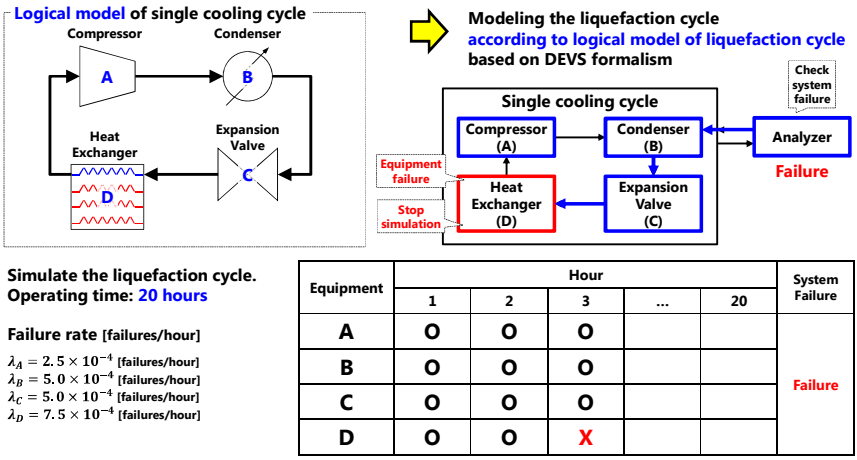


Fig. 5. Procedure of the probability calculation using the RBD

First, check each equipment for equipment failure. To do this, a random value between 0 and 1 is generated for each equipment. If the generated value is less than the failure rate of the equipment, there is equipment failure. All the equipment in the liquefaction cycle are to be checked for equipment failure, as shown in Fig. 5(a). Thereafter, the analyzer model checks for system failure. The analysis model sends a signal to the connected model, a condenser in Fig. 5, and this signal follows the system. If the signal returns to the analyzer model, the system is successfully operating during the simulation.

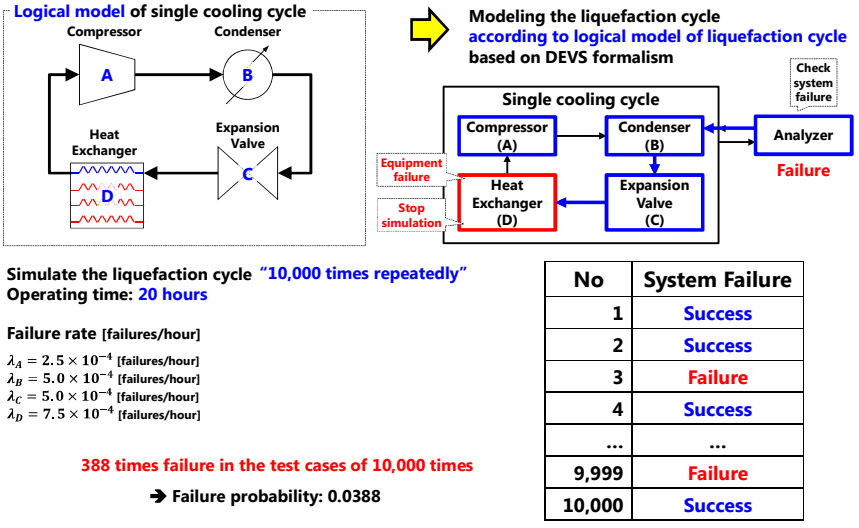


Equipment	Hour					System Failure
	1	2	3	...	20	
A	O	O	O			Failure
B	O	O	O			
C	O	O	O			
D	O	O	X			

Fig. 6. Procedure of the probability calculation using the RBD

Fig. 6 shows an example of system failure during probability calculation. As mentioned in the previous paragraph, equipment failure is first determined. In this example, the heat exchanger fails and is marked as such. The analyzer model sends a signal to check for system failure, and the signal will be stopped at the heat exchanger. Thus, the analyzer determines that the system has failed.

This operation is repeated until the operating time. Fig. 7 shows the calculation of the failure probability by performing the simulation repeatedly. The failure probability is simply calculated based on the ratio of the number of system failures to the total number of simulations.



No	System Failure
1	Success
2	Success
3	Failure
4	Success
...	...
9,999	Failure
10,000	Success

Fig. 7. Failure probability calculation via Monte Carlo simulation, using the RBD

3.2 Models for Reliability Analysis Based on DEVS Formalism

The models for the reliability analysis are categorized into four groups: equipment, repair and maintenance, collector, and analyzer. This section gives a detailed description of these models used on the reliability analysis.

All equipment are modeled with a single inlet and a single outlet. As shown in Fig. 8(a), the model for equipment with a single inlet and a single outlet is configured based on the DEVS formalism. It is characterized by variable states: THROW, WAITING, ANALYZE, and FAIL. The THROW state is used to determine equipment failure.

The model at the THROW state throws a dice to generate a random value between 0 and 1 every unit time, and checks for equipment failure according to the generated value and the failure rate. If the equipment is operating after a dice is thrown at the THROW state, the state will be changed to the WAITING state. In the WAITING state, the equipment waits until the signal from the previous equipment is sent. When the equipment receives the signal from the previous equipment, the current state will be changed to the ANALYZE state. In the ANALYZE state, the physical model confirms the equipment failure. If the equipment is operating, the current state will be changed to the THROW state. When the state is changed from ANALYZE to THROW, the signal will be sent to the outlet of this model. On the other hand, if the equipment fails, the current state will be changed to the FAIL state, and the equipment will stay in such state until the simulation is stopped. Certain equipment can have multiple inlets and outlets. These are expressed by using a collector model, and it is configured based on the DEVS formalism.

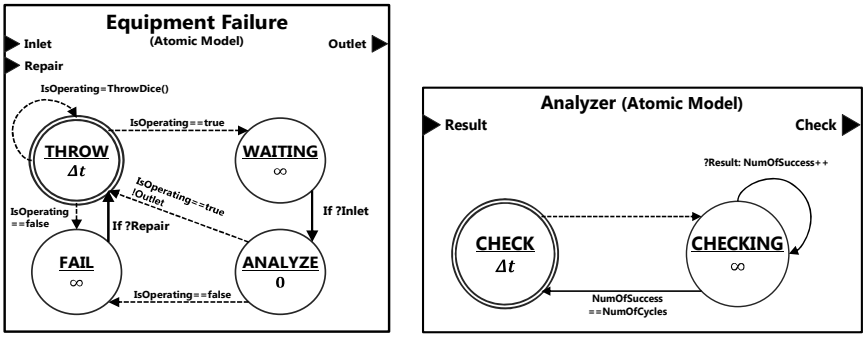


Fig. 8. (a) Equipment Failure model based on the DEVS formalism: check equipment failure or not, (b) Analyzer model based on the DEVS formalism: checking for failure of the whole system

To check for the failure of the whole system for every unit time, the analyzer model is configured based on the DEVS formalism, as shown in Fig. 8(b). The analyzer model has two states: CHECK and CHECKING. At the end of each unit time, the analyzer model changes its state from CHECK to CHECKING. At the same time, a signal to check for system failure is sent to the "check" port. At the CHECKING state, the analyzer model waits for the success signal from the last equipment of the

liquefaction cycle. If the success signal is received, the state will be changed to the CHECK state. If not, the simulation will be stopped by the execution rules of the DEVS formalism. If the liquefaction cycle has multiple cooling cycles, the model in the CHECKING state will wait until all the signals from each cycle arrive.

4 Verification of the Reliability Analysis Method Proposed in This Paper

To verify the reliability analysis method proposed in this thesis, the following three examples are simulated to assess the reliability of each system:

- firewater pumping system and
- power switching system.

4.1 Firewater Pumping System

Fig. 9 shows the configuration of a firewater pumping system. The system has two fire pumps, one engine, and one valve. The engine is necessary to run the fire pump. If the valve or the engine fails, the system will fail. The failure of one fire pump is allowed because the other fire pump will still be running.

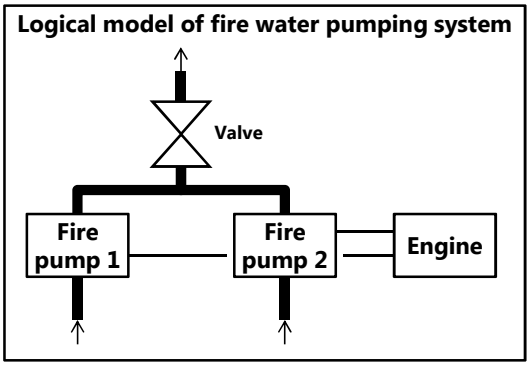


Fig. 9. Firewater pumping system

The FT of the firewater pumping system is configured as shown in Fig. 10(a). The failure of the valve directly leads to system failure. The failure of the engine contributes to system failure by combining the fire pumps.

Fig. 10(b) shows the RBD of the firewater pumping system. All the equipment in this diagram are defined based on the DEVS formalism. As can be seen, the configuration of the RBD is the same as that of the configuration of the system, except for the analyzer model.

Table 1 compares the reliability analysis results for the firewater pumping system. The operating time of the system is assumed to be 1,000 hours, and the simulation is performed 50,000 times. The probability of system failure is similar to the result using the FT. The execution time using the RBD based on the DEVS formalism was shown to be the longest among the three methods.

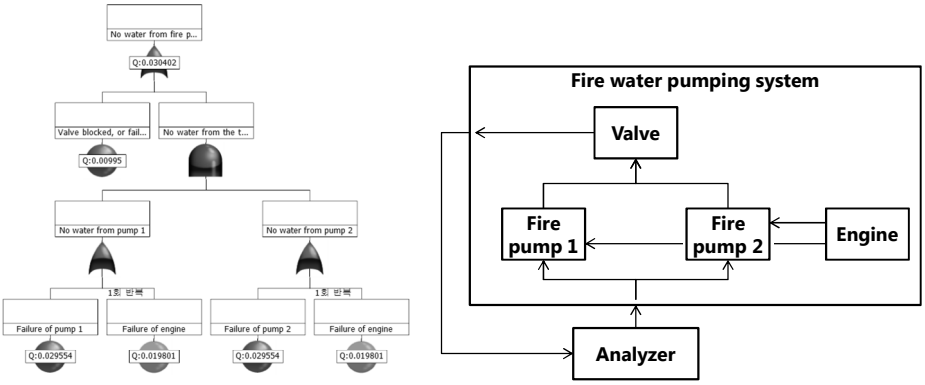


Fig. 10. (a) FT of the firewater pumping system, (b) RBD of the firewater pumping system based on the DEVS formalism

Table 1. Comparison of reliability analysis results for the firewater pumping system

Reliability analysis method	Probability of system failure	Execution time [s]
FTA and Markov chain	3.040%	0.49 [sec]
FTA and the Bayesian network	3.025%	0.23 [sec]
RBD based on DEVS	3.016%	4 [sec]

4.2 Power Switching System

Fig. 12 shows the configuration of a power switching system. The system has two powers, two relays with one controller, and a motor. Power 1 is the primary system, and power 2 is the standby system. Each power is dependent on each relay, and all the relays are dependent on the controller. At least one power is necessary to run the motor.

The FT of the power switching system is configured as shown in Fig. 13(a). If both powers fail, the system will fail. As the relays are dependent on the controller, the failure sequence of the relays and the controller leads to system failure. This is expressed by the use of the PAND gate in the FT.

Fig. 13(b) shows the RBD of the power switching system. All the equipment in this diagram are defined based on the DEVS formalism. As can be seen, the configuration of the RBD is the same as that of the configuration of the system, except for the analyzer model.

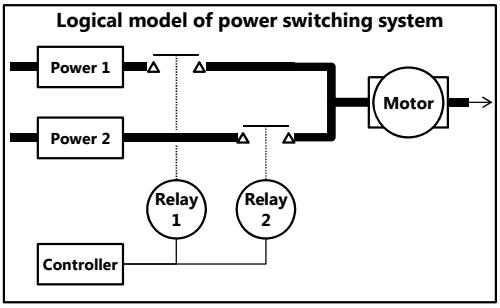


Fig. 11. Power switching system

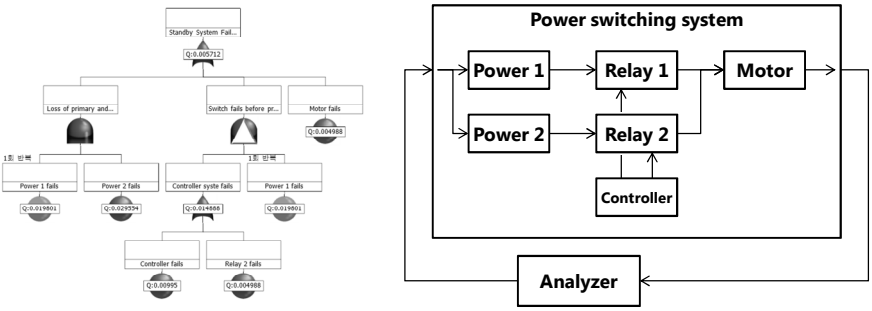


Fig. 12. (a) FT of the power switching system, (b) RBD of the power switching system based on the DEVS formalism

Table 2 compares the reliability analysis results for the power switching system. The operating time of the system is assumed to be 1,000 hours, and the simulation is performed 50,000 times. The probability of system failure is similar to the result using the FT. The execution time using the RBD based on the DEVS formalism is better than that using the Markov chain because the number of equipment increases, resulting in the state space explosion of the Markov chain.

Table 2. Comparison of reliability analysis results for the power switching system

Reliability analysis method	Probability of system failure	Execution time [s]
FTA and Markov chain	0.571%	11.6 [sec]
FTA and the Bayesian network	0.568%	2.1 [sec]
RBD based on DEVS	0.562%	6.9 [sec]

5 Conclusion

The reliability block diagram (RBD) method, which is a graphical representation of a system diagram connecting the subsystems of the components according to their reliability relationships, is used in this paper. A quantitative analysis of RBD can be conducted using Monte Carlo simulation, and the equipment model for reliability simulation is modeled based on the DEVS formalism. Reliability simulations are performed for the certain systems, such as the firewater pumping system and power switching system.

Acknowledgement. This study was partially supported by the Industrial Strategic Technology Development Program (10035331, Simulation-based Manufacturing Technology for Ships and Offshore Plants) funded by the Ministry of Knowledge Economy (MKE) of the Republic of Korea, the Engineering Research Institute of Seoul National University, and the Research Institute of Marine Systems Engineering of Seoul National University.

References

1. Bang, K.W.: Combined discrete event and discrete time simulation framework for ship-building process planning. Master Thesis, Seoul National University (2006)
2. Bobbio, A., Portinale, L., Minichino, M., Ciancamerla, E.: Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. Reliability Engineering & System Safety 71(3), 249–260 (2001)
3. Boyd, M.A.: Dynamic Fault Tree Models: Techniques for Analysis of Advanced Fault-Tolerant Computer Systems. Doctoral Thesis, Dept. of Computer Science, Duke Univ. (1991)
4. Cha, J.H., Lee, K.Y., Ham, S.H., Roh, M.I., Park, K.P., Suh, H.W.: Discrete event/discrete time simulation of block erection by a floating crane based on multibody system dynamics. In: Proceedings of 19th International Offshore and Polar Engineering Conference (ISOPE), pp. 678–685. Osaka International Convention Center, Osaka (2009)
5. Chang, K.P., Rausand, M., Vatn, J.: Reliability assessment of reliquefaction systems on LNG carriers. Reliability Engineering & System Safety 93(9), 1345–1353 (2008)
6. Distefano, S., Puliafito, A.: Dependability evaluation with dynamic reliability block diagrams and dynamic fault trees. IEEE Transactions on Dependable and Secure Computing 6(1), 4–17 (2009)
7. Stamatelatos, M.: Fault Tree Handbook with Aerospace Applications Version 1.1 (2002)
8. Vesely, W.E., Goldberg, F.F., Roberts, N.H., Haasl, D.F.: Fault tree handbook, NUREG-0492, US Nuclear Regulatory Commission (1981)
9. Zeigler, B.P.: Object-oriented simulation with hierarchical, modular models. Academic Press, Inc. (1990)
10. Zeigler, B.P., Praehofer, H., Kim, T.G.: Theory of modeling and simulation. Academic Press, New York (2000)

A Yaw Rate Tracking Control of Active Front Steering System Using Composite Nonlinear Feedback

M. Khairi Aripin[1,*], Y.M. Sam[2], A.D. Kumeresan[2], Kemao Peng[3], Mohd Hanif Che Hasan[4], and Muhamad Fahezal Ismail[5]

[1] Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
khairiaripin@utem.edu.my
[2] Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Johor, Malaysia
{yahaya,kumeresan}@fke.utm.my
[3] Temasek Laboratory, National University of Singapore, Singapore
kmpeng@nus.edu.sg
[4] Faculty of Engineering Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
Hanif.hasan@utem.edu.my
[5] Industrial Automation Section, Universiti Kuala Lumpur Malaysia France Institute, Selangor, Malaysia
fahezal@mfi.unikl.edu.my

Abstract. In this paper, the composite nonlinear feedback (CNF) technique is applied for yaw tracking control of active front steering system with the objectives to improve the transient performance of yaw rate response. For lateral and yaw dynamics analysis, nonlinear and linear vehicle models are utilized as actual vehicle plant and for controller design respectively. The designed controller is evaluated using J-turn cornering manoeuvre condition in computer simulation. The simulation results demonstrate that the application of CNF for yaw rate tracking control improves the yaw stability and vehicle handling performances.

Keywords: composite nonlinear feedback, yaw rate control, active front steering, vehicle yaw stability.

1 Introduction

Yaw stability control system is one of the approaches utilized for lateral dynamics motion control of road vehicles. The vehicle yaw rate is an important variable to be controlled to ensure vehicle lateral stability. Thus, the main objective of yaw stability control is to ensure the proposed controller is able to track a desired yaw rate i.e. the actual response of yaw rate is close to a desired response that is generated by reference model. The yaw rate tracking control can be realized by implementing an active front steering (AFS) for vehicle handling improvement especially during low to mid-range of lateral acceleration. In AFS system, the front wheel steer angle is a sum of steer angle commanded by the driver and corrective steer angle that is generated by the proposed controller.

* Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 231–242, 2013.

Various control strategies have been developed and implemented for vehicle yaw stability control based on AFS [1-8]. The yaw stability controller based on sliding mode and back stepping algorithm is designed to ensure the vehicle yaw rate follow its reference [1]. As implemented in [2], the model predictive control (MPC) technique is adapted for yaw stability control based on active differential braking to improve the vehicle stability during critical manoeuvre while in [3], the mixed-sensitivity minimization technique is applied as feedback control loop to track the desired yaw rate as close as possible. To cater the uncertainties of vehicle parameters, the robust yaw control is designed based on second order sliding mode control (SOSM), internal mode control (IMC) and these control performances was compared each other as discussed in [4-6]. As reported in [7], the yaw rate control of active front steering is designed based on fuzzy logic control to improve the vehicle stability. Similarly, the yaw rate controller is developed based on fuzzy logic control in order to track the desired yaw rate as discussed in [8]. In point of view of tracking control, the transient response performance is vital. However, based on above review, the improvement of transient response performance in yaw rate tracking control is not well emphasized and an appropriate control technique should be proposed for this purpose.

The composite nonlinear feedback (CNF) control is one of nonlinear control technique that has been developed in last decade based on state feedback law. This technique was introduced in [9] for tracking control of 2^{nd} order linear system had been improved for higher order MIMO linear system in [10]. It was further explored and extended for linear system with actuator nonlinearities in [11], general multivariable system with input saturation in [12], hard disk drive servo system and servo positioning system with disturbance in [13-16]. Recently, the CNF have been applied in vehicle dynamics control particularly for active suspension system in order to improve suspension deflection, velocity of car body, tire deflection, velocity of car wheel and body acceleration [17]. In principle, the use of CNF control could improve the performance of transient response based on variable damping ratio concept. The CNF control keep low damping ratio during transient and varied to high damping ratio as the output response close to the reference set point. To realize this concept, the CNF control that consists of linear and nonlinear feedback control law is designed in three important steps which will be discussed later. Therefore, based on previous studies and above discussion, an advantages of CNF control technique especially for improving transient response of tracking control is not yet been examined for vehicle yaw rate tracking control and should be further explored.

In this paper, the CNF control is introduced for vehicle yaw rate tracking control of AFS system. The lateral and yaw dynamics of nonlinear two track model is used as vehicle plant and linearized single track model is utilized for the controller analysis and design. To evaluate the performance of propose controller, the vehicle handling test of cornering manoeuvre are performed in computer simulation.

This paper is organized in 5 sections as the following: Section 1 provides an overview of yaw stability control, its existing control strategies and a review of the CNF

control technique. In Section 2, the dynamics of a nonlinear vehicle model and a linear single track model are discussed. The theory of CNF control and design procedures are explained in Section 3. The simulation results and discussion are presented in Section 4. Finally, a conclusion and future works is presented in Section 5.

2 Vehicle Dynamic Models

In this section, dynamic equations of a nonlinear two track vehicle model and a linear single track model are presented and discussed. These two models are constructed for vehicle plant and controller design respectively, whose performance is analyzed using a computer simulation tool.

2.1 Nonlinear Two Track Model

A nonlinear two track model as shown in Figure 1 is used as the actual vehicle plant for controller evaluation. The nonlinear dynamics for lateral and yaw motion are describe as in equations (1) and (2) respectively;

$$\dot{\beta} = \frac{1}{mv}\left(\cos\beta \sum F_y - \sin\beta \sum F_y\right) - r \tag{1}$$

$$\dot{r} = \frac{1}{I_z}\left[l_f (F_{y1}\cos\delta_f + F_{y2}\cos\delta_f + F_{x1}\sin\delta_f + F_{x2}\sin\delta_f) - l_r (F_{y3} + F_{y4}) + M_z\right] \tag{2}$$

where the sum of longitudinal forces $\sum F_y$, sum of lateral forces $\sum F_x$ and yaw moment M_z in the above equations are given as follows;

$$\sum F_y = \cos\delta_f (F_{x1} + F_{x2}) - \sin\delta_f (F_{y1} + F_{y2}) \tag{3}$$

$$\sum F_y = \sin\delta_f (F_{x1} + F_{x2}) + \cos\delta_f (F_{y1} + F_{y2}) \tag{4}$$

$$M_z = \frac{d}{2}\left(F_{x1}\cos\delta_f - F_{x2}\cos\delta_f - F_{y1}\sin\delta_f + F_{y2}\sin\delta_f + F_{x3} - F_{x4}\right) \tag{5}$$

The vehicle parameters involved in equations (1) - (5) above are vehicle speed v, vehicle mass m, vehicle width track d, distance of front axle to center of gravity (CG) l_f and distance of rear axle to CG l_r. The front wheel steer angle δ_f is the input to the system while the nonlinear longitudinal tire forces F_{xi} and lateral tire forces F_{yi} can be described using Pacejka tire model. Notice that the vehicle speed v is always assumed constant when no braking and accelerating are involved.

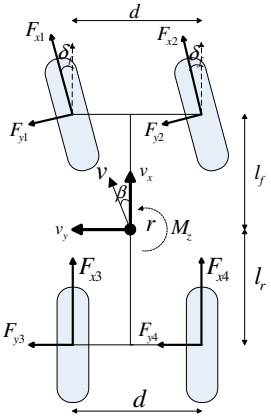


Fig. 1. Two track model

2.2 Tire Model

In the equations (1) - (5) above, the longitudinal tire force F_{xi} and lateral tire force F_{yi} may exhibit nonlinear characteristics. The pure longitudinal and lateral tire forces during pure side slip can be described using the Pacejka tire model as described in the following equations (6) and (7) respectively

$$F_{xi} = D_{xi} \sin\left[C_{xi} \tan^{-1}(B_{xi}.\lambda_i - E_{xi}(B_{xi}.\lambda_i - \tan^{-1}(B_{xi}.\lambda_i)))\right] \tag{6}$$

$$F_{yi} = D_{yi} \sin\left[C_{yi} \tan^{-1}(B_{yi}.\alpha_f - E_{yi}(B_{yi}.\alpha_f - \tan^{-1}(B_{yi}.\alpha_f)))\right] \tag{7}$$

where the parameters $D_{xi}, D_{yi}, C_{xi}, C_{yi}, B_{xi}, B_{yi}, E_{xi}$ and E_{yi} are known as tire model parameters that depending on tire characteristics, road surface and vehicle conditions while λ_i and α_i are longitudinal wheel slip and tire sides slip angle respectively that given by the following equations

$$\lambda_i = \frac{R\omega_i - V_i}{V_i} \tag{8}$$

$$\alpha_1 = \alpha_2 = \delta_f - \tan^{-1}\left(\beta + \frac{l_f.r}{v}\right) \tag{9}$$

$$\alpha_3 = \alpha_4 = \tan^{-1}\left(-\beta + \frac{l_r r}{v}\right) \tag{10}$$

where in equation (8), R is wheel radius, ω_i is wheel angular velocity and V_i is ground contact speed for each tire.

2.3 Linear Single Track Model

To design the controller for yaw rate tracking control of active front steering, a linear single track model as shown in Figure 2 is utilized. This model is linearized from the two track nonlinear vehicle model based on few main assumptions: tire force operates in linear region, very small of front steer angle δ_f and vehicle side slip β ,vehicle speed v is constant and two tires at front and rear axle are lumped into single tire at the centre line of vehicle [18].

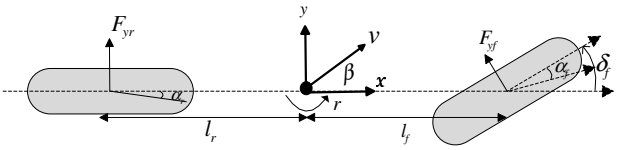


Fig. 2. Single track model

The dynamics equation for the lateral and yaw motions are described as follows

$$mv(\dot{\beta} + r) = F_{yf} + F_{yr} \tag{11}$$

$$I_z \dot{r} = l_f F_{yf} - l_r F_{yr} \tag{12}$$

As assumed above which tire forces are operates in linear region, front lateral tire force F_{yf} and rear lateral tire force F_{yr} are exhibit linear characteristics as described in the following equations

$$F_{yf} = C_f \alpha_f \tag{13}$$

$$F_{yr} = C_r \alpha_r \tag{14}$$

where C_f and C_r are front and rear tire cornering stiffness respectively. For linear tire forces, sideslip angle of front and rear tire are given in the equation (15) and (16) respectively as follows

$$\alpha_f = \delta_f - \beta - \frac{l_f r}{v} \tag{15}$$

$$\alpha_r = -\beta + \frac{l_r r}{v} \tag{16}$$

By re-arrange and simplify the equations (11) – (16), the differential equations of sideslip and yaw rate variable can be simplified as linear state space model as shown in equation (17). Notice that the parameters are same as discussed in section 2.1 above.

$$\dot{x} = Ax + Bu$$

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{-C_f - C_r}{mv} & -1 + \dfrac{C_r l_r - C_f l_f}{mv^2} \\ \dfrac{C_r l_r - C_f l_f}{I_z} & \dfrac{-C_f l_f^2 - C_r l_r^2}{I_z v} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} \dfrac{C_f}{mv} \\ \dfrac{C_f l_f}{I_z} \end{bmatrix} \delta_f \tag{17}$$

2.4 Yaw Rate Reference Model

The main objective in yaw rate tracking control of active front steering is to bring the actual response of vehicle yaw rate close to desired response. The desired yaw rate response is determined as a function of vehicle speed v and front wheel steer angle δ_f in steady state condition as follows

$$r_d = \frac{v}{l + k_u v^2} . \delta_f \tag{18}$$

where k_u is known as cornering stability factor and define as follows

$$k_u = \frac{m(l_r C_r - l_f C_f)}{(l_f + l_r) C_f C_r} \tag{19}$$

However, due to lateral acceleration of the vehicle in g unit could not exceed the maximum road friction coefficient μ, the steady state value of yaw rate must be limited as express in the following equation

$$\left| r_{ss} \right| \leq \frac{\mu g}{v} \tag{20}$$

3 Composite Nonlinear Feedback for Active Front Steering

In this section, the composite nonlinear feedback (CNF) control design procedures for yaw rate tracking control of active front steering (AFS) are presented. In general, The CNF control technique is applicable for systems with or without external disturbances. In this paper, the plant of vehicle dynamics is considered without an external disturbance and all states variable are assumed available for measurement. The following subsection will discuss the design procedures for CNF control which have been established in [11].

3.1 CNF Control Design

A linear time-invariant continuous system with input saturation is considered as follows

$$\begin{aligned} \dot{x} &= Ax + Bsat(u) \quad x(0) = x_o \\ y &= Cx \end{aligned} \tag{21}$$

where $x \in \Re^n$, $u \in \Re$, $y \in \Re$ are the state variable, control input and controlled output respectively. A, B and C are constant system matrices with appropriate dimensions and $sat : \Re \to \Re$ represent actuator saturation defined as follows

$$sat(u) = \text{sgn}(u) \min\left\{u_{\max}, |u|\right\} \tag{22}$$

with $u_{\max}$ is saturation level of the input. To design and apply the CNF control, the following assumptions of system matrices are considered;

1. (A,B) is controllable/stabilizable
2. (A,C) is observable/detectable
3. (A, B, C) is invertible and has no zero at s=0

To design the CNF for tracking control of step input without large overshoot and without adverse actuator saturations effects, the CNF control laws is design in three important steps. The details for step by step design process are discussed as follows

Step 1: design a linear feedback control law

$$u_L = Fx + Gr \tag{23}$$

where r is step input reference and F is feedback matric chosen such that $A + BF$ is an asymptotically stable matrix and closed loop system $C(sI - A - BF)^{-1}B$ has certain desired properties such as small damping ratio. G is a scalar which given as follows

$$G = -[C(A + BF)^{-1}B]^{-1} \tag{24}$$

The selection of matric F is not unique where it can be determined using linear control design techniques such as pole placement assignment.

Step 2: design a nonlinear feedback control law

$$u_N = \rho(r, y)B'P(x - x_e) \tag{25}$$

where $\rho(r, y)$ is nonpositive function locally Lipschitz in y that used to change the damping ratio of closed loop system as output approaches the step command input. $P > 0$ is a solution of the following Lyapunov equation

$$(A+BF)'P+P(A+BF)=-W \tag{26}$$

for some given $W>0$ and x_e is defined as

$$x_e := G_e r \tag{27}$$

$$G_e := -(A+BF)^{-1}BG \tag{28}$$

The nonlinear function $\rho(r,y)$ is not unique. To adapt the variation of tracking target, the nonlinear function $\rho(r,y)$ in [19,20] is utilized as follows

$$p(r,y)=-\gamma e^{-\varphi\varphi_o|y-r|} \tag{29}$$

where

$$\varphi_o=\begin{cases}\dfrac{1}{|y_o-r|}, & y_o\neq r\\ 1, & y_o=r\end{cases} \tag{30}$$

Step 3: complete CNF control law

Both control laws in step 1 and 2 are combined to form the complete CNF control law as follows

$$u=u_L+u_N=Fx+Gr+\rho(r,y)B'P(x-x_e) \tag{31}$$

3.2 Tuning Parameters of Nonlinear Function

The selection of design parameters of nonlinear function is essential for CNF control design so that the performance of closed loop system is improved as the controlled output approaches the reference set point. In this paper, the parameter γ and φ of nonlinear function are tuned according to the method as proposed in [20] as follows

1. choose the desired steady state damping ratio ξ_{ss}
2. determine γ by letting the steady state system has a damping ratio of ξ_{ss}
3. determine an optimal φ by solving minimization some appreciable criterions of integral of absolute error (IAE) and integral of time-multiplied absolute value of error (ITAE) which given by equations

$$\min_\varphi \int_0^\infty |e|dt \quad \text{or} \quad \min_\varphi \int_0^\infty t|e|dt \tag{32}$$

where $e=y-r$ is tracking error.

3.3 Active Front Steering Based on CNF Control

The AFS based on the CNF control technique for yaw rate tracking control is illustrated in Figure 3. The front wheel steer angle is a sum of steer angle commanded by the driver, δ_f and corrective steer angle, δ_c that generated by the CNF control. Based on vehicle parameters and linear single track model as discussed previously, the parameters of CNF control are obtained as follows

$$F = \begin{bmatrix} 0.5 & -0.05 \end{bmatrix},\ G = 0.2321,\ P = \begin{bmatrix} 0.8224 & 0.0562 \\ 0.0562 & 0.1535 \end{bmatrix},\ G_e = \begin{bmatrix} -0.1711 \\ 1 \end{bmatrix},\ \gamma = 0.2,$$

$$\varphi = 0.03$$

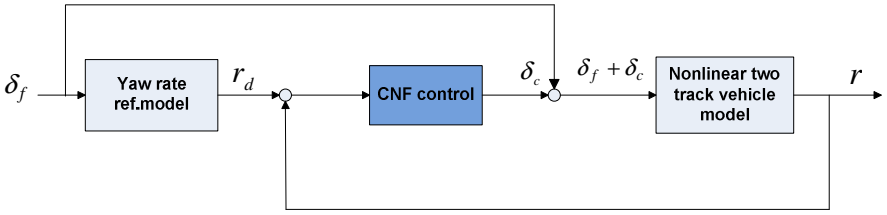


Fig. 3. Active front steering based CNF control

4 Simulation Results

The computer simulation of AFS based on the CNF control for yaw rate tracking control is conducted in Matlab/Simulink. The vehicle parameters used are taken from [21] as follows;

$$m = 1704.7kg,\ I_z = 3048.1kgm^2,\ l_f = 1.035m,\ l_r = 1.655m,\ C_f = 105{,}800N,$$

$$C_r = 79{,}000N,\ d = 1.54m$$

The road surface adhesion coefficient is assumed for dry road $\mu = 1$ and vehicle speed $v = 100$ km/h is assumed constant. To evaluate the performance of propose CNF control for AFS, a J-turn manoeuvre test is conducted in order to analyse the response of yaw rate tracking control. J-turn cornering manoeuvre which similar to a step input is a simple handling test to evaluate the transient and steady state behaviour of yaw stability control. The response obtained is compared to uncontrolled and classical PID controller. Figure 4 shows the input of 1° steer angle that generate $0.32g$ lateral acceleration which is considered an appropriate level for AFS. The response of yaw rate and yaw rate tracking error are shown in Figures 5 and 6 respectively.

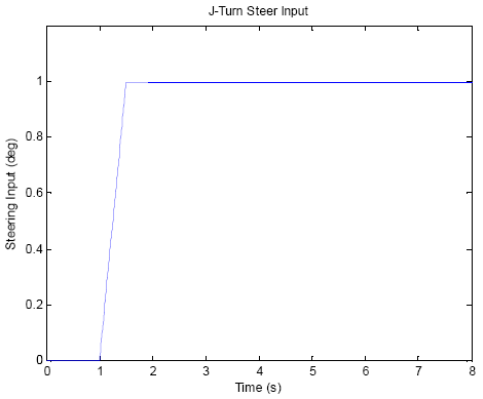


Fig. 4. J-turn steer input at 1° of steer angle

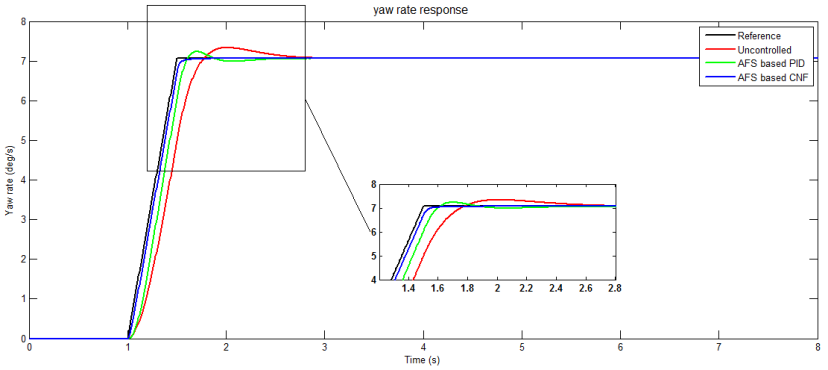


Fig. 5. Yaw rate response of J-turn manoeuvre at 1° steer angle

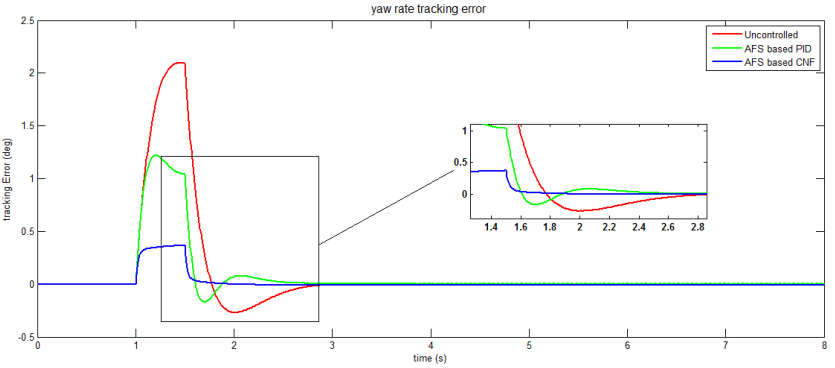


Fig. 6. Yaw rate tracking error of J-turn at 1° steer angle

Table 1. Transient response parameters of J-turn manoeuvre

controllers	amplitude	% OS	t_r (t)	t_s (t)
CNF	Nil	0	0.0524	0.107
PID	7.57	7.10	0.137	0.46
uncontrolled	7.39	4.53	0.299	1.03

From the Figures 5 and 6, it is observed that the CNF control could track the yaw rate reference with fast transient response compared to PID controller and uncontrolled vehicle. The performances of transient response parameters of this valuation are tabulated in Table 1.

The simulation of AFS based on the CNF control technique has been presented. From Table 1, it is observed that the yaw rate response with the CNF control technique have no maximum peak of amplitude i.e. 0% overshoot compared to PID controller and uncontrolled response that obtained 7.1% and 4.53% overshoot respectively. In term of rise time t_r, the CNF control performed better with 0.0524s which is obviously fastest than PID controller i.e. 0.137s and 0.299s of uncontrolled response. The CNF control also achieves fastest sampling time t_s i.e. 0.107s compared to 0.46s of PID controller and 1.03s of uncontrolled response.

5 Conclusion

A new technique for vehicle yaw rate tracking control of active front steering is proposed based on the composite nonlinear feedback (CNF) control scheme. From the results obtained, it is shown that the CNF control is capable to achieve fast response of yaw rate tracking control where it is able to track the desired yaw rate with minimum overshoot and fast settling time for a J-turn cornering manoeuvre. As a conclusion, the CNF control is able to improve the transient response of vehicle yaw rate. For future works, the CNF control technique will be evaluated with other cornering manoeuvre. It will improve to cater external disturbances such as crosswind and uncertainties of vehicle parameters. Co-simulations with vehicle dynamics software such as CarSim will be conducted to validate the proposed controller.

References

1. Zhou, H., Liu, Z.: Vehicle Yaw Stability-Control System Design Based on Sliding Mode and Backstepping Control Approach. IEEE Transactions on Vehicular Technology 59, 3674–3678 (2010)
2. Zhou, H., Liu, Z.: Design of Vehicle Yaw Stability Controller Based on Model Predictive Control. In: IEEE Intelligent Vehicles Symposium, pp. 802–807. IEEE Press, Xian (2009)
3. Cerone, V., Milanese, M., Regruto, D.: Yaw Stability Control Design Through A Mixed-Sensitivity Approach. IEEE Transactions on Control Systems Technology 17, 1096–1104 (2009)
4. Canale, M., Fagiano, L., Ferrara, A., Vecchio, C.: Comparing Internal Model Control and Sliding-Mode Approaches For Vehicle Yaw Control. IEEE Transactions on Intelligent Transportation Systems 10, 31–41 (2009)

5. Canale, M., Fagiano, L., Ferrara, A., Vecchio, C.: Vehicle Yaw Control via Second-Order Sliding-Mode Technique. IEEE Transactions on Industrial Electronics 55, 3908–3916 (2008)
6. Canale, M., Fagiano, L., Milanese, M., Borodani, P.: Robust Vehicle Yaw Control Using An Active Differential and IMC Techniques. Control Engineering Practice 15, 923–941 (2008)
7. Li, Q., Shi, G., Lin, Y., Wei, J.: Yaw Rate Control of Active Front Steering Based on Fuzzy-Logic Controller. In: Second International Workshop on Education Technology and Computer Science, Wuhan, pp. 125–128 (2010)
8. Tekin, G., Ünlüsoy, Y.S.: Design and Simulation of an Integrated Active Yaw Control System for Road Vehicles. International Journal of Vehicle Design 52, 5–19 (2010)
9. Lin, Z., Pachter, M., Ban, S.: Toward Improvement of Tracking Performance - Nonlinear Feedback for Linear Systems. International Journal of Control 70, 1–11 (1998)
10. Turner, M.C., Postlethwaite, I., Walker, D.J.: Non-Linear Tracking Control for Multivariable Constrained Input Linear Systems. International Journal of Control 73, 1160–1172 (2000)
11. Chen, B.M., Lee, T.H., Peng, K., Venkataramanan, V.: Composite Nonlinear Feedback Control for Linear Systems With Input Saturation: Theory and an Application. IEEE Transactions on Automatic Control 48, 427–439 (2003)
12. He, Y., Chen, B.M., Wu, C.: Composite Nonlinear Control with State and Measurement Feedback for General Multivariable Systems with Input Saturation. Systems and Control Letters 54, 455–469 (2005)
13. Lan, W., Thum, C.K., Chen, B.M.: A Hard-Disk-Drive Servo System Design Using Composite Nonlinear-Feedback Control with Optimal Nonlinear Gain Tuning Methods. IEEE Transactions on Industrial Electronics 57, 1735–1745 (2010)
14. Cheng, G., Peng, K.: Robust Composite Nonlinear Feedback Control with Application to A Servo Positioning System. IEEE Transactions on Industrial Electronics 54, 1132–1140 (2007)
15. Cheng, G., Jin, W.: Parameterized Design of Nonlinear Feedback Controllers for Servo Positioning Systems. Journal of Systems Engineering and Electronics 17, 593–599 (2006)
16. Peng, K., Chen, B.M., Cheng, G., Lee, T.H.: Modeling and Compensation of Nonlinearities and Friction in A Micro Hard Disk Drive Servo System with Nonlinear Feedback Control. IEEE Transactions on Control Systems Technology 13, 708–721 (2005)
17. Ismail, M.F., Sam, Y.M., Peng, K., Aripin, M.K., Hamzah, N.A.: A Control Performance of Linear Model and The Macpherson Model for Active Suspension System using Composite Nonlinear Feedback. In: IEEE International Conference on Control System, Computing and Engineering, Penang, pp. 227–233 (2012)
18. Jazar, R.N.: Vehicle Dynamics: Theory and Application. Springer, Heidelberg (2008)
19. Lan, W., Thum, C.K., Chen, B.M.: Optimal Nonlinear Gain Tuning of Composite Nonlinear Feedback Controller and its Application to A Hard Disk Drive Servo System. In: 48th IEEE Conference on Decision and Control Held Jointly with 28th Chinese Control Conference, Shanghai, pp. 3169–3174 (2009)
20. On Selection of Nonlinear Gain in Composite Nonlinear Feedback Control for a Class of Linear Systems. In: 46th IEEE Conference on Decision and Control, pp. 1198-1203. New Orleans (2007)
21. He, J., Crolla, D.A., Levesley, M.C., Manning, W.J.: Coordination of Active Steering, Driveline, and Braking for Integrated Vehicle Dynamics Control. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 220, 1401–1421 (2006)

Two-Stage Nested Optimization-Based Uncertainty Propagation Method for Uncertainty Reduction

Xiaochao Qian, Wei Li, and Ming Yang*

Control and Simulation Center, Harbin Institute of Technology,
150080 Harbin, P.R. China
everqxc@gmail.com, {fleehit,yangming_csc}@163.com

Abstract. A new method of epistemic uncertainty reduction is investigated according to the uncertainty of modeling and simulation. First, technical background of the uncertainty propagation in modeling and simulation is introduced. Uncertainty propagation procedure is divided into three major steps. Next, an epistemic uncertainty reduction method based on two-stage nested sampling uncertainty propagation is proposed, Monte Carlo Simulation method for the inner loop is applied to propagate the aleatory uncertainties and method based on optimization method is applied for the outer loop to propagate the epistemic uncertainties. The optimization objective function is the difference between the result of inner loop and the experiment data. Finally, the thermal challenge problem is given to validate the reasonableness and effectiveness of the proposed method.

Keywords: simulation, uncertainty propagation, epistemic uncertainty, reduction.

1 Introduction

Realistic modeling and simulation of complex systems must include the nondeterministic features of the system and the environment [1]. 'Nondeterministic' mean that the response of the system is not precisely predictable because of the existence of uncertainty in the system or the environment, or human interaction with the system. Uncertainty in modeling and simulation can be classified as either (a) aleatory –the inherent variation in a quantity that, given sufficient samples of the stochastic process, can be characterized via a probability density distribution, or (b) epistemic – uncertainty due to lack of knowledge by the modelers, analysts conducting the analysis, or experimentalists involved in validation [2-4]. Epistemic uncertainty can be reduced by increasing information and knowledge. Uncertainty influence the accuracy of the simulation, and accuracy is the key in simulation credibility assessment and fidelity study. Simulation system which does not satisfied the accuracy requirement is incredible. Therefore, using increasing information and knowledge to reduce the epistemic uncertainty could improve the accuracy and credibility of the simulation system.

* Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 243–252, 2013.

When experimental observations are available for uncertainty analysis, Analysts would often like to adjust uncertainty model parameters in order to improve the agreement between the model predictions and observations; it is the process of reducing epistemic uncertainty, which is also a part of model calibration process. And uncertainty associated with model inputs directly implies uncertainty associated with model outputs. Thus, the process of epistemic uncertainty reduction is actually an opportunity to calibrate the parameters' epistemic uncertainty by using the simulation output which is estimated by the uncertainty propagation. Recently, several methods to calibrate the simulation models have been introduced, including Bayesian calibration, generalized-likelihood-uncertainty-estimation (GLUE) procedure and Kennedy and O'Hagan's method [5]. However, most of the calibration methods did not consider the uncertainties of the model inputs, especially the epistemic uncertainty. Thus, calibration method which is dealing with the epistemic uncertainty of model input should be considered.

This paper addresses the uncertainty problem through the example of the Thermal Challenge Problem [6-7]. We interpreted the challenge problem to involve two goals. The first goal is to develop an uncertainty propagation method which can deal with both of the two types of uncertainty. The second goal is to use the propagation method to reduce the effecting of the epistemic uncertainty. Technical background of the uncertainty propagation in modeling and simulation is introduced in Sec.2. An epistemic uncertainty reduction method based on two-stage nested sampling uncertainty propagation is investigated in Sec.3. Thermal challenge problem is given to validate the reasonableness and effectiveness of the proposed method in Sec.4. Finally, conclusions are provided in Sec.5.

2 Technical Background of the Uncertainty Propagation

Roy and Oberkampf [8] provided a comprehensive framework for treating all sources of uncertainty and their effects on the predicted system responses quantities of interest. There are three major steps about the uncertainty propagation, include (1) the identification of all sources of uncertainty, (2) characterization of model input uncertainties, (3) propagation of input uncertainties through the model to obtain uncertainties in the outputs. In this section, the background of the three steps will be introduced.

2.1 Sources of Uncertainty in Modeling and Simulation

Oberkampf et al. [9] divided the modeling and simulation into six phases: conceptual modeling of the physical system, mathematical modeling of the conceptual model, discretization and algorithm selection for the mathematical model, computer programming of the discrete model, numerical solution of the computer program model, and representation of the numerical solution. And general sources of uncertainty, both aleatory and epistemic, are identified of each phase. Some researches categorized the sources of uncertainty as occurring in model inputs, numerical approximations, and in the form of the mathematical model. The computing method of the three types of uncertainties can be found in Ref. [8]. In this paper, we are only interested in the

model input uncertainties. And the model input uncertainties also can be classified as either purely aleatory, purely epistemic, or a mixture of the two.

2.2 Characterize Uncertainties

As discussed earlier, the model input uncertainties can be classified as either purely aleatory, purely epistemic, or a mixture of the two. For purely aleatory uncertainties, the uncertainty is characterized as a precise distribution. For purely epistemic uncertainties, the uncertainty can be characterized by interval, possibility theory, evidence theory or probability theory [4]. An imprecise distribution is given to represent an uncertainty that is characterized as a mixture of aleatory and epistemic uncertainty. This mathematical structure is a distribution where the parameters of the distribution are not scalars, but are either distributions themselves or interval-valued quantities.

2.3 Uncertainties Propagation through the Model

The use of an analysis approach to estimate the effect of uncertainties on model output is referred to as uncertainty propagation. Uncertainty propagation methods are general divided into two categories. The first category is the conventional sample-based approach such as Monte Carlo Simulation (MCS). The disadvantage of sample-based approach is computational cost. This is especially the case if long-running models are under consideration or probabilities very close to zero or one must be estimated. The second category of uncertainty propagation approach is based on sensitivity analysis. Most of these methods only provide the information of mean and variance based on approximations. The level of accuracy is not sufficient for applications in the calibration of the epistemic uncertainties of model input. There is no probability or frequency of occurrence associated with it based on how the sample was chosen of the epistemic uncertainties propagation. And sampling an aleatory uncertainty implies a sample is taken from a random variable and that each sample is associated with a probability. As mentioned above, the propagation of the aleatory and epistemic uncertainties are segregated. It is a two-stage nested MCS: sampling of values of the epistemic variables (outer loop) and nested conditional sampling of values of the aleatory variables (inner loop). However, the computational model may frequently be very complex and expensive to run such a nested MCS, alternative sampling techniques such as quasi-Monte Carlo simulations, Latin hypercube sampling, and experiment design method have been proposed to reduce the computer cost. Meta-model method as the surrogate model of a complex simulation model was proposed for the same reason.

3 Epistemic Uncertainty Reduction Method

As mentioned in section 2, the propagation of the aleatory and epistemic uncertainties is segregated. And the propagation method is a two-stage nested sample-based simulation, the inner loop propagates the aleatory uncertainties and the outer loop

propagates the epistemic uncertainties. In this paper, a two-stage nested sample-based uncertainty propagation method is introduced for the epistemic model input uncertainty reduction. MCS method for the inner loop is applied to propagation the aleatory uncertainties and method based on optimization is applied for the outer loop to propagation the epistemic uncertainties. The objective function is the difference between the result of inner loop and the experiment data and the constraint is the epistemic uncertainty space of the model input parameters. With the propagation of uncertainties, set values of model input parameters with epistemic uncertainties will be found and the epistemic uncertainties will be reduced.

3.1 MCS Method for the Aleatory Uncertainty Propagation

The MCS method for the aleatory uncertainty propagation is briefly summarized as follows:

- Generation of sample: Assume that $D_1, D_2, \cdots D_n$ is the distributions of the aleatory uncertainty, n is the total dimension of aleatory uncertainty space. Generate random numbers $x^i = [x_1^{(i_1)}, x_2^{(i_2)}, \cdots, x_n^{(i_n)}]$, where i is the single index ordered by the graded lexicographic method, $i = 1, 2, \cdots M$ according to the given distributions.
- Propagation of sample through the model: Propagation of the sample through the model to produce the $[x^{(i)}, y(x^{(i)})]$ mapping from model inputs to model outputs. This procedure is a DO loop around the model that (i) supplies the sampled input to the model, (ii) runs the model, and (iii) stores model outputs for later analysis.
- Estimate the required solution's statistics: For example, the solution's mean can be estimated as:

$$\mu_y(t) \simeq \overline{y}(t) = \frac{1}{M}\sum_{i=1}^{M} y(t, x^{(i)}) \tag{1}$$

3.2 Optimization Method for the Epistemic Uncertainty Propagation

The propagation method for epistemic model input uncertainty reduction is an optimization problem as follows:

$$\begin{aligned} &\min\ D(y(U_a), f(U_e, U_a)) \\ &s.t.\ \ U_e \in U^I = [U^L, U^R]\ U_{e_i} \in U_{e_i}^I = [U_{e_i}^L, U_{e_i}^R] \end{aligned} \tag{2}$$

Where $D(y(U_a), f(U_e, U_a))$ is the objective function of model output and experiment data. U_e is the design vector of epistemic uncertainty with q dimensions. The epistemic uncertainty is expressed by the interval U^I. U_a is the aleatory uncertainty propagated by MCS method. The optimization problem can be solved by most of the optimization algorithm. A flowchart for the uncertainty propagation method of epistemic uncertainty reduction is shown in figure 1.

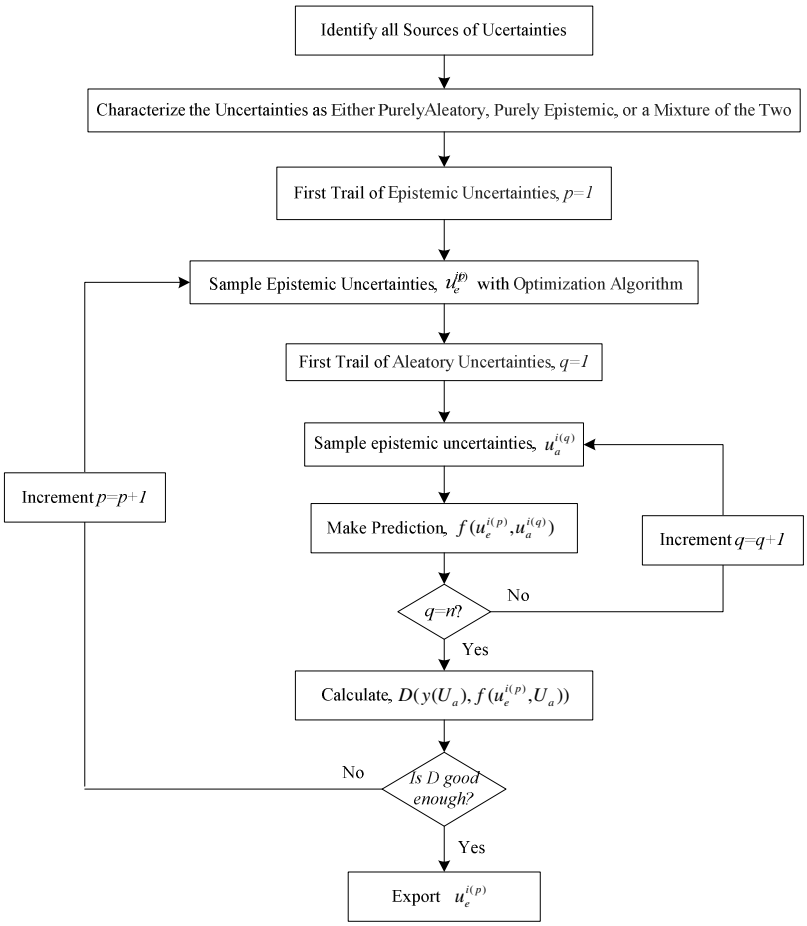


Fig. 1. Flowchart for the uncertainty propagation method of epistemic uncertainty reduction

4 Application with the Thermal Challenge Problem

This section presents a thermal challenge problem for the application of this paper. The thermal problem involves validating a model for heat conduction in a solid. The mathematical model is based on one-dimensional, linear heat conduction in a solid slab, with heat flux boundary conditions. Experimental data from a series of material characterization, validation, and accreditation experiments related to the mathematical model are provided. The objective is to use this problem to calibrate the model input parameters with epistemic uncertainty, and compared the results with the other methods.

4.1 Introduction of the Thermal Challenge Problem

The mathematical model is of the temperature under heating of a device constructed of some material and has the form:

$$T(x,t) = T_i + \frac{qL}{k}\left[\frac{(k/\rho C_p)t}{L^2} + \frac{1}{3} - \frac{x}{L} + \frac{1}{2}\left(\frac{x}{L}\right)^2 - \frac{2}{\pi^2}\sum_{n=1}^{6}\frac{1}{n^2}e^{-n^2\pi^2\frac{(k/\rho C_p)t}{L^2}}\cos\left(n\pi\frac{x}{L}\right)\right] \quad (3)$$

where T is temperature, x is location within the material, t is time since the onset of heating, T_i is the initial ambient temperature, q is the heat flux, L is the thickness of the material, and k, ρC_p are properties of the material, k is the thermal conductivity and ρC_p is the heat capacity. The regulatory requirement is

$$\Pr ob\left(900°C < T_{x=0\ cm, t=1000\ s, T_i=25\ °C, q=3500\ W/m^2, L=1.90\ cm}\right) < 0.01 \quad (4)$$

The formulation and numerical details of the thermal challenge problem are given in [4] and will not be reiterated here except in briefest outline. The problem consists of a mathematical model, three sets of experimental data including material characterization experiment, ensemble validation experiment and accreditation validation experiment which differ in size ('low', 'medium' and 'high'), and a regulatory requirement. In this paper, only high-level experiments data is applied to validate the reasonableness and effectiveness of the proposed method.

The predicted temperature would be a distribution because, although the values of x, t, q, L, and T_i are prescribed for us in the statement of the problem, the values of the material properties k and ρC_p are only known by sample data.

4.2 Material Characterization

On the use of the material parameters, Liu et al. [10] assumed that all the material parameters followed Gaussian distributions, and only applied the thermal conductivity data bigger than 500℃; McFarland et al. [11] classified the material parameters according to the different temperature; Xiong et al. [12] applied all of the material data through statistical analysis; Hills [13] considered the thermal conductivity as linear function of temperature. Epistemic uncertainty is not considered in all of these studies of material. In this paper, epistemic uncertainty of thermal conductivity and heat capacity is considered. Scatter plot of thermal conductivity and heat capacity experiment data is shown in Figures 2.

Uncertainty of thermal conductivity and heat capacity is characterized based on the experiment data. Nonlinear regression is applied here to analyze thermal conductivity; power function ax^b is chosen for the regression equation. We assume both of the two parameters contain epistemic uncertainty, and uncertainty space is the 95% confidence interval of them. The Regression curve and confidence interval is shown in figure 3. Heat capacity is assumed to follow Gaussian distribution, the distribution

is $\rho C_p \sim N(\mu_{\rho C}, \delta_{\rho C}^2)$, parameters of this distribution is estimated by using the experimental data, 95% confidence interval of the mean and variance $[\mu_{\rho C}^L, \mu_{\rho C}^R], [\delta_{\rho C}^L, \delta_{\rho C}^R]$ is considered as the epistemic space. As describe in this paper, thermal conductivity contains epistemic uncertainty, and the heat capacity contains both aleatory and epistemic uncertainty.

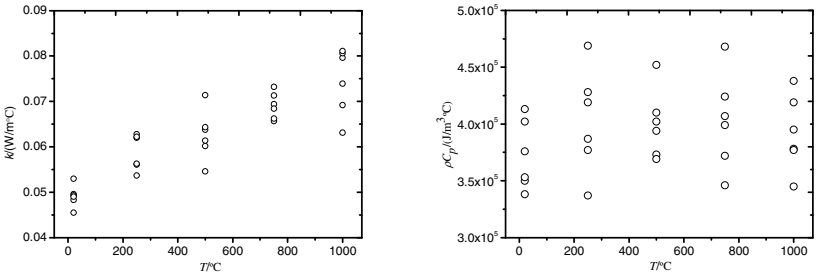


Fig. 2. Scatter plot of thermal conductivity and heat capacity experiment data

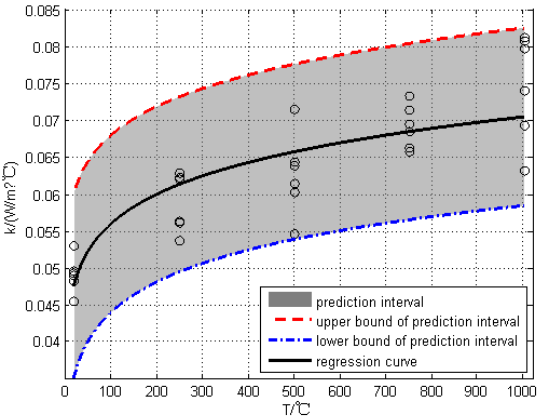


Fig. 3. Regression curve and confidence interval of the thermal conductivity

4.3 Epistemic Uncertainty Reduction

The uncertainty propagation method presented in section 3 is applied here for the epistemic uncertainty reduction. Aleatory uncertainty of thermal conductivity is propagated in the inner loop, epistemic uncertainty of thermal conductivity and heat capacity is propagated in the outer loop.

Difference between ensemble experimental data and aleatory uncertainty propagation results is used here as the objective function of epistemic uncertainty reduction. The objective function is as follow:

$$RMSE_T = \frac{1}{n}\sum_{i=1}^{n}\sqrt{y_{En_i}^2 - y_{S_i}^2} \tag{5}$$

Where n is the experiment number at different heating time t of the ensemble experiment, y_{En_i} is the output temperature of ensemble experiment at time t_i, y_{S_i} is the mean of the experiment results at t_i.

$$y_{S_i} = \frac{1}{m}\sum_{j=1}^{m} y_{Sa_j} \tag{6}$$

Genetic algorithm (GA) is applied in this paper for the optimization algorithm of epistemic uncertainty reduction, after the reduction procedure, the epistemic uncertainty of the material parameters is disappeared.

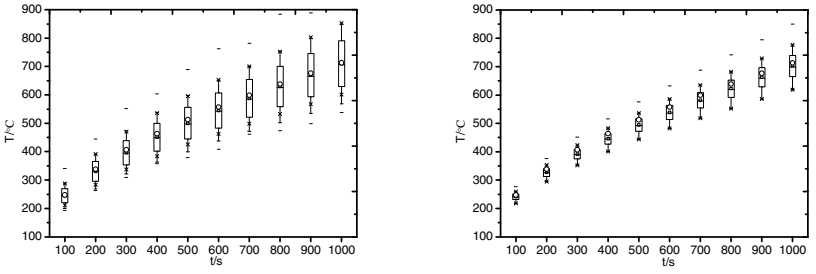


Fig. 4. Accreditation experiment data and the uncertainty propagation result

Accreditation experiment data is applied for the validation of the epistemic uncertainty reduction. Accreditation experiment contents two group of data, in this paper, we applied the mean of them. Uncertainty parameters before and after epistemic uncertainty reduction are propagated by using MCS method. The accreditation experiment data and the uncertainty propagation result is show as figure 4. In this figure, round dot denotes the accreditation data and box plot denotes the statistic characteristics of model outputs. As shown in the figures, uncertainty of the model outputs decreased significantly after the reduction.

4.4 Model Prediction

Failure probability of the thermal challenge problem is calculated with the material parameters before and after the reduction. With the parameters before the reduction, the material parameters contents aleatory and epistemic uncertainty, a two-stage

nested MCS method is applied here to propagate the uncertainty. With the parameters after the reduction, there is only aleatory uncertainty and a one-stage MCS method is applied to propagate the uncertainty. The cumulative distribution curve and failure curve is shown as figure 5. Detailed comparison results are shown in table 1.

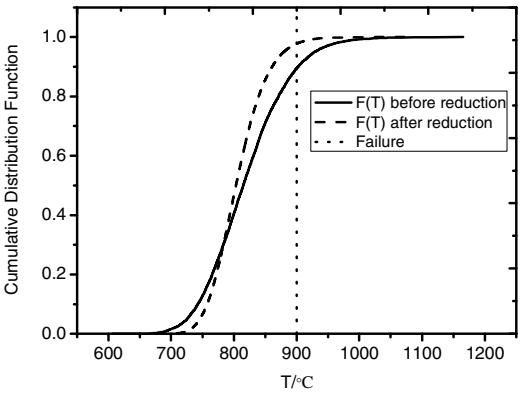


Fig. 5. Cumulative distribution and failure curve

Table 1. Comparison results of the model prediction

	RMSE	90% CI for $T(1000)$	$P(T(1000) > 900^{\circ}C)$
Before reduction	39.43	[725.4, 928.5]	0.106
After reduction	12.15	[745.7, 879.1]	0.022

Before epistemic reduction, failure probability that the predicting temperature over 900℃ is 0.106, after the reduction, the failure probability reduced to 0.022. The model accuracy has improved significantly after the reduction. 90% confidence interval of the predicting output is obviously minished too, the reasonableness and effectiveness of the proposed method is validated.

5 Conclusions

According to the uncertainty of modeling and simulation, an epistemic uncertainty reduction method is proposed. Technical background of the uncertainty propagation in modeling and simulation is introduced. Uncertainty propagation procedure is divided into three major steps including (1) the identification of all sources of uncertainty, (2) characterization of model input uncertainties, (3) propagation of input uncertainties through the model to obtain uncertainties in the outputs. A two-stage nested sampling uncertainty propagation method is investigated for the reduction of epistemic uncertainty. With this reduction method, the accuracy and credibility of the simulation system will be improved.

The selection of objective function for the optimization algorithm is directly influences the optimization result. In this paper, RMSE between model output and experiment data is selected as the objective function. However, the simulation output and the experiment may not one correspondence. How to choose the objective function at this situation is the focus of future research. Genetic algorithm is applied in this paper for the optimization algorithm, genetic algorithm optimized the epistemic uncertainty interval to a point, but in the real situation someone may want to get a smaller interval of the epistemic uncertainty, in this situation, interval optimization algorithm should be applied in this proposed method.

Acknowledgments. This work is supported by the Innovative Team Program of the National Natural Science Foundation of China under Grant No. 61021002.

References

1. Booker, J.M., Ross, T.J.: An evolution of uncertainty assessment and quantification. Scientia Iranica 18(3), 669–676 (2011)
2. Pilch, M., Trucano, T.G., Helton, J.C.: Ideas underlying the Quantification of Margins and Uncertainties. Reliability Engineering & System Safety 96(9SI), 965–975 (2011)
3. Sankararaman, S., Mahadevan, S.: Model validation under epistemic uncertainty. Reliability Engineering & System Safety 96(9SI), 1232–1241 (2011)
4. Helton, J.C., Johnson, J.D.: Quantification of margins and uncertainties: Alternative representations of epistemic uncertainty. Reliability Engineering & System Safety 96(9SI), 1034–1052 (2011)
5. Ferson, S., Oberkampf, W.L., Ginzburg, L.: Model validation and predictive capability for the thermal challenge problem. Comput. Methods Appl. Mech. Engrg. 197(29-32), 2408–2430 (2008)
6. Yuan, J., Ng, S.H.: A sequential approach for stochastic computer model calibration and prediction. Reliability Engineering & System Safety 111, 273–286 (2013)
7. Dowding, K.J., Pilch, M., Hills, R.G.: Formulation of the thermal problem. Comput. Methods Appl. Mech. Engrg. 197(29-32), 2385–2389 (2008)
8. Roy, C.J., Oberkampf, W.L.: A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. Comput. Methods Appl. Mech. Engrg. 200(25-28), 2131–2144 (2011)
9. Oberkampf, W.L., Deland, S.M., Rutherford, B.M., et al.: Error and uncertainty in modeling and simulation. Reliability Engineering & System Safety 75(3), 333–357 (2002)
10. Liu, F., Bayarri, M.J., Berger, J.O., et al.: A Bayesian analysis of the thermal challenge problem. Comput. Methods Appl. Mech. Engrg. 197(29-32), 2457–2466 (2008)
11. McFarland, J., Mahadevan, S.: Multivariate significance testing and model calibration under uncertainty. Comput. Methods Appl. Mech. Engrg. 197(29-32), 2467–2479 (2008)
12. Xiong, Y., Chen, W., Tsui, K.L., et al.: A better understanding of model updating strategies in validating engineering models. Comput. Methods Appl. Mech. Engrg. 198(15-16), 1327–1337 (2009)
13. Hills, R.G., Dowding, K.J.: Multivariate approach to the thermal challenge problem. Comput. Methods Appl. Mech. Engrg. 197(29-32), 2442–2456 (2008)

Intelligent Identification Methods for Rotor Resistance Parameter of Induction Motor Drive

Moulay Rachid Douiri and Mohamed Cherkaoui

Mohammadia Engineering School, Department of Electrical Engineering,
Ibn Sina Avenue, Agdal-Rabat 765, Morocco

Abstract. This paper presents two intelligent identification approaches for rotor resistance of an indirect vector controlled induction motor drive. First approach is based on fuzzy logic control (FLC) able to compensate for variations and errors, FLC scheme was employed to overcome the lack of a precise mathematical model of the process. In the second approach is based on artificial neural networks (ANNs), the error between the rotor flux linkages based on a neural network model and a voltage model is back propagated to adjust the weights of the neural network model for the rotor resistance estimation. The performances of the two intelligent approaches are investigated and compared in simulation.

Keywords: fuzzy logic, neural networks, indirect vector control, induction motor, rotor resistance.

1 Introduction

Field-oriented control has been used widely to obtain improved control performance of induction motor drives [1]-[2]. In general, field-oriented control can be classified into direct and indirect field orientation [3], the main difference being that direct field-oriented control is accomplished with the help of measurement or estimation of flux linkages to obtain the flux angle while indirect field-oriented control is accomplished by utilization of the feed-forward slip command and feedback velocity signal to obtain the decoupling angle [4]. However, the transient performance of both field orientations is determined primarily by the accuracy of the motor parameters used in field orientation. To achieve high-performance field-oriented control of induction motors requires knowledge of the motor parameter sensitivities [2]-[3].

The rotor resistance is an important parameter which is involved in rotor flux estimation and the control law to compensate for the nonlinearity of system [5]-[6]. However, this parameter varies with machine temperature. In addition it was demonstrated that a poor estimate of this parameter affects the regulation (pursuit of flux trajectory and rotor speed) and even it can introduce oscillations [6]-[7]. This difficulty has been the main source of our motivation for this research; in which one propose two intelligent approaches based on fuzzy logic and artificial neural networks to identify the rotor resistance. Several authors have contributed to estimating rotor resistance [5]-[6]-[7]-[8]-[9]-[10]-[11].

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 253–263, 2013.

Fuzzy logic as many of the linguistic terms used by human involve degree of fuzziness and relative significance, it is desirable to address the impact of fuzziness on the solutions made by experts for complex problems. In 1965, fuzzy logic theory was developed by Zadeh [12]-[13], to deal with uncertainties that are not statistical in nature. The concept of fuzzy sets theory differs from that of the conventional crisp sets mainly in the degree by which an object belong to a set. In crisp sets, objects are either included or excluded from a set [14]. In fuzzy set, on the other hand, objects are described in such a way so as to allow gradual transition from being a member of a set to non member. Each object contains a degree of membership ranging from zero to one, where zero indicates non membership while one indicates full membership [14], [15].

An artificial neural network (ANN) is essentially a way to learn the relationship between a set of input data and the corresponding output data. That is, it can memorize data, generalize this information when given new input data, and adjust when the relationship changes [16]. The training is normally done with input-output examples. After training, ANNs have the capability of generalization. That is, given previously unseen input data, they can interpolate from the previous training data. They were inspired by models of neural cell structure in organic brains [17]. They became popular in the research community when architectures were found to enable the learning of nonlinear functions and patterns [16].

This paper is organized as follows: The principle of indirect field oriented control of induction motor drive is presented in the second part, section three presents an fuzzy logic rotor resistance estimator, neural networks rotor resistance in the four part, and the five parts is devoted to illustrate the simulation performance of this control strategy, a conclusion and reference list at the end.

2 Indirect Field Oriented Control

In field orientation concept of induction motor, the goal is to obtain an electromagnetic torque proportional to the quadrature component of stator current i_{sq} (at constant flux) and can control the flux by acting on the direct component of this current i_{sd} [2]. This can be accomplished by choosing ω_e to be the instantaneous speed of ψ_r and locking the phase of the reference system such that the rotor flux is entirely in the d-axis (flux axis), resulting in the mathematical constraint:

$$\psi_r = \psi_{rd}\,; \quad \psi_{rq} = 0. \tag{1}$$

The Fig. 1 illustrates the principle of indirect method using phase diagram. At any instant, d electrical axis is in angular position θ_e relative to α axis. The angle θ_e is the result of the sum of both rotor angular and slip angular positions, as follows:

$$\begin{cases} \theta_e = \theta_r + \theta_{sl} \\ \omega_e t = \omega_r t + \omega_{sl} t = (\omega_r + \omega_{sl})t \end{cases}. \tag{2}$$

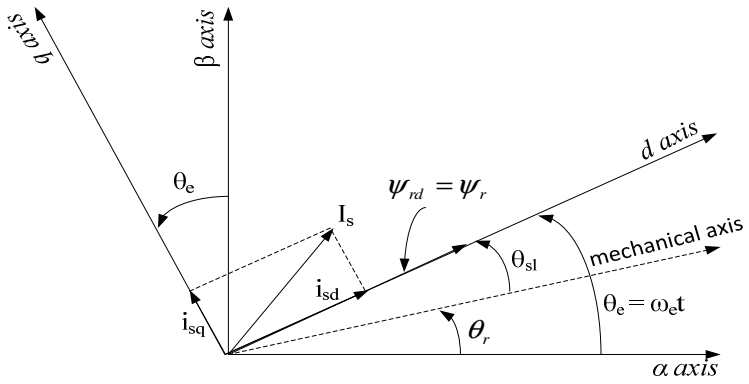


Fig. 1. Vector diagram for indirect field oriented control

The rotor dynamics are given by the following equations:

$$\frac{\mathrm{d}\psi_r}{\mathrm{d}t} = \frac{L_m}{\tau_r} i_{sd} - \frac{1}{\tau_r}\psi_r. \tag{3}$$

$$\frac{\mathrm{d}\omega_r}{\mathrm{d}t} = \frac{3n_p{}^2 L_m}{2JL_r}\psi_r i_{sq} - \frac{f}{J}\omega_r - \frac{n_p}{J}T_l. \tag{4}$$

$$T_e = \frac{3n_p L_m \psi_r}{2L_r} i_{sq}. \tag{5}$$

$$\rho = \int \omega_e \mathrm{d}t = \int \left(\omega_r + \frac{R_r L_m i_{sq}}{L_r \psi_r} \right) \mathrm{d}t. \tag{6}$$

The rotor flux magnitude is related to the direct axis stator current by a first-order differential equation; thus, it can be controlled by controlling the direct axis stator current. Under steady-state operation rotor flux is constant, so Eq. (3) becomes:

$$\psi_r = L_m i_{sd}. \tag{7}$$

Indirect vector control can be implemented using the following equations:

$$i_{sd}^* = \frac{\psi_r^*}{L_m}. \tag{8}$$

$$i_{sq}^* = \frac{2L_r T_e^*}{3n_p L_m \psi_r^*}. \tag{9}$$

$$\omega_{sl}^{*} = \frac{R_r L_m i_{sq}^{*}}{L_r \psi_r^{*}}. \tag{10}$$

$$\rho^{*} = \int \omega_e^{*} \mathrm{d}t = \int (\omega_r + \omega_{sl}^{*}) \mathrm{d}t. \tag{11}$$

3 Fuzzy Rotor Resistance Estimator

The input variables of the estimator should wear explicitly or implicitly information related to the change in resistance. We find that the torque could be the candidate. We can estimate the actual torque from the stator flux. This method is simple but it is not recommended for low-speed operation since in this region it is very difficult to estimate exactly the stator flux.

We will use a function Φ, which is a modification of the function used in Eq. (3):

$$\Phi = \frac{1}{\omega_e}\left(i_{qs} \frac{d\psi_{dr}}{dt} - i_{ds} \frac{d\psi_{qr}}{dt}\right). \tag{12}$$

As the resistance variation with temperature is very slow, we can estimate in steady state. We can demonstrate that steady state; the function Φ can be calculated as follows:

$$\Phi = -i_{ds}\psi_{dr} = -i_{ds}\hat{\psi}_{dr} = F_{est}. \tag{13}$$

where $\hat{\psi}_{dr}$ is the estimated flux of the *d*-axis.

The actual value of the function Φ is calculated:

$$\Phi_{act} = \frac{L_r}{L_m}\left[\frac{1}{\omega_e}(v_{ds} i_{qs} - v_{qs} i_{ds}) + L_s \sigma (i_{ds}^2 + i_{qs}^2)\right]. \tag{14}$$

In Eq. (14) the function Φ is calculated from stator voltage and current. The rotor voltage will be available to our estimation algorithm.

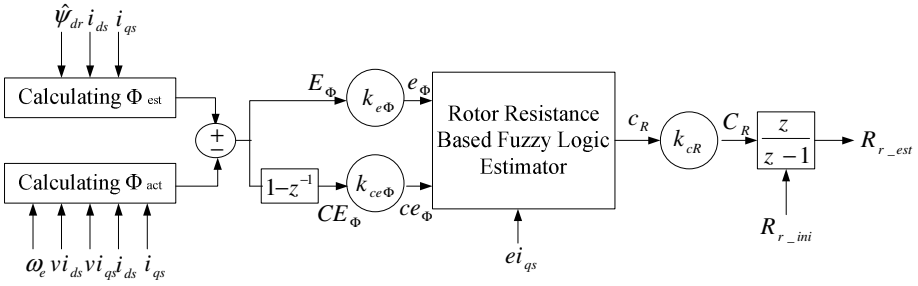


Fig. 2. Structure of the fuzzy rotor resistance estimator

Fig. 2 shows the configuration of rotor resistance estimation. The functions Φ_{est} and Φ_{act} it's first calculated. The error between Φ_{est} and Φ_{act} and its derivation are the inputs of the estimator e, they are then calculated as follows:

$$E_{\Phi}(k) = \Phi_{est}(k) - \Phi_{act}(k). \tag{15}$$

$$CE_{\Phi}(k) = E_{\Phi}(k) - E_{\Phi}(k-1). \tag{16}$$

The internal structure of the estimator involves three steps: fuzzification, inference and defuzzification. The signals $e_{\Phi}(k)$ and $ce_{\Phi}(k)$ are normalized and deducted the signals $E_{\Phi}(k)$ and $CE_{\Phi}(k)$ by multiplying by the factor $k_{e\Phi}$ and $k_{ce\Phi}$.

The estimated value of the resistance increment is obtained by multiplying the estimator output $c_R(k)$ by the gain k_{cR}. Resistance is finally integration increment:

$$R_r(k) = R_r(k-1) + k_{cR}\, c_R(k). \tag{17}$$

Note that the initial value of the resistance R_r is its nominal value.

This resistance is estimated to be used in indirect vector control algorithm to ensure optimal performance.

The rules table contains 49 rules (7 x 7) as shown in Table 1.

Table 1. Fuzzy linguistic rule table

C_R		e_{Φ}						
		NB	NM	NS	Z	PS	PM	PB
	NB	NB	NB	NB	NB	NM	NS	Z
	NM	NB	NB	NB	NM	NS	Z	PS
	NS	NB	NB	NM	NS	Z	PS	PM
ce_{Φ}	AZ	NB	NM	NS	Z	PS	PM	PB
	PS	NM	NS	Z	PS	PM	PB	PB
	PM	NS	Z	PS	PM	PB	PB	PB
	PB	Z	PS	PM	PB	PB	PB	PB

The fuzzy sets are characterized by standard designations: NB (negative big), NM (negative medium), NS (negative small), Z (zero), PS (positive small), PM (positive medium), and PB (positive big).

As the time constant of the resistance variation as a function of temperature is much greater than the electric time constant of the motor, we can estimate the resistance in steady state, where there is no load variation or change control signal. This ensures that signal variations E_{Φ}(k) and $CE_{\Phi}(k)$ are caused by R_r variation only.

4 Neural Rotor Resistance Estimator

In Model Reference Adaptive System (MRAS) or Model Reference Adaptive Control (MRAC), there is a reference model, an adjustable model, and an adaptation mechanism.

Reference model:

$$\frac{d}{dt}\begin{bmatrix}\psi_{rd}\\ \psi_{rq}\end{bmatrix}=\frac{L_r}{L_m}\left(\begin{bmatrix}v_{sd}\\ v_{sq}\end{bmatrix}-R_s\begin{bmatrix}i_{sd}\\ i_{sq}\end{bmatrix}-\sigma L_s\frac{d}{dt}\begin{bmatrix}i_{sd}\\ i_{sq}\end{bmatrix}\right) \tag{18}$$

Adjustable model

$$\frac{d}{dt}\begin{bmatrix}\hat{\psi}_{rd}\\ \hat{\psi}_{rq}\end{bmatrix}=\begin{bmatrix}-\frac{R_r}{L_r} & -\hat{\omega}_r\\ \hat{\omega}_r & -\frac{R_r}{L_r}\end{bmatrix}\begin{bmatrix}\hat{\psi}_{rd}\\ \hat{\psi}_{rq}\end{bmatrix}+\frac{L_m R_r}{L_r}\begin{bmatrix}i_{sd}\\ i_{sq}\end{bmatrix} \tag{19}$$

In neural rotor resistance estimator technique (Fig. 3), there is a reference model that provides the desired output, and a neural network model has two layers based on back propagation technique which provides for the estimated output, both outputs are compared and the total error between the desired state variable and estimated is then back-propagated to adjust the weight (rotor speed) of the neural model [9], [10].

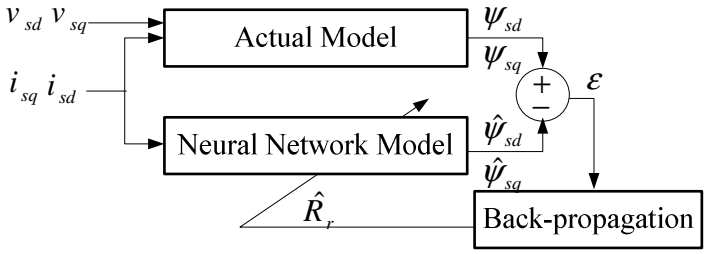


Fig. 3. Structure of the neural rotor resistance estimator

The Adjustable model Eq. (20), after discretization is written as follows:

$$\begin{bmatrix}\hat{\psi}_{rd}(k)\\ \hat{\psi}_{rq}(k)\end{bmatrix}=w_1 I\begin{bmatrix}\hat{\psi}_{rd}(k-1)\\ \hat{\psi}_{rq}(k-1)\end{bmatrix}+w_2 J\begin{bmatrix}\hat{\psi}_{rd}(k-1)\\ \hat{\psi}_{rq}(k-1)\end{bmatrix}+w_3\begin{bmatrix}i_{sd}(k-1)\\ i_{sq}(k-1)\end{bmatrix} \tag{20}$$

where $w_1=1-\frac{R_r}{L_r}t_s$, $w_2=w_r t_s$, $w_3=\frac{L_m R_r}{L_r}t_s$, $I=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, $J=\begin{bmatrix}0 & -1\\ 0 & 1\end{bmatrix}$

Eq. (20) can also be written as:

$$\hat{\psi}_r(k)=w_1 A_1+w_2 A_2+w_3 A_3 \tag{21}$$

The Eq. (21) can be represented as a neural model (Fig. 4) with two layers where w_1, w_2, w_3 represent the weights of the networks $A_1 = \begin{bmatrix} \hat{\psi}_{rd}(k-1) \\ \hat{\psi}_{rq}(k-1) \end{bmatrix}$, $A_2 = \begin{bmatrix} -\hat{\psi}_{rq}(k-1) \\ \hat{\psi}_{rd}(k-1) \end{bmatrix}$, $A_3 = \begin{bmatrix} i_{sd}(k-1) \\ i_{sq}(k-1) \end{bmatrix}$ are the three inputs to the network. The rotor speed signals were involved in weight w_2.

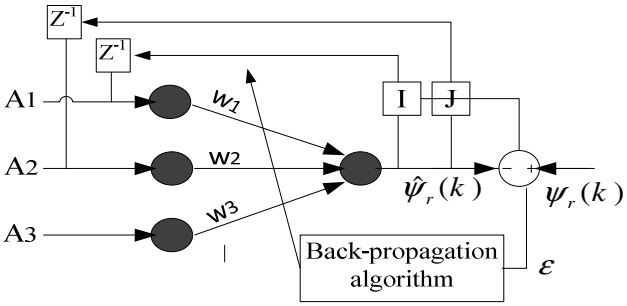


Fig. 4. Two layered neural network model

The total square error on the output layer can be calculated as:

$$E = \frac{1}{2}\varepsilon^2(k) = \frac{1}{2}\left(\psi_r(k) - \hat{\psi}_r(k)\right)^2 \tag{22}$$

Variable weight w_i can be given as follows:

$$\Delta w_i = -\frac{\partial E}{\partial w_i} = -\frac{\partial E}{\partial \hat{\psi}_r(k)}\frac{\partial \hat{\psi}_r(k)}{\partial w_i} = \rho A_i = \left(\psi_r(k) - \hat{\psi}_r(k)\right)^t I \hat{\psi}_r(k-1) \tag{23}$$

where

$$\rho = \frac{\partial E}{\partial \hat{\psi}_r(k)} = \left(\psi_r(k) - \hat{\psi}_r(k)\right)^t \tag{24}$$

either

$$w_i(k) = w_i(k-1) + \eta \Delta w_i(k) \tag{25}$$

with

$$\Delta w_i(k) = -\eta \rho A_i + \alpha \Delta w_i(k-1) \tag{26}$$

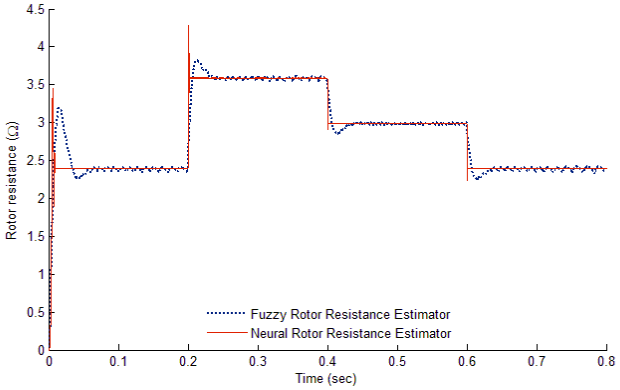
where α determines the effect of past weight changes and η is the training coefficient. The weights are adjusted so as to minimize a square error energy function and the new weight w_i is given by:

$$w_i = (k) = w_i(k-1) - \eta\rho A_i + \alpha\Delta w_i(k-1) \tag{27}$$

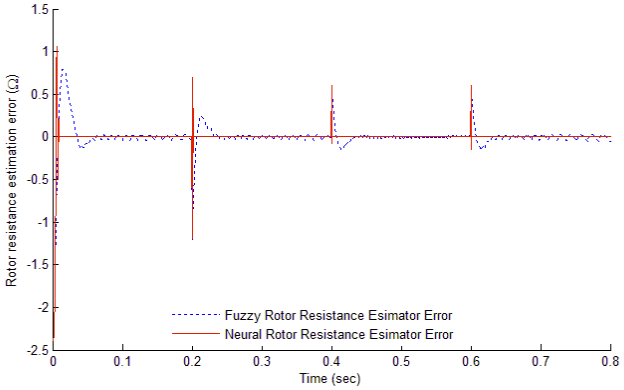
The rotor resistance R_r can be found from either w_1 or w_3 using Eq. (28) or (29):

$$R_r = \frac{L_r w_3}{L_m t_s} \tag{28}$$

$$R_r = \frac{L_r(1-w_1)}{t_s} \tag{29}$$

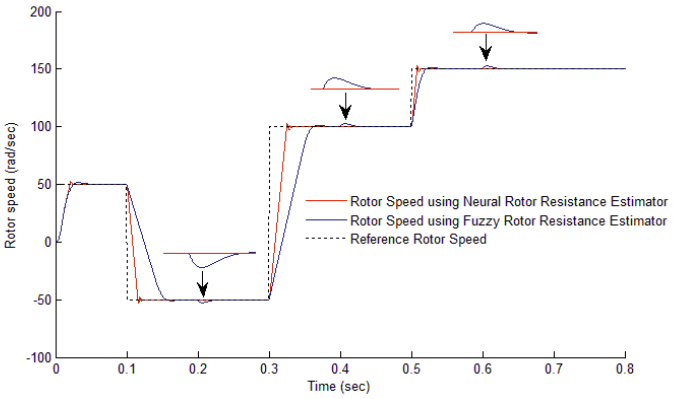


(a)

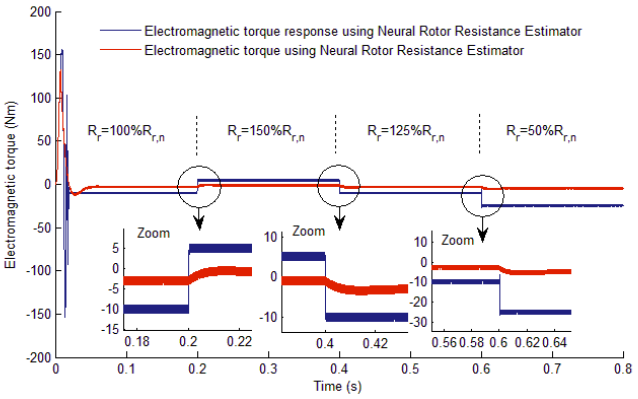


(a')

Fig. 5. Fuzzy rotor resistance estimator compared to neural rotor resistance estimator. (a) rotor speed estimation error; (a') rotor resistance estimation; (b) rotor speed; (c) electromagnetic torque.



(b)



(c)

Fig. 5. (*Continued.*)

Simulation works are carried out to evaluate the performance of the new proposed technique. The simulations are performed using the MATLAB/Simulink simulation package. The parameters and data of the induction motor used for simulation procedure are listed as follows:

P_n = 3kW, V_n = 230V, R_s = 2.89Ω, R_r = 2.39Ω, L_s = 0.225H, L_r = 0.220H, L_m = 0.214H, J = 0.2kg·m^2, n_p = 2.

Figs. 5(a) and 5(a') presents the estimation and error estimation results of fuzzy rotor resistance estimator compared to neural rotor resistance estimator.

The rotor resistance is changed abruptly during steady state operation of the drive. Its value is increased from the nominal value of 2.39Ω to 3.585Ω at 0.2sec, and then decreased to 2.987Ω at 0.4sec, and then decreased to its nominal value at 0.6sec.

This figure shows a good operation of the estimator does not depend, again, the initial value of R_r chosen in the algorithm. This is very important since in reality we do not know the exact value of resistance when the estimation algorithm starts. The convergence of this method is thus confirmed. In reality, the actual rotor resistance varies much more slowly, this means that the estimated rotor resistance can better monitor the actual rotor resistance.

Fig. 5(b) shows the rotor speed variations, the real speeds show a steady state disturbance error before rotor resistance estimation process begins. A short time after estimation process, the disturbance is reduced to zero. The electromagnetic torque is showing in Fig. 5(c), well as the decoupling between the flux and torque is verified.

As can be seen from the comparison, the dynamic response of neural rotor resistance estimator is much better than the fuzzy rotor resistance estimator.

5 Conclusion

The expected results of this research were primarily to improve the performance of drives addressed by conventional methods of indirect rotor flux orientation. This improvement should be achieved by developing a new blueprint indirect more robust with respect to the parameter variation. The basic idea of this command structure is the online correction of rotor resistance variation to maintain the decoupling controller in perfect agreement with the actual conditions of motor operation. Considering the results of the simulation, the proposed neural rotor resistance estimator has better dynamic performance than the fuzzy rotor resistance estimator presented in the literature. The simulation results show the effectiveness of this estimator. The optimal control vector is then obtained and the torque/current is kept at the maximum value corresponding to a given load torque.

References

1. Leonhard, W.: Control of Electrical Drives. Springer (1996)
2. Blaschke, F.: The Principle of Field Orientation as Applied to the New Transvector Closed-Loop Control System for Rotating Field Machines. Siemens Review 34(5), 217–223 (1972)
3. Krause, P.C., Wasynczuk, O., Sudhoff, S.D.: Analysis of Electric Machinery and Drive Systems. Wiley Interscience, John Wiley & Sons, NY (2002)
4. Hasse, K.: On the Dynamics of Speed Control of a Static AC Drive with a Squirrel-Cage Induction Machine. Dissertation for the Doctoral Degree, Darmstadt (1969)
5. Hadj Saïd, S., Mimouni, M.F., M'Sahli, F., Farza, M.: High Gain Observer Based On-Line Rotor and Stator Resistances Estimation for IMs. Simulation Modelling Practice and Theory 19, 1518–1529 (2011)
6. Bartolini, G., Pisano, A., Pisu, P.: Simplified Exponentially Convergent Rotor Resistance Estimation for Induction Motors. IEEE Transactions on Automatic Control 48(2), 325–330 (2003)
7. Kojabadi, H.M.: Active Power and MRAS Based Rotor Resistance Identification of an IM Drive. Simul Modell Practice Theory 17(2), 376–389 (2009)

8. Abbasian, T., Salmasi, F.R., Yazdanpanah, M.J.: Improved Adaptive Feedback Linearization Control of Induction Motors Based on Online Estimation of Core Loss and Rotor Resistance. In: International Symposium on Power Electronics, Electrical Drives, Automation and Motion, SPEEDAM 2006, Taormina-Sicily, Italy, pp. S32/22–27 (2006)
9. Toliyat, H.A., Levi, E., Raina, M.: A Review of RFO Induction Motor Parameter Estimation Techniques. IEEE Trans Energy Convers 18(2), 271–283 (2003)
10. Toliyat, H.A., Wlas, M., Krzemiriski, Z.: Neural-Network-Based Parameter Estimations of Induction Motors. IEEE Transactions on Industrial Electronics 55(4), 1783–1794 (2008)
11. Zidani, F., Nait-Said, M.S., Benbouzid, M.E.H., Diallo, D., Abdessemed, R.: A Fuzzy Rotor Resistance Updating Scheme for an IFOC Induction Motor Drive. IEEE Power Engineering Review 21(11), 47–50 (2001)
12. Zadeh, L.A.: Fuzzy Logic. IEEE Computer Magazine 1(4), 83–92 (1988)
13. Zadeh, L.A.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)
14. Zimmermann, H.J.: Fuzzy Sets, Decision Marking, and Expert Systems, Boston, Dordrecht, Lancaster (1987)
15. Chen, S.-M.A.: Fuzzy Approach for Rule-Based Systems Based on Fuzzy Logics. IEEE Trans. Syst. Man and Cybernetics 26(5), 769–778 (1996)
16. Livingstone, D.J.: Artificial Neural Networks: Methods and Applications. Humana Press Inc. (2009)
17. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the Theory of Neural Computation. Addison-Wesley (1991)

Optimal Design of the Jumping Robot Based on Velocity Directional Manipulability and Dynamic Directional Manipulability

Qi Yang

Institute of Systems Engineering, China Academy of Engineering Physics, Mianyang, China
yangqi1224@163.com

Abstract. Taking velocity directional measure and dynamic directional measure as the jumping performance evaluation indexes, the improvement of jumping performance of the jumping robot is pursued in the point view of mechanism design. On the basis of the jumping robot mechanism model, the kinematic and dynamic equations in take-off phase are established, and the velocity mapping relationship from the joint space to the centroid movement space and the acceleration mapping relationship from the joint driving torque space to the centroid acceleration space are obtained. In order to improve the robot's jumping performance, velocity directional measure and dynamic directional measure combining with the optimal algorithm are used to optimize the mechanism parameters of the jumping robot. Through the calculation of example, the optimal results show that the study on the mechanism parameters optimal design of the jumping robot is feasible by introducing the theory of velocity directional measure and dynamic directional measure.

Keywords: optimal design, jumping robot, mechanism parameters, velocity directional manipulability, dynamic directional manipulability.

1 Introduction

Jumping is an important motion style for many creatures, such as crickets, frogs, and kangaroos. It has clear advantages in terms of the terrain adaptability [1]. For a robot with jumping gait, its motion range can be greatly extended. Jumping robots have good application prospect because these can cross uneven terrains, jump over obstacles and ditches efficiently [2-3]. These properties make them suitable for operations in the inhuman environment and dangerous applications, such as space exploration, archaeology survey, military reconnaissance, geological prospecting, emergency rescue and disaster relief [4]. Such prospect has inspired robotic scientists and engineers to investigate and develop jumping robots by mimicking biological jumping principles of different creatures [5].

Reasonable mechanism design is the fundament to realize the jumping robot. To a new robot model, how to choose mechanism design parameters can make the motion performance better, proposes higher requirements to robot design. But it hasn't formed

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 264–275, 2013.

systemic evaluation indexes for the robot's jumping performance yet. The robot manipulability which represents the global transformation capability of motion and forces has been widely used in the aspects from robot kinematic and dynamic control to robot configuration optimization and other aspects in the study of the manipulator [3-8]. But the research on the mechanism parameters design which influence the manipulator motion performance is relatively little, not to mention the robot's jumping performance.

In this paper, the jumping robot in the ground phase, including take-off and landing, is reasonably simplified and then can be considered as a manipulator; its kinematic and dynamic equations in the take-off phase are established; taking velocity directional manipulability and dynamic directional manipulability as the indexes which evaluate the jumping performance in terms of mechanism inherent characteristics, its mechanism parameters are optimized by applying the indexes, so that the robot's jumping performance in the task orientation has been improved.

2 Mechanism Model of the Jumping Robot

A whole jumping process includes take-off phase, flight phase and landing phase. We focus on the robot's jumping performance in the take-off phase here. The jumping robot studied here is shown in Fig. 1, which is a kind of leg-like jumping robot. It is made of aluminum alloy material and comprised of 4 joints driven by DC servo motors and 4 links with timing belt transmission. Fig. 2 shows the simplified mechanism model of the jumping robot in the take-off phase.

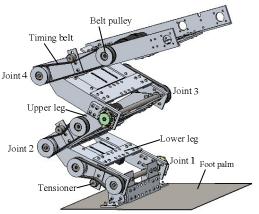


Fig. 1. The jumping robot

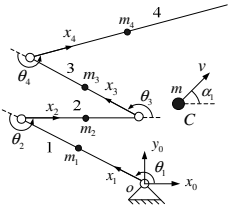


Fig. 2. The mechanism model of the jumping robot in take-off phase

3 Velocity Transmission Capability of the Jumping Robot

3.1 Velocity Manipulability

The robot's velocity manipulability refers to the velocity transmission characteristics along the specified direction under the current configuration state. Under the robot's current position and orientation, if the system with lower joint velocity can make the end-effector obtain larger movement rate along the specified direction, it is considered that the robot's velocity manipulability along this direction is good; if the end-effector can't move or moves with low velocity, it is considered poor [7].

Assuming the robot's end-effector velocity to $\dot{\boldsymbol{x}} \in R^{m\times 1}$, joint velocity to $\dot{\boldsymbol{\theta}} \in R^{n\times 1}$, Jacobian matrix to $\boldsymbol{J}(\boldsymbol{\theta}) \in R^{m\times n}$ in the operation space, for robot with redundant DOF $m < n$, the robot's kinematic equation is

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{1}$$

The unit sphere in the joint space R^n described as

$$\dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{\theta}} = 1 \tag{2}$$

is mapped to an ellipsoid in the operation space described as

$$\dot{\boldsymbol{x}}^T (\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{J}^T(\boldsymbol{\theta}))^{-1} \dot{\boldsymbol{x}} = 1 \tag{3}$$

Eq. (3) is the generalized velocity ellipsoid. When the assumption which the joint velocity in the space R^n is a unit sphere is untenable, the joint velocity need to be normalized [9].

The velocity manipulability w is defined to be

$$w = \sqrt{\det(\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{J}^T(\boldsymbol{\theta}))} \tag{4}$$

According to Eq. (3), the robot's end-effector velocity in R^m forms an ellipsoid with volume $\left\{\pi^{m/2} / \Gamma[(m/2)+1]\right\} w$, where $\Gamma(\bullet)$ is the gamma function. It can be seen that w is proportional to the volume of the velocity ellipsoid.

The velocity manipulability has evaluated the robot's flexibility and measured the robot's overall manipulability. Taking w as the kinematic optimization object is to pursue the isotropic movement, is essentially to pursue the largest velocity ellipsoid volume, or the lengths from the ellipsoid center to every point on the ellipsoid surface as long as possible. Such a same movement requirement in the non-task and task direction not only wastes part of the optimization capability, and is potential to weaken the movement ability in the task direction [7].

3.2 Velocity Directional Manipulability

The velocity of robot's end-effector in the operation space is a vector, but the velocity manipulability only describes the manipulability of the numeric value, its direction is determined by the task. The robot's end-effector velocity can be described by

$$\dot{\boldsymbol{x}} = B\boldsymbol{p} \tag{5}$$

Where, B is a scalar, represents the velocity magnitude; $\boldsymbol{p} = [\cos\alpha_1, \cos\alpha_2, \cdots, \cos\alpha_m]^T \in R^{m\times 1}$ represents the movement direction, $\alpha_1, \alpha_2, \cdots, \alpha_m$ are the angle between the velocity and the axes positive direction respectively. The physical meaning of B is the distance from the ellipsoid center to the ellipsoid surface along direction $\boldsymbol{p}$. It reflects the robot's transmission efficiency between the joint velocity and the end-effector velocity along direction $\boldsymbol{p}$.

Since $\dot{\boldsymbol{x}}$ is a point on the velocity ellipsoid, substituting Eq. (5) into Eq. (3) can obtain

$$(B\boldsymbol{p})^T(\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{J}^T(\boldsymbol{\theta}))^{-1}(B\boldsymbol{p}) = 1 \tag{6}$$

From the above equation, we can obtain

$$B = [\boldsymbol{p}^T(\boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{J}^T(\boldsymbol{\theta}))^{-1}\boldsymbol{p}]^{-1/2} \tag{7}$$

B is defined as the robot's velocity manipulability in direction $\boldsymbol{p}$, that is velocity directional manipulability (VDM). VDM defined above is to measure the velocity transmission capability in the operation space from the point view of the inherent nature of the system.

3.3 Velocity Directional Manipulability of the Jumping Robot

VDM is used to measure the system's velocity transmission capability from the joint space to the operation space of the end-effector. For the jumping robot, the conversion relationship between the joint space and the centroid movement space is established, and VDM is used to measure its velocity transmission capability.

The jumping robot's kinematic model was shown in Fig.2, which describes the mechanism movement using the D-H method, with the fixed coordinate system $o - x_0 y_0 z_0$ setting at the point on the ground where the robot's lower leg contacts. The link lengths of the robot's each part are l_1, l_2, l_3, l_4 respectively; the mass distributions (the ratio of the distance between the centroid and the lower joint to the link length) are a_1, a_2, a_3, a_4 respectively, the masses of each link are m_1, m_2, m_3, m_4 respectively. The radius vector from the robot's centroid to the origin of the coordinate system is set as

$$\boldsymbol{r}_c = [x_c, y_c]^T \tag{8}$$

The radius vectors of each link's centroid are

$$\boldsymbol{r}_{ci} = [x_{ci}, y_{ci}]^T \quad (i = 1,2,3,4) \tag{9}$$

Then the radius vector components of the robot's centroid can be expressed as

$$\begin{cases} x_c = \sum_{i=1}^{4} x_{ci} m_i / m = \sum_{i=1}^{4} x_{ci} \gamma_i \\ y_c = \sum_{i=1}^{4} y_{ci} m_i / m = \sum_{i=1}^{4} y_{ci} \gamma_i \end{cases} \tag{10}$$

Where, $m = \sum_{i=1}^{4} m_i$ is the total mass of the four links, $\gamma_i = m_i / m \quad (i = 1,2,3,4)$ are the ratios of each link mass to the whole mass.

Differentiating Eq. (10) with time can obtain the centroid velocity

$$\dot{\boldsymbol{r}}_c = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{11}$$

Where, $\dot{\boldsymbol{r}}_c = [\dot{x}_c, \dot{y}_c]^T \in R^{2\times 1}$ is the velocity vector of the robot centroid; $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]^T \in R^{4\times 1}$ is the velocity vector of the generalized joint velocity; $\boldsymbol{J} \in R^{2\times 4}$ is the Jacobian matrix of the centroid velocity.

According to Eq. (11), the Jacobian matrix here expresses the mapping relationship between the joint space and the centroid velocity space. Substituting Eq. (11) into Eq. (7) can obtain the velocity manipulability B of the jumping robot in direction $\boldsymbol{p}$.

4 Acceleration Transmission Capability of the Jumping Robot

4.1 Dynamic Manipulability Measure

The dynamics equation in closed form of robot in joint space is generally described by

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{g}(\boldsymbol{\theta}) = \boldsymbol{\tau} \tag{12}$$

Where, $\ddot{\boldsymbol{\theta}} = d^2\boldsymbol{\theta} / dt^2$; $\boldsymbol{M}(\boldsymbol{\theta}) \in R^{n\times n}$ is the positive definite symmetric inertia matrix; $\boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in R^n$ is the centrifugal and Coriolis force vector; $\boldsymbol{g}(\boldsymbol{\theta}) \in R^n$ is the gravity vector; $\boldsymbol{\tau}$ is the joint driving torque vector.

Differentiating Eq. (1) with time can obtain

$$\dot{\boldsymbol{v}} = \boldsymbol{J}\ddot{\boldsymbol{\theta}} + \boldsymbol{a}_r(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{13}$$

Where, $\boldsymbol{a}_r(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}}$. New vectors $\tilde{\boldsymbol{\tau}} \in R^n$ and $\dot{\tilde{\boldsymbol{v}}} \in R^m$ are defined as

$$\tilde{\boldsymbol{\tau}} = \boldsymbol{\tau} - \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - \boldsymbol{g}(\boldsymbol{\theta}) \tag{14}$$

$$\dot{\tilde{\boldsymbol{v}}} = \dot{\boldsymbol{v}} - \boldsymbol{a}_r(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{15}$$

Substituting Eq. (14), Eq. (15) into Eq. (12) and Eq. (13) respectively, after arrangement, we can obtain

$$\dot{\tilde{\boldsymbol{v}}} = \boldsymbol{J}\boldsymbol{M}^{-1}\tilde{\boldsymbol{\tau}} \tag{16}$$

A scale value w_d given by

$$w_d = \sqrt{\det(\boldsymbol{J}(\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{J}^T)} \tag{17}$$

is defined to be the dynamic manipulability measure. Its basic idea is to quantify the degree of the acceleration change under certain joint driving torque constrains according to Eq. (16), so it can be taken as a measure of robot manipulability.

According to Eq. (16), a joint driving torque sphere $\tilde{\boldsymbol{\tau}}^T\tilde{\boldsymbol{\tau}} = 1$ in joint space is mapped to an acceleration ellipsoid in the operation space described by

$$\dot{\tilde{\boldsymbol{v}}}^T(\boldsymbol{M}\boldsymbol{J}^+)^T(\boldsymbol{M}\boldsymbol{J}^+)\dot{\tilde{\boldsymbol{v}}} = 1 \tag{18}$$

Where, $\boldsymbol{J}^+$ is the pseudo inverse of $\boldsymbol{J}$. If the joint driving torques dissatisfies the condition $\|\tilde{\boldsymbol{\tau}}\| \le 1$ or other assumptions, each variable should be normalized [9].

In consideration of the robot dynamics, the dynamic manipulability measure provides a good way to analyze, design and control robot.

4.2 Dynamic Directional Manipulability of the Jumping Robot

The dynamic manipulability measures the dynamic properties of the robot system in its entirety. It describes the numerical size of the dynamic manipulability for each direction of the robot in the operate space. This measure has considered the robot's comprehensive performance with all the possible movement.

But the jumping trajectory was pre-planned for the jumping robot researched in this paper. Therefore, it is necessary to measure the dynamic performance in the task direction of the movement for the robot. Then the dynamic manipulability measure is extended here by introducing the directional manipulability with reference to VDM, and the dynamic performance on the task direction during movement of the jumping robot is measured by the dynamic directional manipulability.

As shown in Fig. 2, the jumping robot mechanism can be considered as a robot manipulator in the take-off phase. Its centroid acceleration in the operation space is a vector which can be expressed as a scalar and a vector product format, noted as

$$\dot{\boldsymbol{v}}_c = A\boldsymbol{d} \tag{19}$$

Where, A is a scalar, denotes the size of the acceleration; $\boldsymbol{d} = [\cos\varphi_1, \cos\varphi_2]^T \in R^2$ is the acceleration direction of the robot centroid; φ_1, φ_2 are the angles between the acceleration direction and the positive direction of x, y axis respectively. A represents the distance from the center of the dynamic manipulability ellipsoid to the ellipsoid surface along direction $\boldsymbol{d}$ as shown in Fig. 3. It reflects the transmission efficiency between the joint driving torque and the robot centroid acceleration in the task direction. And it is a measurement index to measure the robot dynamic performance in the task direction.

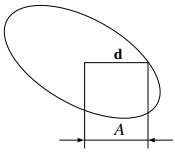


Fig. 3. The schematic diagram of the dynamic directional manipulability

Because that $\dot{\boldsymbol{v}}$ is a point on the dynamic manipulability ellipsoid, substituting Eq. (19) into Eq. (18) ($\dot{\tilde{\boldsymbol{v}}} = \dot{\boldsymbol{v}}$, when $\dot{\boldsymbol{\theta}} = \boldsymbol{0}$) can obtain

$$(A\boldsymbol{d})^T (\boldsymbol{M}\boldsymbol{J}^+)^T (\boldsymbol{M}\boldsymbol{J}^+)(A\boldsymbol{d}) = 1 \tag{20}$$

From Eq. (20), A can be expressed by

$$A = [\boldsymbol{d}^T (\boldsymbol{M}\boldsymbol{J}^+)^T (\boldsymbol{M}\boldsymbol{J}^+)\boldsymbol{d}]^{-1/2} \tag{21}$$

which is defined to be the dynamic directional manipulability (DDM) of the jumping robot along direction $\boldsymbol{d}$ of the acceleration vector $\dot{\boldsymbol{v}}_c$ in the operate space. It can be explained as the size measurement of the reachable acceleration in the given task orientation in the operation space, under certain constraints on the joint driving torque.

A is the function of joint generalized coordinates and mechanism parameters of the jumping robot, it depends entirely on the inherent property of the robot itself. Thus from the viewpoint of the robot system's inherent property, DDM measures the acceleration transmission capability of the jumping robot in the task direction in operation space.

5 Mechanism Parameters Optimal Design of the Jumping Robot Based on VDM & DDM

The jumping robot does a free parabolic movement during the flight phase of the jumping process, so the jump height and length depends on the size and orientation of the centroid velocity at the take-off time. After the task direction of the jump is determined, when the velocity manipulability becomes maximum, the robot's velocity transmission characteristics is best, namely the larger movement rate along the task

direction can be obtained. Therefore, VDM can be used as an evaluation index to measure the jumping performance and to seek the mechanism parameters to optimize the jumping performance, namely to jump higher and farther under given jumping posture and direction.

In the process of take-off, the better the joint torque transmission of the jumping robot performance is, the larger ground reaction force impulse can be obtained [5], thus the better the robot jumping performance can be reached. Eq. (6) shows that under certain driving torque constraints, the joint torque transmission performance can be converted to the degree of the acceleration change through the mapping relation. According to the pre-planned jumping trajectory, the centroid acceleration of the jumping robot achieves maximum during the initial time of the movement, which implies that joint motors should provide considerable torques. Then from the viewpoint of DDM, the mechanism parameters optimization aims to maximize the centroid acceleration under the initial posture and the constraints and motion conditions to improve the robot jumping performance, by adjusting the mechanism parameters to change the mapping relation between the joint driving torque and the centroid acceleration under the joint driving torque constraints in the pre-planned trajectory.

In this paper, with comprehensive consideration of the kinematics and dynamics of the jumping robot, VDM and DDM are taken together as the jumping performance evaluation indexes to expect to better improve the jumping robot's performance.

5.1 Mechanism Parameters Optimal Model

The jumping trajectory is set to be the process from the initial posture $\boldsymbol{\theta}_0$ at t_0 time to the take-off posture (instantaneous before take-off) $\boldsymbol{\theta}_t$ at t time. According to the simplified model of the jumping robot as shown in Fig. 2, the mechanism parameters optimization of the jumping robot based on VDM & DDM can be expressed as:

$$
\begin{aligned}
\max \quad & F(l_i, a_i, m_i) = \omega_1 A^2(l_i, a_i, m_i) + \omega_2 B^2(l_i, a_i, m_i) \\
s.t. \quad & 0 < a_i < 1 \\
& l_{i\min} \le l_i \le l_{i\max} \\
& m_{i\min} \le m_i \le m_{i\max} \\
& \sum_{i=1}^{4} l_i = l \\
& \sum_{i=1}^{4} m_i = m \\
& a_{c0y} / a_{c0x} = \tan\varphi_1, a_{c0} = [a_{c0x}, a_{c0y}]^T \\
& J_2\dot{\theta}_t / (J_1\dot{\theta}_t) = \tan\alpha_1, J = [J_1, J_2]^T \\
& a_{ct} = [0, -g]^T
\end{aligned}
\tag{22}
$$

Where, $i = 1,2,3,;\omega_1$ and ω_2 represent DDM and VDM index weights respectively; $m_{i\min}$ and $m_{i\max}$ are the minimum and maximum mass of link i respectively; the constraint $\sum_{i=1}^{4} m_i = m$ represents that the total mass of four links is invariant; a_{c0x}, a_{c0y} are components of the centroid acceleration along the axis direction x and y respectively at the initial time, and the constraint $a_{c0y} / a_{c0x} = \tan\varphi_1$ represents that the pre-planned centroid acceleration direction is satisfied; the constraint $\boldsymbol{J}_2\dot{\boldsymbol{\theta}}_t / (\boldsymbol{J}_1\dot{\boldsymbol{\theta}}_t) = \tan\alpha_1$ represents that the pre-planned centroid velocity direction is satisfied; $\boldsymbol{a}_{ct}$ is the robot centroid acceleration at the instantaneous time before take-off from the ground, and $a_{ct} = [0, -g]^T$ is the condition to take-off from the ground.

5.2 Calculation and Analysis of Example

From the pre-planned jumping trajectory, the known conditions under the take-off initial posture at t_0 time are as below.

Jonit angle: $\boldsymbol{\theta}_0 = [165^\circ, -150^\circ, 150^\circ, -150^\circ]^T$;

Joint angular acceleration: $\ddot{\boldsymbol{\theta}}_0 = [-168.64, 109.15, -95, 116.65]^T \text{ rad/s}^2$;

Centroid acceleration direction : $\boldsymbol{d} = [\cos 63^\circ, \cos 27^\circ]^T$.

And the known conditions under the posture before take-off from the ground at t time are as below.

Jonit angle: $\boldsymbol{\theta}_t = [130^\circ, -117^\circ, 129^\circ, -119^\circ]^T$;

Joint angular acceleration: $\ddot{\boldsymbol{\theta}}_t = [33.84, -23.6, 23.57, -22.44]^T \text{ rad/s}^2$;

Centroid velocity direction: $\boldsymbol{p} = [\cos 59^\circ, \cos 31^\circ]^T$.

The mechanism parameter constrains are as below.

Mass constrains: $m = 2.85\text{kg}$, $m_{1\min} = 0.35\text{kg}$, $m_{1\max} = 0.55\text{kg}$, $m_{2\min} = m_{3\min} = 0.5\text{kg}$, $m_{2\max} = m_{3\max} = 0.7\text{kg}$, $m_{4\min} = 1.1\text{kg}$, $m_{4\max} = 1.3\text{kg}$;

Link length constrains: $l = 0.7125\text{m}$, $l_{1\min} = 0.0625\text{m}$, $l_{1\max} = 0.1625\text{m}$, $l_{2\min} = l_{3\min} = 0.1\text{m}$, $l_{2\max} = l_{3\max} = 0.2\text{m}$, $l_{4\min} = 0.25\text{m}$, $l_{4\max} = 0.35\text{m}$.

Substituting the known conditions into Eq. (22), $\omega_1 = 0.02$ and $\omega_2 = 0.98$ are chosen by taking the relative importance and dimensions of DDM and VDM into account. Using Matlab to programme and invoking the optimization toolbox to solve the problem, the optimal results are obtained as shown in Table 1; motion parameters as shown in Table 2. Where, v_{ct} is the centroid velocity at the instantaneous time jumping off the ground, and a_{c0} is the centroid acceleration at the take-off initial moment.

Fig. 4 to Fig. 9 show the posture change, the centroid velocity change and the centroid acceleration change during take-off phase of the jumping robot under the initial mechanism parameters and optimization mechanism parameters respectively. For clear expression of the posture, only 6 postures are selected evenly during the movement.

Table 1. Mechanism parameters optimal results based on VDM & DDM

	a_1	a_2	a_3	a_4	m_1 / kg	m_2 / kg
Initial design	0.5	0.5	0.5	0.5	0.45	0.6
Optimal design	0.002	0.003	0.998	0.552	0.516	0.529
	m_3 / kg	m_4 / kg	l_1 / m	l_2 / m	l_3 / m	l_4 / m
Initial design	0.6	1.2	0.1125	0.15	0.15	0.3
Optimal design	0.699	1.1	0.134	0.188	0.2	0.295

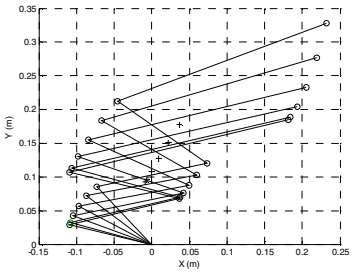


Fig. 4. Posture change under the initial mechanism parameters

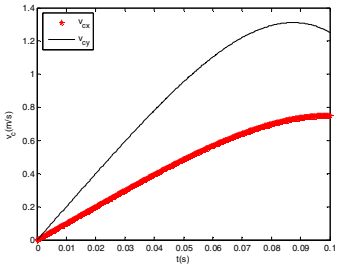


Fig. 5. Centroid velocity change under the initial mechanism parameters

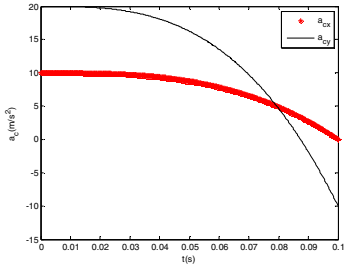


Fig. 6. Centroid acceleration change under the initial mechanism parameters

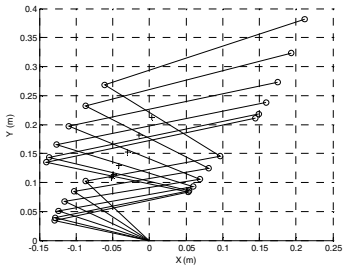


Fig. 7. Posture change under the optimization mechanism parameters

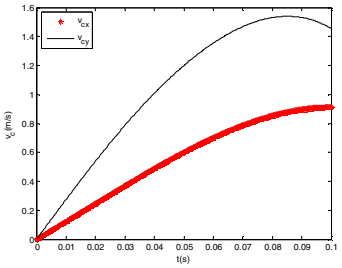


Fig. 8. Centroid velocity change under the optimization mechanism parameters

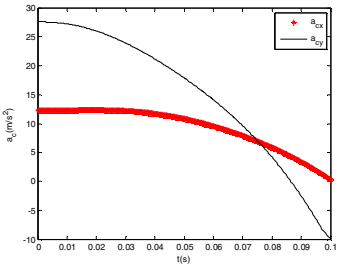


Fig. 9. Centroid acceleration change under the optimization mechanism parameters

Table 2. Motion comparison of the initial design and the optimal design

	v_{ct} (m/s)	a_{c0} (m/s^2)
Initial design	1.46	22.36
Optimal design	1.72	30.19

Table 2 shows that the optimal results obtained by taking VDM & DDM as the optimization object are significantly better than the initial design. The centroid velocity is faster than the initial design by 17.8%, and the centroid acceleration increases 35% under the optimal design. It demonstrates that the mechanism parameters optimal design based on VDM & DDM comprehensively improves the velocity transmission capability and the dynamic manipulability of the jumping robot.

6 Conclusions

In this paper, the jumping robot is taken as the research object, the mechanism model is simplified; the kinematic and dynamic indexes of the jumping robot at the take-off phase are established; the mechanism parameters are optimized by using DDM & VDM indexes. Through the calculation and analysis of the example, the following conclusions can be drawn:

1. VDM is adopted to measure the velocity transmission performance in the task direction through establishing the mapping relationship from the joint space to the centroid motion space.
2. By establishing the transmission relationship between the joint driving torque and the centroid acceleration, DDM is proposed to measure the acceleration transmission performance in the task direction.
3. Taking VDM & DDM as objective, the jumping robot mechanism parameters are optimized by simulation, thus the mechanism inherent property can be fully utilized and the robot jumping performance is improved.
4. The application of the velocity manipulability and the dynamic manipulability theory is extended in this paper. Two evaluation indexes are presented to measure the performance of the jumping robot, which provide a certain reference value to the design and performance analysis of the jumping robot.

References

1. Niiyama, R., Nagakubo, A., Kuniyoshi, Y.: Mowgli: A Bipedal Jumping and Landing Robot with an Artificial Musculoskeletal System. In: International Conference on Robotics and Automation, pp. 2546–2551. IEEE Press, Piscataway (2007)
2. Gao, H.: Dynamic Optimization and Design of One-legged Hopping Robot with Biologically Mimic Kangaroo, pp. 25–35. College of Mechanical and Electrical Engineering, North China University of Technology, Beijing (2008) (in Chinese)
3. Zhang, Q.: Research on the Locomotion Characteristics of the Jumping Behavior for the Bionic Cricket Robot, pp. 37–47. School of Mechanical Engineering, Jiang Nan University, Wuxi (2009) (in Chinese)
4. Wang, M., Zang, X., Fan, J., et al.: Meng Wang, Xizhe Zang, Jizhuang Fan, et al: Biological Jumping Mechanism Analysis and Modeling for Frog Robot. Journal of Bionic Engineering 5(3), 181–188 (2008)
5. Xu, Z., Lv, T., Wang, X., et al.: Motion Optimization for Jumping Robot Based on Inertia Matching Manipulability. Journal of Shang Hai Jiao Tong University 42(7), 1154–1158 (2007) (in Chinese)
6. Yamakita, M., Omagari, Y., Taniguchi, Y.: Jumping Cat Robot with Kicking a Wall. Journal of the Robotics Society of Japan 9, 934–938 (1994)
7. Yao, J., Ding, X., Zhan, Q., et al.: On Task-based Directional Manipulability Measure of Redundant Robot. Robot. 22(6), 501–505 (2000) (in Chinese)
8. Chen, G., Yang, A.: Direction-oriented Manipulability Measures of Dual-arm Robots. Robot. 18(2), 108–114 (1996) (in Chinese)
9. Yoshikawa, T.: Dynamic Manipulability of Robot Manipulators. International Journal of Robotics Research 4, 3–9 (1985)

Dynamic Torque Control Incorporating Tracking Differentiator for Motor-Driven Load Simulator

Kang Chen, Hang Guo, Li Sun, and Jie Yan

College of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China

Abstract. Widely used in the static/dynamic stiffness test of aircraft actuation systems, Motor-driven load simulator (MDLS) simulates the aerodynamic load and exerts the load on actuation system. MDLS endures position coupling disturbance from active motion of actuation system, and this inherent disturbance is called extra torque. However, to eliminate the influence of extra torque is the key issue regarding to the MDLS controller design, as the extra torque may degrade the performance of MDLS seriously. A compound torque control algorithm based on tracking differentiator (TD) is proposed for MDLS in this paper. This algorithm reflects the essential characteristics of MDLS and guarantees transient tracking performance as well as final tracking accuracy. In detail, firstly, the mathematical models of MDLS are derived, and the influence of the extra torque is also studied. Tracking differential filter is then utilized to identify the actuator's velocity, acceleration and jerk, which can compensate the extra torque. Finally, based on the structural invariability theory, a compound controller is developed, which consists of forward path corrector, feed-forward controller and differential tracking filter. Simulation and experiment comparative results are obtained to verify the high-performance nature of the proposed control strategy; the tracking accuracy and bandwidth are greatly improved.

Keywords: flight simulation, electric load simulator, tracking differentiator, compound control.

1 Introduction

Motor-driven load simulator (MDLS) is a widely used hardware-in-the-loop-simulation (HILS) assembly in flight control system development, which could simulate the air load executed on aircraft actuation systems [1]. Applying MDLS to verify the control performance of aircraft actuation systems is gradually becoming a basic flow in the development of aircraft actuation systems.

In order to simulate aerodynamic load, MDLS is usually connected to the actuation system directly with rigid shaft coupling. It is obvious that MDLS inevitably endures strong position coupling disturbance from active rotary motion of actuation system [3]. This disturbance is called extra torque, and it is an inherent disturbance which may even be bigger than the desired torque output sometimes. There are plenty of literatures available which focus on designing appropriate mechanical structure or compensation controller to eliminate extra torque. The basic idea of mechanical

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 276–287, 2013.

structure optimization [4] is to decouple torque servo and position tracking by introducing a synchronizing motor between the system base and loading motor. The stator of synchronizing motor is connected to system base and the rotator is connected to loading motor [5]. In this system, the stator of loading motor is applied to function as position tracking, while the rotator is used to realize torque tracking. However, the disadvantage of this structure is that it complicates the system and also increases the cost. For compensation controller, constant structure control and velocity synchronization are widely used in application of load simulator. Constant structure control was designed by Liu [6], whose idea was feed-forward actuator velocity to eliminate extra torque. Since this method depends on velocity measurement, phase delay influence is inevitable. Velocity synchronizing compensation was presented in Jiao, et al.'s book/paper in Ref. [7], which foreknows the actuator motion by collecting its servo command signal and compensated extra torque in advance. Feed-forward and compensation were introduced to eliminate extra torque in electro-hydraulic load simulator. Plummer [8] presented a valve cross-compensation design approach to improve force tracing accuracy, also based on velocity synchronizing control essentially. The aforementioned methods can improve the tracking performance and/or the robustness of MDLS with motion disturbance; however, they ignore the extra torque caused by the acceleration and jerk of the actuator. Thus, these methods could hardly meet the demand of the testing requirement, particularly when the smart-compact aircraft actuator moves at high frequency. Inexact loading will lead to the degradation of the testing confidence level.

Besides, there are many robustness improvement methods considering actuator movement as disturbance. Nam and Hong [2] proposed a robust force control system using quantitative feedback theory designed for loading system to enhance the robustness of force control system. Chantranuwathana and Peng [9] presented a modular adaptive robust control technique to improve the force control performance of vehicle active suspensions. To tackle with parametric uncertainties in MDLS, quantitative feedback theory (QFT) method [10], self-tuning proportional-integral-derivative (PID) control [12] and neural-net-work-based methods [13-14] are widely employed. Combined with a fuzzy PID controller, a self-tuning grey predictor was proposed by Truong and Ahn [11] to improve the robustness of force control of hydraulic load simulator. However, one disadvantage of this method is that the time consuming is relatively high.

MDLS has a number of characteristics which complicate the development of high-performance closed-loop controllers, including the extra torque, actuator friction and uncertainties. Therefore, the mathematical model of MDLS is firstly derived in this paper. Base on the system model, the sources causing of the extra torque is also discussed. With the use of actuator's velocity, acceleration and jerk, the idea of feed-forward compensation is employed to eliminate the extra torque. A differential tracking filter is introduced to identify the actuator's velocity, acceleration and jerk. A compound torque controller is then constructed to guarantee the tracking bandwidth as well as final tracking accuracy of the MDLS.

2 Dynamic Models and Problem Formulation

The function of MDLS is to simulate aerodynamic load of aircraft rudder for ground testing. In hardware-in-the-loop-simulation process, the actuation system is controlled by onboard flight control computer while the relevant aerodynamic loading torque is calculated by simulation computer according to the height, speed and posture of aircraft. MDLS is designed to exert this loading torque to actuation system.

Generally, in order to simulate aerodynamic load as shown in Fig. 1, in which the left side is the actuation system (i.e., position servo system) and the right side is loading system (i.e., torque servo system), MDLS is connected to actuation system directly. It is obvious that MDLS and actuation system are two mutually coupling systems. For actuation system, this position servo system is required to follow desired angle trajectory in the presence of external disturbances, therefore loading torque can be treated as external disturbance acting on the position closed-loop system. On the other hand, torque servo system needs to exert desired loading torque on actuation system when actuator operates. Moreover, active motion of actuator can be considered as an external disturbance to the torque closed-loop system. To make the torque output track any specified torque trajectory as accurately as possible is regarded as good MDLS design. In this designed system, the torque feedback (torque sensor) and rudder motion position (angle encoder) are available.

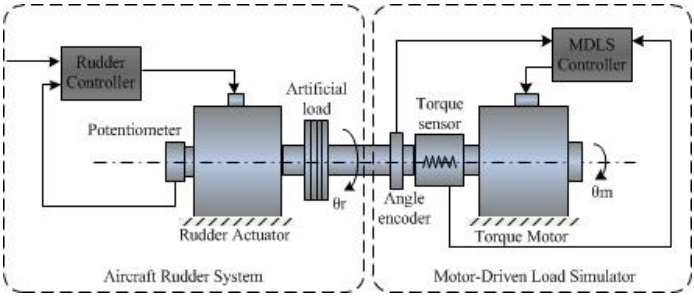


Fig. 1. Architecture of motor-driven load simulator

2.1 Dynamic Models of MDLS

For MDLS, the torque motor can be treated as inductance, resistance of string attached, as shown in fig 2.

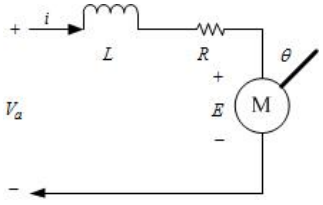


Fig. 2. Simplified model of torque motor

The voltage balance equation is

$$V_a = L\frac{di}{dt} + iR + V_{emf} \tag{1}$$

where i, R, L is the current, resistance, inductance of the armature respectively, V_a is the voltage of the motor windings, and V_{emf} is the back EMF, given by

$$V_{emf} = C_e \phi n = K_e \dot{\theta}_m \tag{2}$$

where K_e is the electro-motive-force (EMF) constant, and $\dot{\theta}_m$ is the motor rotation speed. Meanwhile, the output torque of motor is

$$T = C_T \phi i = K_m i \tag{3}$$

where K_m is the torque constant, and the torque balance equation of motor can be given by

$$T = K_m i = J_m \ddot{\theta}_m + B_m \dot{\theta}_m + T_f \tag{4}$$

where J_m is total inertia of motor rotator and artificial load, B_m is combined coefficient of damping and viscous friction on the load, T_f is loading torque.

In addition, torque sensor is a key component which connects actuation system and load simulator. Since its inertia is very small, it can be considered as an elastic model:

$$T_f = (\theta_m - \theta_r) G_c \tag{5}$$

where θ_r is the rudder position, G_c is total stiffness coefficient of torque sensor. From Eq.(1)-(5), the open-loop transfer function of MDLS can be given by

$$T_f = \frac{G_c [K_m V_a - N(s) \cdot \dot{\theta}_r]}{D(s)} \tag{6}$$

where the coefficients are

$$D(s) = J_m L \cdot s^3 + (J_m R + B_m L) \cdot s^2 + (B_m R + L G_c + K_e K_m) \cdot s + G_c R \tag{7}$$

$$N(s) = J_m L \cdot s^2 + (J_m R + B_m L) \cdot s + (B_m R + K_e K_m) \tag{8}$$

Eq.(6) points that the open-loop transfer function of MDLS consist of two basic components, including the forward path transfer function $G_c \cdot K_m / D(s)$,and the extra torque transfer function $G_c \cdot sN(s) / D(s)$. It is indicated that the torque load that MDLS exerts on the rudder is not only the function of the voltage which drive the torque motor, but also the function of the velocity, acceleration and jerk of the rudder.

2.2 Analysis of the Extra Torque

From Eq.(8) , it is known that the extra load is produced by the velocity, acceleration and jerk of the rudder. In order to analyze the influence of these parameters to the extra torque, three variable are defined in this paper, given by

Velocity disturbance:

$$d_v(s) = \frac{G_c \cdot (B_m R + K_e K_m) \dot{\theta}_r}{D(s)} \tag{9}$$

Acceleration disturbance:

$$d_a(s) = \frac{G_c \cdot (J_m R + B_m L)\ddot{\theta}_r}{D(s)} \tag{10}$$

Jerk disturbance:

$$d_{aa}(s) = \frac{G_c J_m L \dddot{\theta}_r}{D(s)} \tag{11}$$

Using these three variables, each extra torque component can be calculated. It is oblivious that extra torque is a function of the movement frequency of the rudder, thus a simulation block can be designed to analyze these three extra torque components, as shown in fig .3.

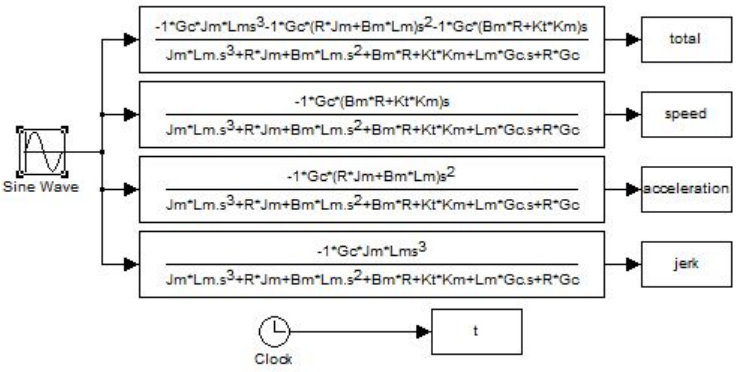


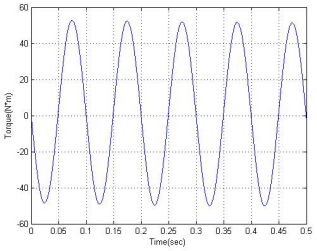
Fig. 3. Extra torque analysis block diagram

In fig 3, four channels represent the total extra torque, extra torque caused by Velocity, extra torque caused by acceleration and extra torque caused by jerk respectively. Table 1 shows the torque motor parameters and MDLS parameters that used in the simulation.

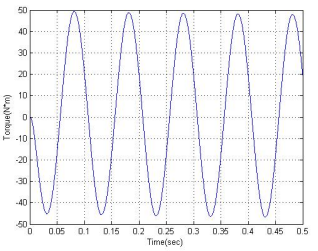
Table 1. MDLS parameters

Parameters	Value	Parameters	Value
$J_m / (kg \cdot m^{-3})$	0.12	$B_m / (N \cdot m \cdot s / rad)$	0.232
$L_m / (H)$	5.0×10^{-3}	$K_t / (N \cdot m \cdot A)$	7.835
$R_m / (\Omega)$	4.23	$K_m / (V \cdot s / rad)$	8.556
$G_c / (N \cdot m / ^o)$	100		

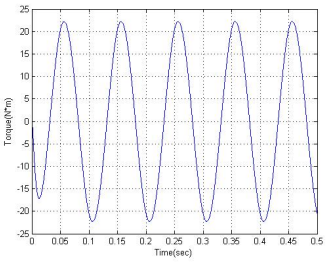
The simulation is completed in Matlab, the movement angle of the rudder is set to 1 degree and the movement frequency is set to 10Hz, 20Hz. the simulation results are shown in fig.4.



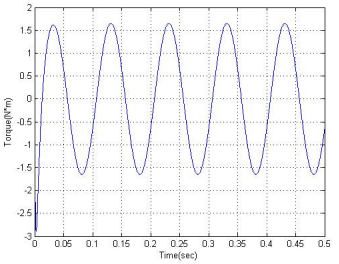
(a) Total extra torque(10Hz)



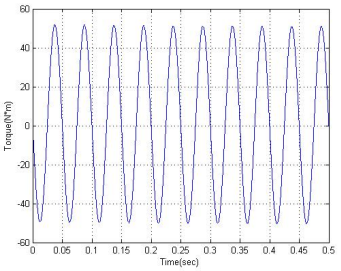
(b) Extra torque caused by velocity(10Hz)



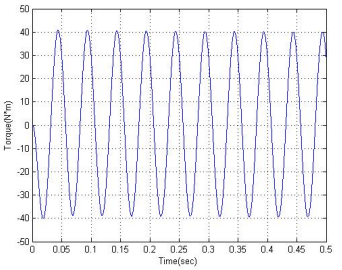
(c) Extra torque caused by acceleration(10Hz)



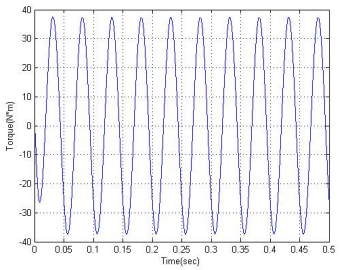
(d) Extra torque caused by jerk(10Hz)



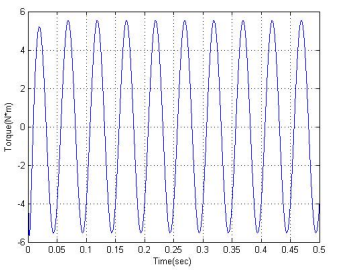
(a) Total extra torque (20Hz)



(b) Extra torque caused by velocity (20Hz)



(c) Extra torque caused by acceleration (20Hz)



(d) Extra torque caused by jerk (20Hz)

Fig. 4. Extra torque simulation results

As shown in fig.4, the extra torque has following features:

1. Three extra torque components have different phases, and may counteract each other.
2. The extra torque caused by velocity of the rudder is the main component of the total extra torque.

3. With the increasing movement frequency, the extra torque caused by the acceleration and jerk increases obliviously, and in order to realize high track performance, these three extra torque component must be compensate.

3 Compensator and Compound Controller Design

Considering a MDLS in the form of Eq. (6), the desired load torque is given as:

$$T_c = G_l \theta_r \tag{12}$$

where G_l is the load grad. The objective is to design a controller, which can eliminate the extra torque and output torque as closely as possible to the command torque.

3.1 Extra Torque Compensator Based on Tracking Differentiator (TD)

The standard form of Tracking Differentiator is given by

$$\begin{cases} \dot{v}_1 = v_2 \\ \dot{v}_2 = -r \times sat(v_1 - v + \dfrac{v_2 |v_2|}{2r}, \delta) \end{cases} \tag{13}$$

where

$$sat(x, \delta) = \begin{cases} sign(x), |x| \geq \delta \\ x / \delta, |x| \leq \delta \end{cases} \tag{14}$$

In Eq.(13),Variable $v_1(t)$ will track the input signal $v(t)$ as fast as possible under the limit of acceleration r, $v_2(t)$ is the differential of $v_1(t)$, and can also be treated as the differential of $v(t)$.

The Tracking Differentiator is actually some kind of signal process method, which can track the differential signal of the input signal with little phase lag, and make the transient process smoother. The ability of extracting differential signals can be used to obtain an ideal transient process and improve stability of the system.

The final discrete expression of the Tracking Differentiator is shown below:

$$\begin{cases} v_1(k+1) = v_1(k) + h r_2(k) \\ v_2(k+1) = v_2(k) + r_v sat(v_1(k) - v(k), v_2(k), \delta, h) \end{cases} \tag{15}$$

where h_v is the step length, r_v is the tracking gain, v_1 is the output of tracking, v_2 is the output of tracking differential; h_v can be selected as control cycle, and r_v should be designed appropriately to ensure the rapidity and stability of the track.

In order to design the compensator to eliminate the extra torque, the velocity, acceleration and jerk signal of the rudder must be obtained. But in the MDLS, only the rudder position can be measured. According to the theory above, a series tracking differentiator can be used to get these signals, as shown in fig 5.

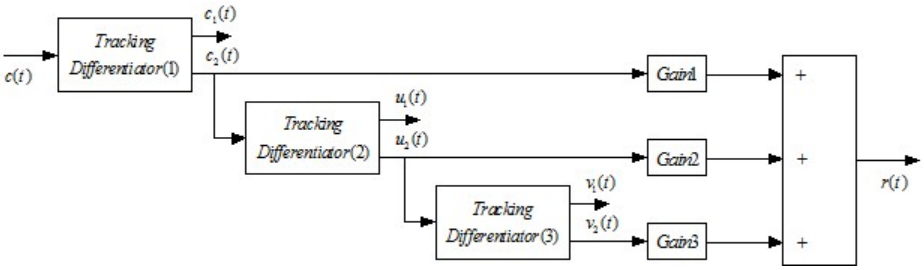


Fig. 5. Series Tracking Differentiator block diagram

In fig.5, $C(t)$ is the input position signal of the rudder, Tracking Differentiator 1 will output the velocity signal $C_2(t)$, meanwhile $C_2(t)$ is the input signal of Tracking Differentiator 2, and the acceleration signal will be output, Tracking Differentiator 3 input the acceleration signal and output the jerk signal. With these three signals and appropriate gain coefficients, an extra torque compensator is completed. The Tracking Differentiator's tracking outputs and differential signals generalized by sine wave signal input are illustrated by the fig 6.

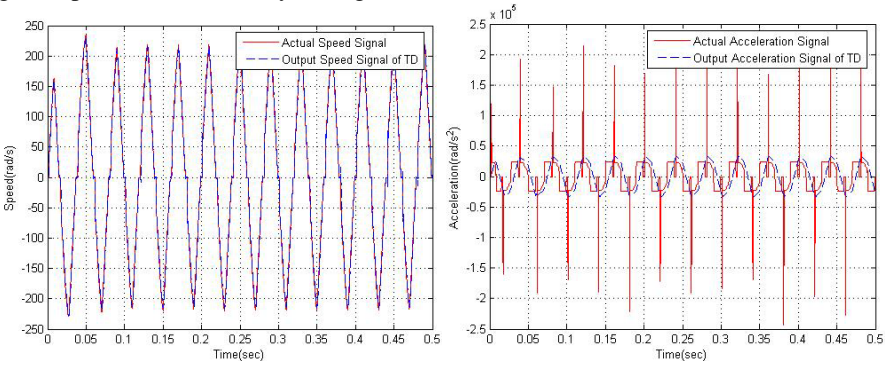


Fig. 6. The comparison of Tracking Differentiator's output and actual signal

3.2 Compound Controller Design

Based on the structural invariability theory, the MDLS controller is composed by three parts: extra torque compensator, command torque feed-forward and PID controller. Illustrated by fig 7:

As shown in fig 7, the TD compensator is used to eliminate the extra torque caused by the movement of the rudder, which is described is section 3.1. $v(t)$ is the command torque, and is induced to the controller directly, so the system rapidity can be improved. The PID controller part is used to eliminate the error between the command torque and the actual torque. It should note that another tracking differentiator

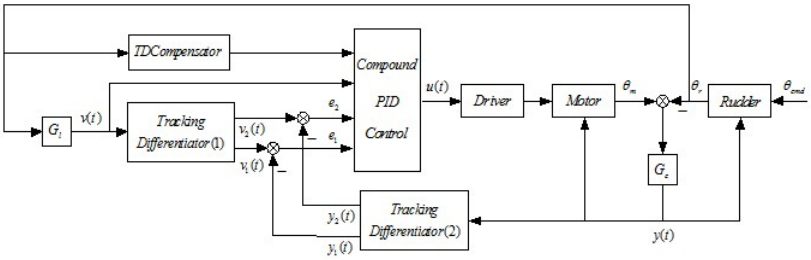


Fig. 7. MDLS compound controller block diagram

is set in the actual torque feedback loop, this tracking differentiator has the actual torque as the input signal, and also outputs two signals: actual torque and its differential signal. The differential signal is induced into the compound controller to realize the differential front PID controller, with its help, the stability of the system can and the feed-forward gain of the controller be improved, resulting in the improvement of the tracking performance. And the final control law is given as

$$u = k_p e_2 - k_d e_1 + k_i \int_0^t e_2 dt + k_{cmd} v_{cmd} + k_{vff} v_r + k_{aff} a_r + k_{jff} j_r \tag{16}$$

where e_2 is the torque error, e_1 is the differential signal of torque error, v_{cmd} is the torque command, v_r, a_r and j_r are the velocity ,acceleration and jerk signals of the rudder generated by TD respectively.

4 Application Example

The proposed compound controller is implemented on a MDLS. The goal is to illustrate the effectiveness of the controller, coping with extra torque and also achieving desire torque tracking in a practical MDLS. Then comparative experimental results are carried out as follows.

4.1 Experiment Setup

As shown in Fig. 8, a direct drive rotary torque motor IDAM-RI13 by IDAM is used as drive component in this MDLS system and is driven by a Metronix digital servo amplifier ARS 2310. A torque sensor AKC-17B with measurement range of 300 Nm is used to measure the loading torque. A Heidenhain rotary encoder ECN125, with Heidenhain PC counter card IK220, is used to measure the rotary displacement of rudder. A 16 bit AD card PCI-9114 by AdLink is used to sample torque signal and a 16 bit DA card PCI-6208 to send out control voltage. The original designed real-time control program is based on RTX real-time operating system and the servo control cycle is 0.5ms. VisualStudio is applied to build the UI program.

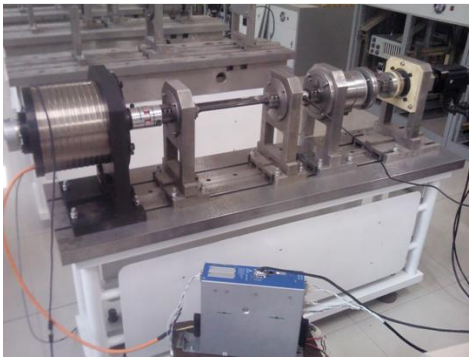


Fig. 8. Experimental setup of MDLS

According to MDLS parameters in Table 1 and control performance of practical experiment, the compound controller parameters are chosen, as shown in table 2:

Table 1. Controller parameters in MDLS

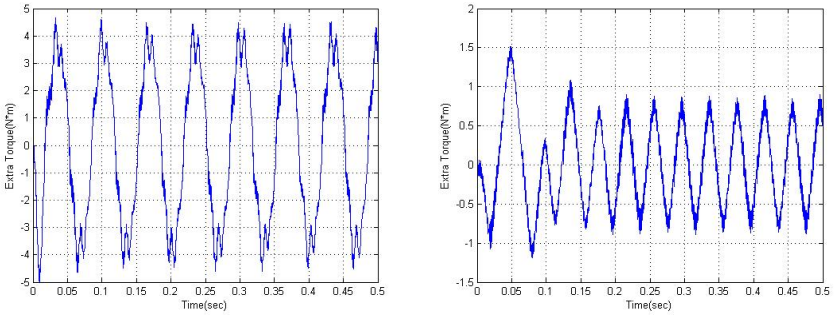
Parameters	Value
Kp	0.3
Ki	0.07
Kd	0.02
K_{tff}(command feed-forward coefficient)	0.12
K_{vff}(velocity feed-forward coefficient)	0.02
K_{aff}(accelerationfeed-forward coefficient)	0.016
K_{iff}(jerk feed-forward coefficient)	0.012
h_v(compensator filter time constant)	0.008
r_v(tracking gain)	0.09

In the following text, two typical experiments for MDLS are carried out.

4.2 Eliminating Extra Torque Experiment

As the ability to eliminate extra torque is a key issue for MDLS, the measurement of the exact extra torque is then very important. An aviation rudder is used in this experiment and its shaft is connected to MDLS by a coupling. Assume MDLS desired loading torque T_{cmd}=0 and the actuator operates with the desired sinusoidal angle command, then the measured torque output is extra torque.

With the help of the angle encoder, the actual rudder angle signal is easy to obtain. The angle signal is input into the TD compensator, then the estimated actuator velocity, acceleration and jerk can be obtained. Set the desired angle trajectory of rudder system as $\theta_r = 1.0\sin(157.08t)$, which represents the rudder moves at 25Hz. Firstly, MDLS operates with only velocity feed-forward, then the TD compensator algorithms, which are mentioned in Section 3.1, is used to eliminate extra torque, respectively. The comparison between two control strategies is shown in Fig. 9.



(a) Extra torque using velocity feedforward (b) Extra torque using TD

Fig. 9. Extra torque comparison in experiment of eliminating extra torque

According to the final tracking error, velocity feed-forward control can only eliminate about 87% of extra torque while TD compensator can eliminate about 98%. Clearly seen, TD compensator has a better performance in eliminating the extra torque in MDLS.

4.3 Gradient Loading Experiment

Gradient loading is the most common use of MDLS, in which loading torque command is proportional to the rudder actual angle. For example, the desired command of rudder is $\theta_r = 1.0\sin(157.08t)$, and the torque gradient coefficient is gl=10N.m/rad, thus the desired loading torque can be calculated by the formula $T_c = G_l \cdot \theta_r$, as shown in Fig 10, and torque tracking errors are shown in Fig 11.

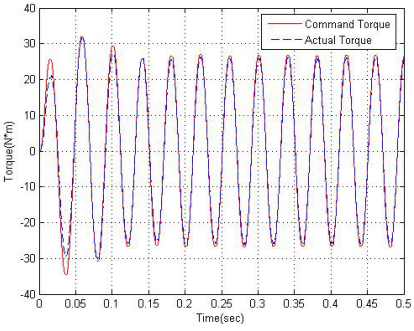


Fig. 10. Torque command and torque outputs at 25Hz

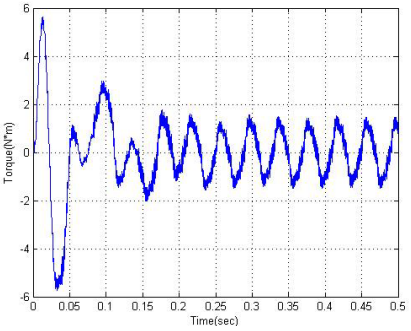


Fig. 11. Tracking errors

The experimental results indicate that the compound controller proposed in this paper is very effective in gradient loading testing for rudder, and the controller can achieve perfect tracking performance for MDLS under coupling disturbance from rudder.

5 Conclusion

In this paper, instead of many previous linear controls for load simulator, a compound torque control has been proposed based on a nonlinear TD compensator in which the velocity, acceleration and jerk of the rudder can be obtained. These three rudder parameters are taken into account by the proposed controller to eliminate the extra torque. Uncertain nonlinearities such as un-modeled friction forces, external disturbance and ignored high-frequency dynamics are effectively handled via certain robust feedback and compound controller. The controller achieves a good transient performance and grantees final tracking accuracy for the output tracking. Extensive simulation and experiment results are obtained for an MDLS test rig to verify the high-performance nature of the proposed controller.

References

1. Jiao, Z.X., Gao, J.X., Hua, Q., et al.: The velocity synchronizing control on the electrohydraulic load simulator. Chinese Journal of Aeronautics 17(1), 39–46 (2004)
2. Nam, Y., Hong, S.: Force control system design for aerodynamic load simulator. Control Engineering Practice 10(5), 549–558 (2002)
3. Li, H.: Development of hybrid control of electrohydraulic torque load simulator. Journal of Dynamic Systems Measurement and Control 124(3), 415–419 (2002)
4. Zhou, J.: Control strategy of the synchrodrive electro-hydraulic servo system. In: Proceedings of International Conference on Fluid Power, pp. 365–367 (2006)
5. Wu, S.L.: Study on control of twin-hydraulic motor synchrodrive system. Chinese Journal of Mechanical Engineering 6(3), 165–170 (1993)
6. Liu, C.N.: The optimized design theory of hydraulic servo system. Metallurgical Industry Press, Beijing (1989) (in Chinese)
7. Yao, J.Y., Shang, Y.X., Jiao, Z.X.: The velocity feed-forward and compensation on eliminating extra torque of electro-hydraulic load simulator. In: Proceedings of the Seventh International Conference on Fluid Power Transmission and Control, pp. 462–465 (2009)
8. Plummer, A.R.: Control techniques for structural testing: a review. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 221(12), 139–167 (2007)
9. Chantranuwathana, S., Peng, H.: Adaptive robust force control for vehicle active suspensions. International Journal of Adaptive Control and Signal Processing 18(2), 83–102 (2004)
10. Mare, J.C.: Dynamic loading systems for ground testing of high speed aerospace actuators. Aircraft Engineering and Aerospace Technology 78(4), 275–282 (2006)
11. Truong, D.Q., Ahn, K.K.: Force control for hydraulic load simulator using self-tuning grey predictor—fuzzy PID. Mechatronics 19(2), 233–246 (2009)
12. Kanellakopoulos, I., Kokotovic, P.V., Morse, A.S.: Systematic design of adaptive controllers for feedback linearizable systems. IEEE Transactions on Automatic Control 36(11), 1241–1253 (1991)
13. Yang, Y.S.: Direct robust adaptive fuzzy control (DRAFC) for uncertain nonlinear systems using small gain theorem. Fuzzy Sets and Systems 151(1), 79–97 (2005)
14. Yang, Y.S., Zhou, C.J.: Adaptive fuzzy H∞ stabilization for strict-feedback canonical nonlinear systems via backstepping and small-gain approach. IEEE Transactions on Fuzzy Systems 13(1), 104–114 (2005)

A Simulation Study of Radial Slits Pressure Regulator for Hydrogen Gas

Takahiro Mizuno[1], Chongho Youn[1], Yoshinari Nakamura[2], and Toshiharu Kagawa[1]

[1] Precision and Intelligence Laboratory, Tokyo Institute of Technology,
4259 Nagatsuta, Midori-ku, Yokohama-shi, Kanagawa, Japan
youn.c.aa@m.titech.ac.jp
[2] KYB Corporation, Mechanical Component Engineering Sect.,
12-1 Asamizo-dai 1-chome Minami-ku, Sagamihara-shi, Kanagawa, Japan

Abstract. Noise generation during decompressing hydrogen in fuel cell vehicles is an issue requiring technological solution. This paper studies noise reducing effect and the flow characteristic of radial slits installed in a pressure reducing valve. Simulation is carried out to estimate the pressure–flow characteristic of the radial slit as a noise reducing mean. The simulation indicates a decrease of Reynolds number in the downstream-side of the radial slits structure, which causes noise reduction. Moreover, we proposed installing radial slits to the downstream of a conventional valve. Noise reduction by this new design was verified with a series of experiments. As a result, the valve with radial slits has a noise reduction effect of about 12.1 dB (A) compared with the conventional valve.

Keywords: Radial Slits, Hydrogen Gas, Pressure Regulator, CFD.

1 Introduction

Currently, around 80% of the energy consumed globally is generated from fossil fuels. Developing countries such as China and India are predicted to have large increases in their demand for energy. Because of a shortage of fossil fuels is predicted in the near future, hydrogen is expected to be an important new energy source. In Japan, NEDO basic technology development program for safe hydrogen utilization was launched in 2003 for research on the earlier hydrogen stations and fuel cell vehicles [1]. Vehicle manufacturers released a joint statement and promised commercial supplies of these vehicles, with larger cruising distances than current electrical vehicles by 2015 [2].

Fuel cell vehicles charge their tanks with hydrogen, that is pressurized up to 35 MPa. For generating electrical power and propulsion, the hydrogen is depressurized to less than 1 MPa and reacted with oxygen in the air. A loud noise is also generated when hydrogen is depressurized. Fuel cell vehicles have no combustion engines, which are loud; therefore, the loud noise of hydrogen depressurization is uncomfortable for drivers. As a result, there is a need for quieter pressure regulators.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 288–297, 2013.

Several theoretical and experimental analyses have been conducted on the generation of aerodynamic noises [3-4]. Those studies focused on methods for noise containment such as designing optimal forms of valve plugs and bases, wrapping the pipes with sound-deadening materials, or using cellular metals to absorb sonic energy [5-9].

Because of these factors, the authors proposed a radial flow structure with radial slits between parallel circular plates [10]. Radial slits are valves composed of two circular parallel plates that are proximately coupled (compared to their radiuses); the air flows radially between the plates. By letting a compressible fluid pass through thin channels, a viscosity resistance is applied, which minimizes turbulence and shock-waves and accordingly reduces noise. We analyzed the flow and noise-generation characteristics of thin radial slits with a width of 50 μm, as the depressurizing structure, and confirmed that the slits, can reduce noise by more than 35 dB compared to orifices [11-12]. We also proved that radial flow channels have a 6 dB noise-reduction effect compared to parallel channels.

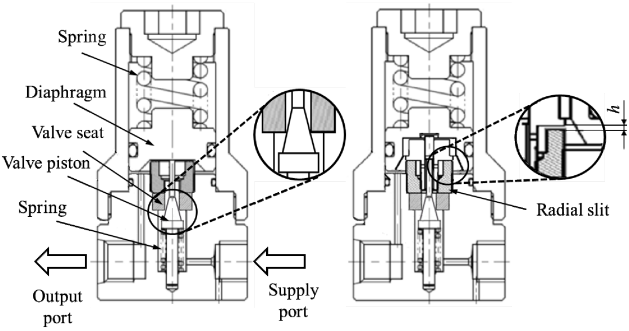
In this paper, we analyze the flow field within the radial slits by simulation, and find out the relationship between their physical and flow characteristics. Additionally, we develop a low-noise pressure regulator for high pressure hydrogen by applying a radial slit structure. We then perform experiments to compare the noise generated by the low noise regulator with that generated by a conventional regulator.

2 Design of Radial Slits Valve

2.1 Conventional Regulator

Figure 1 shows a conventional regulator and the same regulator with a radial slits (RSV) downstream. The conventional valve in Fig. 1(a), is usable up to 35 MPa(G), and has a downstream pressure range of 0.01-1.0 MPa, with the maximum flow rate being 500 L/min. The valve is made from stainless steel (SUS316L) because hydrogen embrittlement is less likely to occur, and the steel is more resistant to corrosion than are other metals such as iron. The length of the conventional valve is 101 mm, and its width is 53 mm. The maximum displacement of the piston is 0.5 mm. The valve base is designed as right-angle-shaped, as shown in the enlarged view of Fig. 1(a). The air flows from the inlet on the right part to the outlet on the left through the valve base. The flow is controlled by adjustment spring, which is compressed by the preset pressure and in turn is adjusted by a screw placed in the upper part of the regulator.

The valve in Fig. 1(b) has radial slits placed downstream of the conventional regulator.



(a)Conventional Type (b) Radial Slit Type

Fig. 1. Structure of Valves

2.2 Radial Slit Structure

Figure 2 and Figure 3 show the structure of the radial slits installed in the regulator. The slits are composed of two parts. The lower half of Part 1 is threaded and is designed to be coupled with Part 2. The outer part of Part 2 is also threaded and is designed for attaching the radial slit unit to the regulator. Very shallow radial trenches are cut in Part 2, which from slits when Part 2 is coupled with Part 1. Hydrogen flows through the radial slits as shown by the arrows in the left side of Fig. 2.

2.3 Simulation

The slit volume for CFD analysis is shown in Fig. 4. The inner diameter of slit is 4 mm, the outer diameter is 8 mm, the angle is 30°, and the thickness of the gap is 0.1 mm. Although radial slits with thicknesses of 0.2, 0.3, and 0.4 mm were also studied experimentally, only the 0.1 mm gap slit was analyzed, because it had the largest pressure drop.

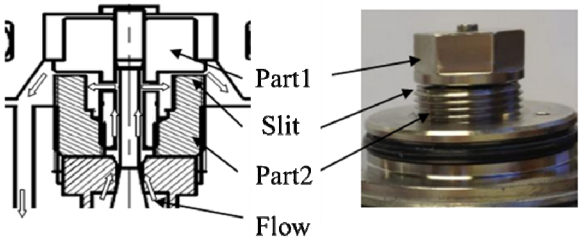


Fig. 2. Cross Section and Side View of Radial Slits Structure

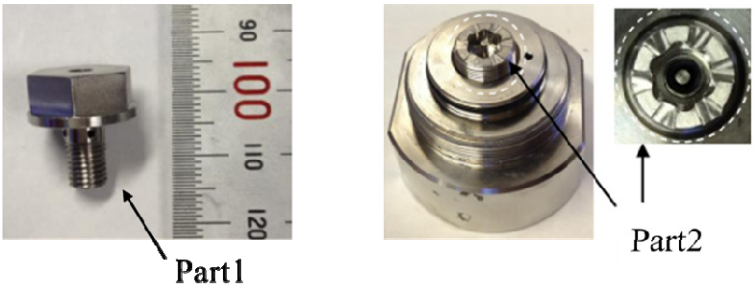


Fig. 3. Photos of Radial Slits Structure

ANSYS FLUENT software was used for this analysis. The flow was modeled as a three dimensional stationary compressible flow, and the k-epsilon model was used for turbulent flow. The explicit difference method was used to consider the generation of shock waves. Because only one of the six slits shown in Fig. 3 was analyzed, the boundary conditions were set as one-sixth of the flow rates of 100, 200, 300, 400, and 500 L/min at the inlet, and as a pressure of 0.6 MPa(G) at the outlet.

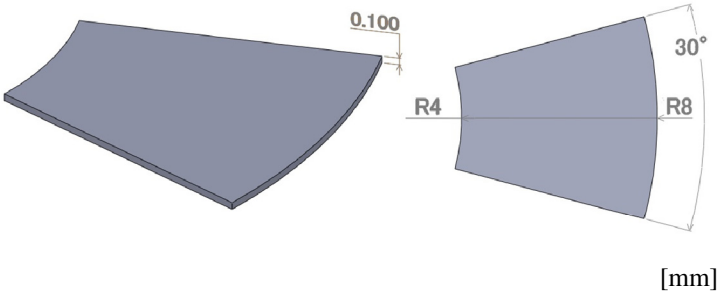


Fig. 4. Volume for CFD Analysis

The pressure-flow characteristics of the radial slits are shown in Fig. 5. Even if the flow rate is 500 L/min, the difference between the upstream and downstream pressures is as little as around 0.2 MPa(G). This proves that most of the regulator function is from the upstream valve, and that radial slits have little effect on the pressure-flow characteristics of the valve.

The pressure and flow velocity distributions from the case where the flow rate was set to 400 L/min (nor), are shown in Figs. 6 and 7, respectively. The pressure and velocity of the center line are shown in Figs. 8 and 9 respectively. The velocity is reduced within the flow, and is speculated to be influenced by viscous friction. As a result, the turbulence is reduced, and the noise is suppressed.

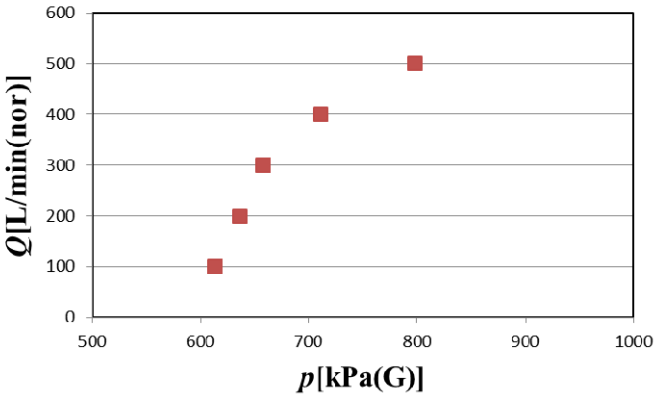


Fig. 5. PQ Characteristics

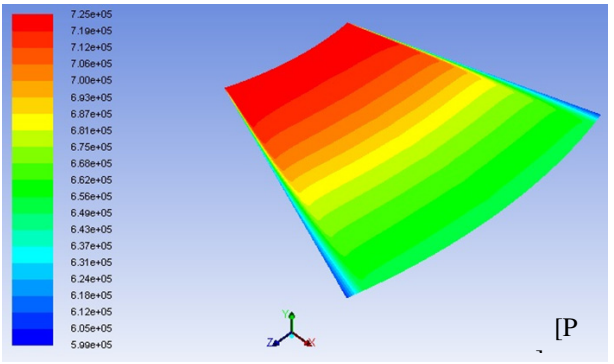


Fig. 6. Pressure Distribution [400L/min(nor)]

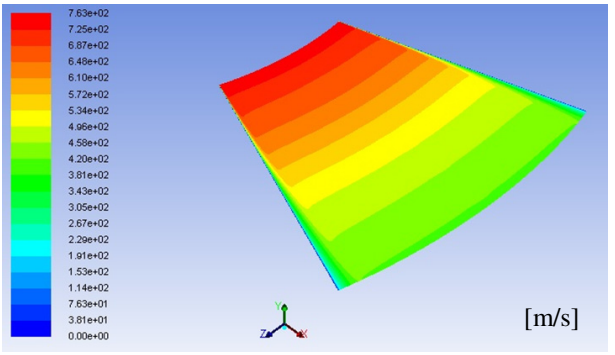


Fig. 7. Velocity Distribution[400L/min(nor)]

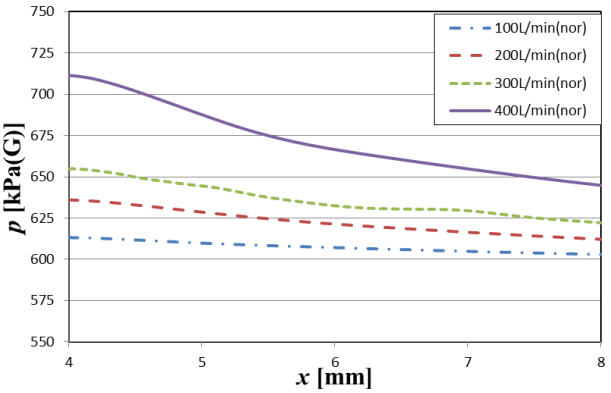


Fig. 8. Pressure of The Center Line

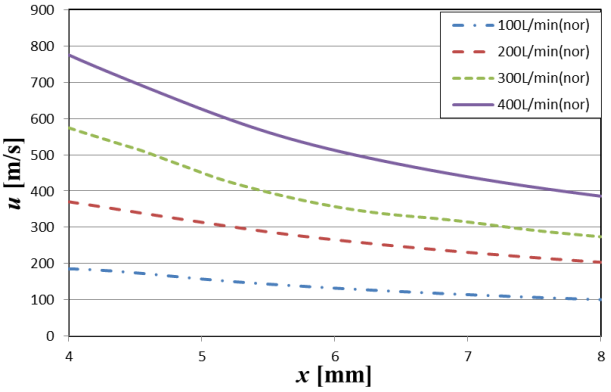


Fig. 9. Velocity of The Center Line

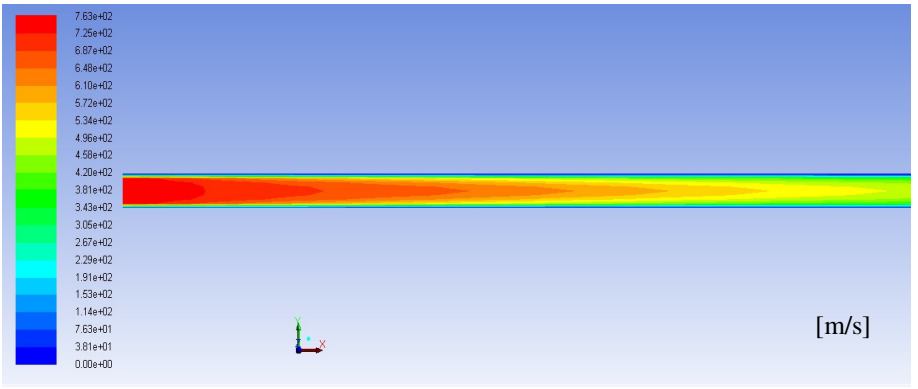


Fig. 10. Velocity Distribution [400L/min(nor)]

The flow velocity in the height direction is shown in Fig. 10. And the Reynolds number about 400 L/min (nor) is shown in Fig. 11. On the inlet boundary, the Reynolds number is around 9,400 and on the outlet boundary, it is around 5,000. This proves that the turbulence effect is reduced in the radial slits. In addition, the decrease in Reynolds number shows that the noise is decreasing as a result of the radial slit.

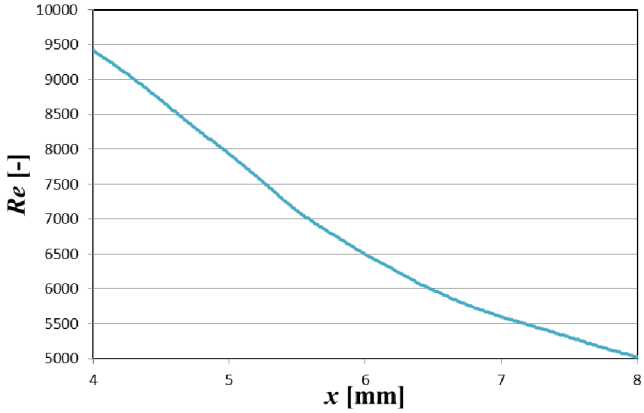


Fig. 11. The Reynolds Number [400L/min(nor)]

3 Experimental Study

3.1 Experimental Rig

For the experiments in this study, it was assumed that the pressure regulators were actually used in a fuel cell vehicle. Therefore, it was necessary to use high-pressure hydrogen [about 35 MPa (G)]. Therefore, experiments were conducted using the facilities of the Hydrogen Energy Test and Research Center (HyTReC) in Fukuoka [13].

Figure 12 shows a circuit diagram of the experiment. The test valve is installed in an explosion resistant cover. In addition, so as not to measure the outside noise, we covered the test valve with a sound absorbing wall; this allowed us to measure only the noise of the test valve.

For ease in switching between experiments with various valves, six valves were built in to the system: the conventional pressure reducing valve, four Radial slits valves, and one spare. Figure 13 shows the position of the test valves. The interval between the test valves is set at 100 mm, with the distance corrected after measuring, and the sound pressure level is defined as the average noise level of the two microphones. The noise measurement was carried out by the each type of the Radial slits valve shown in Table1.

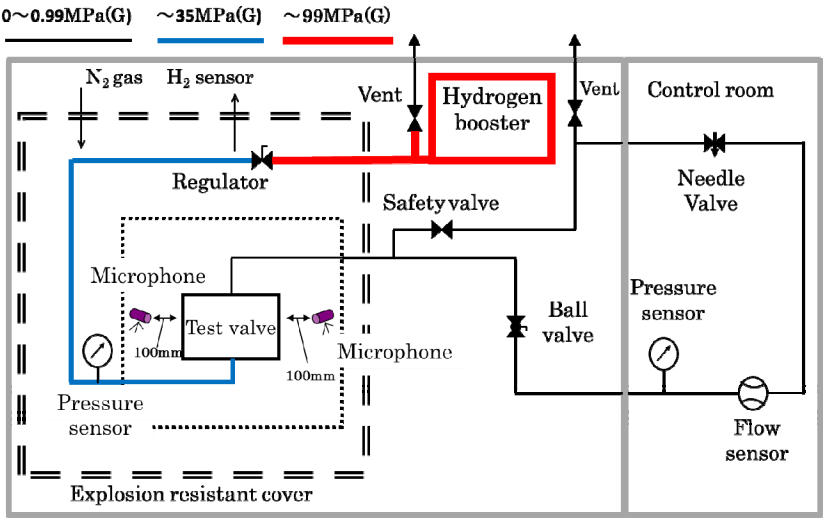


Fig. 12. Experimental Setup for The Valve Test

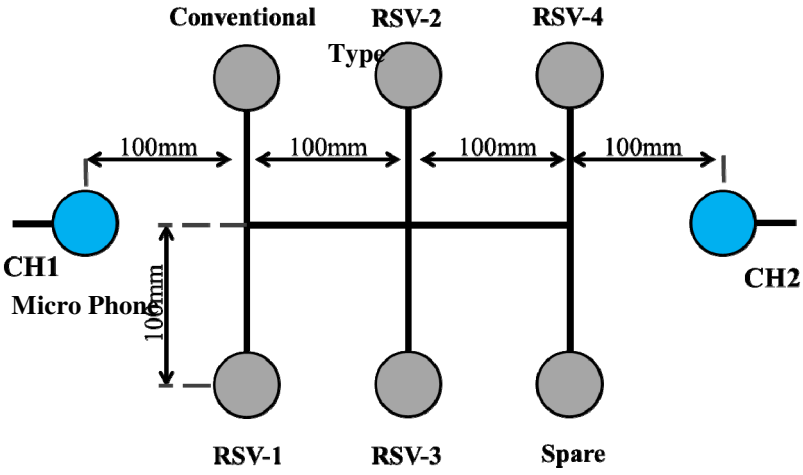


Fig. 13. Positions of The Valve

Table 1. Type of Radial Slits Valve

Type	h [mm]	A_{in} [mm^2]
RSV - 1	0.1	1.28
RSV - 2	0.2	2.57
RSV - 3	0.3	3.85
RSV - 4	0.4	5.14

3.2 Noise Level Results

Figure 14 shows the noise results for each valve at 300 L / min (nor). The results indicate that noise is reduced with each of the four RSVs, as compared with the conventional type valve alone. Further, the noise reduction increases as the height of the slit gap decreases. It is thought that viscous resistance increases as the gap size decreases, reducing the air turbulence and shock waves that are the cause of the noise. As a result, when the slit gap was smallest h = 0.1 mm, the RSV was confirmed to have a silencing effect of 12.1 dB (A).

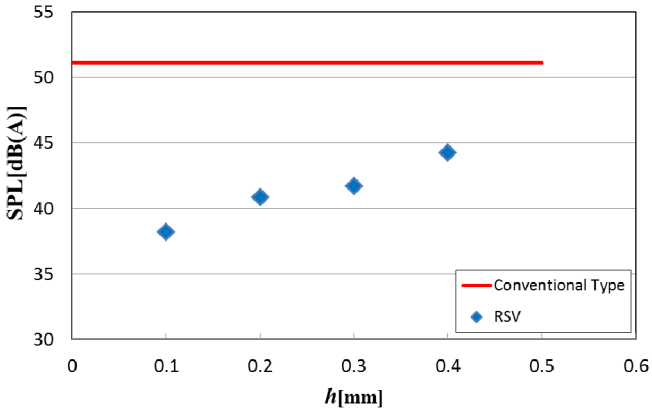


Fig. 14. Sound Pressure Level

4 Conclusions

In this study, we developed a low noise pressure reducing valve for high pressure hydrogen, by adding radial slits structure to the downstream side of a conventional valve. Because the radial slits have a very narrow gap, we expect a low noise effect.

As described in this paper, we used FLUENT to simulate the flow in a radial slit, and obtained the following results.

1. The pressure loss in the radial slit unit is very small relative to that across the whole pressure reducing valve; therefore, radial slits have little effect on the pressure-flow characteristics of the valve.
2. Reynolds number decreased in the radial slit structure, which means that the turbulence effect was reduced at the outlet of the slit. In addition, we conducted a noise experiment on four radial slit valves (RSV) with different gap sizes. The results of the experiment showed that the most effective RSV had a sound reduction effect of about 12.1 dB(A) compared with the conventional type valve.

References

1. NEDO: Infrastructure technology development, such as safe use of hydrogen (2003) (in Japanese), http://www.nedo.go.jp/activities/ZZ_00265.html
2. Ministry of Economy Trade and Industry: Joint Announcement by 13 Japanese companies (2011) (in Japanese), http://www.meti.go.jp/press/20110113003/20110113003.html
3. Lighthill, M.J.: On Sound Generated Aerodynamically - I. General Theory. Proceedings of the Royal Society of London (Series A) 211(1107), 564–587 (1952)
4. Nishimura, N.: Aerodynamic Noise Reducing Technique by Flow Control. The Japan Society of Fluid Mechanics 20, 221–230 (2001) (in Japanese)
5. Sugiyama, K., Doi, S., Ogasawara, T.: Air Flow Noise Generated by a Nozzle with a Sudden Enlargement in Cross-Section. Transactions of the Japan Society of Mechanical Engineers (Series B) 49(443), 1420–1426 (1983)
6. Nakano, M., Ota, E., Tajima, K.: Aerodynamic Study on Structure of Generating Noise and Vibration in High Pressure Gas Valves - I. The Japan Society of Mechanical Engineers (Series B) 45(391), 350–360 (1979) (in Japanese)
7. Nakano, M., Tajima, K., Kumaido, K.: A Study on the Improvement of Noise Suppression in a Low-noise Valve of the Multiple Type- I. Transactions of the Japan Society of Mechanical Engineers (Series B) 54(500), 907–991 (1988)
8. Nishimura, M., Kudo, T., Nishioka, M.: Aerodynamic noise reducing techniques by using pile fabrics. Fluid Dynamics Research 42 (2010)
9. Nishimura, M., Fukatsu, S.: Development of low-noise valve use porous metal. Transactions of the Japan Society of Mechanical Engineers (Series A) 920(78), 431–433 (1992)
10. Kagawa, T., Sekita, K., Murayama, K., Naitou, K.: Control valve, 10-169792
11. Youn, C., Muramatsu, H., Kawashima, K.: Research by visualization of Low Noise Pressure Reducing Structure using Slit Flow. The Japan Fluid Power System Society 36(3), 59–65 (2005) (in Japanese)
12. Youn, C.: Research on the pneumatic system pressure reduction by a slit-type structure,a doctor thesis. Tokyo Institute of Technology (2004) (in Japanese)
13. Hydrogen Energy Test and Research Center (HyTReC) (in Japan), http://hytrec.jp

Symbiotic Simulation of Assembly Quality Control in Large Gas Turbine Manufacturing

Xiangrui Meng, Linxuan Zhang, and Mian Wang

National CIMS Engineering Research Center, Department of Automation,
Tsinghua University, Beijing, China
mxr11@mails.tsinghua.edu.cn, lxzhang@mail.tsinghua.edu.cn,
wang-m05@mails.tsinghua.edu.cn

Abstract. Assembly quality control is a vital problem in large gas turbine manufacturing. Assembly quality may not meet the requirements though each manufactured part's precision is within its range. The paper discusses the assembly quality analysis and control in large gas turbine manufacturing. An intelligent assembly quality control solution using symbiotic simulation is proposed, which combines ON-line Simulation Module (ONSM) with OFF-line Simulation Module (OFFSM). Its mechanisms and partial techniques are discussed. In the end, a measurement simulation of a corporation's gas turbine using *Virtual Assembly Supported System* (VASS) is realized, which verifies the mechanisms of the solution.

Keywords: assembly quality control, symbiotic simulation, assembly process, tolerance analysis, point cloud, virtual training.

1 Introduction

Large gas turbine needs complex assembly technology in product development. The quality of assembly process has a particularly vital effect on large gas turbine's performance, efficiency, safety and service life. For example, the parts' deformation by gravity during assembly process cannot be ignored. The dimension control, assembly process optimization are key factors that affect large gas turbine's parameters during assembly process. Now related enterprises have proposed strict operating procedures for assembly process of large gas turbine. But the assembly process still costs much and brings challenges for assembly process operators. Therefore, necessary measures should be taken to optimize the assembly process, and technical training should be enhanced to improve assembly operators' skills.

Researches on assembly quality control can be divided into two kinds of efforts: one is traditional method, i.e., strict lifecycle quality management on product design, manufacturing, testing, repairing, etc.; another is innovation method, i.e., new technology such as simulation as a tool to analyze and supervise assembly procedures and to train operators. And some of the researches focus on large gas turbine manufacturing. Aguilar, et al. [1] analyze the error sources and influences on the dimensional

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 298–309, 2013.

quality control of the whole production of car body. As a result, they describe an optimization procedure for the system by experimental and simulation tests. Asha, et al. [2] propose a non-dominated sorting genetic algorithm (NSGA), which can find the best combination to get the minimum clearance variation in selective assembly. Su, et al.[3] deal with operator error problem systematically in the copier assembly. They propose two defect-rate prediction models, called Hinckley model and Shibata model, which are helpful to the improvement of assembly quality. Zhang [4] carries out finite element model calculation and simulates the rotor's deformation statically and dynamically. Deformation control technologies are proposed in casings assembly, rotor lifting and rotor alignment processes, etc.

Apart from the traditional methods, researchers also propose new simulation methods or even new simulation paradigm. Wu, et al.[5] build the assembly parallel control system (APCS) for aerospace complex precision mechanical systems, which can ensure the dynamic characteristics of the target system. Huang, et al.[6] develop a generic, agent-based, symbiotic simulation system architecture and mechanisms to support dynamic coupling between the simulation and physical system. And they conduct pilot case studies of the simulation framework. The symbiotic simulation focuses on the close relationship between simulation system and physical system [7], and it is suitable for analyzing large, complex, and uncertain systems. There are some concepts which are relevant to symbiotic simulation, such as on-line simulation, real-time simulation, hardware-in-loop simulation and Dynamic Data Driven Application Systems [8] (DDDAS, proposed by NSF). The symbiotic simulation emphasizes the relationship between the physical system and the simulation system [7], while the on-line simulation doesn't emphasize it and its adaptive ability is not strong enough. The hardware-in-loop simulation is different from on-line simulation because the simulation models in the computer usually don't correspond to the hardware, and the hardware is just a part in the loop. The DDDAS is an advanced simulation concept for complex systems and the symbiotic simulation is compatible with it. In large mechanical equipment industry such as large gas turbine, techniques only using a single type of simulation are usually not enough for assembly quality control. This is because factors effecting assembly quality are various and complex: some of them can be considered by assembly process planning during off-line simulation but the others can hardly be predicted.

This paper focuses on assembly quality control using symbiotic simulation, which combines ON-line Simulation Module (ONSM) with OFF-line Simulation Module (OFFSM). The complementary features of ONSM and OFFSM are united so that assembly quality can be controlled more effectively and efficiently. In Section 2, the architecture of the solution will be introduced. The techniques used in OFFSM or ONSM will be different or expandable in this architecture, and some of them will be discussed in Section 3. In Section 4, we will introduce a case study that reflects the mechanisms of symbiotic simulation. Finally, we summarize the paper in Section 5.

2 Solution of Assembly Quality Control

2.1 Classification of Large Gas Turbine Quality Problems

Factors that influence assembly quality are various, complex and sometimes unpredictable. Generally speaking, they can be classified into four categories: products,

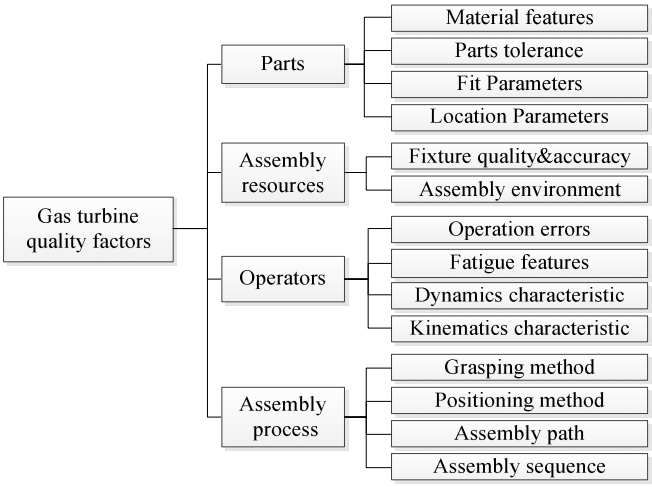


Fig. 1. Factors influencing assembly quality

assembly resources, assembly process and operators. Usually factors of different types are not separated from each other completely, they are coupling and interlaced. Typical influencing factors are listed in Fig.1and described below:

1) Parts factors. Tolerance chain may not meet the requirement even though each part is within its range. For example, the so called "reassembly problem" means that operators often have to reassemble parts after adjustment if the dimensions or other parameters cannot meet the requirement. Besides, in consideration of the physics attributes of large gas turbine, improper clamping force, fixture or other assembly process may lead to deformation of parts. So, finite element analysis should be used as a necessary step to calculate the deformation or vibration of parts.
2) Assembly resources factors. Assembly resources consist of assembly fixture, tools and assembly environments. Fixtures & tools' quality and accuracy may cause technical errors in assembly processing. The environment factors cannot be neglected too, for instance, the gravity may cause parts deformation since the parts are large and heavy.
3) Operators factors. It is a serious test for assembly operators since large gas turbine has strict requirement for assembly quality. Human engineering factors including fatigue factor and human errors should be considered for assembly process and training.
4) Assembly process factors. Assembly process planning consists of assembly sequences, assembly paths, grasping & clamping methods, mating & coordinating methods, etc. Optimizations that enhance accuracy and simplify procedures should be done for them.

2.2 Architecture of the Solution

Based on the considerations above, this paper introduces a solution to assist assembly quality control as shown in Fig.2. This solution for assembly quality control combines ONSM and OFFSM, and reflects the conceptions of symbiotic simulation. *Assembly Quality Symbiotic Control System* (AQPCS) can be built using this architecture, which will optimize assembly process and operators may not need trial assembly any more.

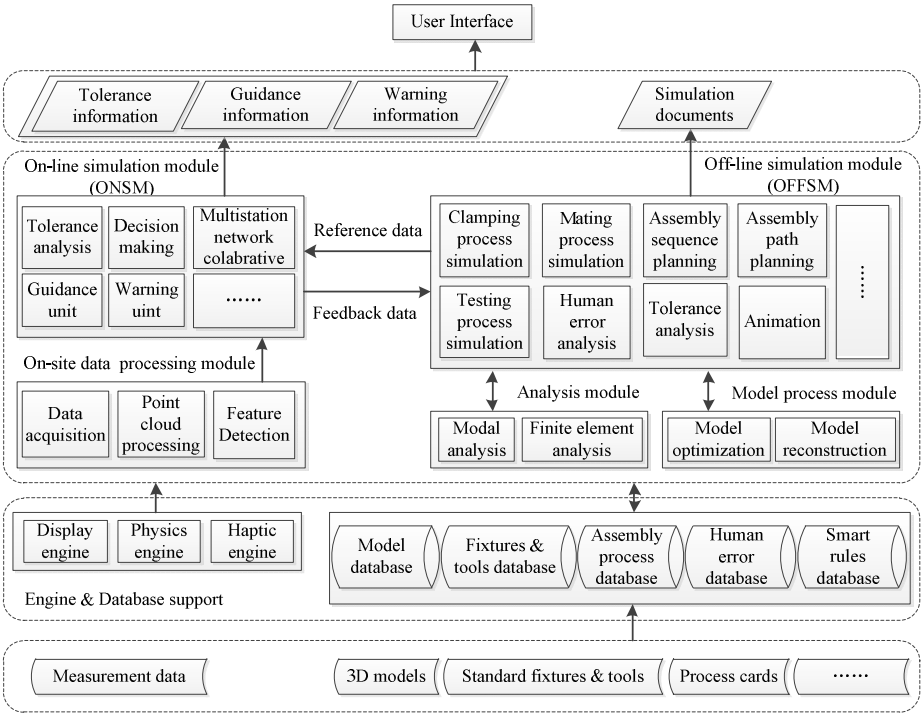


Fig. 2. The symbiotic simulation solution for assembly quality control

The architecture of the solution can be divided into four layers as followings:

1. The bottom layer is the input layer, which provides 3D models, standard fixtures & tools, process cards and measurement data for the system.
2. The second layer from the bottom is engine & database support layer. The engines provide support for virtual environment, such as physical attributes and haptic feedbacks. With the help of databases, assembly process analysis and planning can be much smarter and need fewer human-computer interactions.
3. The basic structure of this solution is in the second layer, combining ONSM with OFFSM. The basic function of OFFSM is assembly process analysis and planning, while guidance and supervision of real-time assembly process are the main functions of ONSM. The model process module reconstructs and optimizes 3D models

for OFFSM. Finite element analysis module can do deformation and vibration analysis of the gas turbine. On-site data processing module acquires point cloud data and outputs them to ONSM after processing.

4. The top layer is the output layer, which is connected with the user interface. Trainees and operators can get simulation documents from OFFSM and necessary information from ONSM.

Based on the basic introduction of the architecture, it is clear that this solution is not simply a combination of on-line simulation or off-line simulation, but it can be viewed as a solution using extended symbiotic simulation. The basic symbiotic simulation mainly improves the on-line simulation but it doesn't emphasize the effect of off-line simulation (sometimes we can call it traditional simulation). But for large and complex systems, both simulations are necessary and they can cooperate with each other. Thus the assembly process can be well controlled. On the whole, there are four main mechanisms of the solution that can be described as followings:

1. Cooperative mechanism. OFFSM can do assembly process analysis and planning, and ONSM provides guidance and warnings for assembly operations. These simulation modules are not binding modules in this system but they cooperate with each other. On the one hand, OFFSM is the preview and reference of ONSM. The problems that may happen in actual assembly process can be predicted in OFFSM (by assembly process planning). And the OFFSM will provide necessary data for ONSM such as assembly sequences, assembly paths. On the other hand, ONSM is the test and feedback of OFFSM. New problems such as human errors or technical errors may be found during ONSM. If the assembly process planning from OFFSM is proved to be improper in real-time assembly process, operators can verify and revise the process planning with the assistance of feedback data. Therefore, assembly quality can be controlled in terms of time, space and process viewed from assembly simulation. They promote and cooperate with each other for assembly quality control. It is described in Table 1.

Table 1. Assembly symbiotic simulation from time, space and process

Simulation module	Time	Space	Process
On-line simulation	Real-time	Hardware-in-loop	Feedback
Off-line simulation	Prediction	Virtual	Guidance &Warning

2. Supervisory mechanism. Evaluation unit, guidance unit and warning unit in ONSM interact with each other and follow a certain procedure, which can be called "supervisory mechanism". Guidance information is provided at specific moments, for example, guiding for the next assembly process. Warning information is provided in real-time case as illustrated in Fig.3. But from another point of view, if the system is too sensitive or rigid, operators may receive many "fake warnings", which

will get operators into trouble. Therefore "real warnings" may be ignored by operators so that warning unit cannot make any sense. We may use machine learning or pattern recognition methods to create a special database to gather necessary data and set up new rules to avoid so many "fake warnings". What's more, multi-station network units can provide coordination assistance for operators working at different places.

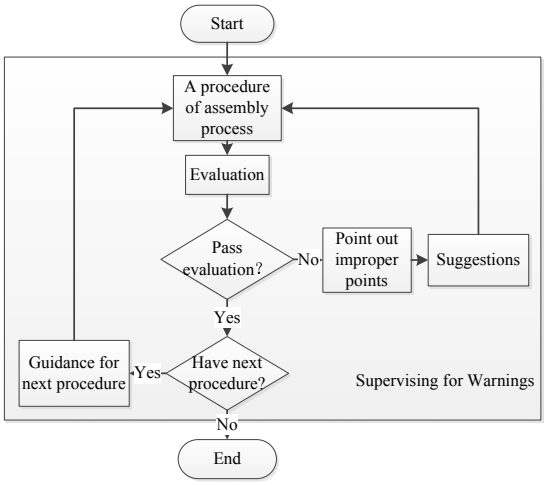


Fig. 3. On-line supervisory mechanism

3. Data interchange mechanism. ONSM and OFFSM can exchange necessary data for cooperation, including reference data and feedback data. The data streams in this solution are in Fig. 4. The left stream is massive measurement data from actual assembly, called "point cloud", swarming into data process module for data processing. Besides, forces values and other parameters that have been measured also follow this stream. As the on-site data process and ONSM can hardly integrate nowadays, a data interface is needed from measurement process module to ONSM. The right stream is the 3D CAD modeling for assembly reconstruction and CAE analysis (i.e. deformation). The analysis results and reconstructed models are used for assembly process planning. These two data streams can be used separately, but interim results (called reference data and feedback data) from ONSM and OFFSM can exchange for cooperation. For example, after getting the actual dimensions of parts, tolerance analysis unit in ONSM calculates the deviations between actual dimensions and reference dimensions. This process can be called "Data Comparison". What's more, the feature detection means that the assembly features can be recognized from massive point cloud, which is necessary and useful for assembly process analysis as well as data comparison.

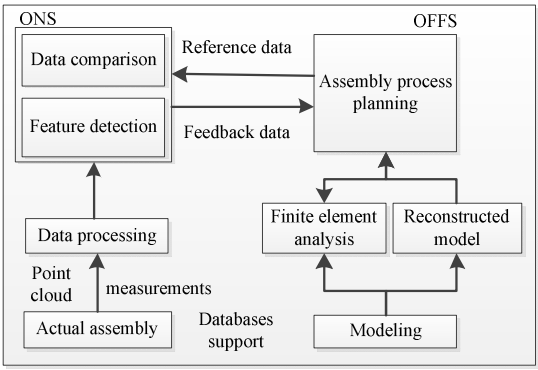


Fig. 4. Data streams in symbiotic simulation

4. Open and extensible mechanism. Factors influencing assembly quality are various, ever-increasing and sometimes unpredictable, so a closed system cannot meet the requirement of assembly quality control. This system is open and expandable from views of structure, function and data. The first extension is model creating and updating. The system can accept 3D CAD models from popular mechanical softwares such as Pro/Engineer, CATIA, UG, and so on. Besides, with the help of ACIS [9] or other modeling kernels, operators can create models within the system, which is a supplementary method for modeling. After reconstruction, the model will contain necessary information such as geometry, topology, physics and tolerance, etc. If the information is not included or incorrect, operators can define, edit or delete them during the reconstruction process. The second extension is rules creating and updating. We will find out proper models to analyze the effects caused by fixtures & tools, operators, etc. For example, we create databases for operators' parameters and human errors. The analysis models related can be updated or replaced. The third is the analysis and simulation expandability. With the help of data structure of databases, we can add necessary units into ONSM or OFFSM for comprehensive analysis and simulation.

3 Partial Key Techniques

3.1 Tolerance Analysis in Symbiotic Simulation

Tolerance plays a vital role in complex mechanical equipment such as large gas turbines. For example, there is a micron-scale clearance between rotor blades and outer cylinder, which is called "radial gap". To reduce the leakage for efficiency, the radial gap is designed as small as possible. Otherwise, it will lead to low efficiency; while if it is too small, the rotor blades and outer cylinder may collide with each other, which will not only decrease efficiency but also endanger safety of production; moreover, if the clearance is inhomogeneous, a turbulent force will generate and it will cause shakiness of gas turbine, which is also a potential safety hazard. Thus, measurement and dimension control are indispensable.

Since large gas turbine has complicated structure, most of the dimensions to be measured are not easy to get. In this case, we can use the so called technique "point cloud" to get the necessary locations or dimensions. Generally speaking, the point cloud technique is used for reverse engineering, but its usage is different in symbiotic simulation.

Non-contact devices and tactile devices can be used in large gas turbine [10]. Non-contact devices can acquire huge data after a short time, and don't need radius compensation, while the tactile devices have higher precision. In most cases, operators only measure the key dimensions or locations. So we can use a non-contact device FaroArm [11] as the measurement device. Creating an object space coordinate system by FaroArm measurement is intuitive for dimension representation, as Fig. 5 shows. Then operators can get point cloud data by measurement.

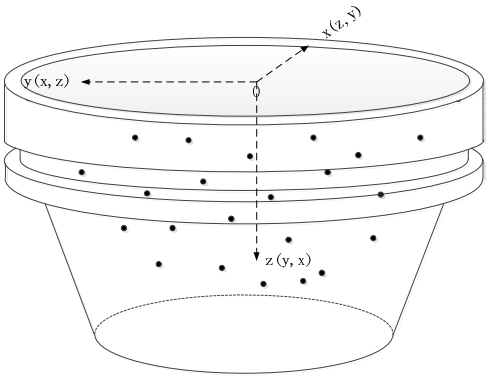


Fig. 5. Acquiring point cloud data from cylinder

The point cloud data cannot be used directly because they contain noise, holes, isolate and invalid data caused by devices or software systems. Before processing, transformations from object space coordinate system to global coordinate system should be made after acquiring point cloud data. After that, typical steps [10] for point cloud processing preparation are needed and they are summarized in Table 2. Researchers have studied these point cloud handling algorithms for some years. Moving Least Squares (MLS) [12] can be used for minimizing the geometric error of the approximation; Radial Basis Functions (RBFs) [13] can be used to repair incomplete area; we can use some of the algorithms based on the successful algorithm ICP [14] for registration and integration; Multi-level Partition of Unity (MPU) [15] is designed for creating surfaces; Pauly et al. [16] introduce a method that can simplify surface representation with lower sampling density. After data processing and simplifying, dimensions used by ONSM can be got.

By point cloud processing and data comparison in ONSM, operators can find out whether the tolerances are within its limits. If not, the simulation system can provide useful information to revise the assembly process, such as advices and warnings. And then operators can use selective assembly technology and even remanufacture the inappropriate parts without trial assembly. Therefore, tolerance can be controlled and assembly process will be simplified with the help of tolerance analysis in the solution.

Table 1. Preparations for using point cloud data

Preparation steps	Description
Filtering	Remove inaccuracy and noise data
Repairing	Fill holes and repair data
Registration	Transformation between different views
Integration	Creating a complete representation after registration
Reconstruction	Reconstruct elements(e.g. reference surface)
Simplification	Reduce memory storage and accelerate computation
Segmentation	Distinguish target areas with its borders

3.2 Training in Symbiotic Simulation System

Strict training to reduce human errors in assembly process is significant, because of complex procedures and high requirement for accuracy. Training now in gas turbine industry is highly dependent on instructions of experienced experts and documents, which is inconvenient, intricate and expensive. With the help of symbiotic simulation, training will be more effective, intelligent and acceptable, as Fig. 6 shows.

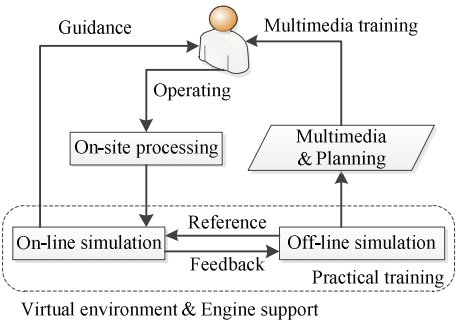


Fig. 6. Training in symbiotic simulation

With the help of ONSM and OFFSM, the training can be classified into off-line training and on-line training, as described below:

1. Virtual environment training and multimedia animation training. Training with haptic feedbacks in virtual environment can improve operators' involvement and acceptance [17]. Practical training under the virtual environment is supported by display engine (e.g. HOOPS [18]), physics engine and haptic engine. Haptic engine calculates force and torque, which is rendered by haptic devices, for instance, PHANTOM devices produced by SensAble Technologies Inc. We can use a multithread method to support physics rendering and haptic feedback [19] . By the way, if the environment is too sensitive for operation, the parts will tremble when touched, because the virtual environment may not distinguish between moving parts and touching parts. Besides, the OFFSM can output simulation files and assembly planning as a convenient way for training. The multimedia animations can help operators understand the assembly process clearly.

2. Real-time training. The real-time training means to train operators just before the next assembly procedure. The guidance in ONSM is not only assistance for actual assembly, but also a new way for real-time training. It is a method to prevent human errors to some extent and it will strengthen the training effect. In this way, it guarantees the assembly process quality.

4 A Case Study: Measurement Simulation for Training

In order to test and verify some conceptions listed above, the paper simulates the measurement process in OFFSM, outputting simulation files for assembly process training. The measurement process simulation belongs to OFFSM, while it is also the basis of ONSM, which assists tolerance analysis of outer cylinder and reflects cooperative and data interchange mechanisms of the solution. In this paper, *virtual assembly supported system* (VASS) [20], whose functions consist of verifying assemblability, creating assembly process planning and outputting simulation files, is used to verify measurement process. One of the problems we meet in the simulation process is that some of the locations to be measured are uneasy to be seen from outside of the cylinder. There are two methods to get targets visible: one is to make a longitudinal section or cross section of cylinder, and the other is to make some of the parts transparent. The paper uses the latter because using longitudinal or cross section may cause operators confused. We finally get the simulation of measurement process as shown in Fig. 7, which describes how the FaroArm measures the cylinder.

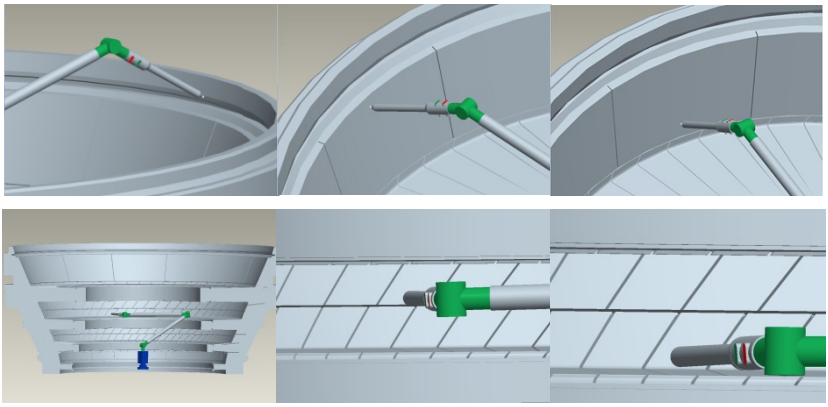


Fig. 7. Measurement simulation by VASS (The longitudinal section picture at the lower left is to show the measurement process clearly.)

5 Summary

Assembly quality control has a vital effect on large gas turbine quality. Firstly, after analyzing reasons that affect gas turbine, we propose the solution using symbiotic simulation for large gas turbine assembly quality control. The paper discusses the

architecture and main mechanisms of the solution. Assembly quality factors can be predicted, checked and revised with the assistance of symbiotic simulation. Secondly, some partial techniques (tolerance analysis, training) are addressed, which play an important role in the solution. Thirdly, an instance that reflects mechanisms and data streams in the solution is realized, which improves skills of operators and the efficiency of assembly process. But the work is just in progress. One of the future researches is to develop proper communication mechanisms between ONSM and OFFSM.

Acknowledgements. This research is supported in part by National Science and Technology Support Program (Grant 2012BAH32F03-1).

References

1. Aguilar, J.J., Sanz, M., Guillomia, D., Lope, M., Bueno, I.: Analysis, characterization and accuracy improvement of optical coordinate measurement systems for car body assembly quality control. Int. J. Adv. Manuf. Tech. 30, 1174–1190 (2006)
2. Asha, A., Kannan, S., Jayabalan, V.: Optimization of clearance variation in selective assembly for components with multiple characteristics. Int. J. Adv. Manuf. Tech. 38, 1026–1044 (2008)
3. Su, Q., Liu, L., Whitney, D.E.: A Systematic Study of the Prediction Model for Operator-Induced Assembly Defects Based on Assembly Complexity Factors. IEEE T. Syst. Man. Cy. A 40, 107–120 (2010)
4. Yuefei, Z.: Study on Assembly Deformation and Control of Gas Turbine Rotor 680. Shanghai Jiao Tong University (2011)
5. Bao-Zhong, W., Jing, Q., Guo-Xi, L., Jing-Zhong, G.: Research on complex precision mechanical system assembly parallel control system. In: 2nd International Conference on Advanced Design and Manufacturing Engineering, ADME 2012, August 16-18, vol. 215-216, pp. 1267–1275. Trans. Tech., Publications, Taiyuan (2012)
6. Huang, S.Y., Cai, W., Turner, S.J., Hsu, W.J., Zhou, S., Low, M.Y.H., Fujimoto, R., Ayani, R.: A generic symbiotic simulation framework. In: 20th Workshop on Principles of Advanced and Distributed Simulation, PADS 2006, vol. 131. IEEE (2006)
7. Aydt, H., Turner, S.J., Cai, W.T., Low, M.: Symbiotic simulation systems: An extended definition motivated by symbiosis in biology. In: Proceedings of the 22nd International Workshop on Principles of Advanced and Distributed Simulation, PADS 2008, pp. 109–116 (2008)
8. Darema, F.: Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3038, pp. 662–669. Springer, Heidelberg (2004)
9. 3D Modeling Components and Solid Modeler | Spatial, http://www.spatial.com/products/3d-acis-modeling
10. Bi, Z.M., Wang, L.H.: Advances in 3D data acquisition and processing for industrial applications. Robot. Cim.-Int. Manuf. 26, 403–413 (2010)
11. Portable Measuring Arm, Arm CMM with Laser Line Probe | FaroArm | FARO Asia, http://www.faroasia.com/products/faroarm/sea/index.php
12. Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T.: Computing and rendering point set surfaces. IEEE T. Vis. Comput. Gr. 9, 3–15 (2003)

13. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3D objects with radial basis functions. In: Computer Graphics Annual Conference (SIGGRAPH 2001), August 12-17, pp. 67–76. Association for Computing Machinery, Los Angeles (2001)
14. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes: Robotics-DL tentative, pp. 586–606. International Society for Optics and Photonics (1992)
15. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.: Multi-level partition of unity implicits. In: ACM SIGGRAPH 2003 Papers, SIGGRAPH 2003, July 27-31, pp. 463–470. Association for Computing Machinery, San Diego (2003)
16. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces.: Visualization. In: Visualization, VIS 2002, pp. 163–170. IEEE (2002)
17. Stork, A., Sevilmis, N., Weber, D., Gorecky, D., Stahl, C., Loskyll, M., Michel, F.: Enabling virtual assembly training in and beyond the automotive industry. In: 2012 18th International Conference on Virtual Systems and Multimedia (VSMM), pp. 347–352 (2012)
18. HOOPS Visualize | Tech Soft 3D, http://www.techsoft3d.com/our-products/hoops-visualize
19. Xia, P.J., Lopes, A.M., Restivo, M.T., Yao, Y.X.: A new type haptics-based virtual environment system for assembly training of complex products. Int. J. Adv. Manuf. Tech. 58, 379–396 (2012)
20. Li, Z., Lin-xuan, Z., Tian-Yuan, X.: Introduction of a Virtual Assembly Supported System. Journal of System Simulation, 1149–1153 (2002)

Profiling Multilevel Partitioning for Asynchronous VLSI Distributed Simulation*

Elias Tsirogiannis[1] and Georgios Theodoropoulos[2]

[1] University of Birmingham, U.K.
[2] Institute of Advanced Research Computing
Durham University, U.K.

Abstract. Partitioning is a crucial factor in VLSI distributed simulation. This paper focuses on the partitioning problem for asynchronous handshake circuits generated by the Balsa asynchronous hardware synthesis environment. A quantitative analysis is presented for multilevel partitioning, as exemplified by the METIS library.

1 Introduction

Distributed simulation is a particularly promising and viable approach to alleviate the simulation bottleneck in VLSI design and has received considerable attention from researchers in mainstream Hardware Description Languages such as VHDL and Verilog and has progressively found its place in innovative commercial hardware design environments. Bailey et al. [1] have identified five major factors which affect the performance of distributed simulation of VLSI: partitioning algorithm, synchronisation algorithm, circuit structure, timing granularity and target architecture. In this paper we focus on the first of these factors.

An exciting development in VLSI design over the last two decades has been the exploration of asynchronous design approaches to address clock skew and energy problems. A number of asynchronous processors have been developed including NSR and Fred at the University of Utah, STRiP at Stanford University, FAM and TITAC at Tokyo University and Institute of Technology respectively, Hades at the University of Hertfordshire, Sun's Counterflow pipeline processor, Sharp's Data-Driven Media Processor, CalTech's processors and Lutonium, the series of asynchronous implementations of the ARM RISC processor (AMULET1, AMULET2e, AMULET3i and SPA) at the University of Manchester, SAMIPS at the University of Birmingham, the "Network-based Asynchronous Architecture" processor at the University of Edinburgh, the ARM996HS and HT80C51 processors from Handshake Solutions, Moore's SEAforth and GA144 processors etc.

Synchronous VLSI modelling and simulation techniques have proved unsuitable for the asynchronous design style and a number of modelling approaches have been investigated, including I-Nets, Petri Nets, Signal Transition Graphs, CCS and in particular the concurrent process algebra Communicating Sequential Processes (CSP). Balsa[1] is a

* Elias Tsirogiannis is now with Nokia Siemens Networks, Bristol, UK.
[1] `http://apt.cs.man.ac.uk/projects/tools/balsa/`

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 310–324, 2013.

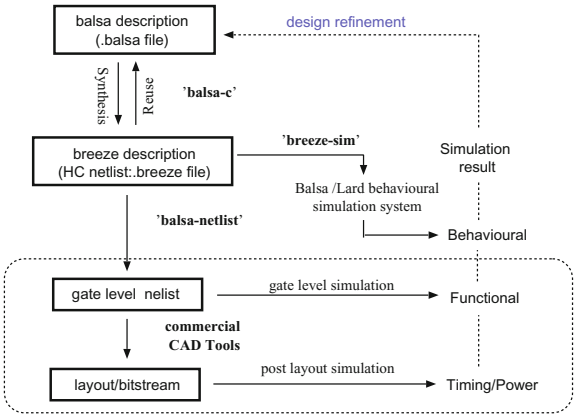


Fig. 1. Balsa System

prominent CSP-based asynchronous circuit synthesis system developed at the University of Manchester.

In [30] we presented PARBREEZE, a distributed simulator for Balsa specifications and we provided a quantitative comparison for two partitioning strategies: manual and multilevel partitioning, as exemplified by the METIS system. We illustrated that METIS delivered better performance, however we identified deficiencies scope for improvement. Taking this work further, in this paper we provide a quantitative profiling of METIS for regular Balsa designs. This exercise constitutes a necessary step in the endeavour to identify and design partitioning algorithms appropriate for asynchronous VLSI systems.

The rest of the paper is structured as follows: Section 2 provides a short overview of the Balsa system; section 3 summarises the multilevel partitioning method; section 4 present a quantitative analysis of METIS for regular Balsa designs and section 5 concludes the paper identifying paths for further work.

2 Balsa and Handshake Circuits

Balsa[2] is both an asynchronous hardware synthesis framework and a CSP-based language for describing asynchronous systems. It has been demonstrated by synthesising at least three major systems: the DMA controller of Amulet3i, SPA, an ARM core for smartcard applications and SAMIPS. Figure 1 shows an overview of the Balsa system. Balsa uses CSP-based constructs to express Register Transfer Level design descriptions in terms of channel communications and fine grain concurrent and sequential process decomposition. Descriptions of designs (*.balsa* file) are translated (***Balsa-c***) into

[2] The latest major release of Balsa version 4.0 took place on June 10, 2010. Balsa is licensed under the GNU GPL Version 2. Source code for Balsa version 4.0 can be found at `ftp://ftp.cs.man.ac.uk/pub/apt/balsa/4.0`.

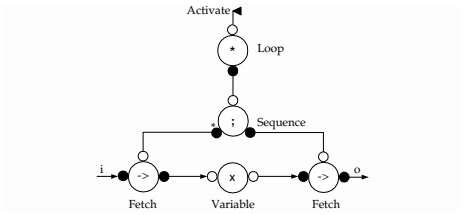
```
import [balsa.types.basic]

procedure buffer (input i:byte;
output o:byte ) is

variable x : byte

begin
loop
i->x - Input communication
; - Sequence operator
o<-x - Output communication
end
end
```

(a)



(b)

Fig. 2. Balsa Code and Handshake circuit for a one-place buffer

implementations in a syntax directed-fashion with language constructs being mapped into networks of parameterised instances of *handshake components* (*.breeze* file) each of which has a concrete gate level implementation. A number of tools are available to process the breeze handshake files. ***Balsa-netlist*** automatically generates CAD native netlist files, which can then be fed into the commercial CAD tools that further synthesize the netlist to the fabricable layout. Balsa has approx. 50 handshake components in total each with a unique name, symbol, definition and several implementations (based on different technologies). In the handshake circuit, components communicate via point-to-point *channels* to exchange control information and optionally data. Balsa uses one-to-one mapping between the language constructs in the specification and the intermediate *handshake circuits* that are produced. Figure 2a illustrates a simple Balsa modelling example of a simple buffer. The corresponding translated handshake circuits are shown in Figure 2b. Three levels of simulation are supported in Balsa, namely behavioural, gate-level and post layout simulation (figure 1). The latter two low simulation levels are carried out by the native simulators of the commercial CAD tools supported by Balsa. At the behavioural level, discrete-event simulation is used to simulate the network of handshake components. Two sequential simulators have been developed for this level, namely LARD [7] and *breeze-sim*[6]. In [30] we described *parbreeze*, a distributed implementation of *breeze-sim* for the distributed simulation of handshake circuits.

3 Multilevel Partitioning

The problem of graph partitioning in the context of parallel processing is an old NP hard problem and has received extensive attention both as a general problem as well as in the context of distributed simulation of VLSI systems (indicative examples include *string*

partitioning [19], *Fanin and Fanout Cone Partitioning* [27] [22] [18] [11], *Corolla Partitioning* [28] and *Concurrency Preserving Partitioning* [16]).

The multilevel partitioning approach may be considered the most promising of those, with efforts starting in 1990's [2][3][4] [9][10] [13][14] [8][20][31]. The key idea of this paradigm consists of three well defined phases (see figure 3):

(1) *Graph coarsening*: the original graph is coarsened down to successively smaller and smaller graphs until a sufficiently small (coarsest) graph is obtained.
(2) *Initial partitioning*: the coarsest graph obtained by the graph coarsening phase is partitioned.
(3) *Uncoarsening*: the partitioned coarsest graph is projected back to the original (finest) graph through a sequence of larger and larger graphs, while the partition is being further refined all the way.

Over the years, a variety of algorithms have been proposed and compared for all the three phases, on a large number of graphs from different domains[2][3][10] [14][13][8] [31][20][5] [25][21], including distributed VLSI simulation [29] [16] [32]. A number of software libraries which incorporate a range of these algorithms have been implemented, the most representative of them including CHACO [10], JOSTLE [31], METIS [13] [14], PARTY [20] and SCOTCH [24]. METIS appears to be the favourite choice when researches consider the multilevel paradigm as a partitioning option and has been repeatedly shown to yield the most promising results amongst them and has been selected for our profiling exercise. METIS multilevel recursive bisection algorithm is used to partition *breeze file* netlists where, vertices of the graph represent the set of the handshake components while the edges are the balsa communication channels.

4 Profiling METIS: Experimental Analysis

As a benchmark, the structure depicted in figure 4 has been used. The main functionality of this structure is to perform linear transformations to the input data [12] and is widely used in different domains including the manipulation of geometrical data, data display in graphics applications, image and signal processing. The benchmark is scalable and modular with inherent parallelism, thus facilitating the parameterisation of the design without affecting its functionality and enabling the isolation of parts of the design for seperate analysis. The system is given an input data vector $\mathbf{u} = [u_1, ..., u_N]^t$ of dimension $N \times 1$, and calculates the output data vector $\mathbf{v} = [v_1, ..., v_M]^t$ of dimension $M \times 1$, using the following matrix multiplication and vertex addition equation

$$\mathbf{v} = \mathbf{A} \cdot \mathbf{u} + \mathbf{k}$$

where $\mathbf{A}$ is an $M \times N$ matrix and $\mathbf{k}$ is an $M \times 1$ vertex, with both $\mathbf{A}$ and $\mathbf{k}$ consisting of constant elements. The i-th element v_i of the vector $\mathbf{v}$ is given by the following equation

$$v_i = \sum_{j=1}^{N} a_{ij} \cdot u_i + k_i, \quad i = 1, ..., M \tag{1}$$

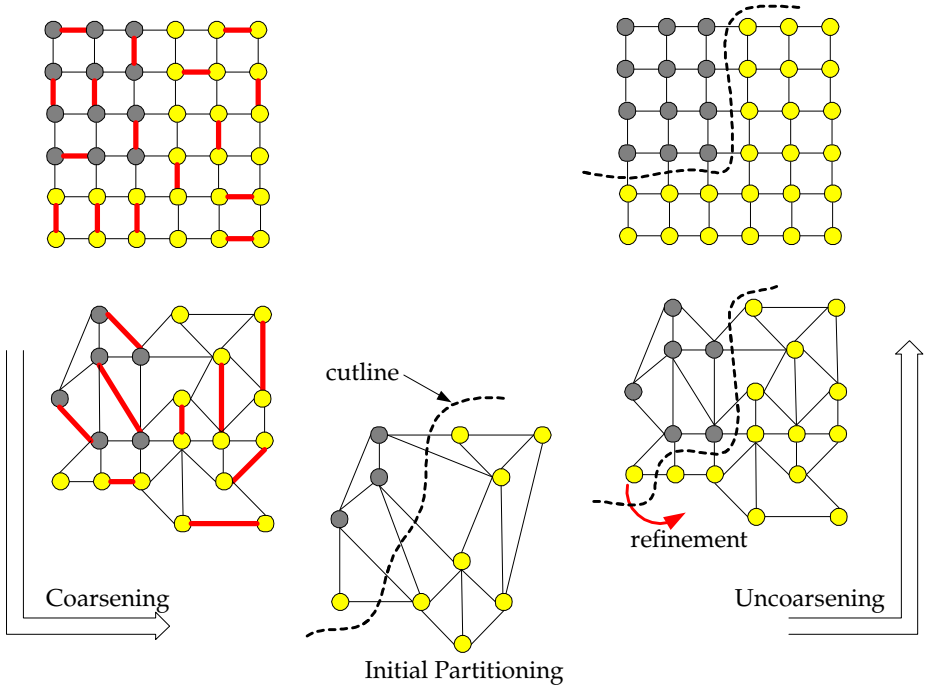


Fig. 3. The multilevel paradigm starts with the coarsening phase whereby the size of the original graph is being successively decreased. A matching algorithm matches heavy edges i.e. vertices with same colour. As it can be seen, the initial partitioning takes place on the coarsest graph with no perfect result. The uncoarsening phase projects the initial partition back to the original graph and manages to refine the erroneous result of the initial partitioning. In this example, we assume that the weight of each dark vertex is twice as the weight of each light vertex.

where a_{ij} is the (i-th, j-th) element of the matrix $\mathbf{A}$ and k_i is the i-th element of the vector $\mathbf{k}$. Each data transformation of a vector $\mathbf{u}$, requires $N \cdot M$ multiplications and $N \cdot M$ additions.

As Balsa does not provide any library for multiplication of non-constant operands [6], we have implemented in Balsa a multi-bit multiplier which involves the use of *carry-save* and *carry-propagate* adders [23], as depicted in figure 5a. All the internal modules are purposely chosen to have fully sequential logic. This practically means, that the entire design of a multiplier does not offer any degree of parallelism that should be exploited by the partitioning algorithms. The overall parallelism of the evaluation design comes from interconnecting modules by forming hybrid (pipeline and parallel) structures. This facilitates our evaluation work, because we know exactly where the parallelism in the design comes from and thus, we are able to assess whether the evaluated partitioning strategy exploits the available parallelism or not. The hardware architecture of each stage of the array structure is illustrated in figure 5b. For our experiments, each stage comprises a 16-bit multiplier and a 32-bit adder. Each 16-bit multiplier has

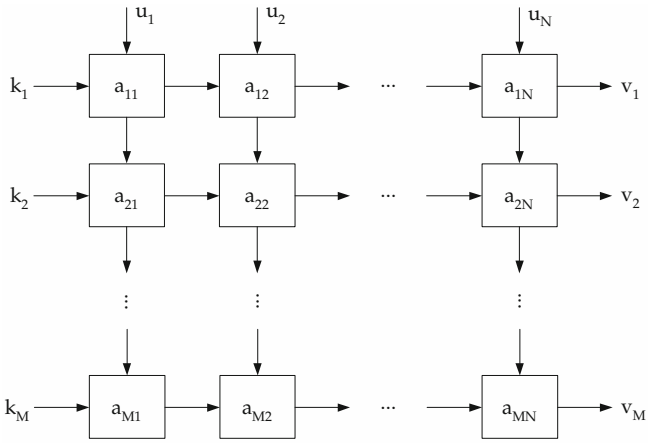


Fig. 4. The schematic diagram for the hardware which performs linear transformations to input data: Successive input values $u_1, ..., u_N$ feed the array from the top to down and successive transformed values of $v_1, ..., v_M$ emerge at the right of the array

a corresponding handshake circuit which consists of 1859 handshake components and 3290 channels, while each adder is implemented by a single handshake component (*BinaryFunc*). Each stage as a whole, consists of 1873 handshake components and 3314 channels. A Balsa*test-harness* provides input stimuli to designs and display the outputs. During a simulation run each stage has received 2000 inputs and has generated 2000 outputs.

Another parameter which determines the way we observe our experiments, is based on the nature of the hardware designs. Typically, hardware designs with intrinsic concurrency comprise three partition patterns, namely *pipeline*, *parallel* and *hybrid* [17]. The selected benchmark can distinguish M distinct pipeline schemes with N pipeline stages each, which work in parallel forming a hybrid pattern. Consequently, we profile METIS in stages, first focusing on a pipeline and then over the entire hybrid design.

In order to partition the Balsa design into PARBREEZE LPs, we have utilised the *METIS_PartGraphRecursive* routine [15] embedded in METIS library which partitions the balsa graph into k equal-sized parts using multilevel recursive bisection, with the objective to minimise the edge-cuts. Based on our experimental analysis practically almost all handshake components have fairly similar computational workloads while, in the absence of any other information, we presume equal edge weights.

The execution platform is a Linux cluster machine with dual- processor Intel Xeon 3GHz nodes and 2 GBytes of memory, interconnected via a Myrinet-2000 switch. Following a performance comparison between MPICH-GM 1.2.6 and MPICH 1.2.6 we have opted for the latter. Results are averaged over 10 runs. We have used 15 nodes with exclusive access, on LP per node. The pipeline design yields 28758 handshake components and 50558 channels while the full hybrid structure 28453 handshake components and 50185 channels.

To evaluate the resulting distributed simulation, in addition to wallclock and speedup, we have used the following metrics:

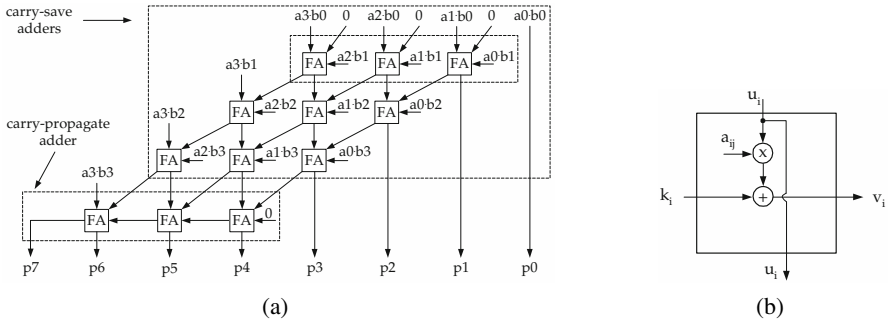


Fig. 5. (a) A 4-bit multiplier: uses 3 carry-save adders and one carry-propagate adder. In this case each adder consists of 3 full- adders. (b) Hardware architecture of each stage comprising a 16-bit multiplier denoted by ***x*** and a 32-bit adder denoted by **+**.

(i) *Efficiency* (E): It is defined as the ratio of speedup to the N number of processing units. It provides the average utilisation of the N processors and in the ideal case, the maximum efficiency being one.

(ii) *Average Parallelism* (A): It is defined as the average number of processors that run concurrently, during the entire execution of a distributed simulation. It is a useful indicator of distributed simulation performance, which provides information about the parallelism in distributed execution. If *shape vector* $\mathbf{P} = [\, p_1, ..., p_N]$ is defined to be a vector of dimension $1 \times N$, in which each element p_i denotes the proportion (normalised to one) of wallclock time which has degree of parallelism i, then the average degree of parallelism can be given by the following equation:

$$A = \sum_{i=1}^{N} i \cdot p_i.$$

Other parameters that help characterise the parallelism in a distributed simulation can be obtained from the shape vector as follows [26]:

- *Minimum parallelism*: $m = i$, such that p_i=min(**P**).
- *Maximum parallelism*: $M = i$, such that p_i=max(**P**).
- *Fraction sequential*: $f = p_1$.
- *Variance in parallelism*: $\sigma^2 = \sum_{i=1}^{N} i^2 \cdot p_i - A^2$.

4.1 Results

Pipeline Structure. Figure 6 shows the wallclock times while speedup and efficiency values are provided in table 1. It is easy to observe that in the majority of cases the attained efficiency is quite low, indicating that there may be room for performance improvement. For instance, the obtained speedup is 1.75 when using 3 processors, whereas the obtained speedup is 1.29 when using 14 processors. In other words, the efficiency of employing 3 processors is 0.58, while the efficiency of employing 14 processors is as low as 0.09. The efficiency rises again at 0.33 when 15 processors are employed.

Table 1. Attained speedup and efficiency

	Number of Processors						
	2	**3**	**4**	**5**	**6**	**7**	**8**
Speedup	0.79	1.75	1.30	2.54	1.62	1.70	2.00
Efficiency	0.40	0.58	0.33	0.51	0.27	0.24	0.25
	9	**10**	**11**	**12**	**13**	**14**	**15**
Speedup	1.83	1.90	1.68	1.60	1.52	1.29	4.96
Efficiency	0.20	0.19	0.15	0.13	0.12	0.09	0.33

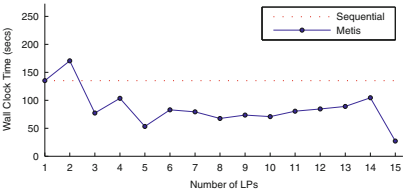
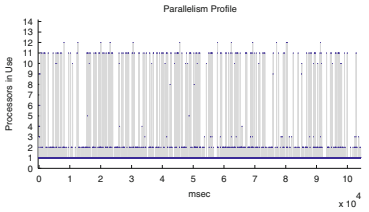
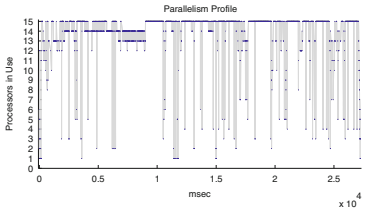


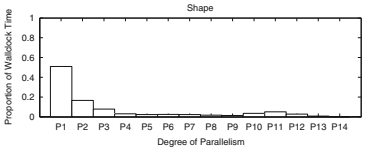
Fig. 6. Performance of METIS for Pipeline



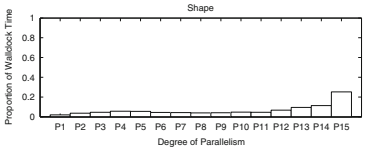
(a) Parallelism profile of distributed simulation on 14 processors



(b) Parallelism profile of distributed simulation on 15 processors



(c) Shape of distributed simulation on 14 processors



(d) Shape of distributed simulation on 15 processors

Fig. 7. Performance Analysis of Pipeline Structure

Figures 7a and 7b, depict the *parallelism profile* of the simulation on 14 and 15 processors respectively (indicating the number of processors used by distributed simulation as a function of the wallclock time) while figures 7c and 7d, provide the *shape* of the simulation. Indicatively, for 14 processors the efficiency is low while for the 15 processor set up, all the 15 processors are used for 25.18% of the wallclock time and 14 processors for 11.39% of the wallclock time. In other words, the inherent parallelism of the design is enough to keep 14 processors busy for non zero proportion of the wallclock time, and therefore, the way the partitioning algorithm distributes the workload among the processors, is to blame when distributed simulation fails to exploit such a concurrency for 14 processors.

It is desirable for distributed simulation to get both the minimum parallelism to the value 1 and the maximum parallelism to the maximum number of the processors available for use. Table 2, presents the average parallelism A, minimum parallelism m, maximum parallelism M, fraction sequential f and variance in parallelism σ^2 metric measurements. Among all parallelism metrics in table 2, we distinguish those obtained using 3, 5 and 15 processors since they exploit the maximum parallelism available

Table 2. Average parallelism A, minimum parallelism m, maximum parallelism M, fraction sequential f and variance in parallelism σ^2 measurements

Number of Processors	A	m	M	f	σ^2
2	1.49	2	1	0.51	0.25
3	2.85	1	3	0.01	0.15
4	2.53	2	4	0.30	1.46
5	4.45	1	5	0.01	0.71
6	3.22	6	1	0.24	2.93
7	3.49	7	1	0.24	4.21
8	4.11	8	4	0.10	3.46
9	3.99	6	1	0.24	8.53
10	4.04	10	1	0.21	6.71
11	3.76	11	4	0.12	3.23
12	3.76	12	1	0.31	8.64
13	3.62	13	1	0.26	7.31
14	3.18	14	1	0.51	11.63
15	10.35	1	15	0.02	20.00

($M = 3,\ 5$ and 15 respectively) and manage to reduce the minimum parallelism to the minimum value ($m = 1$ for all of them). Indeed, as shown in table 1, distributed simulation runs on 3, 5 and 15 processors yield the best efficiency values of 0.58, 0.51 and 0.33 respectively. It is desirable to get both the minimum parallelism to one and the maximum parallelism to the maximum number of the processors available for use. Other set ups seem to fail to achieve this and therefore, the ability to gain high speedups is significantly reduced. For instance, in the case where 4 processors are employed, even though the maximum parallelism reaches the maximum value i.e. $M = 4$, the minimum parallelism is $m = 2$ (i.e. it is not $m = 1$). The fraction sequential f in this case as it is depicted in table 2 reaches the value 0.30[3], indicating that 30% of the wallclock time is spent using only one processor. Table 3 shows how the partitioning algorithm allocates the 15 pipeline stages to 4 processors. It can be seen that stage 9 is split between processor 2 and processor 4: 56.80% of stage 9 is allocated to processor 2, while the rest 43.20% of stage 9 is allocated to processor 4. Table 4 presents how the partitioning algorithm allocates each pipeline stage to maximum 4 processors. In addition to stage 9 which is split into two processors, it can also be seen in table that 78.13% of stage 10 is allocated to processor 1 and the rest 21.87% is allocated to processor 2. Similarly, we note that 84.43% of stage 4 is allocated to processor 3 and 15.52% of the same stage is allocated to processor 4.

Table 5, shows the edge-cuts produced by METIS for a range of 2 up to 15 partitions. As expected, the three smallest edge-cuts obtained when METIS divides the graph into 3, 5 and 15 partitions. Indeed, they are the only three cases, wherein METIS does not

[3] Comparing this value with the corresponding f value (0.01) for 5 processors in table 2, we clearly observe the considerably higher f value for 4 processors.

Table 3. Allocation of pipeline stages to 4 processors

	Pipeline stages allocated to the processor	
	Stage	Percentage
Processor **1**	**5**	100.00%
	6	0.05%
	10	78.13%
	12	99.47%
	13	99.31%
Processor **2**	**1**	100.00%
	2	0.05%
	3	0.05%
	4	0.05%
	6	0.05%
	7	0.05%
	8	100.00%
	9	56.80%
	10	21.87%
	11	100.00%
	12	0.53%
	13	0.69%
	14	0.11%
	15	0.05%

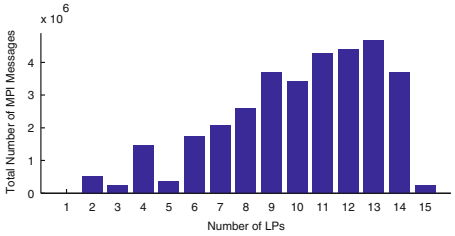
	Pipeline stages allocated to the processor	
	Stage	Percentage
Processor **3**	**4**	84.43%
	6	99.15%
	14	99.89%
	15	99.95%
Processor **4**	**2**	99.95%
	3	99.95%
	4	15.52%
	6	0.75%
	7	99.95%
	9	43.20%

Table 4. Allocation of pipeline stages to maximum 4 processors

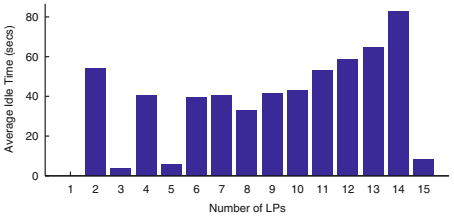
	Pipeline Stages				
	1	**2**	**3**	**4**	**5**
Processor	2(100%)	2(0.05%), 4(99.95%)	2(0.05%), 4(99.95%)	2(0.05%), 3(84.43%), 4(15.52%)	1(100%)
	6	**7**	**8**	**9**	**10**
Processor	1(0.05%), 2(0.05%), 3(99.15%), 4(0.75%)	2(0.05%), 4(99.95%)	2(100%)	2(56.80%), 4(43.20%)	1(78.13%), 2(21.87%)
	11	**12**	**13**	**14**	**15**
Processor	2(100%)	1(99.47%), 2(0.53%)	1(99.31%), 2(0.69%)	2(0.11%), 3(99.89%)	2(0.05%), 3(99.95%)

Table 5. Produced edge-cuts by METIS

	Number of Partitions						
	2	**3**	**4**	**5**	**6**	**7**	**8**
Edge-cuts	59	32	165	42	156	239	292
	9	**10**	**11**	**12**	**13**	**14**	**15**
Edge-cuts	384	312	402	471	487	405	56



(a) Inter-processor communication overhead measured as the total number of MPI messages exchanged between LPs.



(b) Average idle time in LPs.

Fig. 8. Pipeline: Overhead and Idle Time

split n[4] pipeline any stage across LPs and hence, the increase in edge-cuts is avoided. Furthermore, since METIS minimises the edge-cuts only in 3, 5 and 15 LPs we expect the communication overhead to be low only in these three cases. Figure8a which plots the total number of MPI messages exchanged between LPs verifies it, while figure 8b summarises idle times as a result.

Hybrid Structure. Following the same approach, we have profiled METIS for the dull hybrid structure. Figure 9 and table 6 shows the wallclock times obtained speedup and efficiency respectively. The highest speedup and efficiency numbers are obtained for simulation runs on 3 and 5 processors; again, these are the only two cases in which the partitioning algorithm does not split pipeline stages across LPs.

Table 7 presents the parallelism metrics and table 8 the obtained number of edge-cuts per number of partitions. Communication overhead and idle times are shown in figures 10a and 10b.

4.2 Discussion

Clearly, the disadvantage of the multilevel partitioning approach is that it enters the graph coarsening phase without any knowledge of the inherent design relationships of

[4] In fact, there are pipeline stages which are allocated to more than one processor but we consider them as allocation to one processor e.g in the case of 3 LPs, stage 10 is allocated 99.95% to processor 2 and 0.05% to processor 1. Nonetheless, we consider stage 10 to be exclusively allocated to processor 2 as the percentage value 0.05 is negligible.

Table 6. Hybrid: Attained speedup and efficiency

	Number of Processors						
	2	**3**	**4**	**5**	**6**	**7**	**8**
Speedup	0.93	1.77	1.20	2.75	1.60	1.34	1.57
Efficiency	0.47	0.59	0.30	0.55	0.27	0.19	0.20

	9	**10**	**11**	**12**	**13**	**14**	**15**
Speedup	1.82	1.86	1.30	1.50	1.41	1.69	2.62
Efficiency	0.20	0.19	0.12	0.12	0.11	0.12	0.17

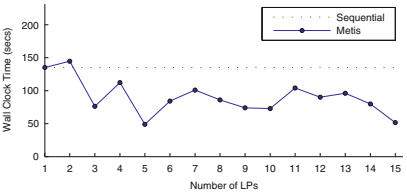


Fig. 9. Hybrid: Performance of METIS

Table 7. Average parallelism A, minimum parallelism m, maximum parallelism M, fraction sequential f and variance in parallelism σ^2 measurements

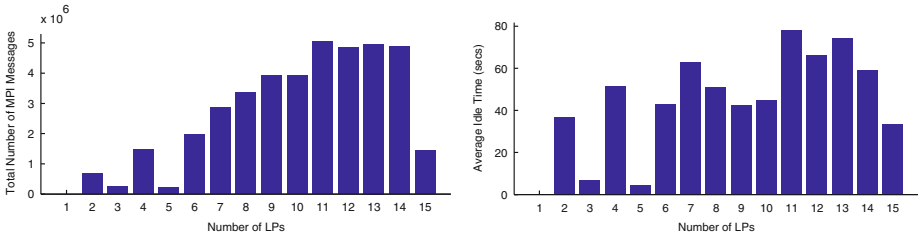
Number of Processors	A	m	M	f	σ^2
2	1.66	1	2	0.34	0.22
3	2.74	2	3	0.11	0.41
4	2.31	2	1	0.40	1.51
5	4.53	1	5	0.01	0.70
6	3.05	5	1	0.30	3.37
7	2.78	6	1	0.41	4.00
8	3.40	5	1	0.35	6.24
9	3.92	9	1	0.20	6.11
10	3.90	10	1	0.21	6.68
11	2.94	11	1	0.42	5.37
12	3.27	12	1	0.34	6.88
13	3.17	13	1	0.47	9.42
14	3.74	14	1	0.27	7.53
15	5.41	11	1	0.39	24.99

Table 8. Produced edge-cuts by METIS

	Number of Partitions						
	2	**3**	**4**	**5**	**6**	**7**	**8**
Edge-cuts	55	35	154	39	164	305	350

	9	**10**	**11**	**12**	**13**	**14**	**15**
Edge-cuts	385	315	546	487	524	514	86

the Balsa graph vertices. The vertices represent handshake components, which, as such, have predefined logic flow and result in predefined flow of events over the simulated handshake circuits. Such information is not incorporated into simple topological information provided by the representative graph which METIS is fed, while the objective to



(a) Inter-processor communication overhead measured as the total number of MPI messages exchanged between LPs

(b) Average idle time in LPs

Fig. 10. Hybrid: Overhead and Idle Time

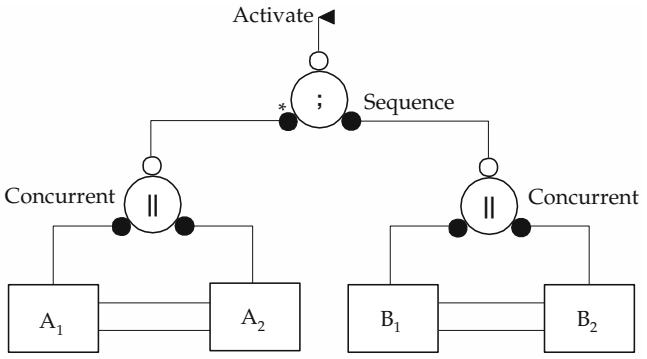


Fig. 11. An example design in which METIS is unable to exploit the inherent parallelism. A 2-way partitioning of the depicted design leads to two partitions, namely $\{A_1, A_2\}$ and $\{B_1, B_2\}$..

produce equal sized partitions with minimised edge-cuts may lead to partitions wherein (a) sequential parts of the design are divided across different LPs and (b) concurrent parts of the design are grouped into the same LP.

This is illustrated in the example of figure 11 showing a handshake circuit with three control flow components and four groups of handshake components A_1,A_2,B_1,B_2. Suppose that each of these four groups has the same number of handshake components and each component in each group is connected with at least two other components belonging to the same group. If we have two processors at our disposal, a correct partitioning decision which exploits the inherent parallelism of the design, is to assign the concurrent groups A_1 and A_2 and the concurrent groups B_1 and B_2 to different LPs. However, as METIS aims at minimising the edge-cut, the result is to bring the parallel groups A_1, A_2 together into one LP and the B_1, B_2. In other words, METIS does not exploit the inherent parallelism of the design and, therefore, fails to gain significant speedup, even in cases in which the communication cost is negligible.

5 Conclusions

In this paper we presented a quantitative analysis of multilevel partitioning algorithms for Balsa asynchronous circuit designs. This is a well recognised mainstream partitioning choice when one deals with the k-way partitioning problem of a graph and the particular library has been shown to deliver the best known outcomes. However, as our analysis has exposed, this approach is unable to capture the ing=herent parallelism of the handshake circuits and therefore yields relatively poor results for these systems. This points to the need for alternative, new partitioning algorithms that would exploit the inherent parallelism of Balsa designs and enable efficient distributed simulation of asynchronous VLSI systems.

References

1. Bailey, M.L., Briner, J.V., Chamberlain, R.D.: Parallel Logic Simulation of VLSI Systems. ACM Computing Surveys (CSUR) 26(3), 255–294 (1994)
2. Barnard, S.T., Simon, H.D.: A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems. In: 6th SIAM Conference on Parallel Processing for Scientific Computing, pp. 711–718 (March 1993)
3. Bui, T.N., Jones, C.: A Heuristic for Reducing Fill in Sparse Matrix Factorization. In: 6th SIAM Conference on Parallel Processing for Scientific Computing. pp. 445–452. SIAM (March 1993)
4. Cong, J., Smith, M.: A Parallel Bottom-Up Clustering Algorithm with Applications to Circuit Partitioning in VLSI Design. In: 30th Design Automation Conference, pp. 755–760. ACM/IEEE (June 1993)
5. Diekmann, R., Monien, B., Preis, R.: Using Helpful Sets to Improve Graph Bisections. In: Interconnection Networks and Mapping and Scheduling Parallel Computations. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 21, pp. 57–73. AMS (1995)
6. Edwards, D.A., Bardsley, A., Janin, L., Plana, L., Toms, W.: Balsa: A Tutorial Guide. Department of Computer Science, 3.5 edn., The University of Manchester (May 2006)
7. Endecott, P., Furber, S.B.: Modelling and Simulation of Asynchronous Systems Using the LARD Hardware Description Language. In: Proceedings of the 12th European Simulation Multiconference on Simulation - Past, Present and Future (ESM 1998), pp. 39–43 (June 1998)
8. Gupta, A.: Fast and Effective Algorithms for Graph Partitioning and Sparse Matrix Ordering. IBM Journal of Research and Development 41(1/2), 171–184 (1997)
9. Hauck, S., Borriello, G.: An Evaluation of Bipartitioning Techniques. In: 16th Conference on Advanced Research in VLSI (ARVLSI 1995), pp. 383–402. IEEE (March 1995)
10. Hendrickson, B., Leland, R.W.: A Multilevel Algorithm for Partitioning Graphs. In: Supercomputing Conference, ACM/IEEE (December 1995)
11. Hering, K., Rünger, G., Trautmann, S.: Modular Construction of Model Partitioning Processes for Parallel Logic Simulation. In: 30th International Workshops on Parallel Processing (ICPP 2001), pp. 99–105. IEEE (September 2001)
12. Jones, G.: Programming in occam. Prentice Hall International, Hemel Hempstead (1987)
13. Karypis, G., Kumar, V.: A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM Journal on Scientific Computing 20(1), 359–392 (1998)
14. Karypis, G., Kumar, V.: Multilevel k-way Partitioning Scheme for Irregular Graphs. Journal of Parallel and Distributed Computing 48(1), 96–129 (1998)

15. Karypis, G., Kumar, V.: METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, 4.0 edn. Department of Computer Science, University of Minnesota (1998)
16. Kim, H.K.: Parallel Logic Simulation of Digital Circuits. Ph.D. thesis, Wright State University (1998)
17. Kim, H.K., Jean, J.S.: Concurrency Preserving Rartitioning (CPP) for Parallel Logic Simulation. In: 10th Workshop on Parallel and Distributed Simulation (PADS 1996), pp. 98–105 (May 1996)
18. Krishnaswamy, V., Hasteer, G., Banerjee, P.: Automatic Parallelization of Compiled Event Driven VHDL Simulation. IEEE Transactions on Computers 51(4), 380–394 (2002)
19. Levendel, Y.H., Menon, P.R., Patel, S.H.: Special Purpose Computer for Logic Simulation Using Distributed Processing. Bell System Technical Journal 61, 2873–2907 (1982)
20. Monien, B., Preis, R., Diekmann, R.: Quality Matching and Local Improvement for Multilevel Graph Partitioning. Parallel Computing 26(12), 1609–1634 (2000)
21. Monien, B., Schambeger, S.: Graph Partitioning with the Party Library: Helpful-Sets in Practice. In: Proceedings of the 16th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2004), pp. 198–205. IEEE (October 2004)
22. Mueller-Thuns, R.B., Saab, D.G., Damiano, R.F., Abraham, J.A.: VLSI Logic and Fault Simulation on General-Purpose Parallel Computers. IEEE Transactions on CAD of Integrated Circuits and Systems 12(3), 446–460 (1993)
23. Omondi, A.R.: Computer Arithmetic Systems: Algorithms, Architecture and Implementation. Prentice-Hall, Englewood Cliffs (1994)
24. Pellegrini, F.: Scotch and libScotch 4.0 User's Guide. University of Bordeaux I (January 2006)
25. Schamberger, S.: Improvements to the Helpful-Set Heuristic and a New Evaluation Scheme for Graphs Partitioners. In: Kumar, V., Gavrilova, M.L., Tan, C.J.K., L'Ecuyer, P. (eds.) ICCSA 2003, Part I. LNCS, vol. 2667, pp. 49–59. Springer, Heidelberg (2003)
26. Sevcik, K.: Characterizations of Parallelism in Applications and Their Use in Scheduling. In: International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 1989), pp. 171–180. ACM (1989)
27. Smith, S.P., Underwood, B., Mercer, M.R.: An Analysis of Several Approaches to Circuit Partitioning for Parallel Logic Simulation. In: International Conference on Computer Design, pp. 664–667. IEEE, Rye Brook (1987)
28. Sporrer, C., Bauer, H.: Corolla Partitioning for Distributed Logic Simulation of VLSI-Circuits. In: 7th Workshop on Parallel and Distributed Simulation (PADS 1993), pp. 85–92. IEEE (1993)
29. Subramanian, S., Rao, D.M., Wilsey, P.A.: Applying Multilevel Partitioning to Parallel Logic Simulation. Parallel and Distributed Discrete Event Simulation, pp. 49–78. Nova Science, Hauppauge (2002)
30. Tsirogiannis, E., Theodoropoulos, G., Chen, D., Zhang, Q., Janin, L., Edwards, D.A.: A Framework for Distributed Simulation of Asynchronous Handshake Circuits. In: Proceedings of 39th Annual Simulation Symposium (ANSS 2006), pp. 214–222. IEEE (April 2006)
31. Walshaw, C., Cross, M.: Mesh Partitioning: A Multilevel Balancing and Refinement Algorithm. SIAM Journal on Scientific Computing 22(1), 63–80 (2000)
32. Zhu, L., Chen, G., Szymanski, B.K., Tropper, C., Zhang, T.: Parallel Logic Simulation of Million-Gate VLSI Circuits. In: MASCOTS, pp. 521–524. IEEE (September 2005)

A Parallel Variant of BiCGStar-Plus Method Reduced to Single Global Synchronization

Seiji Fujino[1,⋆] and Keiichi Murakami[2]

[1] Research Institute for Information Technology, Kyushu University
fujino@cc.kyushu-u.ac.jp
[2] Sumitomo Rubber Industries, Ltd.

Abstract. In this paper, we propose new product-type iterative methods by introducing the BiCGSafe strategy, i.e., utilization of associate residual in place of residual, to the variants of GPBiCG. We refer to BiCGStar (BiCG with stabilization of associate residual) and its improved version of BiCGStar method. BiCGStar and BiCGStar-plus methods outperform compared with the conventional iterative methods. Moreover, our proposed methods are suited to parallel computer with distributed memory systems, since they require single global synchronization per one iteration.

Keywords: iterative method, synchronization, parallel computer.

1 Introduction

The GPBiCG[8] method is well known product type iterative method for solving $A\boldsymbol{x} = \boldsymbol{b}$, where A is $n \times n$ sparse non-symmetric matrix, $\boldsymbol{x}$ and $\boldsymbol{b}$ are a solution and right-hand side vector of order n, respectively. Recently, four variants of GPBiCG based on IDR formulation [7][6] have been proposed by Prof. Abe[1]. Their variants shows better convergence rate than the original GPBiCG in many numerical experiments. On the other hand, there exists an alternative approach to improve GPBiCG. Prof. Fujino[4] has derived BiCGSafe method by reordering the construction of polynomials of the original GPBiCG in order to avoid complexed order of stabilized polynomial sequence.

In this paper, we give new product-type iterative methods by introducing the BiCGSafe strategy to the variants of GPBiCG. Our proposed methods are suitable for distributed memory systems, since they require single global synchronization per one iteration.

2 BiCGStar Method

The kth residual $\boldsymbol{r}_k := \boldsymbol{b} - A\boldsymbol{x}_k$ of product type iterative methods is expressed as

$$\boldsymbol{r}_k = H_k(A)R_k(A)\boldsymbol{r}_0. \tag{1}$$

⋆ Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 325–332, 2013.

Here, H_k and R_k are polynomials of degree k called stabilization polynomial and bi-Lanczos polynomial, respectively. Following [1], we use bi-Lanczos polynomial of the form:

$$\begin{cases} R_0(\lambda) = 1, \quad P_0(\lambda) = 1, \\ R_{k+1}(\lambda) = R_k(\lambda) - \alpha_k P_k(\lambda), \\ P_{k+1}(\lambda) = R_{k+1}(\lambda) - \beta_k P_k(\lambda), \quad k = 1, 2, \ldots \end{cases} \tag{2}$$

Stabilized polynomial is given as the next three-term recurrence.

$$\begin{cases} H_0(\lambda) = 1, \quad H_1(\lambda) = (1 - \zeta_0\lambda)H_0(\lambda), \\ H_{k+1}(\lambda) = (1 + \eta_k - \zeta_k\lambda)H_k(\lambda) - \eta_k H_{k-1}(\lambda), \\ k = 1, 2, \ldots \end{cases} \tag{3}$$

Parameters ζ_k, η_k are determined by means of local minimization of 2-norm of associate residual $\boldsymbol{a\_r}_k := H_{k+1}(A)R_k(A)\boldsymbol{r}_0$. α_k, β_k are computed such that $H_k(A)R_{k+1}(A)\boldsymbol{r}_0$ and $AH_k(A)P_{k+1}(A)\boldsymbol{r}_0$ are orthogonal to initial shadow residual $\tilde{\boldsymbol{r}}_0$. From the eqn. (2),

$$H_k(A)R_{k+1}(A)\boldsymbol{r}_0 = H_k(A)R_k(A)\boldsymbol{r}_0 - \alpha_k AH_k(A)P_k(A)\boldsymbol{r}_0 \perp \tilde{\boldsymbol{r}}_0, \tag{4}$$

$$AH_k(A)P_{k+1}(A)\boldsymbol{r}_0 = AH_k(A)R_{k+1}(A)\boldsymbol{r}_0 - \beta_k AH_k(A)P_k(A)\boldsymbol{r}_0 \perp \tilde{\boldsymbol{r}}_0. \tag{5}$$

Then, parameters α_k and β_k can be computed as

$$\alpha_k = \frac{(\tilde{\boldsymbol{r}}_0, H_k(A)R_k(A)\boldsymbol{r}_0)}{(\tilde{\boldsymbol{r}}_0, AH_k(A)P_k(A)\boldsymbol{r}_0)}, \tag{6}$$

$$\beta_k = \frac{(\tilde{\boldsymbol{r}}_0, AH_k(A)R_{k+1}(A)\boldsymbol{r}_0)}{(\tilde{\boldsymbol{r}}_0, AH_k(A)P_k(A)\boldsymbol{r}_0)}. \tag{7}$$

3 Reduction of Synchronization Points of BiCGStar Method

For reduction of synchronization points of BiCGStar method, we improve formulation of parameter β_k as the above mentioned equation (7). For parameter β_k, we can derive alternative expression. $H_k(\lambda)P_{k+1}(\lambda)$ can be written as

$$\begin{aligned} H_k(\lambda)P_{k+1}(\lambda) &= H_k(\lambda)R_{k+1}(\lambda) - \beta_k H_k(\lambda)P_k(\lambda), \\ &= H_k(\lambda)R_k(\lambda) - \alpha_k\lambda H_k(\lambda)P_k(\lambda) - \beta_k H_k(\lambda)P_k(\lambda). \end{aligned} \tag{8}$$

With the equation (8) and the relation of $(\boldsymbol{x}, A\boldsymbol{y}) = (A^T\boldsymbol{x}, \boldsymbol{y})$, we obtain

$$\begin{aligned} \beta_k &= \frac{(\tilde{\boldsymbol{r}}_0, A(H_k(A)R_k(A)\boldsymbol{r}_0 - \alpha_k AH_k(A)P_k(A)\boldsymbol{r}_0))}{(\tilde{\boldsymbol{r}}_0, AH_k(A)P_k(A)\boldsymbol{r}_0)}, \\ &= \frac{(A^T\tilde{\boldsymbol{r}}_0, H_k(A)R_k(A)\boldsymbol{r}_0) - \alpha_k(A^T\tilde{\boldsymbol{r}}_0, AH_k(A)P_k(A)\boldsymbol{r}_0))}{(\tilde{\boldsymbol{r}}_0, AH_k(A)P_k(A)\boldsymbol{r}_0)}. \end{aligned} \tag{9}$$

Although the equation (9) needs two extra inner products, the coefficients α_k, β_k can be computed at the same place. This means the number of global synchronization points can be reduced. Namely, number of global synchronization of BiCGStar and BiCGStar-plus methods is single only.

We introduce auxiliary vectors $\boldsymbol{p} := H_k(A)P_k(A)\boldsymbol{r}_0$, $\boldsymbol{w}_k := H_{k+1}(A)P_k(A)\boldsymbol{r}_0$, $\boldsymbol{u}_k := H_k(A)R_{k+1}(A)\boldsymbol{r}_0$, $\boldsymbol{q}_k := H_k(A)P_{k+1}(A)\boldsymbol{r}_0$, and put $\boldsymbol{r}' = H_{k+1}(A)R_k(A)\boldsymbol{r}_0$ which is same as the associate residual. Approximate solution $\boldsymbol{x}_k$ is updated with auxiliary vectors $\boldsymbol{t}_k$ and $\boldsymbol{v}_k$ such that $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$, $\boldsymbol{r}'_k = \boldsymbol{b} - A\boldsymbol{t}_k$, $\boldsymbol{u}_k = \boldsymbol{b} - A\boldsymbol{v}_k$. Then, we give an algorithm and name it BiCGStar (BiCG method using STabilized Associate Residual) method:

Algorithm 1. BiCGStar method

1. Let $\boldsymbol{x}_0$ be an initial guess, Compute $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, choose $\tilde{\boldsymbol{r}}_0$
2. Compute $A\boldsymbol{r}_0$, $A^T\tilde{\boldsymbol{r}}_0$, $\boldsymbol{p}_0 = \boldsymbol{r}_0$, $A\boldsymbol{p}_0 = A\boldsymbol{r}_0$, $\boldsymbol{u}_{-1} = \boldsymbol{q}_{-1} = \boldsymbol{v}_{-1} = \boldsymbol{0}$
3. **for** $k = 0, 1, \ldots$ **do**
4. $\quad \boldsymbol{y}_k = \boldsymbol{u}_k - \boldsymbol{r}_k$
5. $\quad$ **if** $\|\boldsymbol{r}_k\|/\|\boldsymbol{r}_0\| \leq \epsilon$ **stop**
6. $\quad \alpha_k = (\tilde{\boldsymbol{r}}_0, \boldsymbol{r}_k)/(\tilde{\boldsymbol{r}}_0, A\boldsymbol{p}_k)$
7. $\quad \beta_k = \dfrac{(A^T\tilde{\boldsymbol{r}}_0, \boldsymbol{r}_k) - \alpha_k(A^T\tilde{\boldsymbol{r}}_0, A\boldsymbol{p}_k)}{(\tilde{\boldsymbol{r}}_0, A\boldsymbol{p}_k)}$
8. $\quad \zeta_k = \dfrac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(A\boldsymbol{r}_k, \boldsymbol{r}_k) - (A\boldsymbol{r}_k, \boldsymbol{y}_k)(\boldsymbol{y}_k, \boldsymbol{r}_k)}{(A\boldsymbol{r}_k, A\boldsymbol{r}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (A\boldsymbol{r}_k, \boldsymbol{y}_k)(\boldsymbol{y}_k, A\boldsymbol{r}_k)}$
9. $\quad \eta_k = \dfrac{(A\boldsymbol{r}_k, A\boldsymbol{r}_k)(\boldsymbol{y}_k, \boldsymbol{r}_k) - (A\boldsymbol{r}_k, \boldsymbol{y}_k)(A\boldsymbol{r}_k, \boldsymbol{r}_k)}{(A\boldsymbol{r}_k, A\boldsymbol{r}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (A\boldsymbol{r}_k, \boldsymbol{y}_k)(\boldsymbol{y}_k, A\boldsymbol{r}_k)}$
$\quad$ (**if** $k = 0$ **then** $\zeta_k = (A\boldsymbol{r}_k, \boldsymbol{r}_k)/(A\boldsymbol{r}_k, A\boldsymbol{r}_k)$, $\eta_k = 0$)
10. $\quad \boldsymbol{r}'_k = (1 + \eta_k)\boldsymbol{r}_k - \zeta_k A\boldsymbol{r}_k - \eta_k \boldsymbol{u}_{k-1}$
11. $\quad \boldsymbol{t}_k = (1 + \eta_k)\boldsymbol{x}_k + \zeta_k \boldsymbol{r}_k - \eta_k \boldsymbol{v}_{k-1}$
12. $\quad \boldsymbol{w}_k = (1 + \eta_k)\boldsymbol{p}_k - \zeta_k A\boldsymbol{p}_k - \eta_k \boldsymbol{q}_{k-1}$
13. $\quad$ Compute $A\boldsymbol{w}_k$
14. $\quad \boldsymbol{u}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k$
15. $\quad \boldsymbol{v}_k = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
16. $\quad \boldsymbol{r}_{k+1} = \boldsymbol{r}'_k - \alpha_k A\boldsymbol{w}_k$
17. $\quad \boldsymbol{x}_{k+1} = \boldsymbol{t}_k + \alpha_k \boldsymbol{w}_k$
18. $\quad$ Compute $A\boldsymbol{r}_{k+1}$
19. $\quad \boldsymbol{q}_k = \boldsymbol{u}_k - \beta_k \boldsymbol{p}_k$
20. $\quad \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} - \beta_k \boldsymbol{w}_k$
21. $\quad A\boldsymbol{p}_{k+1} = A\boldsymbol{r}_{k+1} - \beta_k A\boldsymbol{w}_k$
22. **end do**

Coupled two-term recurrences of Rutishauser[1][5] has been used for stabilized polynomial.

$$\begin{cases} \tilde{G}_0(\lambda) = 0, \quad H_0(\lambda) = 1, \\ \tilde{G}_{k+1}(\lambda) = \zeta_k \lambda H_k(\lambda) + \eta_k \tilde{G}_k(\lambda), \\ H_{k+1}(\lambda) = H_k(\lambda) - \tilde{G}_{k+1}(\lambda), \quad k = 0, 1, \ldots \end{cases} \tag{10}$$

Here, auxiliary polynomial $\tilde{G}_k(\lambda)$ is defined as

$$\tilde{G}_k(\lambda) := H_k(\lambda) - H_{k+1}(\lambda), \quad k = 0, 1, \ldots \tag{11}$$

We refer to the BiCGStar method using the equation (10) instead of the equation (3) as BiCGStar-plus method. Introducing the auxiliary vectors $\boldsymbol{y}_k := \tilde{G}_k(A)R_k(A)\boldsymbol{r}_0$, $\boldsymbol{z}_k := \tilde{G}_{k+1}(A)R_k(A)\boldsymbol{r}_0$, $\boldsymbol{s}_k := \tilde{G}_k(A)P_k(A)\boldsymbol{r}_0$, $\boldsymbol{c}_k := \tilde{G}_{k+1}(A)P_k(A)\boldsymbol{r}_0$, the algorithm of BiCGStar-plus is expressed as below.

Algorithm 2. BiCGStar-plus method

1. Let $\boldsymbol{x}_0$ be an initial guess, Compute $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, choose $\tilde{\boldsymbol{r}}_0$
2. Compute $A\boldsymbol{r}_0$, $A^T\tilde{\boldsymbol{r}}_0$, $\boldsymbol{p}_0 = \boldsymbol{r}_0$, $A\boldsymbol{p}_0 = A\boldsymbol{r}_0$, $\boldsymbol{y}_0 = \boldsymbol{s}_0 = \boldsymbol{t}_0 = \boldsymbol{0}$
3. **for** $k = 0, 1, \ldots$ **do**
4. **if** $\|\boldsymbol{r}_k\|/\|\boldsymbol{r}_0\| \leq \epsilon$ **stop**
5. $\alpha_k = (\tilde{\boldsymbol{r}}_0, \boldsymbol{r}_k)/(\tilde{\boldsymbol{r}}_0, A\boldsymbol{p}_k)$
6. $\beta_k = \dfrac{(A^T\tilde{\boldsymbol{r}}_0, \boldsymbol{r}_k) - \alpha_k(A^T\tilde{\boldsymbol{r}}_0, A\boldsymbol{p}_k)}{(\tilde{\boldsymbol{r}}_0, A\boldsymbol{p}_k)}$
7. $\zeta_k = \dfrac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(A\boldsymbol{r}_k, \boldsymbol{r}_k) - (A\boldsymbol{r}_k, \boldsymbol{y}_k)(\boldsymbol{y}_k, \boldsymbol{r}_k)}{(A\boldsymbol{r}_k, A\boldsymbol{r}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (A\boldsymbol{r}_k, \boldsymbol{y}_k)(\boldsymbol{y}_k, A\boldsymbol{r}_k)}$
8. $\eta_k = \dfrac{(A\boldsymbol{r}_k, A\boldsymbol{r}_k)(\boldsymbol{y}_k, \boldsymbol{r}_k) - (A\boldsymbol{r}_k, \boldsymbol{y}_k)(A\boldsymbol{r}_k, \boldsymbol{r}_k)}{(A\boldsymbol{r}_k, A\boldsymbol{r}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (A\boldsymbol{r}_k, \boldsymbol{y}_k)(\boldsymbol{y}_k, A\boldsymbol{r}_k)}$
   (**if** $k = 0$ **then** $\zeta_k = (A\boldsymbol{r}_k, \boldsymbol{r}_k)/(A\boldsymbol{r}_k, A\boldsymbol{r}_k)$, $\eta_k = 0$)
9. $\boldsymbol{v}_k = \zeta_k\boldsymbol{r}_k + \eta_k\boldsymbol{t}_k$
10. $\boldsymbol{z}_k = \zeta_k A\boldsymbol{r}_k + \eta_k\boldsymbol{y}_k$
11. $\boldsymbol{c}_k = \zeta_k A\boldsymbol{p}_k + \eta_k\boldsymbol{s}_k$
12. Compute $A\boldsymbol{c}_k$
13. $\boldsymbol{w}_k = \boldsymbol{p}_k - \boldsymbol{c}_k$
14. $A\boldsymbol{w}_k = A\boldsymbol{p}_k - A\boldsymbol{c}_k$
15. $\boldsymbol{t}_{k+1} = \boldsymbol{v}_k - \alpha_k\boldsymbol{c}_k$
16. $\boldsymbol{y}_{k+1} = \boldsymbol{z}_k - \alpha_k A\boldsymbol{c}_k$
17. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{v}_k + \alpha_k\boldsymbol{w}_k$
18. $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \boldsymbol{z}_k - \alpha_k A\boldsymbol{w}_k$
19. Compute $A\boldsymbol{r}_{k+1}$
20. $\boldsymbol{s}_{k+1} = \boldsymbol{y}_{k+1} - \beta_k\boldsymbol{c}_k$
21. $\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} - \beta_k\boldsymbol{w}_k$
22. $A\boldsymbol{p}_{k+1} = A\boldsymbol{r}_{k+1} - \beta_k A\boldsymbol{w}_k$
23. **end do**

Table 1 summarizes the number of vector operations and synchronization points per one iteration.

Table 1. Number of vector operations and synchronization points per one iteration

method	Mv	Dot	AXPY	Sync.
GPBiCG	2	8	14.0	3
GPBiCG variant 1	2	8	14.5	2
GPBiCG variant 2	2	8	14.0	2
GPBiCG variant 3	2	8	13.0	2
GPBiCG variant 4	2	8	13.5	2
BiCGSafe	2	8	13.5	2
BiCGStar	2	10	14.5	**1**
BiCGStar-plus	2	10	13.5	**1**

4 Numerical Experiments

In this section, we present numerical results. All computations were performed on Fujitsu Primergy CX400 (Intel Xeon E5-2690, memory: 128GB, OS: Red Hat Linux Enterprise). Fujitsu Fortran Compiler with "-Kfast" optimization option was used. In all cases, the iteration was started with initial guess $\boldsymbol{x}_0 = \boldsymbol{0}$. The right-hand side $\boldsymbol{b}$ was imposed from physical load conditions. We set maximum iterations as 50,000, stopping criterion as $\|\boldsymbol{r}_{k+1}\|_2/\|\boldsymbol{r}_0\|_2 \leq 10^{-8}$. Initial shadow residual was equal to the initial residual. Note that we applied diagonal scaling to the test matrices.

Table 2 shows the specifications of test matrices. All matrices are taken from Florida Sparse Matrix Collection [3]. In Table 2, "nnz" means number of nonzero entries of each matrix. "n" means dimension of matrix.

Table 2. Specifications of test matrices

matrix	n	nnz	nnz/n	analytical field
atmosmodd	1,270,432	8,814,880	6.9	fluid dynamics
epb3	84,617	463,625	5.5	thermal
Freescale1	3,428,755	18,920,347	5.5	circuit simulation
thermomech_dK	204,316	2,846,228	13.9	thermal
xenon1	48,600	1,181,120	24.3	structural
water_tank	60,740	2,035,281	33.5	fluid dynamics
sme3Da	12,504	874,887	70.0	structural

In Table 3, we compared the number of matrix-vector multiplications and CPU time of six iterative methods. "TRR" represents the True Relative Residual defined as $\|\boldsymbol{b} - A\boldsymbol{x}_{k+1}\|_2/\|\boldsymbol{b} - A\boldsymbol{x}_0\|_2$ in the $\log_{10}(\ )$ scale.

We can observe from Table 3 as below.

1. For matrices epb3, xenon1 and water_tank, BiCGStar-plus method converged the fastest among the tested methods.
2. Approximate solutions of variant 2 of GPBiCG and BiCGStar was not enough accurate for matrices thermomech_dK, sme3Da, and GPBiCG also output incorrect solution for sme3Da.

Table 3. Number of matrix-vector multiplications (Mv) and CPU time of six iterative methods

matrix	method	Mv	CPU time [sec.]	ratio	TRR
epb3	GPBiCG	4,042	6.56	1.00	-8.1
	GPBiCG variant 2	4,124	5.98	0.91	-8.0
	GPBiCG variant 4	3,412	4.81	0.73	-8.1
	BiCGSafe	3,758	4.59	0.70	-8.0
	BiCGStar	4,388	6.11	0.93	-8.0
	BiCGStar-plus	3,716	**4.50**	0.69	-8.0
Freescale1	GPBiCG	10,102	838.14	1.00	-8.0
	GPBiCG variant 2	9,332	856.38	1.02	-8.0
	GPBiCG variant 4	8,942	784.89	0.94	-8.0
	BiCGSafe	9,230	**705.00**	0.84	-8.0
	BiCGStar	9,370	782.37	0.93	-8.0
	BiCGStar-plus	9,552	705.60	0.84	-8.0
thermomech_dK	GPBiCG	14,604	136.06	1.00	-8.1
	GPBiCG variant 2	15,392	146.41	1.08	(-6.1)
	GPBiCG variant 4	9,576	90.93	0.67	-8.1
	BiCGSafe	9,436	**84.63**	0.62	-8.0
	BiCGStar	16,808	160.34	1.18	(-6.1)
	BiCGStar-plus	9,246	86.24	0.63	-8.2
xenon1	GPBiCG	1,436	2.79	1.00	-8.0
	GPBiCG variant 2	1,258	2.53	0.91	-8.0
	GPBiCG variant 4	1,306	2.52	0.90	-8.0
	BiCGSafe	1,460	2.64	0.95	-8.0
	BiCGStar	1,362	2.75	0.98	-8.0
	BiCGStar-plus	1,320	**2.48**	0.89	-8.0
water_tank	GPBiCG	2,794	10.41	1.00	-8.0
	GPBiCG variant 2	2,578	9.85	0.95	-8.0
	GPBiCG variant 4	2,782	10.31	0.99	-8.0
	BiCGSafe	3,186	11.40	1.10	-8.0
	BiCGStar	2,756	10.49	1.01	-8.0
	BiCGStar-plus	2,602	**9.56**	0.92	-8.1
sme3Da	GPBiCG	6,086	9.26	1.00	(-6.8)
	GPBiCG variant 2	5,488	7.98	0.86	(-6.0)
	GPBiCG variant 4	3,994	6.64	0.72	-8.0
	BiCGSafe	3,958	**5.46**	0.59	-8.0
	BiCGStar	6,332	9.42	1.02	(-6.1)
	BiCGStar-plus	4,106	5.95	0.64	-8.1

Table 4. Comparison of average ratios of matrix-vector multiplications and CPU time

method	GPBiCG	variant 2	variant 4	BiCGSafe	BiCGStar	BiCGStar-plus
Mv ratio	1.00	0.95	**0.81**	0.86	1.02	0.83
time ratio	1.00	0.95	0.82	0.78	1.01	**0.76**

3. However, TRR of variant 4 of GPBiCG, BiCGSafe and BiCGStar-plus reached to less than 10^{-8} for all test matrices.
4. In matrices Freescale1 and thermomech_dk, the computational time of BiCGStar-plus is significantly less than that of GPBiCG in a single processor and great decreasing the elapsed time ratio of matrix thermomech_dk shown in Fig. 3.

Table 4 presents the comparison of average ratios of matrix-vector multiplications and CPU time. From Table 4, we can learn that the variant 4 of GPBiCG, BiCGSafe and BiCGStar-plus are more stable and faster than other methods.

We present speedup of four iterative methods for matrix atmosmodd as number of processors increased as shown in Figure 1. We see that speedup of four iterative methods is scalable for matrix atmosmodd. In particular, the slope of speedup seems to change around 100 processors. We show also elapsed time ratio of four iterative methods for matrix atmosmodd in Figure 2. Ratio means elapsed time of GPBiCG method to that of other iterative methods at each number of processors. It is clear that ratios of BiCGSafe and BiCGStar-plus methods are effective compared with GPBiCG and GPBiCG_v4. Moreover we demonstrate elapsed time ratio of four iterative methods for seven matrices at 128 processors in Figure 3. It turned out that BiCGStar-plus method outperforms compared

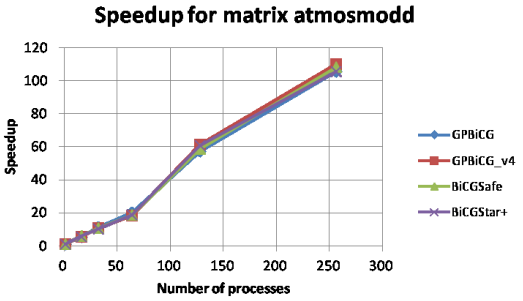


Fig. 1. Speedup of four iterative methods for matrix atmosmodd

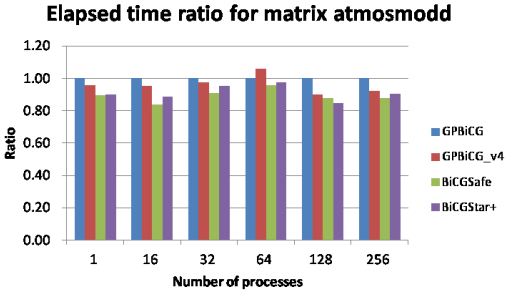


Fig. 2. Elapsed time ratio of four iterative methods for matrix atmosmodd

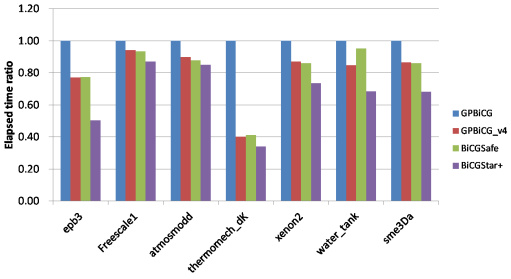


Fig. 3. Elapsed time ratio of four iterative methods for seven matrices at 128 processors

with other three iterative methods due to reduction of number of synchronization points per one iteration from the results as shown in Figure 3.

5 Conclusions

We proposed BiCGStar and BiCGStar-plus method, and evaluated their performance on a single processor. In our numerical examples, BiCGStar-plus method outperforms compared with the original GPBiCG method, and competitive with variants of GPBiCG and BiCGSafe method from the view point of accuracy and computational times. Moreover parallel performance of BiCGStar methods was excellent compared with other iterative methods on parallel computer with distributed memory.

References

1. Abe, K., Sleijpen, G.L.G.: Solving linear equations with a stabilized GPBiCG mehtod. Appl. Numer. Math. 67, 4–16 (2013)
2. Abe, K., Sleijpen, G.L.G.: Hybrid Bi-CG methods with Bi-CG formulation closer to the IDR approach. Appl. Math. Comput. 218, 10889–10899 (2012)
3. Florida Sparse Matrix Collection, `http://www.cise.ufl.edu/research/sparse/matrices/index.html`
4. Fujino, S., Fujiwara, M., Yoshida, M.: A proposal of preconditioned BiCGSafe method with safe convergence. In: Proc. of the 17th IMACS World Congress on Scientific Computation, Appl. Math. Simul. CD-ROM, Paris (2005)
5. Rutishauser, H.: Theory of gradient method, Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems. Mitt. Inst. Angew. Math, pp. 24–49. ETH Zürich, Birkhäuser (1959)
6. Sleijpen, G.L.G., Sonneveld, P., van Gijzen, M.B.: Bi-CGSTAB as an induced dimension reduction method. Appl. Numer. Math. 60, 1100–1114 (2010)
7. Sonneveld, P., van Gijzen, M.B.: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. 31(2), 1035–1062 (2008)
8. Zhang, S.-L.: GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems. SIAM J. Sci. Comput. 18, 537–551 (1997)

Construction of Simulation Environment Based on Augmented Reality Technique

Hanyu Xue[1], Hongyan Quan[1,*], Xiao Song[2], and Maomao Wu[1]

[1] East China Normal University, No. 3663 Zhongshan North Road, Shanghai, China
hyquan@sei.ecnu.edu.cn
[2] Beihang University, No. 37 College Road, Beijing, China
songxiao@buaa.edu.cn

Abstract. To meet the requirement of interaction and real time modeling in simulation scenarios, a strategy for scenarios construction is proposed. We take advantage of the technique of augmented reality to study. In order to constitute the scene we reconstruct fluid surface from video combining the fluid motion vectors with LBM (Lattice Boltzmann Method) and refine the height field by interpolating distribution of the fluid particle. Based on the purpose of virtual and real object combination, this study proposes a method of tracing the gold feature points between frames using continuity of particle movement. The strategy of dichotomy is taken to refine the camera intrinsic parameters and SBA (Sparse Bundle Adjustment) method is taken to optimize the intrinsic and extrinsic parameters. Further experiment results demonstrate that it is a valid and efficient method for simulation environment constructing.

Keywords: simulation, augmented reality, construction, SBA, LBM.

1 Introduction and Related Works

3D realistic fluid modeling, simulation have different applications in various domains, including urban planning and designing, special effects of films and televisions, tourism exhibition, military training system, disasters predicting and engineering visualization.

In order to achieve the seamless integration and interactivity with objects of the scene in real-time, fluid reconstruction techniques can satisfy this requirement. The reconstruction of the natural landscape fluid is necessary to enhance the reality of virtual scene.

Compared to the existing method of reconstruction, the proposed method has the following characteristics: (1)It can carry out the reconstruction with a single video taken by ordinary capturing devices. (2)The reconstruction results meet the physical movement of the fluid. It can improve realism to the simulation scene. (3)It is conducive to interaction with the environment. (4)The mapping textures from video have the realism of objective environment.

* Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 333–339, 2013.

The traditional methods for water surface reconstruction include physical properties based methods, shape from stereo methods and geometric constraint methods. Heas [1] used optimizing method to estimate 3D motions of atmospheric image. Nagai [2] established a large database and map the texture to reconstruct 3D world and reconstructed from single image. With the development of techniques, people proposed a method of Shape from Shading (abbreviated as SFS). Frankot et al.[3] proposed a valid algorithm and get satisfactory reconstruction results. David et al.[4] studied a method for reconstructing water from real video footage, they can obtain more informative 3D results from a wider range of possible scenes.

A new method for simulation scene construction is studied here. We use optimization method to recovery camera parameters and the simulation environment can be constructed.This paper proceeds as follow. Section 2 describes the method of reconstruction from video. Section 3 describes the method of feature points tracking and the method of recovering the intrinsic and extrinsic parameter of camera using dichotomy optimization. In section 4 we demonstrate the experimental result of the proposed method. The performance of running time and comparing with existing method are also discussed in Section 4, and it is concluded in Section 5.

2 Reconstruction of Fluid from Video

Fluid surface reconstruction is a fundamental task in virtual scene modeling. Due to the particularity of fluid motion, the complexity is introduced to the study of fluid reconstruction. We use both fluid motion vectors as well as LBM (Lattice Boltzmann Method, LBM) to study. To obtain smoothing motion vectors field, we initialize the motion vectors using regional correlation method to study. Then we use LBM to calculate the height field from motion vectors of particles.

2.1 Motion Vector Initialization

In this study, a region-based image corresponding method [5] is adopted to initialize the motion vector. There are two stages in the processing, low level process and advanced level process. Then a cluster strategy is taken to obtain more accurate results.

Feature vector is defined as $\boldsymbol{vector} = [x, y, u, v, sigu, sigv]$, Where (x, y) denotes the position of the fluid particles, u and v denote the x-direction component and y-direction component of motion vectors, $sigu$ and $sigv$ denote the sign of u and v respectively. It takes 1,-1 and 0 corresponding to negative, positive and zero components respectively.

The algorithm of fluid motion vector initializing is described as follows:

1) Calculate the fluid motion vector between two adjacent frames using region based correlation method.
2) Cluster the fluid motion vectors obtained from last step.
3) Statistic the counts of each cluster.
4) For each particle, find the closest neighborhood if they have clustered

After these four steps, the initialization results of fluid motion vectors can be obtained.

2.2 Calculating Height Field of Fluid

We then take advantage of D2Q9 model to our research work. In this model each particle is driven by its neighborhoods and for every eight directions, it can also effect on the others. If the join force shows as tensile force, the height of the particle is regarded as 0. If the join force shows as extrusion force, the height is decided by the magnitude of join force. Many literatures has discussed the process of the reconstruction of fluid using motion vector, here we take the method described by Quan [6].From the model distribution function $f_i(\mathbf{r},t)$of position $\boldsymbol{r}$ at time t can be calculated. We use the following formula to calculate the height of the particle.

$$h(x,t)=\sum_{i=0}^{8} f_{iu}(x,t)+\sum_{i=0}^{8} f_{iv}(x,t) \tag{1}$$

$f_{iu}(x,t)$ denotes horizontal component of $f_i(x,t)$ and $f_{iv}(x,t)$ vertical component of $f_i(x,t)$.

3 Construction of Simulation Environment

3.1 Feature Points Extraction

Extraction of feature points is a key step in simulation scene construction. It will play an important role in 3D scene construction. Currently, people usually match the feature points from the algorithm of Harris and SIFT. Because of illumination and shelter errors will be brought in the produce of detecting and matching. In order to obtain more accurate result, we use the continuity of the movement as a constraint to optimize the feature point. Figure1 demonstrates the constraints of motion continuity. It is an example containing 5 frames.

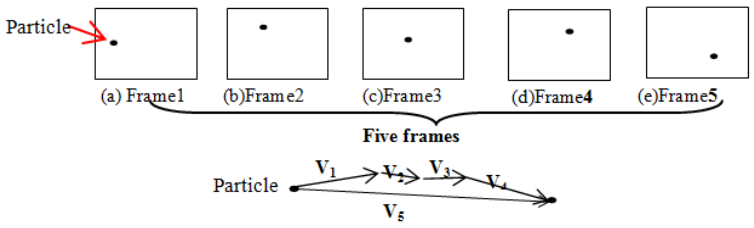


Fig. 1. Constraint of Movement Continuity

In figure 1, the positions of a particle in Frame1, Frame2, Frame3, Fame4 and Frame5 are signed in the figure respectively. $\boldsymbol{V}_1,\boldsymbol{V}_2,\boldsymbol{V}_3$ and $\boldsymbol{V}_4$ denote the motion vectors of each adjacent frames. $\boldsymbol{V}_5$ denotes the motion vector between Frame1 and Frame 5. According to the constraint of motion continuity, the motion vectors should satisfy the following equation.

$$\boldsymbol{V}_5=\boldsymbol{V}_1+\boldsymbol{V}_2+\boldsymbol{V}_3+\boldsymbol{V}_4 \tag{2}$$

In the study we set the threshold as 150 and if the deference between vectors are larger than threshold it is considered as an imprecise value. This method can be applied to extracting feature tracking points from video and can keep from noise. The results can also be used to recover extrinsic camera parameters.

3.2 Recoveries of Camera Parameters

According to epipolar geometry, fundamental matrix can constrain the adjacent views from different viewpoints. Accurate calculation of fundamental matrix is seriously important to intrinsic and extrinsic parameters recovery. There are eight parameters need to calculate in fundamental matrix. In this study, we take advantage of RANSAC (Random Sample Consensus, RANSAC) method to estimate fundamental matrix [6], camera parameters can be obtained further. The linear algorithm proposed in [6] can deal with varying intrinsic camera parameters.

However problems still exists. In the dynamic tracking procedure, it is necessary to obtain more accurate camera parameters and the camera parameters optimization is necessary. We take advantage the combination strategy of dichotomy and SBA to optimize camera parameters. In order to obtain more satisfied results, it adopts reprojection error to measure the accuracy of optimal result. The reprojection error is defined as:

$$error = \frac{1}{n}\sum_{i=1}^{n}\sqrt{d^2(m_i,\hat{m}_i)+d^2(m'_i,\hat{m}_i')} \tag{3}$$

m_i and m_i' are corresponding point, $\hat{m}_i$ and $\hat{m}_i'$ denote the reprojections of 3D reconstruction of the feature points m_i and m_i' respectively. $d(m_i,\hat{m}_i)$ denotes the Euclidean distance between m_i and $\hat{m}_i$, $d(m_i',\hat{m}_i')$ denotes Euclidean distance between m_i' and $\hat{m}_i'$, n denotes the number of feature points.

It is very important to recover focal length of the camera for calculating intrinsic parameter accurately. In this study, we use dichotomy to optimize focal length. The optimizing algorithm of focal length can be described as:

1) Initialize the focal length and the re-projection error using (3).
2) Give an increment to focal length and calculate new focal length f_1' and f_2' using formula 4 and5, respectively.

$$f_1' = f_0 \times (1 - k_1) \tag{4}$$

$$f_2' = f_0 \times (1 + k_2) \tag{5}$$

3) Calculate the re-projection error under f_1' and f_2', obtain the more accurate focal length result with less error.
4) If current process leads to a convergence result and the error is less than threshold, end up the algorithm.
5) If the process can not be convergence result, change the parameter k_1 and k_2, go to step 2.

In the study, we set the argument k_1 and k_2 initializing value 0.3, iterations number 50, and obtain more satisfied results. The algorithm based on the dichotomy is more valid, optimal camera intrinsic parameters can be obtained.

SBA is a fast and robust method used in optimizing intrinsic and extrinsic camera parameters. It uses Levenberg-Marquardt algorithm to minimize reprojection error and can obtain more accurate 3D structure. We use SBA method to optimize the parameters further and can obtain more satisfied results.

4 Experimental Results and Analysis

Based on the aforementioned method, we implement the experiments by using the series of videos in DynTex database [7]. The hardware environment is microcomputer with dual-core CPU and 2GB Memory. The operating system is Windows7.0. We use Visual C ++ 6.0 development environment, OpenCV and OpenGL package to build the experimental platform. In the experiment we test the effectiveness of the algorithm and analyses the performance of the algorithm.

We use the video scene provided by reference [6] as background to construct the simulation environment to test the validity of algorithm studied here. The fluid reconstructed scene is from the video provided by reference [7].

The simulation environment is synthesized using Visual C ++ 6.0 development tool combining with OpenGL engine. The synthesized simulation environment are shown as Fig.2

Fig. 2. The Construction Results of Simulation Environment. The results of the first line are first three frames synthesizing the fluid with the “Road”. Those of the second line are first three frames synthesizing the fluid with the “Lawn”.

It can be seen from Fig.2 that the construction results of simulation environment have the feature of strong realistic. In the synthesis environment, the background and the virtual fluid have consistency geometry. The recovered camera parameters have more accurate characteristics and camera tracking position is relatively stable without jitter phenomenon. These can verify that the method proposed here is validity.

In order to verify the time performance of the proposed method, we use 20 consecutive frames from video provided by reference [7] to statistic the average computation time of each module. The statistical are shown in Table 1.

Table 1. The Average Time of Each Module (Second)

Video	Extraction of Feature Points	Initialization of Camera Parameters	Optimization based on Dichotomy	Optimization based on SBA	Total Time
Road	0.06620	0.0241	0.0501	0.0900	0.2314
Lawn	0.05075	0.0157	0.0703	0.1287	0.2655
Flower	0.05850	0.0191	0.0352	0.1215	0.2343

It can be seen from Fig.2 that the method proposed here has lower running time. The average running time is 0.24373 seconds, and this can meet the requirement of simulation.

5 Conclusion and Future Works

In order to constitute the scene we proposed a fluid reconstruction method using fluid motion vectors calculation and LBM combination. A method of feature points matching is studied using motion continuity to constrain and further more intrinsic and extrinsic parameters can be recovered and optimized based on dichotomy strategy.

Although, we have gained the preliminary result in the study, there is still some work need further improving. In the camera tracking, the number of feature points is not enough occasionally. In future study, it is possible to consider geometric constraints to improve it. On the other hand, it will be considered that new conditions can be constrained to improve the camera parameters further.

Acknowledgments. We wish to thank Ming-Qi Yu for his comments and suggestions. We would also like to thank the Natural Science Foundation of China (Grant 61070128, 61272199, 61104057) for funding this project, Natural Science Foundation of Shanghai Science and Technology Commission in China (grant 11ZR1411100), and Key Innovation Project Research of Shanghai Education Commission(12ZZ042).

References

1. Heas, P., Memin, E.: Three-dimensional motion estimation of atmospheric layers from image sequences. IEEE Transactions on Geoscience and Remote Sensing 46, 2385–2396 (2008)
2. Nagai, T., Naruse, T., Ikehara, M., et al.: HMM-based Surface Reconstruction from Single Images. In: Proceeding of IEEE International Conference of Image Processing, pp. 561–564 (2002)
3. Frankot, R., Chellapp, R.: A Method for Enforcing Integrability. In: Shape from Shading Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 10, 439–451 (1988)

4. Pickup, D., Li, C., Cosker, D., Hall, P., Willis, P.: Reconstructing Mass-Conserved Water Surfaces using Shape from Shading and Optical Flow. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part IV. LNCS, vol. 6495, pp. 189–201. Springer, Heidelberg (2011)
5. Zhang, Z., Deriche, R., Faugera, Z., et al.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artificial Intelligence Journal 78, 87–119 (1995)
6. Zhang, G.F., Jia, J.Y., et al.: Consistent Depth Maps Recovery from a Video Sequence. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 974–988 (2006)
7. Peteri, R., Huskies, M., Fazekas, S.: DynTex: A Comprehensive Database of Dynamic Textures. Pattern Recognition Letters 31, 1627–1632 (2010)

A Mediate-Based ABS Framework in Large-Scale Military Analytic Simulation

Yang Mei[1], Zhou Yun[1], Yang Shan-liang[1], Yang Zheng-jun[2], and Huang Ke-di[1]

[1] College of Information Systems and Management,
National University of Defense Technology, Changsha, Hunan, P.R. China, 410073
[2] Hunan Expressway Management Bureau, Changsha, Hunan, P.R. China, 410016
yangmei@nudt.edu.cn

Abstract. In this paper, we present a novel mediate-based framework for ABS (Agent-based simulation) which integrates common functions for commanders and operational entities in military analytic simulation. This framework aims to provide architecture for agents that run faster than real-time. Traditional "sense-think-action" cycle for multi-agent is extended to "sense-think-lookahead-action" cycle to reduce computation complexity and avoid possible loss of interactions. And a simulation platform with three mediators is designed for typical actions of military agents. The initial implementation of sense mediator is described and the spending of time for a simple scenario is compared with traditional approaches.

Keywords: Military simulation, Sense mediator, Agent-based simulation, Agent cycle.

1 Introduction

Warfare system is a kind of complex large-scale system with typical characteristics of emergence. Unpredictable phenomena emerge from interactions of entities, and a completely different turning point may come out because of a trivial action, resulted in the nail-biting of the commanders in a battle.

M&S (modeling and simulation) methods have drawn increasing attention to handle the complexity and high cost of warfare simulation. From the perspective of users, the military M&S can be applied in training (such as DIS and HLA), testing, and analysis. The MAS (Military Analytical Simulation) is different from other military simulations in the following aspects: (1)there are many calculation works; (2)the MAS should be faster than real time; (3)MAS is a multi-resolution simulation; (4)it is a hybrid system where some algorithms are continuous and others are discrete; (5)multiple replications runs simultaneously, and the empirical method needs to be designed before running; (6)the data should be provided by physical experiments, and high level MAS is supported by lower level simulation and experiments; (7)it is the combination of qualitative analysis and quantitative analysis[1]. In MAS, the characteristic and the complex interactions of all entities especially those commanders and operational units in a conflict are simulated to obtain the detailed quantitative data. These statistical data are then analyzed to study possible outcomes in different

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 340–347, 2013.

scenarios. It is a natural way to model entities in the battle space using ABMS (Agent-based Modeling and Simulation), since it is focused on the interactions between individuals and is fit for study of emergency.

High Level Architecture/Run-Time Infrastructure (HLA/RTI) is widely used for training simulation. The services of HLA/RTI provide the common functional infrastructure for M&S. We have learned from the studies of parallel RTI-based MAS system that the parallelizing of time management, event management and interaction management is the key point at the level of simulation platform[2][3]. However, as the entities are assembled in several federates and the interactions between entities in different federates are limited by network bandwidth, it is not suitable for taking on large-scale military analytical system with a great deal of interactions. Besides, HLA only provides interface specification for federates but not the inner-details[4], so the implementations of entities can varies with totally different performance for different federate developers, which leads to a phenomena analogous to some cars run slowly on the highway and may make a traffic jam. Lastly, the goal of HLA/RTI is to improve interoperability for distributed simulation, so HLA/RTI is helpful for training and inter-connection of multi-simulations, and the speed up ratio will be hard to increase without modifications in the platform.

The key contributions of this paper are proposing a Mediate–based ABS M&S Framework (MABSF) for MAS, and designing of some typical interaction mediators which account for the interactions between agents and between agent and environment. The sense mediator is based on the study of interest management in HLA/RTI; it limits interactions in specific interest area divided with uniform grid. The movement mediator and engagement mediator carry on their work on the basis of interactions determined by sense mediator.

2 Related Works

Simulation platform, which is also called simulation engine, is separated from simulation application logic to manage the execution of simulation replications, remind the application program when to change a particular state, and keep track of any future commitments in its diary[5]. Because the platform is sensitive to the communication software and the granularity and resolution of the models may affect the structure of the system, there can hardly be a generic simulation platform to be applied to all the areas in military system. There are various works focusing on different factors in military. For example, HLA standard promoted by DoD and finalized by IEEE and OMG has provided interoperability between simulations[4]. Joint Theater Level Simulation (JTLS) is a war-gaming system that can run on the computer[6]. Joint Warfare System (JWARS)/Joint Analysis System (JAS) is a campaign-level model of military operations to support resource allocation decisions and critical operational planning by a closed-form analytic simulation[7]. MANA (Map Aware Non-uniform Automata) is an agent-based model designed for combat operations, but may slow down when the number of agents comes to 600[8]. It can be known from these systems that interaction between entities in battle space is always a vital aspect.

Frameworks of agent-based simulation have also been attached importance to, for example, an agent-based generic framework for symbiotic simulation systems is designed and explored in [9] [10]. With the rapid development of Dynamic Data Driven Application System (DDDAS), dynamic data driven simulation for real-time combat decision support has been drawn much more attention [11]. However, the frameworks introduced previously do not meet the needs of our system, because some of them are not designed for analytic purpose, some of them can not support high level decision, some of them are not large-scale.

Time-stepped and Event-driven simulation can be adopted according to whether the system is real-time or faster than real-time. To simulate ABMs, the states of the agents need to change along the axis of time. So far, most of the ABM frameworks are based on time-stepped method, that is, time axis is discretized using constant delta of time and states are updated at each time step. At each point of time, agents can perform "sense-think-act" cycle[12]. However, the optimizing of time step is a crucial problem because some critical changes of states may be missed if the time step is too big, while some irrelevant updates may result in high computational cost and memory utilization if the time step is too small. In contrast, event-driven method can be utilized to update states of agent when needed. The updating of states occurs when event is processed. Seungman Lee[13] outlines timing mechanisms for hybrid agent-based simulation, where models are built by a combination of both continuous-time and discrete-event models, and their interactions. Bo Zhang[14] combines ABS and Discrete-Event Simulation (DES) with DEVS (Discrete Event System Specification) formalism, and achieves better performance than traditional approaches.

3 Our Approach

3.1 Design of MABSF

The MABSF is a specialized architecture for MAS. It is the basis for planning optimize, planning evaluation, and decision making. The simulation platform is composed of DES/PDES (Parallel Discrete Event Simulation), and several mediators (see Figure 1). DES/PDES is responsible for managing event queues, executing events in the queues. The mediators perform common calculations of cycles among agents.

At the first temptation of Agent-based MAS, we realized that an agent may not move to a position where it planned to due to the interruption of an enemy or the change of the plan in the process – the entities are inter-connected in a large complicated network, and any action of an agent will cause the refresh of combat situation which therefore affects the execution of decision made by the acting agent and any other agents. To solve this problem, a "look-ahead" step is added before it carries out its movement. The longest performable motion of the action will be calculated and the consequence interruption will be scheduled to make the action be sequences of motions. The sequence of action may vary from agent to agent, but the look-ahead is underneath.

Here, the simulation platform is introduced as mediator to coordinate the agents' interactions in cycles (see Figure 1). The functions of simulation platform are

extended to integrate common functions in the cycle of agent. This enlarges the simulation platform and weakens the role of world model to static environment data. From the perspective of agent, the inter-operations are not actually performed by themselves as in the real world, but only send requests provided with values of states and receive the outcomes. Most of the common calculations for interactions are carried out or logically lies in the mediators.

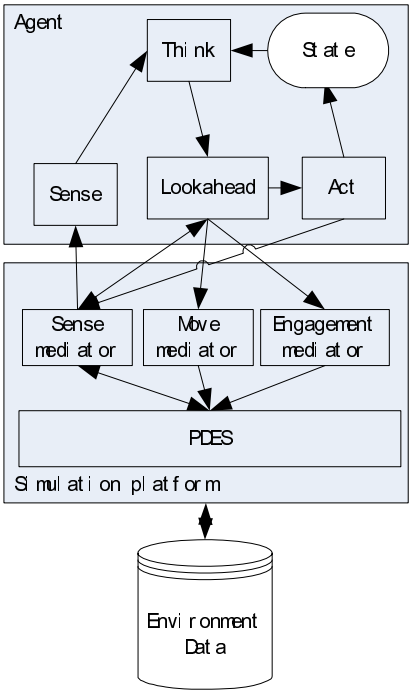


Fig. 1. Framework of MABS

The simulation advances by reiteratively schedule and processing events in the event queue, which is also called FEL (Future Event List) compared to Processed Event List[15-17]. This event queue is generally a priority queue. By contrast to DES, PDES uses several events to be executed simultaneously. According to the amount of event queues for logical process, PDES can be classified as single-queue and multi-queue. For single-queue, one of the possible solutions is to adopt concurrent priority queue[18]. For multi-queue, several event lists are distributed to each logical process or each simulation object[19]. LVT (Local Virtual Time) is calculated to operate on the local event queue, and GVT (Global Virtual Time) can be computed and updated periodically through some algorithms.

The simulation is initiated by a scenario with a series of events being created. After the agents are created and the states are set according to scenario data loaded from configuration files, the sense mediator decides the supposed detecting pairs. That is, the initial situation map is created by each side of the battle. To actually simulate the

behavior of "sense", events of these "detection" are inserted into an event queue at the time of zero with a timestamp of zero (where, time represents simulation time in this paper). These events then are executed and the agents (e.g. aircraft with radar) can get the states (such as position, velocity, etc.) of agents in the search areas of their ISR components from sense mediator.

In the process of Sense, there are two manners for an agent to track the objects when the sensor component is activated. One is that the agent asks the mediator for what it can "see". Otherwise, the sensation occurs when some objects the agent is interested in enter or depart from its detection area. Agent with capability of sense holds and maintains a detected list. Once the sense mediator compute the time of entering and existing in advance, the corresponding events of entering and existing are created and scheduled in the event queue, and the object is added to the detected list of the detecting agent. Then the agent looks up the detected list to get the positions of the objects when needed, for example, aircraft confirm the coordinate of the target in its target list before it makes an attack.

In the process of Think, the agent employs a predefined expert system. The expert system takes the states of agent and current perception as input to make a decision for COA (Course of Action). This module does not interoperate directly with simulation platform, and will not be described in detail.

In the process of Lookahead, the action mediator performs the action in advance to make sure whether or not the action can be carried out completely and determine the maximum safe part of the action.

In the process of Action, the safe part of the action is committed to make an effect on environment and agent itself.

This novel mediate-based simulation platform assists the agent by providing common computations. The architecture can not only greatly relieve the burden of developers of agents, especially for military experts who may be not specialized in programming, but also reduce redundant computation by contrast to the decentralized approach where some agent may take the same calculation. The last but not the least, the Lookahead process avoid rollback owing to causality errors. The following two sections will mainly focus on the sense mediators which is the foundation of the others.

3.2 Sense Mediator

The sense mediator relates to the state of environment and how the agents perceive. It provides a component to determine when an agent can see others and what it can see. It is designed to reduce both the synchronization and the updating of irrelevant data. The computations for sensation are drawn toward the sense mediator, resulted in the decrease of redundant computation among agents.

According to the time stream mechanism, simulator can be time-stepped or event-driven[20]. As we mentioned previously, the system is built upon DES/PDES, without updating states at constant intervals. For example, an agent A detected by agent B moves from P1 to P2 at Time [T1, T2), see Figure 2. The agent A will leave detection area of B at position of P3 at Time T3, and enter into the sight of agent C at Time T4. When it comes to T5, the agent C need to refresh the states of targets it has detected to

make sure that the following attacking is based on the newest states. In this case, agent A updates its state at specific time and synchronized with other agents due to the effect of the changing state on them. The state updating for agent A and synchronization of agent A-D are shown in Figure 3.

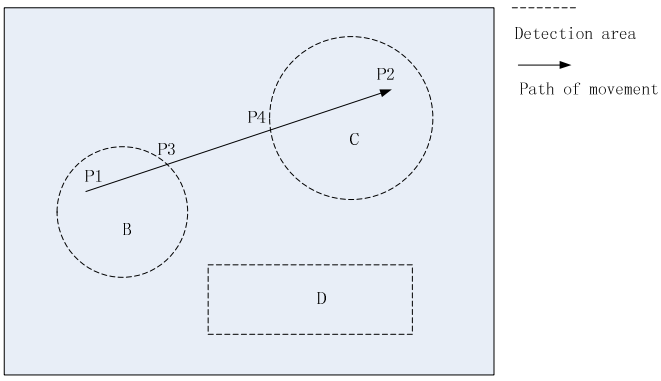


Fig. 2. An example for sense mediator

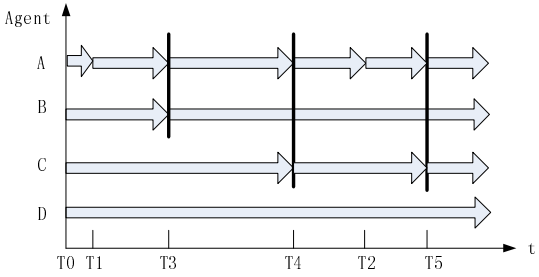


Fig. 3. Updating and synchronization of agents

We should notice that these synchronizations are not always simultaneous with modification of states. The change of states for A occurs at T1, T3, T4, and T2; the updating of states for synchronization occurs at T3, T4, and T5; the starting and end of movement from P1 to P2 for agent A are ignored by other agents; and agent D does not require the information of agent A.

In our hierarchical agent-based MAS[21], an agent can be composed of C2, sensor, platform, communication, or weapon. Once activated, the sensor deployed on an agent can let the agent become aware of passively or actively. In active sensation, an agent first creates an event and put it into the event queue to create a sensation. When processing this event, the agent receives the information it can see from the world. It is the sensor who knows when the event should be created. In the passive mode, the idea is that change in the world leads to newly sensation or out of sensation, which are two most important events the sensor cares about. As it cannot predict state of the world, the sensor is notified by the mediator, the world, or the changing agent, and is submissive to the change.

The computations to determine *who sees whom* is a key issue to achieve scalable operation[22]. In MAS, the sensations are typically among spaces and can be understood in terms of *publish and subscribe*. The publish area is the area that bounds the location of the platform, while the subscribe area is the space that the sensor can perceive. The agent B can detect agent A only when the subscribe area of B intersects the publish area of A. It is implemented similarly to a spatial-based interest management in HLA[23].

4 Conclusions and Future Work

This paper has presented a framework of mediate-based architecture for agent-based M&S in military analytic simulation. A Lookahead process is added in the common Sense-Think-Act loop of an agent. The proposed MABSF and agent cycles can work together to speed up the simulation. However, there are still some performance issues when it is implemented on platform such as cluster and GPU and the efficiency of event processing may be the bottleneck. The performance on different platforms will be measured in our future work, and methods for improving performance will be studied later.

Acknowledgements. This work is supported by National Natural Science Foundation of China: Research on theory and method of dynamic data driven simulation for real-time operational decision support (61074108), Research on theory and method of agile simulation for decision-making of complex systems (61374185).

References

1. Ke-di, H., Xin-ye, Z., Shan-liang, Y., Mei, Y., Feng-hua, H., Ying, C.: System design description infrastructure overview for military simulation and analysis system. Journal of System Simulation 24, 2439–2447 (2012)
2. Mei, Y., Xin-ye, Z., Ying, C., Ke-di, H.: Key technologies of high performance simulation for analytical simulation evaluation. Journal of System Simulation 24, 2463–2467 (2012)
3. Qiang, H., Xiao-cheng, L., Chun-guang, P., Jian-guo, H.: Research and implementation of parallel simulation engine for component based-on BOM. Journal of National University of Defense Technology 33, 154–158 (2011)
4. IEEE standard for modeling and simulation (M&S) high level architecture (HLA)– framework and rules (2010)
5. Pidd, M.: Simulation worldviews - so what? In: Proceedings of the 2004 Winter Simulation Conference (2004)
6. JTLS Executive Overview, `http://www.rolands.com/jtls/j_over.html` (April 12, 2013)
7. George, F., Stone, I., McIntyre, G.A.: The joint warfare system (JWARS): a modeling and analysis tool for the defense department. In: Proceedings of the 2001 Winter Simulation Conference (2001)

8. Justin, L.R.: A Comparative Study of Simulation Software for Modeling Stability Operations. In: Proceedings of the 2012 Symposium on Military Modeling and Simulation (2012)
9. Aydt, H., Turner, S.J., Cai, W., Low, M.Y.H.: Research issues in symbiotic simulation. In: Proceedings of the 2009 Winter Simulation Conference (2009)
10. Heiko Aydt, S.J.T.: Wentong Cai, Malcolm Yoke Hean Low: An agent-based generic framework for symbiotic simulation systems. In: Uhrmacher, A.M., D.W. (eds.) Agents, Simulation and Applications. Taylor & Francis (2008)
11. Yun, Z.: Research on the theory and methods of dynamic data driven simulation for real-time combat decision support. Phd Thesis. National University of Defense Technology, Changsha, Hunan, P.R.China (2010)
12. Wen-Guang, Y.: Research on parallel agent-based simulation on multi-core CPU and GPU heterogeneous platform. PhD Thesis. National University of Defense Technology, Changsha, Hunan, P.R.China (2012)
13. Lee, S., Pritchett, A., Goldsman, D.: Hybrid agent-based simulation for analyzing the national airspace system. In: Proceedings of the 2001 Winter Simulation Conference (2001)
14. Zhang, B., Ukkusuri, S.V., Chan, W.K.: Agent-based discrete-event hybrid space modeling approach for transportation evacuation simulation. In: Proceedings of the 2011 Winter Simulation Conference (2011)
15. Bagrodia, R.L., Liao, W.-T.: Maisie: a language for the design of efficient discrete-event simulations. IEEE Transactions on Software Engineering 20, 225–238 (1994)
16. Preiss, B.R., Loucks, W.M., Macintyre, I.D.: Effects of the checkpoint interval on time and space in time warp. ACM Transactions on Modeling and Computer Simulation 4, 223–253 (1994)
17. Goldberg, A.: Virtual time synchronization of replicated processes. In: Proceedings of the 6th Workshop on Parallel and Distributed Simulation, pp. 107–116 (1992)
18. Rao, V.N., Kumar, V.: Concurrent access of priority queues. IEEE Transaction on Computers 37, 1657–1665 (1988)
19. Yao-cheng, Z.: General purpose parallel discrete event simulation environment and the study of relevant techniques. PhD Thesis. National University of Defense Technology, Changsha, Hunan, P.R.China (2008)
20. Fujimoto, R.M.: Parallel and distributed simulation systems. John Wiley & Sons, Inc. (2000)
21. Mei, Y., Shan-liang, Y., Xin-ye, Z., Zheng-jun, Y., Yun, Z.: Hierarchical agent in military analysis simulation for on-line decision support system. Journal of Computer Science 40, 22–26 (2013)
22. Steinman, J.S.: Scalable publish and subscribe data distribution, `http://www.warpiv.com/documents/papers/pubsub.pdf` (May 06, 2013)
23. Mei, Y., Xin-ye, Z., Yun, Z., Ying, C.: An interest management in large-scale analytic simulation. In: 2013 International Conference on Computational Intelligence and Information (2013)

Confrontation Scenario Simulation Using Functional Programming Model

Lin Tang, Minggang Dou, Ze Deng, and Dan Chen

School of Computer Science
China University of Geosciences
Wuhan, China
dan.chen@gmail.com

Abstract. Simulation is an important approach to the study of scenarios of confrontation among antagonistic groups. It remains a research issue to explore the influence of the crowd size and imbalance between groups' sizes on the process of confrontation. In this study, a multi-agent simulation system has been developed with functional programming model (FPM). FPM can easily formulate the simultaneous behaviors/actions of individuals. It also provides "communication backbone" for agents' interactions. A timing system has been designed to drive the simulation procedure. Our simulations focus on whether/how a confrontational scenario may remain stable. Experimental results indicate that with the increment of the overall size of the crowd in confrontation, the possibility of the scenario getting out of control rises. A relatively small scale of crowd is much more controllable.

Keywords: Crowd Simulation, Multi-agent System, Functional Programming Model, Confrontation Scenario.

1 Introduction

Confrontational scenarios have long been attended by researchers from multiple disciplines and government agencies for the risks to society and numerous disasters caused by such scenarios in history. Simulation study on confrontation scenarios has gained more and more attentions in recent years for the obvious advantages over traditional approaches [5].

Existing works along this direction are in general agent-based, which use autonomous and intelligent agents for crowd modeling [7]. Decision-making mechanisms have been built focusing on casualty [2], triggers of violence [3], individual factors (e.g., hostility, personal price, punishments, participants anonymity, cohesiveness) that influence violence level [1][6]. However, the influences of the number of the crowd and imbalance between the sizes of groups in confrontation have been largely ignored.

In this study, we developed a multi-agent simulation framework for confrontation scenarios. The system takes advantages of functional programming model (FPM) [4] for its merits of functional parallelism, asynchronous message, pattern matching.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 348–354, 2013.

Our confrontation scenario simulation comprises three types of processes: agents, the virtual world, and the terrain. Each agent possesses autonomous capability to improvise their action. This study concerns four behaviors: assaulting, assembling, bluffing, and retreat.

Experiments have been performed to investigate influences of the number of the crowd and the imbalance between groups on the confrontation scenario. The results indicate that within relatively small scale of the crowd, even if one group is much larger than the other, it is usually controllable.

2 Functional Programming Model

Functional programming model treats computation as the evaluation of mathematical functions. The follows are the features of FPM that eases the modeling of confrontation scenarios.

Functional Parallelism: FPM maintains a number of light-weight processes each behaving like an autonomous object in the real world.

Asynchronous Message: The communication among processes is asynchronous that one process would send message to the other but not just wait for the response.

Pattern Matching: The pattern matching helps agents analyze complicated information from the simulated scenario.

As shown in Fig. 1, FPM sustains three main components of the confrontation simulation: (1) a world process, (2) agent processes, and (3) a terrain process.

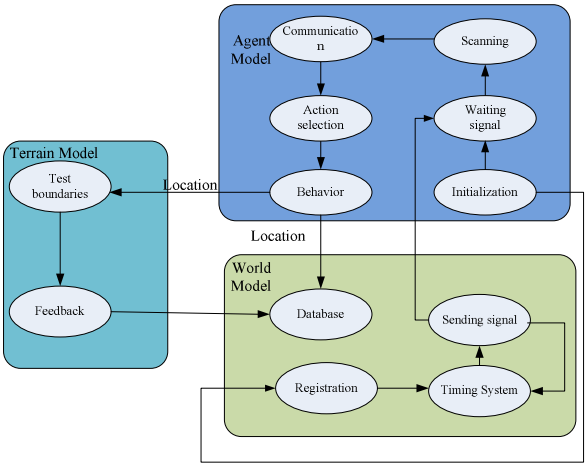


Fig. 1. Simulation System Architecture

The world process [8] manages the records of location of all registered agents with a database. The world process would initiate the timing system to send the signal marking the start of simulation. Then the world process waits for the feedbacks of all agents and the terrain process and update. The terrain process [8] maintains the environment information. Every agent [8] belongs to a light-weighted process with a built-in record of the agent's dynamic information.

3 Agent Modeling for Confrontation Scenario

This section introduces the agent's properties and decision making mechanism. The agent model focuses on the underlying mechanism of agent's behavior selection.

3.1 Agent's Properties

An agent has four fundamental properties, i.e., position, emotion, personality, and velocity. Position (*x*, *y*, *radius*) is simply the current position the agent locates with a radius indicating its range of sensing. Emotion properties include aggressiveness and rationality. The agent's velocity defines (1) *dir* (direction), i.e., the direction of last movement, (2) *v* indicates the agent's current speed, and (3) v_{max}, i.e., the highest speed the agent can achieve.

Personality: Each agent would be initiated with a value. The value of personality is used as the parameter θ to regulate the value of emotion. The parameter θ can be calculated by formula (1)

$$\theta = \frac{p}{p_{\max}}. \tag{1}$$

where p represents the value of agent's personality, while the $p_{\max}$ is the maximum personality value.

Emotion: the emotion comprises two parts: aggressiveness and rationality. The aggressiveness increases by ΔA which is calculated by formula (2)

$$\Delta A = \begin{cases} 0, & |N_{own} - N_{opposite}| < NL \\ (N_{own} - N_{oppo}) \times \dfrac{N_{b(oppo)}}{N_{oppo}} \times \dfrac{N_{b(own)}}{N_{own}}, & |N_{own} - N_{opposite}| \geq NL \end{cases} \tag{2}$$

where N_{own} represents the number of their own people within radius, while the N_{oppo} means the number of people of the opposite group within radius. $N_{b(oppo)}$ represents the opposite group's people in bluffing behavior, and the $N_{b(own)}$ means the own group's people in bluffing behavior. *NL* is the limit value of the difference between two groups' numbers. The rationality value is determined by the personality value.

The emotion value is calculated by formula (3)

$$E = A \cdot \theta - R \cdot (1 - \theta) \tag{3}$$

where *E* represents the value of agents' emotion, *A* means aggressiveness, and *R* means rationality.

Before the introduction to Velocity, the calculation of density which is denoted by ρ_t is shown in formula (4)

$$\rho_t = \frac{N}{N_{\text{sum}}} \tag{4}$$

where ρ_t represents the place of certain agent's density in time frame *t*, the *N* is the number of people within the radius of agent, and N_{sum} is the total number in the confrontation scenario.

In addition, within agent's radius, there also is a place representing the max density which is denoted by ρ' calculated in formula (5)

$$\rho' = \frac{N_{(x,y)}}{N_{\text{sum}}} \tag{5}$$

where $N(x,y)$ represents maximum number of own people in one of the coordinates within current agent's radius.

Velocity: the speed v is calculated by the segmented function written as formula (6)

$$v = \begin{cases} v_{\max} \cdot \left(1 - \frac{\rho}{\rho_{\max}}\right), E \in [0, E_1) \\ v_{\max} \cdot \left(\frac{E}{E_{\max}} + \left(1 - \frac{\rho}{\rho_{\max}}\right) \cdot \left(1 - \frac{E_1}{E_{\max}}\right)\right), E \in [E_1, E_{\max}] \end{cases} \tag{6}$$

where is the maximum crowd density, E_1 is the threshold of emotion, and $E_{\max}$ represents the maximum emotion value.

The direction of agent is determined by formula (7)

$$\alpha = \frac{y - y_{(t-1)}}{x - x_{(t-1)}}$$

$$dir_{(t)} = \frac{dir_{(t-1)} + \tan[\alpha \cdot (1 - \frac{E}{E_{\max}})]}{1 - dir_{(t-1)} \cdot \tan[\alpha \cdot (1 - \frac{E}{E_{\max}})]} \tag{7}$$

where x and y come from ρ', $x_{(t-1)}$ and $y_{(t-1)}$ are from last time frame, $dir_{(t)}$ represents current direction, and $dir_{(t-1)}$ is the last time frame's direction.

3.2 Agent's Behavior Model

The behavior selection is based on three rules: threshold rule, emotion rule, and attention rule. The threshold rule is mainly responsible for the state transition of agents. The emotion rule represents the emotion propagation among crowd. The attention rule controls the moving direction of agents. There are four kinds of behaviors including assembling, assaulting, bluffing, and retreat.

Assembling: agents seek for companies and approach them. Agents' emotion values tend to increase.

Assaulting: agents follow the confirmed target(s) closely and send messages to targets as "assault." Targets record the times of being assaulted and change the behaviors to respond.

Bluffing: agents are immobilized. Other agents would detect the existence of bluffing ones.

Retreat: agents change their directions and approach the place with lowest density.

Fig. 2 illustrates the agent's behavior selection procedure. The agent would change their behavior according to the emotion value by comparing with the threshold value. Agents will be removed from the scenario when being excessively assaulted.

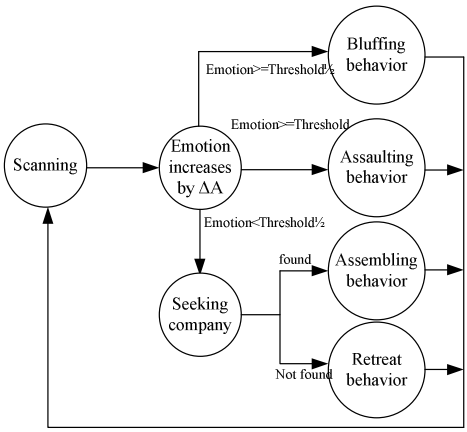


Fig. 2. Agent's state transition in terms of behaviors

4 Experiments and Results

Experiments have been performed with focuses on the imbalance between the numbers of two groups. The simulation scenario can contain individuals with different personalities in a group. The environment was abstracted as a closed Euclidean space (200 by 200 meters) with each space unit resembles one meter.

As presented in Table 1, the overall crowd size can be 100, 200 and 300. We have made 100 simulation runs for each configuration.

As shown in Fig. 3 (a), there are two groups (marked pink and green) with 50 agents respectively. In Fig. 3(b), the highlighted area shows that with a high value of aggression, the pink agents directly broke into the green group. Fig. 3(c) indicates that the green agents were fleeing away from the pink ones. As shown in Fig. 3(d), the two groups were in confrontation with properties of emotion below the threshold.

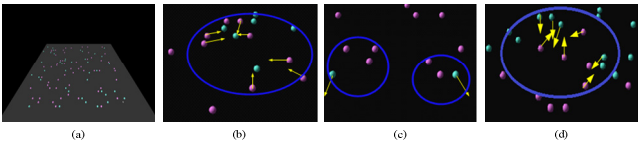


Fig. 3. Snapshots of the confrontational behaviors, (a) The overall simulation scenario, (b)the green party in the circle getting surrounded by the pink party, (c) the green party in fleeing, (d) some members of the green party in bluffing

As presented in the right column of the Table 1, the proportions of bluffing people declines with the increasing crowd size. The larger the crowd size is, the more possibly the scenario tends to lose control, that is, one group collapses. It can also be observed that the proportion of bluffing agents decreases with the increasing crowd size.

In the final experiment (200 vs. 100 agents), the number of bluffing agents in the 2nd group is zero, and this means the confrontation is completely out of control and ends up with collapse of the 2nd group. This is a contrast to the second experiment, in which the proportion of bluffing agents reaches 40%. This suggests that when the overall crowd scale is small, the asymmetry on the size of two groups is less significant.

Table 1. Configurations of simulation run and results

Crowd sizes	Agent distribution in two groups	Numbers of the bluffing agents (averaged from 100 experiment runs)
100	50 versus 50	26.63 : 26.52
	65 versus 35	25.54 : 18.30
200	100 versus 100	38.10 : 37.27
	125 versus 75	25.27 : 19.72
300	150 versus 150	42.29 : 43.09
	200 versus 100	10.87 : 0.00

5 Conclusions

In this study, we have successfully developed a simulation framework for confrontation scenarios using Functional Programming Model (FPM). The simulation framework takes advantage of the merits of FPM in functional parallelism, asynchronous message, and pattern matching. Modeling of confrontation scenarios has then been simplified comparing to simulation techniques. A series of simulations has been performed to investigate the factors that influence the violence level in confrontation scenarios. Experimental results have been collected with various crowd sizes and the proportions groups inside each crowd. The results indicate that with the increment of the overall size of the crowd in confrontation, the possibility of the scenario getting out of control rises. A relatively small scale of crowd is much more controllable.

Acknowledgments. This research was supported in part by the National Natural Science Foundation of China (No. 61272314), the Program for New Century Excellent Talents in University (NCET-11-0722), the Excellent Youth Foundation of Hubei Scientific Committee (No. 2012FFA025), the Specialized Research Fund for the Doctoral Program of Higher Education (20110145110010), and the Fundamental Research Funds for the Central Universities (CUG, Wuhan, No. CUG120114).

References

1. Bu, F.: Modeling and Simulation of Mass Violence Event Based on Agent Technology. In: 2nd IEEE International Conference Emergency Management and Management Sciences, ICEMMS (2011)
2. Chao, W., Li, T.: Simulation of Social Behaviors in Virtual Crowd. Computer Animation and Social Agents (2010)
3. Collins, R.: Violence: A Micro-sociological Theory. Princeton University Press (2009)
4. Field, A.J., Harrison, P.G.: Functional Programming. Addison-Wesley, Reading (1988)
5. Hill, J., Randall, W.: Modeling Perceptual Attention in Virtual Humans. In: Proceedings of the 8th Conference on Computer Generated Forces and Behavioral Representation, pp. 563–573 (1999)
6. Fridman, N., Kaminka, G.A., Zilka, A.: Towards Qualitative Reasoning for Policy Decision Support in Demonstrations. In: Dechesne, F., Hattori, H., ter Mors, A., Such, J.M., Weyns, D., Dignum, F. (eds.) AAMAS 2011 Workshops. LNCS, vol. 7068, pp. 19–34. Springer, Heidelberg (2012)
7. Pan, X., Charles, S.H., Ken, D., Kincho, H.L.: Human and Social Behavior in Computational Modeling and Analysis of Egress. Automation in Construction 15(4), 448–461 (2006)
8. Ulf, E.: Design Patterns for Simulations in Erlang/OTP. Computing Science 178 Examensarbete DV3 (2000) ISSN 1100-1836

Research on Simulation-Experimentation-Based Effectiveness Evaluation of Digitized Force's Command and Control System

Zhao Fang[1,2], Zhang Tao[3], Wei Xin[2], and Zhao Xin Jun[2]

[1] College of Automation Science and Electrical Engineering,
Beijing University of Aeronautics and Astronautics, Beijing, 100191, China
[2] Science and Technology on Complex Simulation Land Systems Simulation Laboratory,
Beijing, 100072, China
[3] Naval Academy of Armament, 100036, China

Abstract. Pitched at problems needing to be solved for evaluating the digitized force's command & control (DF C2) system, this paper proposes the major contents and processes for evaluating the C2 system's effectiveness based on simulation experimentation, lays down the Measures of Efficiencies(MoEs) system of effectiveness evaluation (EE) for the C2 system, and analyzes, through the design of experiment samples and the planning of the evaluation process, how to use the simulation-experiment approaches of comparison and statistics to conduct effectiveness evaluation, contribution analysis and optimization analysis of the C2 system.

Keywords: C2 system, Simulation Experiment, Effectiveness Evaluation, MoEs system.

1 Introduction

The C2 system's effectiveness refers to the extent to which the C2 system utilized in particular conditions to implement a specified mission can reach the expected possible objective, and the evaluation of the C2 system is a process mainly focusing on evaluating whether the C2 system has met the set requirements. It is the process for such entities as the organization to use the C2 system evaluation results, the appraisal organization for C2 system finalization or the C2-system experimentation agency to measure the extent to which the C2 system carrying out the required mission in the set or specific environment and constraint conditions can reach the expected objective(s).

This study conducted effectiveness evaluation of the DF C2 system by means of simulation experimentation to verify whether the system can meet the needs for weapon-materiel construction and application; established a red force-blue force engagement simulation system, which was notably based on the operational process; used the comparison approach; designed experiment samples; conducted direct acquisition, statistics and analysis of the experiment data; solved some important problems in the C2 system's effectiveness by comparing the DF C2 system with the TF C2 system in terms of the values of the various MoEs and the eventual comprehensive

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 355–360, 2013.

effectiveness; showed the weak links in design and schemes as well as the critical nodes that may increase the efficiency; and assessed the system effectiveness according to certain principles and measures.

2 Evaluation Contents and Process

2.1 Evaluation Contents

To solve the problems in evaluating the digitized C2 system, we conducted evaluation in the following three aspects:

1. Effectiveness evaluation: we analyzed in the background of typical engagement, how well the various measures in the MoEs system for C2-system evaluation performed;
2. Contribution analysis: we analyzed in the background of typical engagement, to what extent has the C2 system contributed to the overall operational effectiveness;
3. Optimization analysis: we conducted optimization analysis of the C2 system by analyzing the influence of such core parts of the C2 system as the commanding process, situation sharing and software complexity on operational effectiveness.

2.2 Evaluation Process

The simulation experiment based effectiveness evaluation of the C2 system involves complicated system engineering covering such steps as MoEs design, method selection, sample design, data preparation, simulation-system preparation as well as data acquisition and analysis.

1. Evaluation-task identification: identifying the problems to be dealt with and setting the evaluation objectives according to requirements;
2. Evaluation-scheme design: led by the evaluation objectives, selecting the MoEs, identifying the evaluation approach and design the experiment samples, covering such areas as the simulation scenario, the constraint conditions as well as sampling and sample implementation strategies;
3. Simulation experimentation: preparing the relevant data and systems, running the simulation system and acquiring the data;
4. Analysis of data from the running: analyzing the data against the MoEs, the contribution degrees of the core parameters and the credibility of the evaluation results.
5. Evaluation-conclusion production.

The aforesaid 5 steps can be taken repeatedly, with simulation experimentation and effectiveness evaluation conducted with incremental iteration.

3 MoEs Selection

As the C2 system is the core of C^4ISR, the established set of sufficient and proper MoEs for evaluating the C2's effectiveness can be used for in-depth and comprehensive analysis of the massive simulation-result data obtained from simulating the C2 system, for evaluating the C2 system's effectiveness, and for providing data support of the C2 system's approach argumentation, construction and optimization.

Although there were already studies of the C2 system with the corresponding MoEs-systems for effectiveness evaluation established, most of them adopted mathematical approaches with the measurements mostly from expert scorings and many measures set with communications-system capabilities taken into consideration. Establishing the MoEs system specifically for the military equipment of the C2 system, however, was rarely seen. Our evaluation approach, however, was based on engagement simulation, taking from the simulation results the EE-typical measures with direct physical senses. We generally recognized and adopted the GAO Fei et al [1]- supplied structure and algorithm to establish the testable and measurable MoEs for evaluating the C2 system's effectiveness, characterizing the C2 system's effectiveness with the three measures of situation sharing, operation commanding and battlefield survival. On the situation-sharing sub-measures, however, we had some different opinions from those of GAO Fei et al. We characterized situation sharing with the four submeasures of the enemy-situation updating time, the self-situation updating time, the comprehensive-situation forming time and the situation-information accuracy. MoEs system for effective evaluation of the C2 system shown as in Fig1.

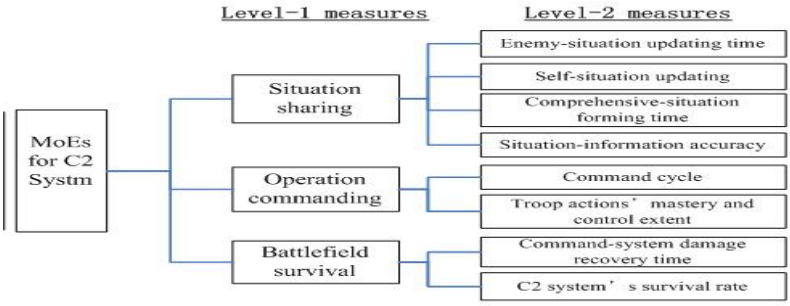


Fig. 1. MoEs system for effectiveness evaluation of the C2 system

1. Situation sharing's effectiveness was described in 4 level-2 measures:

- Enemy-situation updating time: the time for updating all the enemy situations, including situations about the enemy's command posts and the urgent air situation (in seconds). Here, the situation refers to all the target information about the enemy force currently mastered by the corresponding intelligence reconnaissance system, and the updating time refers to the interval between the intelligence reconnaissance system having survailed information change to the target and the corresponding command posts putting it on the situation map and having it shown.

- Self-situation updating time: the time for finishing the updating of all the self-situation including our side's commands, air situations, etc. (in seconds).Here, the updating time in this paper refers to the interval between our force's situation information getting changed and the corresponding command posts putting it on the situation map and having it shown.
- Comprehensive-situation generating time: the time for finishing all the situation processing and generating the comprehensive situation (in minutes).Here, the generating time in this paper refers to the interval between the command-seat in question generating the comprehensive situation map and the corresponding command posts putting it on the situation map and having it shown.
- Situation-information accuracy: the extent of accuracy of the situation-information display (in percentage).Here, the situation refers to the situation displayed on the comprehensive situation map, and the accuracy refers to the ratio of the quantity of situation information actually onto the map to the quantity of situation information that should be displayed on the map.

2. Operation commanding's effectiveness was characterized with two level-2 measures:

- Command cycle: the interval between the C2 system receiving the case sent from the intelligence reconnaissance system and the C2 system receiving the feedback on how the command was implemented from the implementation unit(s). Here, the case mainly refers to the occurrence of threat or the coordination event.
- Troop actions' mastery and control extent: the ratio of the number of troops whose state of actions were mastered and controlled to the number of troops needed to be effectively mastered and controlled (in percentage).

3. The battlefield survival's effectiveness was characterized with two level-2 measures:

- C2 system damage recovery time: the time for the C2 system to be recovered from its getting damaged (in seconds). Here, recovery refers to the realization of all the needed functions getting totally recovered or of functioning at a lowered level.
- C2 system survival rate: the ratio of the number of the surviving C2 units to the total number of C2 units (in percentage). Here, the C2 unit refers to the entities constituting the command post, such as the single vehicle and the single machine.

4 Evaluation Process for Simulation Experimentation

4.1 Design of Samples

To meet the needs for comparison analysis and optimization analysis, we designed samples for the C2 system's simulation experiments:

1. Sample 1 for describing the DF C2 system's core parameters;
2. Sample 2 for describing the TF C2 system's core parameters;
3. Optimization analysis oriented, sample 3 was for describing the core parameters of a C2 system with its situation-sharing capability further enhanced;

4. Optimization analysis oriented, sample 4 was for describing the core parameters of a C2 system with its situation-sharing capability seriously weakened;
5. Optimization analysis oriented, sample 5 was for describing the core parameters for the DF C2 system's command process;
6. Optimization analysis oriented, sample 6 was for describing the core parameters for the TF C2 system's command process;
7. Optimization analysis oriented, sample 7 was for describing the core parameters of a C2 system with its software complexity further simplified;
8. Optimization analysis oriented, sample 8 was for describing the core parameters of a C2 system with its software complexity much enhanced.

In the C2-system models described in the same simulation system, different samples were related to different variables (parameters), their particular data and corresponding models omitted from description and discussion in this paper.

4.2 Process for Simulation Experimentation

For the work flow for simulation experimentation, We experimented under the scenario of the same background of engagement and the same weapon materiel except the C2 system. Conducting comparison-oriented analysis of requirements, we set the data for the models for sample 1 and sample 2, respectively. According to the experimentation phases set by the simulation scenario, simulation experiments were started. After the experiments, the corresponding MoEs data were acquired directly from the simulation-experiment results with no expert scoring or personal experience involved. Data corresponding to sample 1 and 2 were recorded with the data cleaned and singular values discarded.

Comparison was conducted over the entire process of engagement or a particular operational phase in value change of such MoEs as the situation updating time and the comprehensive-situation generating time. The range of MoE changes got from statistics was compared with the corresponding system-needed values and the overall requirement values for system R&D, respectively, for determining through analysis whether the C2 system had satisfied or reached the operation needs and the requirements for weapon-system development. Meanwhile, regarding the value of each level-3 measure of the MoEs system for C2 effectiveness, sample 1 and 2 were analyzed to find by how many folds the value from sample 1 had been raised or reduced against the corresponding value from sample 2.And finally, with the value of each measure analyzed, the simple algorithm for calculating the average value was used to calculate the value of each level-2 measure, and the comprehensive comparison and analysis results between sample 1 sample 2 were eventually obtained.

In the simulation-experiment process for sample1 and sample 2, we recorded and acquired the red force's and the blue force's mission data with each action assigned to the red force and the blue force regarded as a mission and the effectiveness of the finished mission used to measure the contribution that the C2 system had made to the operation system's operational capability. Through the analysis of the experiment data, we obtained the situation of mission finishing by the red force and the blue force, from which we discovered the C2 system's multiplication effect over the entire SoS operation.

We also conducted the optimization-oriented comparison-based analysis of requirements, and set parameter data and the corresponding models for sample 3 to sample n. In the same typical scenario, we ran the samples and acquired the red force's and the blue force's mission-finishing results data with the data-cleaning program run and the singular values discarded. Through comparison-based analysis of sample 1 and respectively each element of the "sample 3, sample 4, … , sample n" set, we got the result of whether the optimized parameters can influence the operational effectiveness and, if there was influence, in what way and to what extent.

5 Conclusion

With the DF's requirement of the C2 system getting increasingly tough, the effectiveness evaluation of the DF C2 system has become increasingly important as the evaluation directly influences the entire process of the system from requirement argumentation, through design and development, to application.

The effective simulation experimentation based approach for comparison-based evaluation has such features as low risk, high efficiency, low cost, repeatable experimentation and readiness for quantitative analysis, and therefore has become a powerful means for analyzing and evaluating the C2 system's effectiveness. Adopting this approach, this ongoing project has achieved early research results, which have proved the feasibility of this approach.

References

1. Gao, F., Gao, F., Wang, Y., Chen, S.: Construction of Measures of Effectiveness System for Digitized Army's C2 System. Ship Electronic Engineering 3, 1–3 (2012)
2. Liu, J.-X., Xiu, S.-L., Ouyang, Z.-Z., Luo, X.-S.: Study on the Index System Structure of C4ISR System Effectiveness Evaluation. Fire Control and Command Control 30(6), 18–21 (2005)
3. Liu, X.-Y., Du, X.-M., Wang, L., Su, X.-B.: MoEs System for Evaluating Digital Force's Equipment-Command Effectiveness. Sichuan Ordnance Journal 31(7), 127–130 (2010)
4. Wang, Y.-M., Xu, P.-D.: Research on Evaluation of C2 System's Operational Effectiveness Using ANP. Fire Control and Command Control 32(11), 77–80 (2010)
5. Zhu, M.-J., Zhang, W.: Modeling and Simulation on Workflow of Combat Command and Control. Fire Control and Command Control 33(12), 45–48 (2006)
6. Li, T., Guo, Q.-S., Duan, L., Hu, J.: Construction of Evaluation System of Digital Force Combat Capacity. Journal of Academy of Armored Force Engineering 22(2), 14–19 (2008)
7. Song, X., Zhang, S., Qian, L.: Opinion dynamics in networked command and control organizations. Physica A: Statistical Mechanics and its Applications 392(20), 5206–5217 (2013)
8. Zhu, M.-J., Wang, J.-Y.: Study on Modeling and Simulation of Command Organization. Journal of System Simulation 19(21), 5008–5014 (2007)

SanTrain: A Serious Game Architecture as Platform for Multiple First Aid and Emergency Medical Trainings

Axel Lehmann[1], Marko Hofmann[2], Julia Palii[1], Alexandros Karakasidis[2], and Patrick Ruckdeschel[1]

[1] Institut für Technische Informatik
Universität der Bundeswehr München,
Germany
[2] Institut für Technik Intelligenter Systeme (ITIS)
Universität der Bundeswehr München,
Germany
axel.lehmann@unibw.de

Abstract. Rapid innovations of ICT enable ubiquitous, mobile and smart computer applications almost anywhere at any time. Permanently growing pervasiveness of these systems and applications at reasonable costs leads to significant changes of learning and training platforms and environments. While virtual simulation and virtual reality have played a dominant role as training platforms over past years, serious games are now receiving more and more attention due to added visualization, AI capabilities, and their availability on smart mobile devices.

As design and development of a smart game is extremely complex and expensive, its sale depends on its attraction for a broad gamer community (ranging from "digital nerds" to novice users) and on multiple usability. Therefore, the cost-effective development of a game-based learning and training platform for very specific domains like in our case for military, medical or first aid applications with limited number of potential users require the design of flexible, cost-effective game architectures for multiple use. This paper proposes such a flexible serious game architecture as first aid and medical emergency training platform for military personal. It reports on a research project sponsored by the German Federal Armed Forces (Bundeswehr).

1 Introduction

Our developing information and knowledge societies require life-long learning and location-independent training capabilities, which create needs for innovative learning and training environments and concepts. The current development of computer and communication technologies (ICT) towards pervasive and smart computing offers new possibilities for learning and training, e.g. for blended learning, virtual simulation or serious game applications. As a consequence of these ICT trends, access to and application of information and expert knowledge is increasingly feasible almost any time at any location, thereby providing solutions to meet those learning needs. Though younger users can mostly cope with this smart and ubiquitous computing, and

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 361–366, 2013.

as experts of a specific domain are having different learning requirements compared to novice users, such learning platforms have to be adaptable to the needs of a wide range of users.

2 Project Requirements for a Cost-Effective Training Platform

Modern military forces have sought to reduce deaths on the battlefield by training large numbers of ordinary troops to offer fast and excellent battlefield first aid of particularly lethal injuries even when a medic is not present. As recent military conflicts have shown, intensive training of Tactical Combat Casualty Care (TCCC) principles can save lives on a battlefield (see Ref. (1)). To train these TCCC principles, trainees have to learn to follow a sequence of simple life saving steps and strict priorities (triage, diagnosis, treatment). To be effective, those training methods have to be grounded on various kinds of simulated tactical scenarios which are expected to lead the trainee to apply correct first aid or medical treatment in case of an emergency under stress.

With respect to current ICT trends mentioned above, the primary goal of our research project is to develop a concept and demonstrator of a flexible, adaptable and user-friendly learning and training platform for these kinds of application. Regarding major requirements for multiple first aid and emergency medical trainings in a typical military scenario, the project and its included research had to meet requirements summarized in Fig. 1. A second goal of the project is to forward a concept of a modular training platform flexible enough to be adaptable to various kinds of scenarios (e.g. non-military ones), training purposes (injury diagnosis, treatments) and trainee profiles (first aid responders or medics).

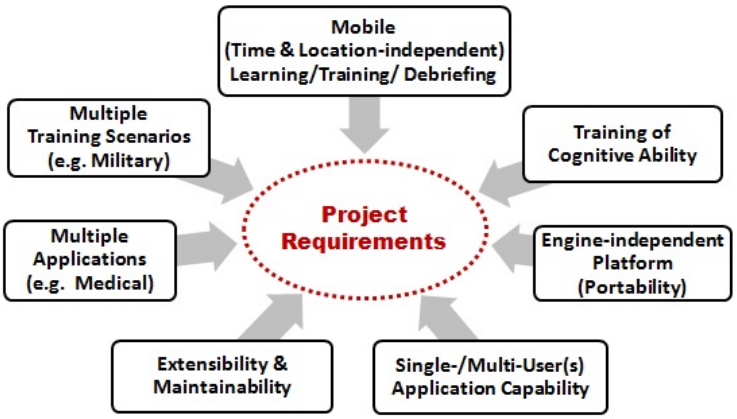


Fig. 1. General Project Requirements

Taking together all the requirements mentioned above and in consensus with the project sponsor we decided to focus our research and concept development towards a serious game based training platform. While powerful and sophisticated interactive, virtual simulation applications and virtual reality platforms are also widely used for these kinds of training, these require the availability of trainer capacities for briefing, trainee guidance and debriefing, which is not available everywhere or anytime. In case of virtual reality training environments, a non-standard and often expensive equipment and infrastructure are required in addition. Meanwhile, serious game platforms offer excellent visualization capabilities for demonstrating realistic stories in domain specific scenarios, and enable interactions between players. They also offer AI capabilities to implement some rule-based or case-based pedagogic guidance, as well as briefing and debriefing capabilities reacting on the trainees activities during the game.

3 Proposed Modeling Approach and SanTrain Architecture

According to the project goals and constraints mentioned in section 2, our conceptual approach for the development of a serious game architecture had to consider the following major demands:

- The concept has to offer an effective blended learning and mobile training environment for trainees with different cognitive skills and abilities (e.g. first aid responders or medics).
- As interactions between trainee and a personal trainer will be not available in gaming mode, abilities and didactic experiences of a trainer have to be modeled and represented in the game as far as possible (Media Didactic Model).
- As the training platform should train cognitive skills of trainees regarding certain types of injuries, pathophysiology of the human body has to be modeled very precise as far as relevant for diagnosis and treatment for those injuries, e.g. for first aid actions as well as intensive medical care (Physiology Model).
- In different scenarios – especially in, but not limited to military ones - trainees have to learn how to react tactically in a specific setting and scenario before starting first aid actions (Tactical Model).

As result of our research we propose a modular serious game architecture for such a training platform, described in Fig. 2 as generic SanTrain architecture.

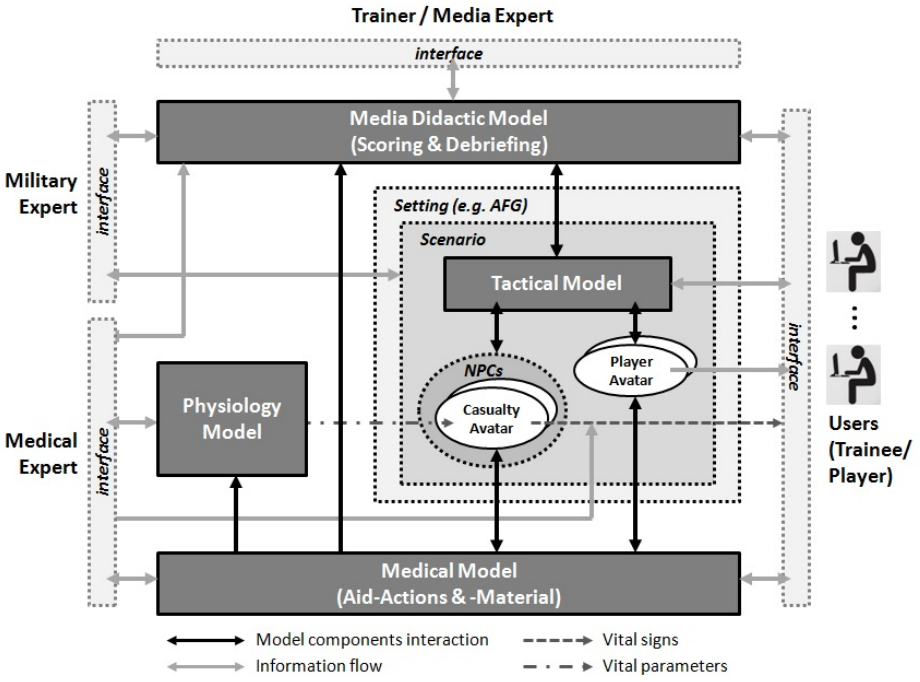


Fig. 2. Proposed Generic SanTrain Architecture

As the trainees background and skills as well as their respective training goals can be very different, the Pathophysiology Model of the vital system has to be able to represent all visible vital signs at the patient´s body important for diagnosis, as well as all vital parameters that are relevant for modeling human pathophysiology, consequences of injuries, as well as effects of first aid or medical treatments (Medical Model):

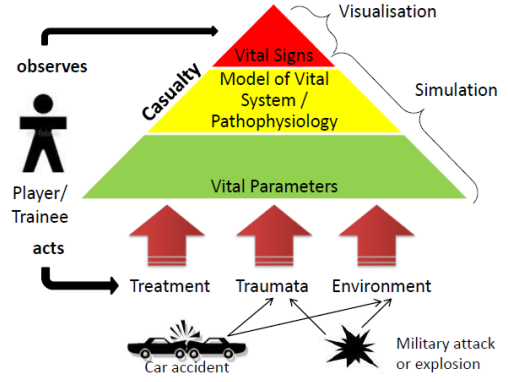


Fig. 3. Basic Relations between Scenario, Player, and Pathophysiology Model

4 SanTrain Demonstrator

As proof of concept, we have already implemented a SanTrain demonstrator with single-user interface and 2D-visualization. This demonstrator includes implementations of a Physiology Model and a Medical Model for first aid treatment (especially for extremity hemorrhage, tension pneumothorax, and airway obstruction), according to TCCC, as well as simplified Tactical and Media Didactic Models. Fig. 4 shows a screen shot of its current user interface. This demonstrator was and still is used for validation and plausibility checks of the models involved, especially for the Physiology and Medical models which are safety critical core models and implementations for correct training (see Ref. (3) and (4)).

Fig. 4. SanTrain Demonstrator: Visualization for the Trainee

Basic modeled components of a casualty – visible as well as invisible ones - are shown in Fig. 5

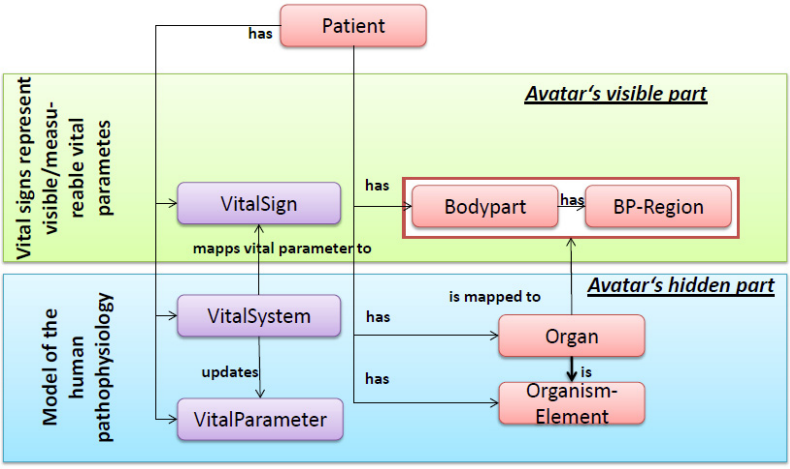


Fig. 5. Basic Components of a Casualty Avatar

5 Conclusions and Future Works

The proposed SanTrain concept and architecture are forming the basis for fulfilling the above mentioned general and specific project requirements and constraints. The separation of pathophysiological, medical and tactical models which interact through well-defined interfaces offers the required flexibility for singular model respectively modul maintenance, extensions, or exchange, which might be required for different skill levels of trainees or training goals, as well as for others than military application domains. Feasibility of the SanTrain conceptual modeling approach and its proposed architecture could be already successfully demonstrated by our 2D-demonstrator implementation, by function tests, as well as by some major plausibility checks and validation activities.

Our current Research and Works are Focused on:

- Further intensive plausibility checks and validation of the Physiology and Medical Models,
- 3D-implementation of SanTrain as a single-player serious game for TCCC training based on the demonstrator experiences,
- conceptualization of a mobile version with off-the-shelf smart products,
- development of a multiplayer version,
- empirical analyses regarding user acceptance and training effectiveness of the training platform.

Acknowledgements. The authors like to thank Sanitätsamt der Bundeswehr (German Federal Armed Forces) for funding this research study, Lars Schneidereit and Kevin Röhrborn for their valuable contributions and support as military experts, as well as Hwa Feron for her contributions to this paper.

References

1. Butler, F.K.: Tactical Combat Casualty Care 2007: Evolving Concepts and Battlefield Experience. Military Medicine 172, 11:1 (2007)
2. Zyda, M.: From Visual Simulation to Virtual Reality to Games. Computer 38(9), 25–32 (2005)
3. Feron, H., Hofmann, M.: Tactical Combat Casualty Care: Strategic issues of a serious simulation game development. In: Laroque, C., Himmelspach, J., Pasupathy, R., Rose, O., Uhrmacher, A.M. (eds.) Proceedings of the 2012 Winter Simulation Conference. Institute of Electrical and Electronics Engineers (IEEE), Inc., Piscataway (2012)
4. Feron, H., Lehmann, A., Josse, F.: TCCC Serious Game Simulation Challenges: Design Requirements, Verification, Methodology, Technology. SAGE Publications, Thousand Oaks (to be published 2013)

Development of an Interactive Game as Educational Tool Using an Off-shelf Simulation Application

David Chan[1] and Poey Ching Neo[2]

[1] Advent2 Labs Consultation Pte Ltd, Managing Consultant, Singapore
chanyk@advent2labs.com
[2] Advent2 Labs Consultation Pte Ltd, Consultant, Singapore
poeychingneo@yahoo.com

Abstract. With recent development of personal computer, interactive learning tools for educational purposes have evolved, from the use of blackboard and physical objects to the recent use of graphical animations on Personal Computers and Tablet PCs. The use of technology not only increases the effectiveness of personal computer as an educational tool, it also helps in engaging a learner. In this paper we attempt to create a computerized version of the popular beer program albeit with the use of 3D animation using an off-shelf simulation tool namely Flexsim^{TM} Simulation System. The approach has expanded the usage of an industrial simulation tool from just a simulation and modelling tool to be used as a platform for interactive game development for educational purpose. With the built-in feature in 3D graphic animation it also shorten the time required to develop the game model.

Keywords: Game, SCM, Simulation.

1 Introduction

The purpose of this paper is to present how a simulation tool can be used to develop educational interactive tool. Discrete computer simulation software has been used in solving operation management problems in many industries. With the advancement of computer technology and the speed of graphic image processing, there is great improvement in the development of 3D graphic presentation to animate processes in the simulation modelling [1]. This enhances the presentation and understanding of the simulation model which is supposed to present the actual process flow. In order to equip the school leaver for the industries, institutions have been using it to teach, train and conduct research in Simulation and Modelling. The demand for computer interactive educational tool increases with time.

Hence we felt that simulation tool could also contribute to the development of such educational tool due to the available development features such as customization using code, the ability to handle images and animations and the possibility to interact with the users and other devices. Our approach is to attempt to create a popular beer game using one such simulation tool.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 367–372, 2013.

2 Beer Game

Beer Game was first designed as an interactive game in the 1960s by MIT [2]. The game was adopted by many Supply Chain Management studies to demonstrate the extreme change in the supply position upstream in the supply chain which could be generated by a small change in demand downstream in the supply chain (known as "Bullwhip Effect") [3]. In the past 50 odd years, new technology has being employed in the recreation of the Beer Game albeit with improvements. The original Beer Game introduced by MIT in the 1960s was done using paper and pencils with groups of learners playing different roles in the supply chain. In year 2007, a 2D single player Beer Game was created and written in C++ programing language, where a paper has been written and discussed [4]. In recent years, a web-based version in which allows participation of multiple players has been created by MIT students using Java programing language.

2.1 2D Beer Game

The 2D Beer Game, a single player game, consists of 4 supply chain layers: factory, distributor, wholesaler and retailer. A single player can play the manager role of any of these 4 layers, and the computer will play the other layers' role. It is assumed that order quantities of each layer are not known to the other layers. The game is played for a number of rounds. In each round, the manager will have to decide on its order size and communicate only this to his upstream supplier. Information is transferred with delays because of lead time as in real operations and also resulted in delay.

Under these conditions, even with almost stable consumer demand, inventories, backlogs and their respective costs incurred in each supply chain layers accumulate considerably. This game successfully showcases the bullwhip effect, which is a result of local decision made with insufficient information.

A typical view of the beer game that had being created using 2D graphic representation is shown in Fig. 1 [2]. It uses graphics of a side view of a truck to represent the 2 units of delay times, and a side view of a building to represent warehouse that contains the inventory.

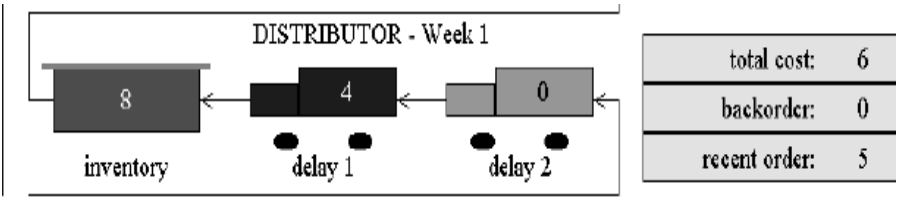


Fig. 1. Snapshot of a 2D beer game

2.2 3D Beer Game Requirements

Creation of a 3D beer game requires the following abilities in the software program.

a) Firstly, it needs to allow for import of 3D graphics. This gives us free hand on design of the presentation of the game, where lots of existing 3D graphics can be found on the web. For example, how our carton of beer is represented.
b) Secondly, it needs to be capable of animating the movement of imported 3D graphics, such as the movement of trucks or ships transporting cartons of beer between different locations, for example, between factory and distributor, during the game play.
c) Thirdly, a mechanism that allows creation of decision points to process or compute the cost, order fulfilment quantity and shipment quantity of every supply chain layer.
d) Fourthly, it needs to provide capability of customizing graphical user interface. The graphical user interface will includes dialog boxes, where user can input information such as the order size. The program should include the presentation of results using graphs and/or table. It will be helpful in the presentation of the KPI, that is, the cost incurred by the supply chain layer.

Base on the above requirements, there can have a few approaches in creating the 3D beer program. The often used approach is to identify a 3D game engine which fulfils the first 2 requirements preferably provides as much capability either through directly providing the third and fourth requirements, or allowing of plugin libraries to achieve them. This approach requires familiarity in the development of 3D game engine. As we are creating an educational tool for study of supply chain management, it may not be easy to find someone that possesses both the knowledge of supply chain management and development skill in 3D engine environment. Though we can have project that involves cross faculty experts, it definitely is a better approach if experts in supply chain management can also be the developer of supply chain management education tool.

2.3 3D Beer Game

The above 4 requirements can be met by a commercial supply chain simulation tool FlexSim™ (http://www.flexsim.com). This software has several levels of animation – virtual reality (VR), 3D and 2D. The engine enables fast and dramatic presentation of the animation and graphics. This software has discrete event simulation engine which fulfils the third requirement, whereby when an order is received, this event will trigger a series of events such as delivering goods to downstream, receiving goods from upstream and update of stock in warehouse. We have used this simulation software and created a version of the beer game, and here onwards we shall refer to it as the 3D Beer Game.

The 3D Beer Program is a single player (desktop) game. The game begins with a user interface which allows the player to choose the game configuration. There are two possible configurations. The first one consists only of a Retailer Layer and

Factory Layer. It simulates a "centralized mode" as describe in [5]. The other has all 4 layers, that is, Retailer, Distributor, Wholesaler and Factory. The "centralized mode" is devised in an improved version of beer game created by Kaminsky. It collapses the factory, distributor and wholesaler into a single layer. This takes into consideration where the same owner operates all 3 layers, that is, factory, distributor and wholesaler.

Fig.2 shows the 2 game configurations. The left figure shows the first configuration with Retail and Factory Layers. The right figure shows the configuration with Retail, Distributors, Wholesales and Factory. The coloured frame details the boundaries of each supply chain layer, that is, cost incurred by manager of each supply chain layer.

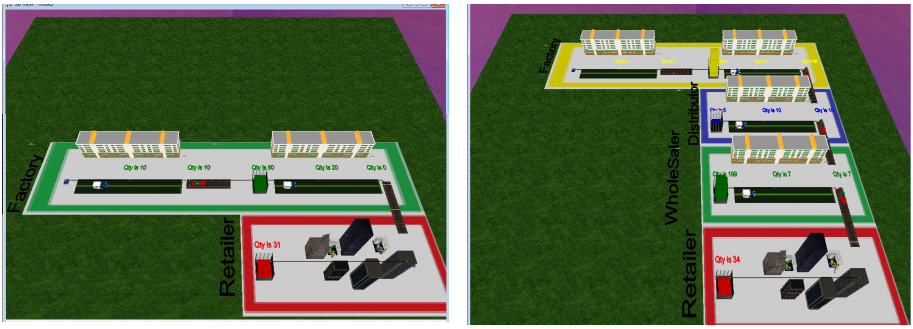


Fig. 2. Overall Representation of various "Play" modes

The left hand picture of Fig. 3 shows the Factory with trucks and trains moving towards the warehouse. The game presents the feature of transport delay time visually, with animated movement of trucks and trains representing time needed to "move" products from one geological location to another. The animated movement of stacking goods in the pictorial warehouse gives intuitive information of inventory in various layer of supply chain, while the actual inventory amount is displayed nearby. The quantity of goods in transport is displayed near the vehicle.

The right hand picture of Fig. 3 shows the City View which represents the Retail Layer. The high rise buildings with "people" provide visual representation of the consumer purchasing products. These provide more contextual information compare to a typical 2D game where lines are used to represent connection and static pictures with text use to label various layers.

This game will run for a number of rounds. Each round, the duration of the game is determined by the simulation time counter. When the predefined simulation time is reached, the model will pause and brings up the dialog box that allows the player to enter his order quantity. The player can continue with the game after the "Run" button is clicked.

At the end of the game, a final reporting dialog box will show the final costs incur in the game, as shown in Fig. 4. There are two parts in the final report. The first part is the inventory, order and fulfilment status for each layer in each round. The second part is Total Cost, Inventory Holding Cost and Backlog Cost for each layer derived from the inventory condition in each round.

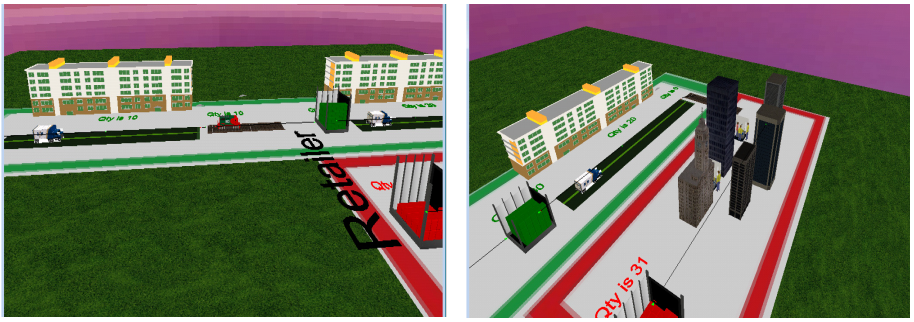
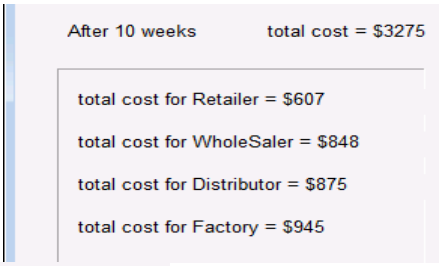


Fig. 3. Transport View and City View



nventory Information

	Retailer					Distributor			Factory		
Week No	Inventory Inl	New Custome	Backlog Qty	Goods Receiv	Order Placed	Total Cost	Inventory Co	Backlog Cost	Total Cost	Inventory Co	Backlog Cost
1	9	6	0	0	21	7.50	7.50	0.00	7.50	7.50	0.00
2	4	5	0	0	7	30.00	15.00	15.00	18.50	18.50	0.00
3	0	8	4	0	7	82.50	22.50	60.00	130.50	40.50	90.00
4	0	30	19	15	89	154.00	26.00	128.00	382.50	202.50	180.00
5	0	50	69	0	90	251.50	33.50	218.00	665.00	364.00	301.00
6	0	34	88	15	7	372.00	41.00	331.00	826.50	525.50	301.00
7	0	24	112	0	7	607.50	55.50	552.00	977.00	676.00	301.00
8	0	5	110	7	7	835.50	130.50	705.00	1277.00	766.00	511.00
9	0	3	105	8	7	1056.00	221.00	835.00	2004.00	1276.00	728.00
10	0	2	100	7	7	1182.50	307.50	875.00	2674.50	1729.50	945.00

Fig. 4. Summary and Detail Report of the game

3 Conclusion

It is common to find simulation tool in an education institute. Simulation technology is a mature technology that is being employed in many different fields of study. In Supply Chain Management studies, simulation is often used to study the effect that changes in variable(s) have in overall performance of a supply chain [5]. Recent development of supporting 3D graphics in simulation tools enables the presentation of the intricate relationship between the different layers in the supply chain. The simulation model will further enhance the learning bullwhip effect in supply chain management.

The 3D beer game is created using only features provided within the simulation tool itself. The simulation model is relatively simple model. The customized logic is created using FlexScript™ language, which is utilized in the creation of the scheduled events. The graphical interface is provided therein and with some logic customization using FlexScript™ to present the final calculated cost. As all the features are provided

within the same tool, the development depends on familiarity of the tool. With the further enhancement in this tool and its technology, it is possible that game developers could develop computer game even with external input device such as joystick using this platform [6] [7].

References

1. Beaverstock, M., Greenwood, A.: Applied Simulation: Modelling and Analysis Using FlexSim 3D Simulation Software, 2nd edn. FlexSim Software Product, Inc., USA
2. Tobail, A., Crowe, J., Arisha, A.: Learning by Gaming: Supply Chain Application. In: Jain, S., Creasey, R.R., Himmelspach, J., White, K.P., Fu, M. (eds.) Proceedings of the 2011 Winter Simulation Conference, pp. 184–191. Institute of Electrical and Electronics Engineers, Inc., Piscataway (2011)
3. Massachusetts Institute of Technology, MIT Forum for Supply Chain Innovation, http://beergrame.mit.edu/
4. Cos, J.F.: Bullwhips effect. APICS Dictionary 13th edn. APICS: http://www.apics.org/dictionary/dictionary-information?ID=454 (accessed on Auguest 18, 2013)
5. Knolmayer, G., Schmidt, R., Rihs, S.: Working Paper No 197: Teaching Supply Chain Dynamics Beyond the Beer Game. In: Working Paper. Institute of Information Systems (2007)
6. Kaminsky, P., Simchi-Levi, D.: A New Computerized Beer Game: A Tool for Teaching the Value of Integrated Supply Chain Management. Northwestern University (2007)
7. Onyesolu, M.O., Ezeani, I., Okonkwo, O.R.: A Survey of Some Virtual Reality Tools and Resources, Virtual Reality and Environments, C.S. Lányi (Ed.). InTech (2012), doi: 10.5772/39062, ISBN: 978-953-51-0579-4, Electronic document, http://www.intechopen.com/books/virtual-reality-and-environments/a-survey-of-some-virtual-reality-tools-and-resources (accessed July 20, 2013)
8. Pellegrin, O.: Boat Simulation in FlexSim (2012), https://www.youtube.com/watch?v=p2PBu9LrO2Y (accessed on July 20, 2013)

Particle-Based Simulation of Fluid-Solid Coupling

Xinyi Yin, Xukun Shen, Fengquan Zhang, and Guanzhe Huang

State Key Laboratory of Virtual Reality Technology and Systems,
Beihang University, China

Abstract. We present an efficient framework for fluid-solid coupling, including both rigid and elastic bodies. Based on a unified particle model, we apply different coupling scheme for fluid-solid and solid-solid coupling respectively. Realistic and versatility are achieved by coupling all the objects adaptively. With the help of GPU accelerating, we gain a real-time simulation.

Keywords: SPH, fluid, rigid, elastic, plastic.

1 Introduction

The fluid simulation has always been a hot topic in computer graphics due to its varied behaviors, including wave, vortex, droplet, bubbles and foam. And it gets more interesting when fluid interacts with solid, which is universal in the real world: A boat float in the lake, a balloon imbues with water, coffee pours on the cloth, there is a inevitable need for fluid-solid coupling simulation which could be widely used in computer games, virtual surgery, film-making and so on.

In order to capture the diverse behaviors between fluid and solid, various methods have been developed in this field. Both the fluid and solid simulation could be divided into two main categories: mesh-based and mesh-free methods. The Lagrangian method is a based on particles, whose free nature enables more details of fluid behavior to be presented. The Smoothed Particle Hydrodynamics (SPH) could be extended to simulate different kinds of solids, as well as phase changing. Undeniable, Mesh-based method gains noticeable achievements in these areas. However, the grid structure makes it hard for reconstruction when simulating complex physical phenomena such as solidifying and fusion. Besides, the development of GPU technologies also contributes to the wide use of Lagrangian method.

The main contribution of this work is a Lagrangian simulation framework which captures the behavior of two-way fluid-solid coupling. With a unified particle representation of both fluid and solid, we integrate various physical dynamics to model different kinds of solids. An effective coupling scheme is presented which suits our GPU accelerating platform.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 373–378, 2013.

2 Related Work

SPH is a lagrangian method for fluid simulation. Muller [1] derived force fields from the famous Navier-Stokes equation and designed classical smoothing kernels for viscosity and pressure calculating. Based on previous work, fluid simulation was speed up with more details. Solid simulation is a well-studied subject and numerous methods are established such as FEM, FVM, position-based methods, a detail survey is presented in [2]. Many kinds of solids could be modeled, such as glass, ice, rubber, cloth, accompany with numerous interesting natural phenomena: fracture, melting, fusion, solidification and so on.

The behavior between fluid and solids is complex. Apart from porous materials , the key problem in fluid-solid coupling is to avoid penetration at the boundary of solids and achieve realistic effect. Lennard-Jones penalty force prevailed in the boundary handling while its limitations calls for other innovations. Freezing particles and ghost particles are proposed to solve the problem. Ihmsen [3] discussed these methods and introduced PCISPH algorithm. However, it's one-way coupling and time consuming.

With so many methods presented, our method is inspired by the ideal of [4,5]. We apply a unified particle representation of fluid and solids, specific physical-based models are implemented to model different kinds of solids including rigid, elastic and plastic bodies. We revised the work of [5] to simulate the fluid-solid coupling, while it introduced additional physical engines to handle the interaction of solids. Here we improve the work of modeling granular materials [6] and realize an adaptive collision scheme.

3 Fluid-Solid Coupling

3.1 Framework Overview

In this section, we describe a two-way fluid-solid coupling based on the GPU accelerating platform. Section 3.2 presents a brief introduction of several basic physical models used in our system, followed by a description of our coupling scheme in section 3.3. The results are discussed in Section 4.

3.2 Physical Model

SPH Concept. Our fluid simulation is based on Weakly Compressible Smoothed Hydrodynamics (WCSPH)[7], We revise the classical equations of SPH to highlight the contribution of particle's volume:

$$\rho_i = \rho_0 \sum_j V_j W(r_{ij}, h) \tag{1}$$

$$f_i^{pressure} = -V_i \sum_j V_j (\frac{P_i + P_j}{2}) \nabla W(r_{ij}, h) \tag{2}$$

$$f_i^{viscosity} = \mu V_i \sum_j V_j (\nu_j - \nu_i) \nabla^2 W(r_{ij}, h) \tag{3}$$

Solid Bodies. We present both rigid and deformable bodies in this paper, the former is based on rigid dynamics[8]. The motion of rigid body is divided into translation and rotation. On our parallel platform, We focus on the behavior of single particle rather that the whole body. With the linear and angular velocity of the body calculated, the velocity of single particle is presented by:

$$\nu_i = \nu_{linear} + \omega \times r_i \tag{4}$$

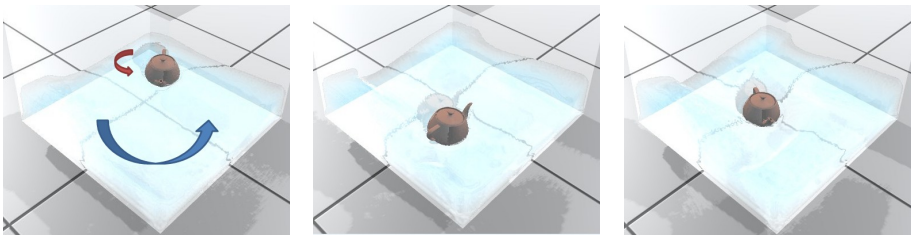


Fig. 1. A teapot is involved to the center of the vortex

Compared with rigid bodies, elastic bodies deform when force is exerted on the body. Muller [9] proposed a point based elasticity model derived from continuum mechanics. We implemented this work and made some modification.Considering we only use the surface particles of solids so the number of neighbor particles decreases, which may lead to errors when computing the displacement gradient with Moving Least Squares. Besides, time spend on the inversion of matrix far exceeds the calculation of vector. So we replaced MLS with SPH method [4]. We revise the elastic force equation in order to gain a symmetric force between a pair of particles, which also suits our parallel platform.

$$f_{ij} = (V_i J_i \sigma_i + V_j J_j \sigma_j) d_{ij} \tag{5}$$

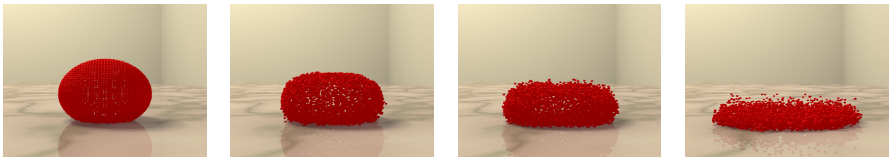


Fig. 2. A ball deforms under gravity, the plastic range is increased from left to right

While there is no pure elasticity objects in the world. When the deformation exceeds the threshold, the counter energy is absorbed by the material and the shape is changed permanently. We implement the work of [10] to model plasticity.

3.3 Coupling Scheme

An efficiency method was proposed in [5] to handle the fluid-solid coupling. It takes the contribution of boundary solid particles to influence the behaviors of fluid particles, which is actually an improvement of "freezing particles" method. Only a little modification of classical SPH equations is needed to handle the force between fluid and solid particles:

$$f_{ij} = -V_i V_j{}^{solid}(\frac{P_i + P_j}{2})\nabla W(r_{ij}, h) + \mu V_i V_j{}^{solid}(\nu_j - \nu_i)\nabla^2 W(r_{ij}, h) \quad (6)$$

In our experiments the fluid sticks to the solid due to the viscosity. Here we choose smaller viscosity coefficient to make the fluid flows smoothly on the solid. And the coefficient value depends on the attribute of solid. For example, the fluid behaves more sticky when flowing on the cloth than the glass.

While [5] used bullet to handle the collision between rigid bodies. Here we propose a adaptive coupling scheme for different kinds of solids. Instead of studying the collision of different bodies in the macroscopic view, we focus on the interaction between pairs of particles in the contact region. This discrete method not only simplifies the problem, but also enables parallel computing.

We extend the work of simulating granular materials [6]. In this model, every particle stands for a sphere with a radius R. A little overlap is acceptable and its used to compute the contact force. We set the value of R related to the particle's volume, which prevents penetration in coarsely sampled region. The contact force is based on Hertz theory, which takes the attributes of materials into consideration:

$$F_n = -(k_d \xi^{\frac{1}{2}} \nu_n + k_r \xi^{\frac{3}{2}}) \quad (7)$$

Where ξ is the overlap between two spheres, and ν_n is the magnitude of normal velocity. k_d and k_r is viscous damping and elastic restoration coefficient respectively. And all these variables are depends on physic attributes such as velocity, particle's radius, Young's modulus.

4 Results

We have presented several scenarios on 2.67GHz Intel(R) core(TM) i5 PC with a NVIDIA GeForce GTX 480 graphics card. The rendering is achieved by our own pipeline, as for elastic bodies, we turn to POV-Ray.

Fig. 1 shows a teapot flows in vortex, the two kinds of rotation illustrate the physical description of rigid bodies. The teapot is represented by 7207 surface particles, dropped into 63K fluid particles. The time for just fluid simulation is 0.013 seconds per frame. And it increases to 0.065 when 10 teapots were added.

An elastic ball with 6K particles is shown in Fig. 2. The young's modulus is $13 \cdot 10^6 N/m^2$. We set the elastic limit to be 0.013, and the ball got flat more and more when the plastic limit increased from 0.009 to 0.08. The right-most picture illustrates that the deformation energy is totally absorbed by the material.

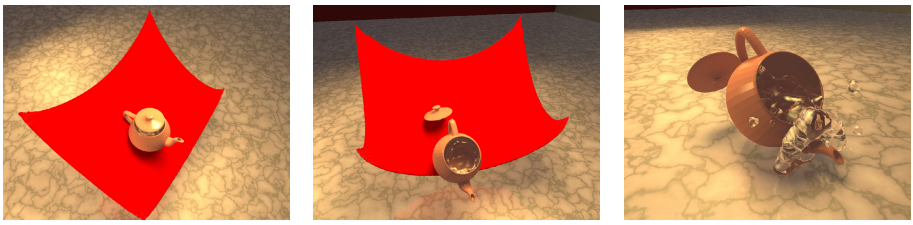


Fig. 3. A teapot is bounced off by an elastic panel whose corners are fixed

In Fig. 3, The red elastic panel is represented by only one layer of 16K particles. Collision exists between the teapot and the panel, as well as the lid and the teapot. The young's modulus of the panel and teapot are $10^7 N/m^2$ and $5500 N/m^2$ respectively. So the collision effect of teapot-lip is different from that of teapot-panel. The average collision radius is 0.006 while the smooth radius of this scenario is 0.037. And the average collision duration is about 0.007 seconds.

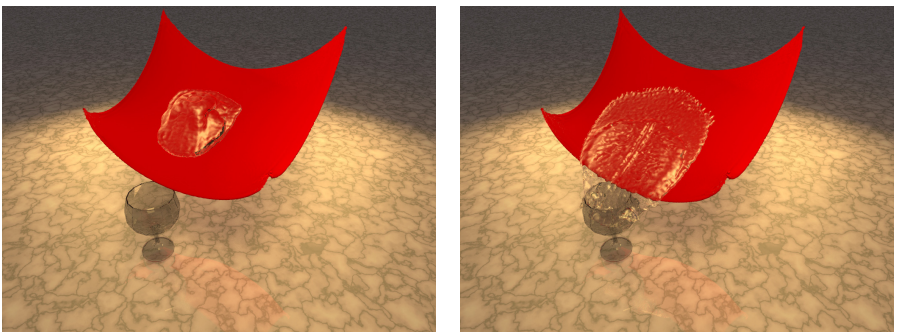


Fig. 4. Water falls into the elastic panel and flows into the goblet

The fluid-rigid and fluid-elastic coupling both exists in Fig. 4, We set the viscosity coefficient of the panel lager than that of the goblet. The water flowed down the elastic panel and stick to the panel a little and spread out, dropped into the goblet and splashed. It takes about 0.021 seconds for each frame.

Compared with [5,11], no physical engines are used and the flexibility of materials is implemented in our system. [4] focus on the state change of solids while pays little attention to the interaction between objects. With the adaptive coupling scheme presented in our paper, realistic and versatility are achieved.

5 Conclusions and Future Work

We presented a unified particle model for fluid-solid simulation. Real-time effect is gained with the acceleration of GPU platform. We improved existing fluid-solid and solid-solid coupling scheme to cater the free natural of particles, which

enables us to present vivid behaviors between fluid and solids. Air particles may be introduced into our system in the future. While problems exist in our system. The elastic particles oscillate in regions undergoing large deformation. This happens when the impact velocity is high in collision. Overall, our method is a unified particle solution of fluid-solid coupling.

Acknowledgements. This research work is supported by National High Technology Research and Development Program of China under Grant No.2013AA013803.

References

1. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 154–159. Eurographics Association (2003)
2. Nealen, A., Müller, M., Keiser, R., Boxerman, E., Carlson, M.: Physically based deformable models in computer graphics. In: Computer Graphics Forum, pp. 809–836. Wiley Online Library (2006)
3. Bender, J., Erleben, K., Teschner, M.: Boundary handling and adaptive time-stepping for pcisph. In: Ihmsen, M., Akinci, N., Gissler, M., Teschner, M. (eds.) Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2010)
4. Solenthaler, B., Schläfli, J., Pajarola, R.: A unified particle model for fluid–solid interactions. Computer Animation and Virtual Worlds 18(1), 69–82 (2007)
5. Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., Teschner, M.: Versatile rigid-fluid coupling for incompressible sph. ACM Transactions on Graphics (TOG) 31(4), 62 (2012)
6. Bell, N., Yu, Y., Mucha, P.J.: Particle-based simulation of granular materials. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 77–86. ACM (2005)
7. Zhang, F., Shen, X., Long, X., Zhao, B., Hu, L.: Real-time particle fluid simulation with wcsph. In: Pacific Graphics Short Papers, pp. 29–34. The Eurographics Association (2012)
8. Baraff, D.: An introduction to physically based modeling: rigid body simulation iunconstrained rigid body dynamics. SIGGRAPH Course Notes (1997)
9. Müller, M., Keiser, R., Nealen, A., Pauly, M., Gross, M., Alexa, M.: Point based animation of elastic, plastic and melting objects. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 141–151. Eurographics Association (2004)
10. O'Brien, J.F., Bargteil, A.W., Hodgins, J.K.: Graphical modeling and animation of ductile fracture. ACM Transactions on Graphics (TOG) 21(3), 291–294 (2002)
11. Akinci, N., Cornelis, J., Akinci, G., Teschner, M.: Coupling elastic solids with smoothed particle hydrodynamics fluids. Computer Animation and Virtual Worlds (2013)

A Fast Outlier Detection Method for Big Data

Boyuan Liu, Wenhui Fan, and Tianyuan Xiao

State CIMS Engineering Research Center, Tsinghua University, Beijing, China
Liuboyuan@126.com

Abstract. Outlier in simulation can help people to know the defect of simulation system. With the rapid expansion of data scale, conventional outlier detection methods begin to have trouble dealing with large datasets. In this paper, we propose an Entropy based Fast Detection (EFD) algorithm which incorporates the new ideas in handling big data. The algorithm takes the information entropy measure as the core, with attribute frequency value as the auxiliary. By means of rapid computation of decreased entropy, the outliers can be got quickly. The results show that EFD algorithm can detect the outliers in high efficiency without obvious loss of accuracy.

Keywords: Outlier, Information Entropy, Big data.

1 Introduction

Outlier detection is to find the data which deviate from the other obviously from massive data. What is outlier? Hawkins's definition[1] exposes the essence of outlier to a certain extent. "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism." By means of researching on outlier, people can find some unexpected knowledge but usually very useful. Because of its great value, outlier detection has been widely used in different fields.

In simulation field, because simulation system scale continues becoming larger, the data size researchers can utilize also becomes bigger than ever, and therefore it is usually hard to evaluate the credibility of simulation system directly by analyzing massive data. Outlier can provide additional information to indicate the fault of system and help to improve the system performance.

Facing massive data, many conventional methods are not suitable for large datasets anymore. In Viktor Mayer-Schonberger's "big data", one important shift he put forward in analyzing big data is "mass data permits us to loosen up our desire for exactitude"[3]. Based on this thinking, in this paper we present an efficiently outlier detection algorithm that can perform well for large dataset.

2 Related Works

Previous researches on outlier detection mainly focus on the following directions:

Statistics based methods[4]: Assuming data to be detected meet the model of certain distribution, the data does not follow the distribution is considered to be the outlier.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 379–384, 2013.

Clustering based methods: these methods are generated when applying clustering methods on data mining, and outlier is generated as byproduct.

Neighbor based methods: by means of comparing the difference degree with its neighbor. According to the different measure, the methods can be divided into two kinds: distance-based method[5] and density method[6].

Entropy based methods: according to the measure of information entropy such as Kolmogorov complexity, information entropy and so on, analyze the content of information to decide the outliers.

Statistics based algorithm requires distribution characteristic in advance, which limits its range of application. Rajarman stated that because of "curse of dimensionality", in high-dimensional space the distances between every point are almost the same[7], which indicates the distance based algorithms that no longer suit for large dataset. Entropy based methods is a relatively good way to handle the problem.

3 Entropy Based Fast Detection (EFD) Algorithm

3.1 Problem Formulation

Information entropy put forward by Shannon is used measure the disorder of a system. If X is a random variable, and $S(X)$ is the set of values it can take, and $p(x)$ is the probability function of X, then the entropy $E(X)$ can be defined as follows:

$$E(X) = -\sum_{\mathrm{x}\in \mathrm{S(x)}} \mathrm{p(x)}\ \log(\mathrm{p(x)}) \tag{1}$$

As entropy measures the degree of disorder, entropy based outlier detect algorithm is based on the following hypothesis: the lower the entropy is, the cleaner the data is; the higher the entropy is, the more chaotic the data is. So if the removing of some data will make the entropy of the dataset become lower, these points could be the outlier. The points which will make the highest decreased entropy are the outliers. If k outliers are wanted, what we need to get is a sub dataset O which can satisfy:

$$\begin{cases} \min \mathrm{E(D-O)} \\ O \in D, |O| = k \end{cases} \tag{2}$$

This is an optimization problem in fact. To simplify the question, we assume the independence of different attribute. For a multivariable vector $\mathrm{x} = \{X_1, X_2, \ldots, X_m\}$, Equation (1) can be transformed into Equation (3).

$$\begin{aligned} \mathrm{E(x)} &= -\sum_{x_1 \in \mathrm{S}(X_1)} \cdots \sum_{x_m \in \mathrm{S}(X_m)} p(x_1, \ldots, x_m) \log\big(p(x_1, \ldots, x_m)\big) \\ &= E(X_1) + E(X_2) + \cdots + E(X_m) \end{aligned} \tag{3}$$

He proposed a Local-Search heuristic based algorithm[9] (denoted by LSA) and Fast Greedy Algorithm[8] to solve the problem. Fast Greedy Algorithm has better performance, but facing large data set, its computation process is still complicated and the efficiency is not ideal. Our work is based on this method.

3.2 The Algorithm

If one record in a dataset is removed, the entropy will change. He has proved that after removing one record, the decreased entropy values is[8]

$$DE(i) = \sum_{w=1}^{m} \left(\frac{f(p_i^w)-1}{n_l-1} log \frac{f(p_i^w)-1}{n_l-1} - \frac{f(p_i^w)}{n_l-1} log \frac{f(p_i^w)}{n_l-1} \right) \quad (4)$$

Where p_i^w is the attribute w of record i, and $f(p_i^w)$ is the current frequency count of p_i^w, and n_l is the number of records remained in dataset. Detailed proof can be found on literature[8]. If k outliers are wanted, the decrease value should be calculated for k times, and get one outlier which holds the highest value each time.

In his algorithm this computation is the elementary operation and its time complexity is $O(1)$. Actually, the implementation of function "log" is very complex in computer, and it cannot be finished in few clock cycles. In most cases "log" computation will expand to Taylor series, which will consume a lot of time.

Let us continue the derivation.

$$DE(i) = \sum_{w=1}^{m} \left(\frac{f(p_i^w)-1}{n_l-1} \log (f(p_i^w)-1) - \frac{f(p_i^w)-1}{n_l-1} log(n_l-1) - \frac{f(p_i^w)}{n_l-1} log f(p_i^w) \frac{f(p_i^w)}{n_l-1} + \frac{f(p_i^w)}{n_l-1} log(n_l-1) \right)$$

$$= \sum_{w=1}^{m} \left(\frac{(f(p_i^w)-1) * \log(f(p_i^w)-1) - f(p_i^w) * log f(p_i^w)}{n_l-1} + \frac{log(n_l-1)}{n_l-1} \right)$$

$$= \frac{\sum_{w=1}^{m} \left((f(p_i^w)-1) * \log(f(p_i^w)-1) - f(p_i^w) * log f(p_i^w) \right)}{n_l-1} + \frac{m * log(n_l-1)}{n_l-1} \quad (5)$$

In fact, to mine the outlier, the exact decreased value is not important. What we need to know is the sequence of these values. Considering the result we have deduced, the values of the second term and the denominator in the first term are constant. So the decreased value is determined by this function:

$$f(x) = (x-1) * \log(x-1) - x * \log x \quad (6)$$

Where the variable X is the frequency. The value X can take must be integer, so it can be calculated in advance and stored in memory. When the value is needed, it can be obtained from the memory directly with nearly no delay. It is a time-space tradeoff. Next let us estimate the memory space needed to store these values.

For a double variable to store the function result, it needs 4 bytes. So if 1,000,000 frequencies are needed, the total space is 4*1M=4M. This is perfectly acceptable. And different attributes can use this vector together.

As we mentioned in Section 1, speed is more important than accuracy. Actually for big data, the data is distributed sparse[7]. So if the data size is big enough, the frequencies of these outliers are usually very small. After removing some outliers, the

sequence of the decreased entropy of remaining points in most cases will not change. Therefore it is not necessary to scan the data for many times to get the results. In EFD algorithm, the decreased value for every data will be calculated for only once, and the data take the most k highest value will be regarded as the outliers. This will avoid multiple scans of the data, which will save lots of time certainly.

As the data distribution is sparse, maybe more than one point will be selected at the same time. In this case, it is necessary to rank them and select the most possible outliers. Here Attribute Value Frequency (AVF) is introduced to assist decision.

AVF is proposed by Koufakou[10] which is used as a fast outlier detection strategy. Here AVF is used as an auxiliary criterion to improve the accuracy of our outlier detection. Different from the original definition of AVF, here AVF is:

$$\mathrm{AVF(i)} = \sum_{w=1}^{m} f(p_i^w) \tag{7}$$

It can be seen that AVF equals to the sum of every attribute frequency. If the AVF is low, it means that the attributes of the record tend to be rare, and the chance that the record is the outlier is high. When two or more records have the same decreased entropy, the AVF will be used to sequence these records.

3.3 Time and Space Complexities

It is designated that the size of dataset is n, and the number of attributes is m, and the outlier number k, and the mean number of distinct value of every attribute is p.

The time complexity in initialization phase is $O(nm)$, and the time complexity in computation phase is $O(nm)$, and the time complexity in sorting phase is $O(nk)$. Usually k is bigger than the m, so the total time complexity is $O(nk)$.

To store the frequencies of every attribute, m hash tables are needed, and the space complexity of these tables is $O(pm)$. If the dataset need to be stored in memory, the space complexity is $O(mn)$. So the space complexity of EFD algorithm is $O((p+n)m)$.

4 Experiments

4.1 Experiments Design

EFD algorithm is implemented in C++, and to compare conveniently, Fast Greedy Algorithm is also implemented on our platform. All the experiments are ran on a PC with a Core 2 2.66GHz processor and 4GB of RAM.

Two experiments were carried. Experiment 1 is intended to test the accuracy of EFD algorithm, and experiment 2 is intended to test the efficiency of EFD algorithm.

4.2 Result on Experiment 1

The dataset used for test is the Wisconsin breast cancer data set from UCI Machine Learning Repository. The original dataset has 699 points and 9 attributes. Following Harkins[11], only of every sixth malignant record is kept, resulting in 39 outliers (8%) and 444 non-outliers (92%).

Table 1 shows the result of detection accuracy by different algorithms. The first column is the outlier number wanted, and the other columns is the outlier number that every algorithm can detect.

Table 1. Result On Breast cancer dataset

Outlier number	EFD	Greedy
8	7	8
16	15	15
24	22	22
32	27	29
40	33	33
48	36	37
56	39	39

It can be seen that the performance of algorithm is not as good as that of Fast Greedy Algorithm. As it is mentioned before, the speed is more important than the accuracy. The accuracy of EFD algorithm is little worse, but acceptable.

4.3 Result on Experiment 2

This experiment is to test the speed and scalability of the algorithms. The large dataset is creased by GAClust developed by Dana Cristofor. The dataset is only used for the scalability test, and the detection result makes no sense. For the setup of GAClust, the number of attributes, the number of classes and the random generator seed is set to be 20, 50 and 5. In all 5 datasets are got with data sizes 100,000, 200,000, 300,000, 400,000 and 500,000. The outlier number is set to 0.1%, which is 100, 200, 300, 400 and 500. The test result is shown in Fig. 1.

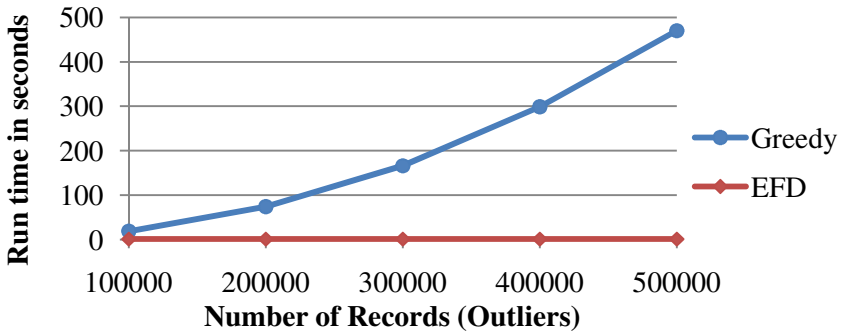


Fig. 1. Result on Artificial Dataset

As shown in Fig.1, the Fast Greedy Algorithm increases at geometric series as the data size grows. Actually, the running time of EFD algorithm in these datasets is always 1 second.

5 Conclusion

With the arrival of the big data age, conventional detection algorithm cannot do well in outlier mining. Based on the new idea during big data handling, the following work are done in this paper: optimizes the computation for decreased information entropy; simplify the mining process of outlier for fast detection; apply AVF factor to EFD algorithm to solve the sequence problem during outlier detection. Experiment results show that EFD algorithm can keep the detection accuracy well meanwhile improve the efficiency markedly.

References

1. Hawkins, D.M.: Identification of outliers. Chapman and Hall, London (1980)
2. Mayer-Sch Nberger, V.C.K.: Big data: a revolution that will transform how we live, work, and think. Houghton Mifflin Harcourt, Boston (2013)
3. Barnett, V., Lewis, T.: Outliers in statistical data (3rd edition). J. Oper Res. Soc. 46, 1034 (1995)
4. Angiulli, F., Fassetti, F.: DOLPHIN: An efficient algorithm for mining distance-based outliers in very large datasets. ACM Transactions on Knowledge Discovery from Data 3 (2009)
5. Breunig, M.M., Kriegel, H., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: 2000 ACM SIGMOD - International Conference on Management of Data, Dallas, TX, United states, vol. 29, pp. 93–104 (2000)
6. Rajaraman, A.U.J.D.: Mining of massive datasets. Cambridge University Press, Cambridge (2012)
7. Zengyou, H., Shengchun, D., Xiaofei, X., Huang, J.Z.: A fast greedy algorithm for outlier mining. Applications of Evolutionary Computing. In: Proceedings of the EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC. LNCS, vol. 3907, pp. 567–576 (2006)
8. He, Z., Deng, S., Xu, X.: An optimization model for outlier detection in categorical data. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 400–409. Springer, Heidelberg (2005)
9. Koufakou, A., Ortiz, E.G., Georgiopoulos, M., Anagnostopoulos, G.C., Reynolds, K.M.: A scalable and efficient outlier detection strategy for categorical data. In: 19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007, October 29-31, vol. 2, pp. 210–217. IEEE Computer Society, Patras (2007)
10. Hawkins, S., He, H., Williams, G.J., Baxter, R.A.: Outlier detection using replicator neural networks. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) DaWaK 2002. LNCS, vol. 2454, pp. 170–180. Springer, Heidelberg (2002)

A Discrete Event Simulation Based Production Line Optimization through Markov Decision Process

Yuan Feng[1,⋆], Wenhui Fan[1], and Yuanhui Qin[2]

[1] State CIMS Engineering Research Center, Department of Automation, Tsinghua University, Beijing, 100084, P.R. China
[2] System Engineering Research Institute, Beijing, 100036 P.R. China
fengyuan1216@gmail.com

Abstract. In this paper, we built simulation model of the production line from one car engine parts plant in Beijing, in order to find proper solutions to raise productivity. The method of Discrete Event Simulation was used to construct the simulation model on account of the fact that production line was a typical discrete event system. Besides, worker heterogeneity, stochastic environment and the effect of worker learning and forgetting were introduced into simulation model to make it closer to reality. We proposed different schedule policies to manage the running of production line with the verification from simulation experiments. Then, by taking advantage of the simulation results obtained previous, we built the optimization model by applying Markov Decision Process (MDP) to seek for the best policy promoting the productivity of production line.

Keywords: discrete event simulation, production line, learning and forgetting, markov decision process.

1 Introduction

The productivity of production line is quite significant because it determines whether the company can make the best use of its facilities to fulfill customers' orders. In this paper, we studied a production line from one car engine parts company, which is located in the suburbs of Beijing. The production line is made up of numbers of stations, and each station is equipped with one machine and one fixed worker. All stations are lined up together from beginning to end in the order of processing sequence. Besides, there is one buffer zone between two adjacent stations to store temporary products. Those temporary products are products that have been processed by the previous station and are waiting to be processed by the afterward station. Moreover, we need to introduce the definition of procedure, which means that the stations that belong to one procedure share the same processing technique.

Discrete Event Simulation (DES) is a method to study the system which events happen at discrete, rather than continuous time points [1]. The system is

⋆ Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 385–390, 2013.

driven forward by the happening of events. As for production line, the status of the system changes when events, such as products coming to stations, products being processed and products leaving stations happen [2].

Obviously, workers need to take a rest when they have worked for a long time. In order to avoid the suspension of production, it is necessary to arrange alternate workers to take their work. Therefore, we encountered two problems here. One is that whether the quick switch among different stations for those alternate workers affects their operation proficiency. The other one is that how to arrange different alternate workers to different vacant positions.

As for the first problem, the alternate workers need to learn more operation skills than fixed workers. Therefore, it will cost them more energy and time to get used to different operation skills [3]. Moreover, if one worker hasn't worked at certain station for a period of time, the worker will forget the operation skill learnt at that station before gradually [4]. In order to simulate this phenomenon, we introduced the effect of learning and forgetting into the model and built the learning and forgetting models for every worker, including all fixed workers and alternate workers within the system.

The study relevant to this effect is broad. Bailey and Yelle conducted a survey about the learning effect [5]. Globerson discussed how the impact of breaks influenced operation skill when one worker is performing a repetitive work [6]. Nembhard and Uzumeri studied the effect based on the data collected by bar code readers and automated acquisition devices from workers in an automotive electronics component plant [7]. Shafer, Nembhard and Uzumeri [8] studied that effect as well, focusing on the relation between heterogeneity of worker pool and assembly line productivity.

The second problem we encountered was solved by Markov Decision Process (MDP). MDP is a discrete time stochastic control process. At one step, MDP calculates the expectations of different actions under current status and finds out the best action. Through iteration, we will obtain the best dynamic action sequence [9].

In this paper, we firstly built the model of production line based on DES with the original data from the plant. Then we introduced the worker models and controlling models, including learning and forgetting effect model into the system. After that, the action models were integrated to take charge of the arrangement of alternate workers when there were available positions to work at. At last, MDP model was applied in the system to optimize the productivity.

2 System Model

2.1 Learning and Forgetting Model

The productivity rate of one worker is

$$\eta_{ws}(t) = \frac{\lambda_{ws}(t)[\xi_{ws}(t)]^{\alpha_w} + p_{ws}}{\lambda_{ws}(t)[\xi_{ws}(t)]^{\alpha_w} + p_{ws} + r_{ws}} \ , w \in \{1, 2, \cdots, W\}, s \in \{1, 2, \cdots S\} \quad (1)$$

$\eta_{ws}(t)$ is the learning and forgetting discount factor for worker w at station s, varying from 0 to 1. $\lambda_{ws}(t)$ is the number of products finished by worker w at station s. α_w is a constant value which represents the learning and forgetting level of worker w. p_{ws} is the experience that worker w gained at station s before production starts. r_{ws} is the fitted parameters for p_{ws} to make $\frac{p_{ws}}{p_{ws}+r_{ws}} = 0.5$. W is the number of workers and S is the number of stations.

$\xi_{ws}(t)$ is the ratio of working time for worker w during the most recent period of time t_0 to t_0, rather than the ratio of average elapsed time to the elapsed time of the most recent unit produced, which was presented by Nembhard and Uzumeri [7]. Because the learning and forgetting model by Nembhard and Uzumeri concentrates on the situation where each worker is only dedicated to one kind of operation. However, in our model, alternate workers need to master more than one operation skills.

2.2 Worker/Machine Energy Model

It is expressed above in section 1 that workers need to rest and machines need maintenance after operating continuously for a relative long time. The worker/machine energy model is the command center which controls the states of all workers and machines. When workers or machines are working, their energy drops. When workers or machines are resting, their energy increases.

2.3 Policy Model

The function of policy model is to manage the alternate workers and arrange them to different stations which fixed workers are resting. Based on our survey [10], we raised five policies (actions) as follows.

1. Action 1: Working time priority.
2. Action 2: Buffer quantity priority.
3. Action 3: Load (Working time×Buffer quantity) priority.
4. Action 4: Experience priority.
5. Action 5: Experience priority (worker rests when energy halves).

3 Simulation Model

Figure 1 presents the overall structure of the simulation model and an example of specific station structure of station 5. The solid arrows in figure 1 stand for the directions of dataflow. From left to right, Simulation Control Model is the model that controls the running status of each station. Worker Control Model and Machine Control Model determine the operation time of every station. The main function of Worker/Machine Energy Model is updating the conditions of all workers and machines, based on the data from Storage Area. Learning and Forgetting Model calculates every worker's working efficiency in real time. Worker Status Statistics Model collects data from simulation model and store data in Storage Area. Policy Model manages the behaviors of alternate workers.

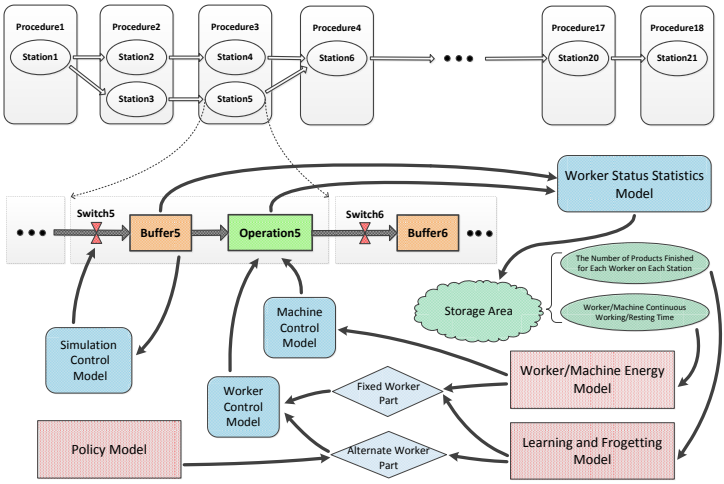


Fig. 1. The structure of production line simulation model

4 Optimization Model

A typical MDP relies on a five dimension tuple, (S, A, $\{P_{sa}\}$, γ, R). Within our model, there are S machines, S fixed workers, 2 alternate workers and S buffers. Each machine has two states. Each fixed worker also has two states. The states of one alternate worker are $S+1$. Every buffer has two states. Therefore, the state space is: $2^S \times 2^S \times (S+1)^2 \times 2^S$.

However, we can find out that the state space is enormous according to the parameters of production line. It is larger than 4×10^{21}. Apparently, it is not practical and possible to implement MDP under such a huge state space. Therefore, we simplified the state space to 3 states: high level output speed, medium level output speed and low level output speed, (S1, S2 and S3). The elements of action set A are the five actions stated above in section 2.3, $A = \{a_1, a_2, a_3, a_4, a_5\}$.

We assume that we have known many state transfer paths as follows:

$$s_2 \xrightarrow{a_1} s_1 \xrightarrow{a_5} s_1 \xrightarrow{a_3} s_3 \xrightarrow{a_4} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \cdots$$

Based on the transfer path above, $\{P_{sa}\}$ can be defined as follows:

$$P_{sa}(s^{'}) = \frac{N(sas^{'})}{N(sa)} \tag{2}$$

$N(sas^{'})$ is the times when we take action a in state s and state s transfers to $s^{'}$ under action a. $N(sa)$ is the times when we take action a in state s.

We assume function $V(s), s \in S$ is the value function of MDP. At the beginning of simulation, $V(s) = R(s), \forall s \in S$. Then, when each time period ends, $V(s), \forall s \in S$ is updated as the function below:

$$V(s) := R(s) + \underset{a \in A}{max}\, \gamma \sum_{s'} P_{sa}(S^{'}) V(s^{'}) \tag{3}$$

The $a_{max} \in A$ makes the right part of equation 3 reach the maximum value. So, a_{max} is the optimal action for the coming time period. Moreover, as the iteration continues, $V(s)$ will become convergence.

5 Simulation Result

Experiments were put forward by contrasting MDP and other five policies (each policy is linked with only one kind of action). The productivity is judged by the output of production line during a relative long enough simulation time. In this paper, the simulation time was set to 4×10^5 (second). And it costs 106.5 seconds to run one time on our computer (Intel^R Core^{TM}2 Quad Q9400, 4GB RAM), and the outputs are as follows in table 1.

Table 1. Output of production line

	MDP	$a_1 \in A$	$a_2 \in A$	$a_3 \in A$	$a_4 \in A$	$a_5 \in A$
1	3199	2597	3158	2834	2769	2988
2	3318	2582	3152	2839	2755	2988
3	3317	2588	3174	2849	2743	2996
4	3313	2593	3173	2844	2762	3011
5	3201	2585	3143	2829	2712	2997
μ	3269.6	2589.0	3160.0	2839.0	2748.2	2996

We conducted 5 groups of simulation, and each group contains 6 simulation processes under 6 different alternate worker assignment policies. It is stated above that there are 5 basic assignment policies, action 1 to action 5. The 6th policy is the method of MDP, which is the mixture of those 5 actions. Through comparison, it is not hard to conclude that the productivity under MDP is the best because the first row has the maximum output.

6 Conclusion

In this paper, we built the simulation model of the production line from one car engine parts company. Firstly, through analysis, we proposed the system

model, integrated with learning and forgetting effect model and alternate worker assignment policy model. The reason why we introduced learning and forgetting effect was that this effect precisely describes the situation where some workers need to learn multiple operation skills rather than one.

Then we simplified the state space and action set so as to obtain the five dimensions tuple of MDP and make it practical to realize. In order to verify the effect of MDP, we conducted several experiments with different configurations in groups. After comparing the results of simulation, it was proved that the application of MDP truly advanced the productivity.

Acknowledgements. This research is supported by National Science and Technology Support Program (Grant No. 2012BAF15G00).

References

1. Kelton, W.D., Law, A.M.: Simulation modeling and analysis. McGraw Hill, Boston (2000)
2. Yeralan, S., Eginhard, J.M.: A general model of a production line with intermediate buffer and station breakdown. IIE Transactions 19(2), 130–139 (1987)
3. Nembhard, D.A.: The effects of task complexity and experience on learning and forgetting: A field study. Human Factors: The Journal of the Human Factors and Ergonomics Society 42(2), 272–286 (2000)
4. Brucoli, M., Carnimeo, L., Grassi, G.: Discrete-time cellular neural networks for associative memories with learning and forgetting capabilities. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 42(7), 396–399 (1995)
5. Bailey, C.D.: Forgetting and the learning curve: A laboratory study. Management Science 35(3), 340–352 (1989)
6. Globerson, S., Levin, N., Shtub, A.: The impact of breaks on forgetting when performing a repetitive task. IIE Transactions 21(4), 376–381 (1989)
7. Nembhard, D.A., Uzumeri, M.V.: Experiential learning and forgetting for manual and cognitive tasks. International Journal of Industrial Ergonomics 25(4), 315–326 (2000)
8. Shafer, S.M., Nembhard, D.A., Uzumeri, M.V.: The effects of worker learning, forgetting, and heterogeneity on assembly line productivity. Management Science 47(12), 1639–1653 (2001)
9. Puterman, M.L.: Markov decision processes: discrete stochastic dynamic programming. Wiley-Interscience (2009)
10. Stratman, J.K., Roth, A.V., Gilland, W.G.: The deployment of temporary production workers in assembly operations: a case study of the hidden costs of learning and forgetting. Journal of Operations Management 21(6), 689–707 (2004)

Fidelity Evaluation for DELS Simulation Models

Yukun Liu and Jing Chen

Logistics Engineering Center, Beijing University of Posts and Telecommunications
Beijing , 100876, CN
lykbupt@163.com, cjlheart@gmail.com

Abstract. The simulation practice in discrete event logistics system (DELS) domain lacks objective and operable methods in fidelity evaluation and management which leads to high costs or inadequate effectiveness of simulations. This paper provides an approach to evaluate the fidelity of simulation models for DELS. Simulation Fidelity is composed of experiment fidelity and model fidelity. Model fidelity is described by the ratio between the simulation model and the understood reality model. Next, fidelity is divided into four dimensions: structural, correlational, temporal and sensorial. Then a three-step operable approach to evaluate fidelity using Fidelity Evaluation Scale is presented by decomposing and analyzing the model in Fidelity Template, enumerating and calculating the indicators of the dimensions. An example is given showing how the approach is used in simulation practice in logistics context. This approach provides an operable and quantitative evaluation of fidelity which will benefit the trade-off between the costs and effectiveness of simulations of DELS.

Keywords: fidelity, evaluation, scale, discrete event, logistics system.

1 Introduction

The simulation practice in discrete event logistics system (DELS) lacks objective and operable methods in fidelity evaluation, design and management issues which results in inadequate simulations or too much costs[1]. In order to make cost-effective simulation, frameworks and operable evaluation methods of fidelity are needed.

Researchers have proposed important frameworks and methods for training simulation[2], distributed interactive simulation[3] and distributed simulation using the HLA architecture [4] . Kim discussed simulation fidelity issues to improve simulation modeling productivity and proposed a formal modeling framework for comparing discrete event system simulation models in terms of fidelity in the manufacturing field[5]. But we have to tailor it or set up a new framework that is more suitable for logistics context. We aim at working out a practical approach to evaluate, design and manage simulation fidelity throughout the process of modeling and simulation within DELS domain. This paper is our first step, focusing on how to measure and evaluate fidelity of simulations of DELS in practice.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 391–396, 2013.

In the reminder of this paper, a formal description of fidelity is given in section 2 and a fidelity dimension division is provided in section 3; then, an operable approach to evaluate fidelity using Fidelity Evaluation Scale is presented in section 4; section 5 shows how Fidelity Evaluation Scale method is used in practice with an example in logistics context; conclusions and future work are discussed in section 6.

2 Concept of Fidelity

The extant literature presents many definitions of the term fidelity, for example: "The degree to which the representation within a simulation is similar to a real world object, feature, or condition in a measurable or perceivable manner" [6]. We noticed that when people discuss fidelity, two levels are usually included, the structure of the simulation model and its behavior when running simulation. In this paper, we describe fidelity from the structural aspect.

Simulation fidelity is composed of experiment fidelity and model fidelity. Experiment fidelity tells how the simulation experiment scheme is designed to achieve the simulation objectives while the model fidelity tells how well the model represents the object system. A model with high model fidelity is capable to serve a simulation analysis with low experiment fidelity though this is not economical, but a model with low model fidelity will fail to serve a simulation analysis with high experiment fidelity. We discuss fidelity on the model fidelity level in this paper.

We describe fidelity as the ratio between the simulation world and the object real world (existing or imagined reality) and fidelity can be formally described as:

$$Fidelity = \frac{M_S}{M_R} \tag{1}$$

where M_S is the simulation model, and M_R is the reference model abstracted from the understood reality[7]. To find an operable approach to describe M_S and M_R,a multi-dimensional metric is always expected. So we will discuss the dimensions next.

3 Dimensions of Fidelity

Before working on the fidelity dimensions, we need to understand the characteristics of simulations of discrete event logistics system(DELS) and their differences from other simulations such as training simulations with a simulator. The DELS simulations possess the following features: the object system is more often a future system in imagination than an existing system; the simulation model usually has no hardware and no man in-the-loop; the purpose of simulation, in most cases, is to understand, analyze, evaluate and optimize the system and in some other cases, to train or teach by simulation; the object system is usually a compound model made up of queuing systems, inventory systems, transport network systems and other systems.

Based on the features of simulation in DELS domain, we divide fidelity into four dimensions: structural, correlational, temporal and sensorial. The structural dimension tells what sub-systems, components and entities the model contains and the structural

characteristics of the entities including position, shape, size, color, etc. The correlational dimension describes the logic relationship between each discrete event that happens to each entity. The logic includes: autogenetic logic for independent stochastic arriving events, time-driven events; interactive logic such as the flow sequence (input/output), triggering, mutual restrictions; the strategies and algorithms used. The temporal dimension tells how well the time features of all discrete events are modeled, namely how well each event happens at the exact moment it should happen. The sensorial dimension describes how well the simulation model is built from the view of the user's feeling. Typically, sights, sounds, and tactile sensation are often involved.

4 Steps to Evaluate the Fidelity of a Simulation Model

4.1 Step 1: Analyze a Model by a Fidelity Template

A Fidelity Template is a table structure used to decompose and record the model by tree-structured items which is built up according to a class of similar object systems.

Firstly, we decompose a model from structural dimension into sub-models, modules, basic process units and enumerate involved entities and their structural parameters. Secondly, from correlational dimension we enumerate all discrete events that will happen to each entity and describe all logic relations for each discrete event. Thirdly, describe the time features of all discrete events and fourthly, select indicators to describe the sensorial dimension.

After step 1 we get two expansion tables from the original template, referred as Fidelity Table, one for the simulation model, the other for the reference model.

4.2 Step 2: Create a Fidelity Evaluation Scale

A Fidelity Evaluation Scale is divided into the same four dimensions. Within each dimension, we use a two-sub-dimension array to measure the fidelity. The first sub-dimension is the model structure, and the second sub-dimension are the indicators of each dimension summarized from the Fidelity Table gained in step 1. See the example of Table 1 in Section 5 .

4.3 Step 3: Calculate the Ratios in the Fidelity Evaluation Scale

The last step is to calculate the ratios of all indicators summarized from the Fidelity Table for the simulation model against the indicators summarized from the Fidelity Table for the reference model accordingly in the Fidelity Evaluation Scale Table.

5 An Example in Logistics Context

5.1 The Example Object System and Its Simulation Models

The example system is a reduced version of a carton cigarette replenishment system in a distribution center in which workers take carton cigarettes from cigarette cases in

the stockpiling area into the feeding ports of an automatic sorting machine according to some replenishment strategies. For the system, we have two models differ in structure and have different fidelity designs for different modules.

In Model 1, the two brands of cigarettes in both carton and case units are modeled separately and each case has 50 cartons for both brands; carton cigarettes are temperately stocked in the sorting machine after fed into the ports and the sorting rates of carton A and B are modeled as two statistical models; the workers take cartons from the cases to the ports and when the quantity of case cigarettes is below the reorder point, case cigarettes are replenished immediately; each brand of cigarette has different inventory replenishment strategy both at the 'case' and the 'carton' level; the manual replenishment process is modeled as a queuing system with workers defined as severs, the carton cigarettes modeled as customers and the service time modeled as a statistical model based on the data from another distribution center; there is no capacity limit for workers; in the stockpiling area, Brand A occupies 3 units of stock space, and Brand B, 2; one empty case occupies 1 unit of stock space; the model is a 3D model and provides animation when running simulation; all entities are cubical with different colors; no render effects are in sensorial dimension.

In Model 2, the two brands of carton cigarettes are modeled as one; the cigarette cases, empty cases and buffer space for empty cases are omitted; at the feeding ports, two statistical models are used for each feeding port: the time when a replenishment requirement is generated, and the quantity needed each time; the manual replenishment process is modeled as an AVG system with workers modeled as a combined mechanism with the arm actions modeled in details and the legs and other body parts modeled without action details; the free route is modeled as the fixed route; there is capacity limit for workers, 12 cartons at most for each worker each time; the model is a 3D model and provides animation when running simulation; all entities have a better 3D shape; the model is rendered by light and shadows, and facial material textures.

5.2 Evaluation of the Two Example Simulation Models

First, we analyze the two simulation models and the reference model by a Fidelity Template. We decompose the model into two modules: the stockpiling process of case cigarettes (M1) and the manual replenishment process of carton cigarettes (M2) and into three levels: the model level, the first and second level of modules. We enumerate all entity categories, types and structural parameter types of each entity. Then we enumerate all discrete events with their logic relationships listed. Next we enumerate the events that adopt simplification on time features and finally select some sensorial indicators to describe the sensorial dimension. Analysis results are listed in the tree-structured Fidelity Table which is omitted here due to page length constraints.

Second, for each of the four dimensions we summarize the indicators to describe it from the Fidelity Table into a Fidelity Evaluation Scale Table on basis of modules.

Third, calculate and fill the ratios of the indicators summarized from the Fidelity Table separately for Model 1 and Model 2 against the indicators for the reference model into the Fidelity Evaluation Scale Table. The evaluation results are shown in Table 1 in section 5.3.

5.3 Result of the Fidelity Evaluation Scale

We give the same weight to each dimension and each indicator when calculating the averages of fidelity. From Table 1, it is difficult to judge which of the two models has a total higher fidelity. Notice that a high model fidelity is capable of supporting a high experiment fidelity analysis. For the average fidelity of the first three dimensions, Module 1 of Model 1 has a higher average fidelity than Module 1 of Model 2 (0.976>0.241) in which two brands of case cigarettes and their reorder points are

Table 1. Fidelity evaluation result by Fidelity Evaluation Scale method

Level 0		Model 1		Model 2	
Dimension	Level -1	Level-1-M1	Level-1-M2	Level-1-M1	Level-1-M2
D-1	Number of levels	2/2	2/3	2/2	3/3
	Number of entity categories	4/4	2/(3+5_lev2) *	0/4	(3+5_lev2)/ (3+5_lev2)
	Entity types	(4×2)/(4×2)	(1×2)/(1×2)	(4×1)/(4×2)	(1×1)/(1×2)
	Number of parameter types	17/17	2/6	1/17	5/6
	Average (D1)	**1**	**0.583**	**0.390**	**0.833**
D-2	Number of event levels	2/2	2/3	2/2	3/3
	Number of event	14/14	2/(8+3 group)	0/14	(8+1group)/(8 +3 group) **
	Number of Strategy/Algorithm	3/3	0/1	0/3	(0.8)/1 ***
	Average (D2)	**1**	**0.264**	**0.333**	**0.844**
D-3	Number of events with exact time	13/14	0/(8+3 group)	0/14	(8+1group)/(8 +3 group)
	Average (D3)	**0.929**	**0**	**0**	**0.733**
Average of D-1~D-3		**0.976**	**0.282**	**0.241**	**0.803**
D-4	Visualized	Yes (1)	Yes (1)	Yes (1)	Yes (1)
	3D	Yes (1)	Yes (1)	Yes (1)	Yes (1)
	3D shape	Basic (0.2)	Basic(0.2)	Good (0.8)	Good (0.8)
	Color &texture	No (0)	No (0)	Yes (1)	Yes (1)
	Light/shadow& other render	No (0)	No (0)	Yes (1)	Yes (1)
	Average (D4)	**0.440**	**0.440**	**0.960**	**0.960**
Average of four Dimensions		**0.842**	**0.321**	**0.421**	**0.843**

* When a level is omitted, we add a coefficient of 0.5. For example: 2/(3+5_lev2) = (2/3) ×0.5=0.333.

** (8+1group)/(8+3 group)=(8/8) ×0.6+(1/3)*0.4=0.733.

*** 0.8 is a coefficient that describe the similarity of the fixed route between real free route.

modeled which can support the experiment analysis of the replenishment strategies of two brands of case cigarettes. But Module 2 of Model 2 has a higher average fidelity than that of Model 1 (0.803>0.282) in which the manual replenishment process is modeled as an AGV system instead of a queuing system which can support the experiment analysis of the detailed activities and the efficiency of workers. Besides, Model 2 also has a higher fidelity for the fourth dimension which might be helpful for marketing and the process of validation and verification. However, the higher the model fidelity is, the more resources are needed in system analysis, modeling, validation and other simulation activities. In order to make simulation cost-effective , fidelity should be designed and managed well before and during modeling and simulation. The operable and quantitative evaluation method provided in this paper lays an important foundation for the design and management of simulation fidelity.

6 Conclusions and Future Work

We use the ratio between the simulation model and the intended object reality (existing or imagined reality) to evaluate simulation fidelity. To make the evaluation operable, we first divide fidelity into four dimensions: structural, correlational, temporal and sensorial. Based on the dimension framework, we use a multi-dimension and multi-indicator Fidelity Evaluation Scale method to measure fidelity with an example in discrete event logistics system(DELS). This method provides an operable and quantitative evaluation of fidelity which serves as a significant basis for our future work including the analysis of the relationship between fidelity and simulation effectiveness and costs as well as how to design and manage fidelity before and during the process of modeling and simulation to make simulations of DELS cost-effective.

References

1. Gross, D.C.: Report from the Fidelity Implementation Study Group. In: 99 Spring Simulation Interoperability Workshop: Paper 167 (1999)
2. Hess, R.A., Marchesi, F.: Analytical Assessment of Flight Simulator Fidelity Using Pilot Models. Journal of Guidance, Control, and Dynamics 32, 760–770 (2009)
3. Liu, F., Yang, M.: Research on distributed simulation's fidelity. Computer Simulation 22, 49–52 (2005)
4. Foster, L., Yelmgren, D.: Accuracy in DoD High Level Architecture Federations. 97 Fall Simulation Interoperability Workshop: Paper 064 (1997)
5. Kim, H.: Reference Model Based High Fidelity Simulation Modeling for Manufacturing Systems. School of Industrial and Systems Engineering, Georgia Institute of Technology, Georgia (2004)
6. Pace, D.K.: Dimensions and Attributes of Simulation Fidelity. In: Proceedings of the Fall 1998 Simulation Interoperability Workshop, Orlando, September 14-18 (1998)
7. Roza, J.C.: Simulation Fidelity Theory and Practice: A Unified Approach to Defining, Specifying and Measuring the Realism of Simulations. Delft University of Technology, Netherlands (2005)

Symbiotic Simulation for the Generation and Simulation of Incident Management Strategies

Vinh-An Vu[1], Giho Park[2], and Gary Tan[1]

[1] Department of Computer Science, School of Computing,
National University of Singapore, Computing 1, 13 Computing Drive, Singapore 117417
{vuvinhan01,dcstansh}@nus.edu.sg
[2] Future Urban Mobility Interdisciplinary Research Group, Singapore-MIT Alliance for Research and Technology, 1 CREATE Way, #10-01 CREATE Tower, Singapore 138602
edward@smart.mit.edu

Abstract. Since its introduction in the Workshop on Grand Challenges for Modeling and Simulation in Dagstuhl 2002, Symbiotic Simulation has proven its versatility in many diverse areas ranging from manufacturing to pedestrian evacuation. This paper presents a new application of Symbiotic Simulation in the simulation and generation of traffic incident management strategies. The framework for the generation and evaluation of incident management strategies is used in which Symbiotic Simulation is the core technique of the Strategy Generation Module. Preliminary experimental result shows the effectiveness of Symbiotic Simulation in helping to simulate and select the best strategy to improve the traffic condition. Challenges are also highlighted with potential research direction.

Keywords: Incident Management Strategies, Symbiotic Simulation, Closed-Loop Framework.

1 Introduction

The importance of incident management strategies in modern transportation networks is well established especially when the networks have evolved and become increasingly complicated [1]. The advance of information and sensing technology has seen the emergence of Advance Traveler Information Systems (ATIS) which open a new opportunity to use collected traffic data containing speed, volume, density, etc. to deduce the important information about current and future traffic condition. This information is then utilized to develop strategic actions in order to enhance drivers' preparedness and improve the network safety and reliability. Existing incident management strategies generation studies are usually divided into two broad categories: Optimization Problem and Artificial Intelligence technique.

In the *Optimization Problem*, the general methodology is to deduce the optimal traffic assignment in order to optimize some pre-specified objective functions subject to the defined constraints. The objective functions here could be to focus on system optimum (SO), user equilibrium (UE) or some variants of them. The Optimization

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 397–402, 2013.

Problem could be solved by analytical approach [2-4] or simulation-based approach [5-7]. The Optimization Problem could also be iterative or feedback in nature. Iterative approach makes use of real time measurements, disturbances and predictions to run a traffic flow model repeatedly over a future time horizon aiming to get some control objectives. In [8], iterative control was used with model predictive control approach to optimize the control of ramp metering and variable speed limit. Feedback control, on the other hand, gets measurements from the system and determines control actions based on the current state of the system in such a way that the performance of the system is improved. Study in [9] employed feedback control to generate the desired splitting rate at the bifurcation nodes subject to the predefined criteria.

In the *Artificial Intelligence* category, the controller tries to imitate human intelligence and thinking while solving the traffic problem by computer programs. Studies in [10-13] widely used different Artificial Intelligence techniques in decision support problem.

To the best of our knowledge, whether the Optimization Problem or Artificial Intelligence technique is used, the capability to simulate incident management strategies before implementation is limited. Fortunately, Symbiotic Simulation, which was introduced in the Workshop on Grand Challenges for Modeling and Simulation in Dagstuhl 2002 [14], could be a solution. A symbiotic simulation system is defined as one that interacts with the physical system in a mutually beneficial way. It is highly adaptive since simulation not only performs What-If experiments that are used to control the physical system, but also accepts and responds to data from the physical system. Symbiotic Simulation has been applied in many fields such as manufacturing [15], pedestrian evacuation [16], unmanned aerial vehicles (UAVs) [17], etc. In [18], Aydt et al. attempted to use Symbiotic Simulation in a smart routing application to determine the optimal route for vehicles. However, the study is still in progress since the usage of nanoscopic simulator in Symbiotic Simulation may be a challenge for its application in real-time decision support. In this paper, Symbiotic Simulation incorporating predictive information from a mesoscopic traffic simulator will be utilized as a core technique in generating and simulating a wide variety of incident management strategies that could work for different types of traffic control and advisory equipment.

The paper is organized as follows. Section 2 presents the implementation of Symbiotic Simulation in Incident Management Strategies Generation Framework. Section 3 reports the preliminary experimental results of the implemented technique. Finally, section 4 summarizes and concludes the paper with challenges and future work.

2 Implementing Symbiotic Simulation in Incident Management Strategies Generation Framework

In order to facilitate the generation and evaluation of incident management strategies, a closed-loop framework proposed by Vu et al. [19] is used. This closed framework (Fig. 1) consists of three main components: the microscopic simulator MITSIMLab acting as a proxy to real world, the mesoscopic simulator DynaMIT estimating and

predicting traffic conditions in MITSIMLab based on surveillance data, and the Strategy Generation Module generating and simulating incident management strategies based on pre-defined objective functions. These components exchange information between each other and operate under the same incident scenarios.

The Strategy Simulation Module is implemented in the framework with four sub-components: Strategies Generator, What-If simulations, Performance Analysis, and Learning. The Strategies Generator uses predictive information from DynaMIT to generate a list of candidate strategies. These candidate strategies are then simulated by What-If simulations. The results of What-If simulations are analyzed to find best strategy subject to predefined objective functions. The Learning sub-component is used to speed up the strategy generation process. It performs features matching to find the previous scenarios with similar traffic condition to the current one. If there is a good match, the previously proposed strategy will directly be applied without going through the whole process.

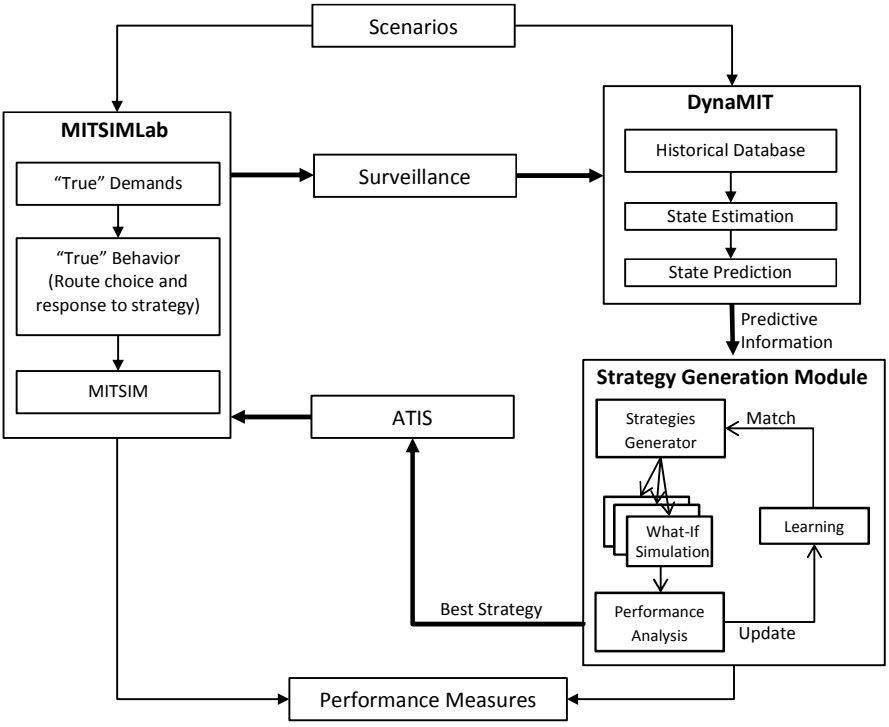


Fig. 1. Symbiotic Simulation in Closed-Loop Framework

3 Preliminary Result

This work is still in progress. Some early result was highlighted in the case study in a synthetic network in [19]. This case study focuses on proposing route guidance

strategies to divert the critical Origin-Destination (OD) flow which contributes the most to the count of vehicles at incident location. This critical OD flow is found by analyzing the predicted traffic condition from DynaMIT. A list of route guidance strategies is generated with one strategy corresponding to one alternative path of the critical OD flow. Symbiotic Simulation is then used to find the best path for diversion among all the alternative paths. The compliance of the drivers to proposed route guidance is assumed to be 100%.

The result shows that in normal condition without incident, the average travel time for all OD pairs is 354.6 seconds. When an incident happens and no route guidance is provided, the travel time drastically increases by 85.7% to 658.3 seconds. With route guidance, the average travel time increases by only 53.8% to 545.5 seconds. This implies that 37.1% of the increase in average travel time is eliminated by the route guidance strategy.

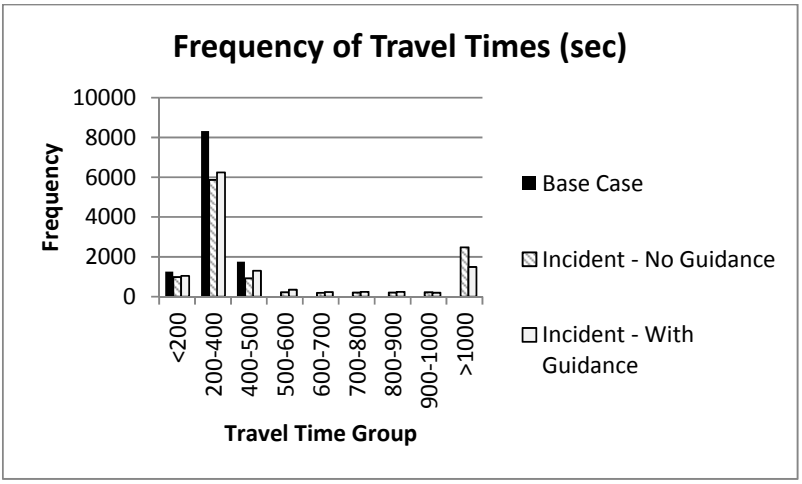


Fig. 2. Frequency of Travel Times

Fig. 2 plots the frequency distribution of travel times for three scenarios: base case without incident, incident without guidance and incident with guidance. In the base case, all the drivers belong to the first four groups with low travel times. When an incident happens and no route guidance is provided, a large number of drivers shift to the last group with very high travel time. When route guidance is provided, the number of drivers with very high travel time significantly decreases. This is a meaningful effect since it improves the satisfaction of the travelers in the network.

The link travel time is also plotted in Fig. 3. The bottom line shows the link travel time in the base case; the dotted line shows high link travel time when an incident happens without route guidance and the dashed line shows the improved link travel time when the incident happens with route guidance. The considerable improvement in link travel time is the result of route guidance strategies trying to divert drivers from congested link to other links with higher residual capacity.

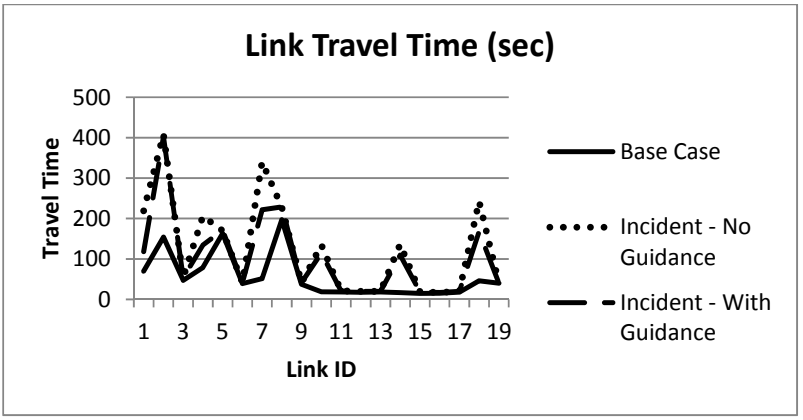


Fig. 3. Link Travel Time for Different Scenarios

4 Conclusions and Future Work

This paper has discussed a potential application of Symbiotic Simulation in the simulation and generation of Incident Management Strategies. The proposed idea was implemented in the closed-loop framework which facilitates the generation and evaluation of generated strategies. Preliminary results of case study in a synthetic network showed positive effect of Symbiotic Simulation in generating and simulating incident management strategies aiming to improve traffic conditions.

Through this preliminary work, some challenges in using Symbiotic Simulation in generating and simulating incident management strategies have been identified. First, a good model of drivers' response to information should be studied for more realistic simulation of strategies. Second, a mechanism to speed up What-If simulations should be investigated since the running time could become bottleneck when the network is large and complicated. Hybrid simulation, parallel and distributed computing and learning are potential solutions. And third, algorithms for more intricate strategies combining different parameters (travel time, speed, density, etc.) for different traffic control and advisory (Variable Message Sign, traffic signal, road pricing, etc.) should be developed.

References

1. Mahmassani, H.: Dynamic Network Traffic Assignment and Simulation Methodology for Advanced System Management Applications. Networks and Spatial Economics 1(3-4), 267–292 (2001)
2. Merchant, D.K., Nemhauster, G.L.: Optimal Conditions for a Dynamic Traffic Assignment Model. Transportation Science 12(3), 202–207 (1978)
3. Ran, B., Boyce, D.E., Leblanc, L.J.: A New Class of Instantaneous Dynamic User-Optimal Traffic Assignment Models. Operations Research 41(1), 192–202 (1993)

4. Chen, H.-K., Hsueh, C.F.: A Model and an Algorithm for the Dynamic User-Optimal Route Choice Problem. Transportation Research 32B(3), 219–234 (1998)
5. Mahmassani, H., Peeta, S.: System Optimal Dynamic Assignment for Electronic Route Guidance in a Congested Traffic Network. In: Proceedings of the 2nd International Capri Seminar on Urban Traffic Networks, Capri, Italy (1992)
6. Ben-Akiva, M., Bierlaire, M., Bottom, J., Koutsopoulos, H.N., Mishalani, R.: Development of a Route Guidance Generation System for Real-time Application. In: Preprints of the 8th IFAC/IFIP/IFORS Symposium on Transportation Systems, pp. 433–439 (1997)
7. Antoniou, C., Koutsopoulos, H.N., Ben-Akiva, M., Chauhan, A.S.: Evaluation of Diversion Strategies Using Dynamic Traffic Assignment. Transportation Planning and Technology 34(3), 199–216 (2011)
8. Hegyi, A., De Schutter, B., Hellendoorn, J.: Model Predictive Control for Optimal Coordination of Ramp Metering and Variable Speed Limits. Transportation Research Part C 13(3), 185–209 (2005)
9. Messmer, A., Papageorgiou, M.: Automatic Control Methods Applied to Freeway Network Traffic. Automatica 30(4), 691–702 (1994)
10. Ritchie, S.G.: A Knowledge-Based Decision Support Architecture for Advanced Traffic Management. Transportation Research Part A 24(1), 27–37 (1990)
11. Lin, I., Chout, S.: Developing Adaptive Driving Route Guidance Systems Based on Fuzzy Neural Network. In: IET International Conference on Intelligent Environments, Seattle, Wa, USA, pp. 1–8 (July 2008)
12. Park, K., Bell, M.G.H., Kaparias, I., Bogenberger, K.: Adaptive Route Choice Model for Intelligent Route Guidance Using a Rule-Based Approach. In: Transportation Research Record: Journal of the Transportation Research Board, No. 2000, pp. 88–97. Transportation Research Board of the National Academies, Washington, D.C (2007)
13. Adler, J.L., Satapathy, G., Manikonda, V., Bowles, B., Blue, V.J.: A Multi-Agent Approach to Cooperative Traffic Management and Route Guidance. Transportation Research 39B(4), 297–318 (2005)
14. Fujimoto, R., Lunceford, W.H., Page, E.H., Uhrmacher, A.: Grand Challenges for Modeling and Simulation: Dagstuhl Report. Technical Report 350, Seminar No. 02351, Schloss Dagstuhl (2002)
15. Low, Y.H., Lye, K.W., Lendermann, P., Turner, S.J., Leo, S., Chin, R.: An Agent-Based Approach for Managing Symbiotic Simulation of Semiconductor Assembly and Test Operations. In: Proceedings of the 2005 International Conference on Autonomous Agent and Multiagent Systems (AAMAS), Utrecht, The Netherlands, July 25-29 (2005)
16. Hetu, S., Tan, G.: Potential Benefits of Symbiotic Simulation to Pedestrian Evacuation. In: 2009 Asia Simulation Conference. Japan Society for Simulation Technology (2009)
17. Kamrani, F., Ayani, R.: Using Online Simulation for Adaptive Path Planning of UAVs. In: Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-time Applications, Chania, Greece, pp. 167–174 (October 2007)
18. Aydt, H., Lees, M., Knoll, A.: Symbiotic Simulation for Future Electro-Mobility Transportation Systems. In: Proceedings of the 2012 Winter Simulation Conference, pp. 1–12 (2012)
19. Vu, V.-A., Park, G., Tan, G., Ben-Akiva, M.: A Framework for the Generation and Evaluation of Incident Management Strategies. Submitted to the 93rd Annual Meeting of the Transportation Research Board, Washington, D.C (2014)

Simulation Job Scheduling on Clusters with Heterogeneous Scheduling Systems

Chi Xing[1], Xudong Chai[2], Qi Wang[2], Yang Chen[1], and Li Tan[2]

[1] College of Automation Science and Electrical Engineering,
Beihang University, Beijing 100083
xingchi09@163.com
[2] Beijing Simulation Center, Beijing 100854, China

Abstract. In complex product design enterprise, different departments establish private simulation centers (including computing clusters, application software, job scheduling system etc.), and the resource utilization of each simulation cluster is uneven. So the whole enterprise simulation resources cannot be fully effective used, resulting in low efficiency of simulation and long time waiting. This paper proposed a method of simulation job cooperative scheduling on the clusters with heterogeneous systems, in order to reduce the waiting time after submitting jobs, so it can achieve efficient use of simulation resources in whole enterprise. The instance of one actual simulation problem in this paper shown the job execution time is reduced, and the scheduling method is highly efficient and reliable.

Keywords: cluster with heterogeneous systems, simulation resources, cooperative scheduling, job transporter.

1 Introduction

In complex product design and manufacturing process, a single disciplinary simulation has an important meaning. Therefore, in the large-scale manufacturing enterprise, different departments always establish their own simulation center (including computing clusters, simulation software, cluster scheduling system etc.) to ensure the capability of afford the simulation tasks in the busiest period of the year. So it will cause significant waste of simulation resources in other the time. In the enterprise actual survey found that the "busy" and "free" time of different departments did not appear in the same period [1]. In a period of time, if "free" resources in the heterogeneous environments can be easily assigned to the "busy" task to use, so the problem of uneven distribution of simulation resources in the enterprise can be solved. It can greatly improve the efficiency on the implementation of the simulation task.

The so-called simulation resource clusters with heterogeneous scheduling system is that different clusters use different scheduling system. Currently, the mainstream cluster scheduling system includes LSF, PBS, Condor, SGE [2,3] and so on, they can be well compatible heterogeneous computers and heterogeneous operating systems in a single

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 403–408, 2013.

cluster, and can be complete the load balance on simulation jobs submission. But they are for single-cluster scheduling and do not have the ability of cooperative scheduling on different clusters with heterogeneous scheduling systems, such as the cluster in PBS cannot submit jobs to the cluster in LSF. Heterogeneous scheduling system is a bottleneck of multi-cluster to share their simulation resources.

Aiming at these problems, our team studied a middleware of simulation job cooperative scheduling on the clusters with heterogeneous systems called JobTransporter, it can move simulation jobs freely between clusters with heterogeneous scheduling system and submit the simulation jobs from the "busy" simulation cluster to "free" simulation cluster although they have different scheduling systems.

2 Realization of Simulation Jobs Cooperative Scheduling

2.1 Single Cluster Dispersion Use Mode

Traditional using model of single simulation resource cluster is that different department has their own simulation cluster, they are not connected and used independently. Such a situation would exist: the simulation cluster of department A is operating at full capacity, and it has large number of jobs in the queue are waiting for calculating, at the same time simulation cluster of department A and B rarely been occupied and even some completely free, as shown in Figure 1. This will result in lower efficiency of the simulation operation execution in department A, meanwhile, the large number of simulation resources in department B and C are wasted.

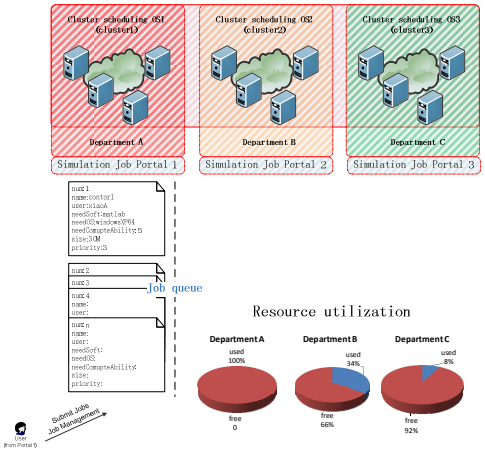


Fig. 1. Single Cluster Dispersion Use Mode

2.2 Cooperative Scheduling in Clusters with Heterogeneous Scheduling System

The aim of achieving simulation jobs scheduling cross-cluster is select the most appropriate remote simulation resources and submit the queued jobs when the local cluster is fully occupied. And it also can monitor the status of jobs in remote cluster, the result will be retrieved after job has been done.

In this paper, through the JobTransporter middleware, each cluster can monitor the local and remote simulation clusters and the status of simulation jobs running on the clusters, and it scheduler the simulation jobs on clusters which has heterogeneous systems. The JobTransporter middleware has three modules, as shown in Figure 2.

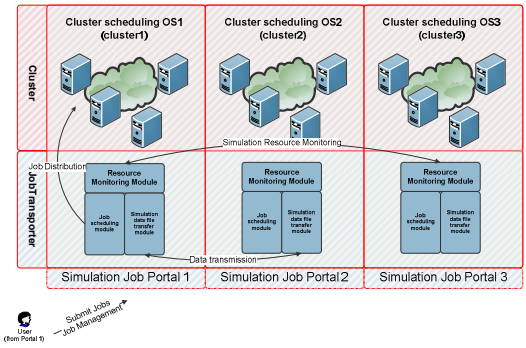


Fig. 2. Cooperative Scheduling in Clusters with Heterogeneous Scheduling System

Job scheduling on clusters with heterogeneous scheduling systems shown in figure 1 consists of three layers, including

1. Simulation Resource Monitoring Module. It is used to monitor local and remote cluster system resource usage and status of simulation jobs running on clusters.
2. Simulation job scheduling module: It is used to submit the simulation jobs on local and remote clusters.
3. Simulation data and file transfer module: It is used to transfer the job data and files between clusters.

Users can submit their simulation jobs through their own department's portal. Firstly, simulation resource monitoring module collects the information from local and remote clusters, and if the local cluster has free simulation resource, the jobTransporter will notice the scheduling system to submit the jobs on the local cluster because of the local owning faster network and it can save uploading files time. If local resources are occupied and there are a large number of jobs in the queue waiting for calculating, simulation resource monitoring module will select the idle simulation resources from remote cluster by calculating, then simulation data and file transfer module transports the simulation files and data from local to remote, and remote jobTransporter receive the file and submit the job on the remote cluster through simulation job scheduling module. After completion of simulation, local cluster takes back all result files rely on communication of simulation job scheduling module between local and remote cluster and user can download the result files.

2.3 Queued Job Selection Policy

How to select which job in the queue (the waiting job list) should submit to a remote cluster simulation? The current strategy is first come first calculate in one cluster, if it submit the first job in the queue to remote, it will occur a situation that the job in local cluster has been done and it has been released resources, the first job in the queue can be submit, while it has chosen the job to send to remote and it is being upload. When the first job finished, the results must be retrieved, so the time cost will be increased.

At this stage, based on previous job Submission history, the paper proposes a static scheduling strategy: Firstly, the first ten jobs in the queue are not treated, they are still in the queue and waiting the local resource. Secondly, collecting the job information from eleventh to twentieth in job queue, choosing a relatively small job file send to a remote cluster for calculate. It will not only reduce waiting time, but also ensure the less transfer time.

3 Simulation Resource Modeling [5]

Simulation resource monitoring module gets the information and status of local and remote clusters, and it will scheduler them as a unified resources to select optimal simulation resources. The computation resource can be described as a unified model [5].

$$Resource =< ND, CP, CPR, CM > \tag{1}$$

Where, $ND = \{Node_1, Node_2, \ldots, Node_k\}$ denotes k physical nodes,

$CP = \{NR_{compute}(i) | i \in ND\}$, $NR_{compute}$ refers to computation performance of the physical node, mainly including CPU speed, core number and memory. It influences the job computing time.

$CM = \{NR_{net}(i,j) | i,j \in ND\}$, NR_{net} is defined as the communication performance between local and remote clusters, including the job pending time and the transmission time.

Under the constraint that simulation system performance meets the minimal demand, the model aims at finding the scheduling solution for co-simulation task with least job execution time (formula 2) based on jobTransporter to improve the efficiency of computation resource, and save energy.

$$\min \mathrm{T} = \sum(\mathrm{a/CP} + \mathrm{b/CM}) \tag{2}$$

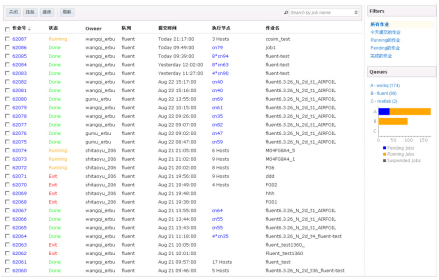
where a,b represent weight of CP and CM.

At present, a, b parameters are fixed values.

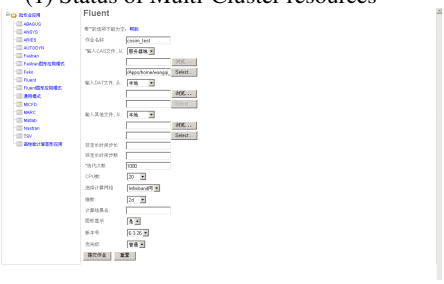
4 Application Example

In this paper, we built and deploy two heterogeneous simulation resources experiment clusters, and the scheduling systems are PBS pro and LSF. Each cluster can submit simulation job, monitor the status of job running, and view the information of two simulation clusters shown in Figure 3.

(1) Status of Multi-Cluster resources



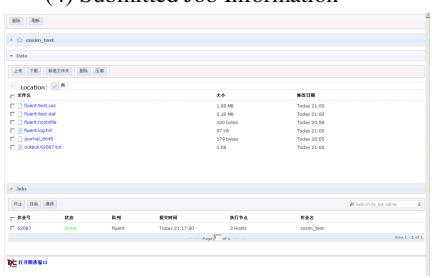
(2) Information of Jobs in the Multi-Cluster



(3) Submit Simulation Job

(4) Submitted Job Information

(5) Submitted Job running status



(6) Simulation Results

Fig. 3. Application Results

1. Status of Multi-Cluster resources. The simulation resources named CN is cluster with LSF scheduling system, and others named NODE is cluster with PBS pro scheduling system. We can see the status of each compute nodes.
2. Information of Jobs in the Multi-Cluster. The operation of all jobs in multi-cluster.
3. Submit Simulation Job. Submit the simulation job through the portal.
4. Submitted Job Information. We can see the information of submitted job, such as submitted time, which compute nodes to run, finished time, resource requirements and so on.
5. Submitted Job running status. We can see that the job is running intermediate results.
6. Simulation Results. It shows that final results and it can be download.

5 Conclusion and Future Works

This paper presents the method of sharing and efficient use simulation resources in multi-cluster with heterogeneous scheduling systems. Firstly, it model for simulation hardware resources, in order to monitor the status of multi-cluster and select the optimal simulation resources by calculating. Secondly, we study and implement the simulation task scheduling middleware JobTransporter to connect multi-cluster, submit the simulation job to remote cluster when the local cluster is fully occupied, and monitor the status of submitted jobs. This method of simulation job cooperative scheduling on the clusters with heterogeneous systems can achieve efficient use of simulation resources in whole enterprise. When the simulation resources are occupied in the local case, the job is submitted to the remote cluster to avoid continuing waiting on the local resources. Finally, the paper also provides job submission application examples and operating results.

The further works are focused on:

1. According to job file size and network environment, How to dynamically adjust the a, b parameters in formula (2) in order to choose more optimal remote resources.
2. Dynamically select the most suitable job in job queue for submission.

References

1. Li, B.H., Chai, X., Zhang, L., Hou, B., Lin, T.Y., Yang, C., Xiao, Y., Xing, C., Zhang, Z., Zhang, Y., Li, T.: New Advances of the Research on Cloud Simulation. In: Kim, J.-H., Lee, K., Tanaka, S., Park, S.-H. (eds.) AsiaSim2011. PICT, vol. 4, pp. 144–163. Springer, Heidelberg (2012)
2. Burks, B., Chansup, D.: A Comparison of Job Management Systems in Supporting Sun HPC Cluster Tools. SUPerG Fall (2000)
3. Hassaine, O.: HPC Cluster Tools Best Practices. SUPerG Meetings (April 2000)
4. Hang, Q., Guyu, H., Wang, L.: Concept, modeling and application of ontology. Journal of PLA University of Science and Technology (Natural Science edition)
5. Yang, C., Chai, X., Zhang, F.: Research on co-simulation task scheduling based on virtualization technology under cloud simulation. In: Xiao, T., Zhang, L., Fei, M. (eds.) AsiaSim 2012, Part II. CCIS, vol. 324, pp. 421–430. Springer, Heidelberg (2012)

Research on a Knowledge Model of Aircraft Design Based on Unified Description Framework

Gong Xiayi[1,2], Li Bo Hu[1,2], Chai Xudong[2], and Gu Mu[2]

[1] College of Automation Science and Electrical Engineering, Beihang University, Beijing 100083
[2] Beijing Simulation Center, Beijing 100854, China
gongwhite56166@sina.com

Abstract. In order to solve the problem that the heterogeneous knowledge in aircraft design is difficult to share, reuse and support to reason each other, a ontology model of knowledge, based on the unified description framework, was proposed. Firstly, the formats of representation for all kinds of knowledge in aircraft design knowledge were studied, and then a unified description framework for this knowledge was proposed. Secondly, the ontology model of knowledge was studied. In this model, basic cell for modeling was defined, which could be used to modeling directly. At last, an example, modeling of pneumatic knowledge in aircraft design verified the validity of the modeling method in this paper.

Keywords: aircraft design knowledge, knowledge modeling, knowledge representation, unified description framework.

1 Introduction

In aircraft design, traditional methods have failed to meet their design digital, intelligent requirements. Design cycle is long, and the lacks experience in aircraft design science accumulated knowledge and experience makes the design inheritance, reuse, and aircraft innovative and intelligent design difficult. The knowledge-based design methodology can promote effective solutions to these problems.

2 Knowledge of Aircraft Design

It is generally believed that design knowledge that can be used for a variety of product design and decision-making of high-value collection of information, this paper defines aircraft design knowledge (Design Knowledge of Aircraft, DKA) that can describe the characteristics of the object field of aircraft design, the intrinsic relationship and the fact that the implementation process, in principle, a collection of models and experiences. Categories of the knowledge are as follows:

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 409–414, 2013.

1. Design Basics (Basic Knowledge, BK): including aircraft design to rely on the basic theory, aircraft design manuals and basic knowledge of aircraft, such as aircraft layout description, aircraft geometry, materials, and also includes that part of the design evaluation criteria, such as knowledge;
2. Design expertise (Professional Knowledge, PK): it refers to the aircraft used to guide the design of a specialized fields of knowledge;
3. Design Collaborative Knowledge (Collaboration Knowledge, CK): aircraft design is a multidisciplinary collaborative design, interactive design process:
   (a) The design constraints of knowledge (Restriction Knowledge, RK): it includes the coupling parameters, sub-function between qualitative description of such effects;
   (b) The design process knowledge (Process Knowledge, PK): it includes vehicle design process, design steps, design techniques and design methods route selection, resource allocation, the design task decomposition.

Aircraft design knowledge content shown in Figure 1.

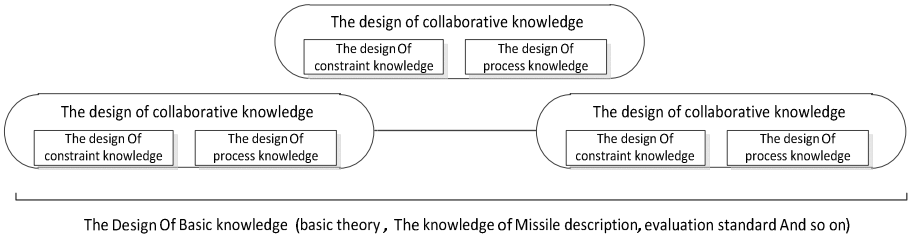


Fig. 1. The content of aircraft design knowledge

3 Knowledge Description of Aircraft Design

For aircraft design knowledge, according to the characteristics of various types of knowledge representation method, this paper selects the appropriate representation of the aircraft design knowledge representation.

3.1 Basic Design Knowledge Representation

Basic knowledge of aircraft design can be used to assist the search for knowledge, matching, fusion and reasoning.First-order predicate logic are expressed as: <a first-order predicate logic> :: = <quantifier> <predicate formula> <connector> <predicate formula>. For the aircraft aerodynamic design, the accuracy for some features defined Precision (x, v) represents the accuracy indicator x v. So there is:

$$\text{Precision}(X,v1) \vee \text{Precision}(Y,v1) \vee \text{Precision}(Mx,v1)\ldots$$

3.2 Professional Design Knowledge Representation

In this paper, its production rule representation in the form of <Rules>:: = if <antecedent>, then <after pieces >. For example, in the aerodynamic shape design, knowledge representation in the form of its rule, a rule knowledge is:

If selected canard aircraft then the aircraft rudder should be placed in the front wing.

Aircraft design in a framework of knowledge is as follows:

frame1：Calculation of the aerodynamic force：

The lift calculation： computational formula: $Y = \frac{1}{2} c_y \rho V^2 S$

The lift coefficient calculation： $c_y = c_{yw} + c_{yB} + c_{ywB} + c_{y\delta}$

Calculation of elastic resistance： computational formula: $X = \frac{1}{2} c_x \rho V^2 S$

Drag coefficient calculation： $c_x = c_{x0} + c_{xi}$

$= k(c_{xfw} + c_{xww} + c_{xfB} + c_{xsB} + c_{xwB}) + k_1(c_{xiw} + c_{xiB})$

Fig. 2. The collaborative knowledge of pneumatic sub-system

3.3 Collaborative Design Knowledge Representation

Petri nets is used in this method of collaborative knowledge that aircraft, specifically, the aerodynamic design process design collaboration knowledge of such a class, which can be expressed as in Figure 3.

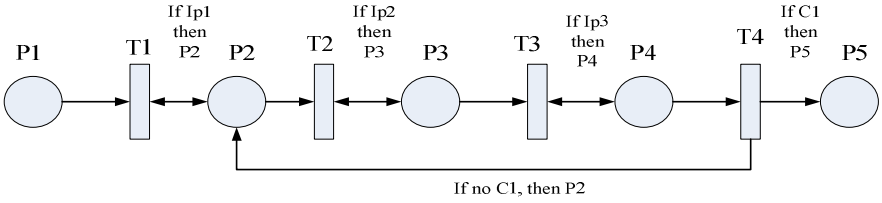


Fig. 3. The collaborative knowledge of pneumatic sub-system

Where P1, P2, P3, P4, P5, respectively, corresponding to the process design phase of each module, such as P2 represents the aerodynamic shape design phase.

4 Knowledge Model of Ontological Aircraft Design Based on Unified Description Framework

4.1 Aircraft Design Knowledge Unified Description Framework

How to get these heterogeneous knowledge unified organization and consistency of expression in order to effectively support knowledge sharing, reuse, knowledge

management and other applications, mechanism of unified description of such knowledge should be studied.

Define a unified description framework shown in Figure 4.

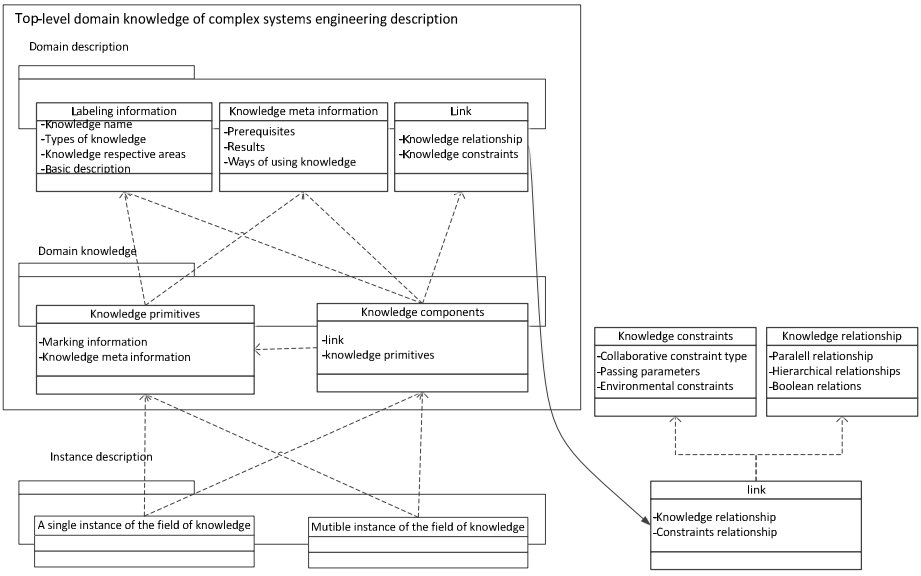


Fig. 4. The unified description framework for aircraft design knowledge

In the framework of this description, three knowledge elements are defined: identification information, knowledge meta information and links used to create complex systems domain knowledge meta-knowledge. Then define the top-level of knowledge in the knowledge base element (Knowledge Element) and knowledge components (Knowledge Component).

In this way, the system can solve complex problems complete knowledge of the knowledge base through knowledge elements and combinations of elements constructed from the layers shown in Figure 5.

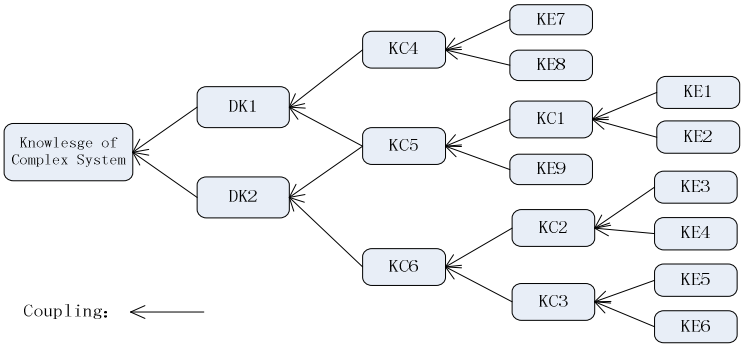


Fig. 5. The component of complex system knowledge

Meta-knowledge layer defines the three aircraft design knowledge knowledge elements, that is:

MKnowledge=::=<IDInfo, KmInfo, Coupling>,

IDInfo :: = <Name, Type, Domain, Description>, means that the identification information of the knowledge base element, is used to describe the meaning of the knowledge and types of primitives;KmInfo: = <ConditionInfo, ResultInfo UseManner>, represent knowledge meta-information, knowledge primitives are used to describe the contents of the main components;

Coupling :: = <Relation, Restriction>,used to describe the knowledge and knowledge of primitive relations and interactions between components constraint information.

Knowledge describes the minimum size of the top-level and sub-granular knowledge: Knowledge primitives (Knowledge Element) and knowledge components (Knowledge Component).

KE::= <IDInfo, KmInfo>, knowledge representation primitives various heterogeneous body of knowledge is the smallest constituent units, which corresponds to the practical application of a formal knowledge;

Besides knowledge components and knowledge components, knowledge primitives can also be combined knowledge components become more complex to solve more complex problems. Knowledge components therefore defined as: KC :: = <KE/KC, Coupling>.

4.2 Aircraft Design Ontology Modeling

Aircraft design ontology model describes the concept of aircraft design, relationships, and other knowledge associated together to achieve the knowledge to solve the problem of aircraft design summary and normalized. Aircraft design knowledge model is shown in Figure 6.

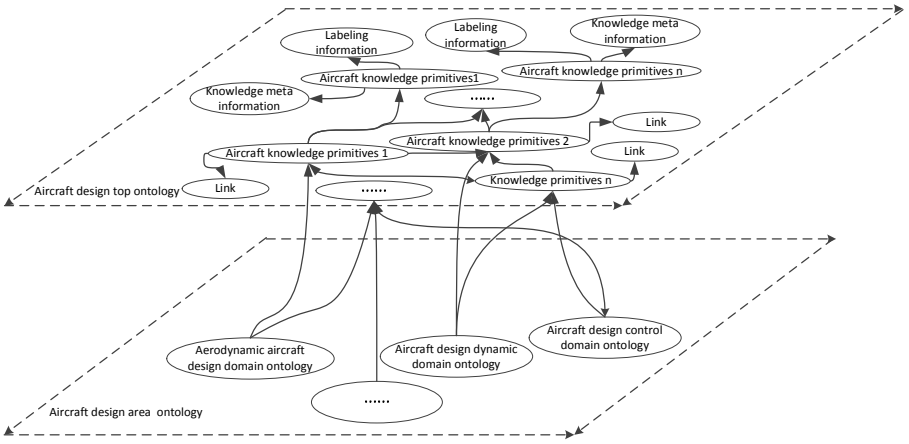


Fig. 6. The knowledge model of aircraft design

Aircraft design knowledge model has two-tier structure, namely top-level ontology layer aircraft design and aircraft design domain ontology layer.

5 Application Examples

Firstly, the use of a template is to get the original knowledge acquisition. Then create aerodynamic design disciplines ontology. The knowledge gained will be created in the input 3.3 ontology. For an aerodynamic design basics (first-order predicate logic representation): PInPara (Ma, vm) ∨ PInPara (H, vh) ∨ PInPara (α, va) ..., the Unified Modeling Ontology is omitted due to space.

6 Conclusion and Outlook

This paper describes a knowledge-oriented unified description of aircraft design framework and the ontological model based on this unified description framework, which can have all kinds of heterogeneous knowledge unified organization and consistency of expression, and effective support aircraft design knowledge sharing, reuse, knowledge application and knowledge reasoning. Finally, through an application example it verifies the validity of this research.

References

1. Liu, X., Liang, B., Liu, L., et al.: The Theory, Method and Technology for modeling of complex system. Science Press, Beijing (2008)
2. Shi, Z.: Knowledge Engineering. Tsinghua University Press, Beijing (1988)
3. John, K.C., Kinston: Designing Knowledge Based Systems: The Common KADS Design Model. Knowledge-based Systems 11(5), 311–319 (1998)
4. Nicolas, S., Frits, T.A.: Product Modeling Approach to Building Knowledge Integration. Automation in Construction 8(3), 269–275 (1999)
5. Xin, S.: Research on Knowledge Modeling and Model Reuse Method for Complex Production Design. Doctor Thesis of Tianjin University (2005)
6. Research on Modeling and Managing of Complex Product Development Process. Doctor Thesis of Hefei University of Technology (2011)
7. Girod, M., et al.: Decision making in conceptual engineering design: an empirical investigation. Journal of Engineering Manufacture 217(9), 1215–1228 (2003)
8. Azize, S.S., Chassapis, C.: A decision- making framework model for design and manufacturing of mechanical transmission system development. Engineering with Computers 21(2), 164–176 (2005)
9. Chen, W., Chen, S.: Knowledge engineering and knowledge management. Tsinghua University Press (2010)
10. Mineau, G.W., Rokia, M.: Conceptual Modeling for Data and Knowledge Management. Data & Knowledge Engineering 33(2), 137–168 (2000)

A 3D Visualization Framework for Real-time Distribution and Situation Forecast of Atmospheric Chemical Pollution

Haibo Wang[1], Jingeng Mai[1], Yi Song[2], and Chaoshi Wang[1], Lin Zhang[1], Fei Tao[1], and Qining Wang[2]

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[2] Intelligent Control Laboratory, College of Engineering, Peking University, Beijing 100871, China

Abstract. The visualization system of pollutant distribution contributes to scientific decision-making in the emergency pollution affairs. In this paper, we propose a framework which supports 3D visualization for real-time distribution and situation forecast of atmospheric chemical pollution. The core of our framework is a distributed system infrastructure, which is designed for massive data storage and parallel computing. The stored data includes terrain elevation, vector maps, satellite maps, meteorological data and concentration data from gas sensors. High-performance computing generates gridded data for visualization. Web-based 3D visual applications with B/S structure support cross-platform terminal access.

Keywords: visualization, distributed system, situation forecast, web-based, chemical pollution.

1 Introduction

The atmospheric chemical pollution has a significant hazard to human health. The system for real-time monitoring, forecast and visualization of the pollution level in urban areas is an important tool for evaluating the atmospheric environment and dealing with the emergency pollution accidents.

James J.Q. Yu *et al.* developed a method of collecting data from gas sensors deployed on the public traffic system[1]. Jinqiu Xiao *et al.* also attempted to monitor the atmospheric chemical pollution via the telecommunication network [2]. In addition, many other methods have been used to realize real-time monitoring [3].

Besides real-time data, the system needs the support of more information, including terrain elevation, satellite maps and vector maps, to estimate pollutant distribution around 3D space and forecast the diffusion situation [4].

Displaying the pollution level in 2D view is a common environment evaluation method at present [5]. Many people have attempted 3D visualization with terrain data [6-8].However, the combination of real-time pollution distribution and situation forecast with 3D terrain remains a great challenge.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 415–420, 2013.

Łukasz Kamiński *et al.* proposed a web-based GIS visualization architecture and indicated that the system integrated with sensors providing current on-line data containing measurement results was practical[9].

We propose a 3D visualization system for atmospheric chemical pollutant real-time distribution and situation forecast. The gas sensors are deployed on a mobile blimp with a large monitoring range for real-time measurement. The distributed system infrastructure is used to realize massive data storage and parallel computing. The pollutant distribution and situation forecast are estimated according to the diffusion model and real-time measurement data, and displayed on the cross-platform terminals to realize web-based 3D visualization.

The rest of the paper is organized as follows. The architecture of the system is described in Section 2, and the infrastructure of data processing is shown in Section 3. In Section 4 and 5, the diffusion model and structure of 3D visualization platform are discussed. We conclude in Section 6.

2 Architecture

As shown in Fig.1, the system consists of three parts: flight platform for sensors deployment, ground station for data storage and processing, and access terminals for 2D and 3D view. And as shown in Fig.2, the data access layer provides the storage, access and management interfaces for structured data (e.g. GPS, velocity, temperature, humidity, atmospheric pressure, wind speed, wind direction and chemical concentration) and unstructured data (e.g. video and images); in the business logic layer, a variety of data modeling methods are taken for data gridding Based on the pollution data of discrete points from gas sensors, and various visual structured data are generated from visualization mapping of gridded data, such as points, lines and planes; in the user interface, the contour lines and planes are drawn and rendered according to visual structured data to display atmospheric chemical pollution distribution, and the cross-platform visualization applications are realized based on WebGL technology.

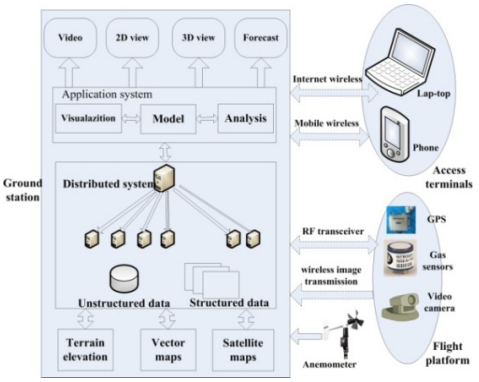


Fig. 1. Functional Structure

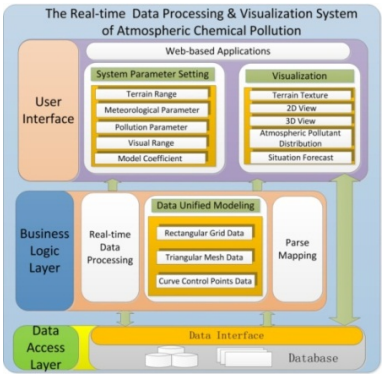


Fig. 2. Hierarchical Structure

3 Data Processing Infrastructure

As shown in Fig.3, the data processing infrastructure consist of three parts:1) Data source The terrain data in the system includes terrain elevation, satellite maps and vector maps. The terrain elevation data is about 30GB from SRTM (Shuttle Radar Topography Mission). The satellite maps data is about 3TB in total. The vector maps data in the form of MapInfo is about 3GB, including the coordinate information of landmarks, such as buildings, roads, etc. The real-time monitoring data is divided into two types according to the format. One is video image in the form of streaming media; the other is structured data, including GPS, velocity, temperature, humidity, atmospheric pressure, wind speed, wind direction and chemical concentration, etc. 2) Data storage The distributed system based on Hadoop includes one NameNode and several DataNodes. The data of terrain elevation and satellite maps, which is not frequently read and written, is stored in the DataNodes of Hadoop system. The vector maps data is stored in the NameNode. Besides, the structured data, video data and visualization data from the model computing within a given period (e.g. one month) are all stored in the NameNode. The outdated data will be immigrated to the DataNodes of Hadoop system as historical data. 3) Data interface The data of terrain elevation and satellite maps are provided in the form of web map service, while the vector maps data is provided by the MapX. The real-time database supports ODBC interface, and the real-time video data supports video streaming playback interface. The historical data stored in Hadoop system can be accessed via historical data service.

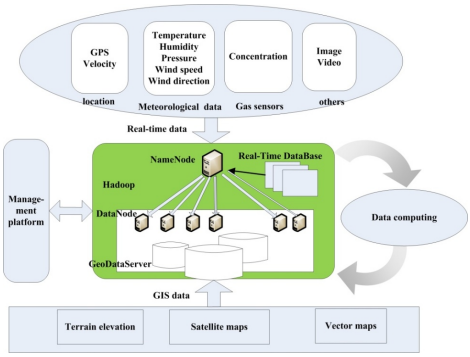


Fig. 3. Data Processing Infrastructure

4 Diffusion Model

After comprehensive comparison among Gaussian plume model [10,11], RIMPUFF (Lagrange atmospheric diffusion smoke model) [12], ATSTEP (the mixture of Gaussian model and Segmented plume model) [13,14], and CALPUFF (three-dimensional unsteady Lagrange smoke diffusion system) [15], Gaussian plume model is chosen as the diffusion and situation forecast model of the system. The model can not only simulate current distribution of atmospheric pollutants diffusion

concentration in the leeward direction, but also forecast pollutant distribution sometime in the future according to real-time and historical data [16-19]. Headwind diffusion is considered and added to Gaussian plume model for optimization [20, 21]. The optimized model is more suitable for continuous point source diffusion. Pasquill diffusion parameter determination method combined with atmospheric parameters is taken to determine the diffusion coefficient. Subsequently, the concentration distribution of chemical pollutant and diffusion trend are visualized via model computing to realize diffusion analysis and situation forecast.

5 3D Visualization

As shown in Fig.4, 3D real-time visualization displays real-time distribution of chemical pollutant and provides the reference for situation forecast. It is related to the construction of 3D terrain platform, real-time data processing and visualization technology.

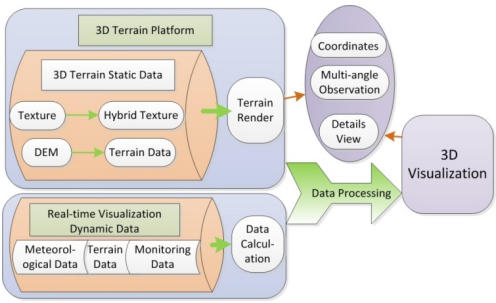


Fig. 4. 3D Real-time Visualization Platform

The construction of 3D terrain platform can be divided into 3 steps. Firstly, 3D terrain static data is retrieved from the database and calculated with corresponding texture data for the hybrid textures. Secondly, terrain elevation is used to divide terrain blocks according to terrain multi-resolution structure [22], and terrain data is obtained from building Quad-tree Spatial Data Model [23]. Finally, the hybrid texture and terrain data are used in the construction of 3D terrain platform via Semi-CLOD (Semi-Continuous Level Of Details) algorithm [24] and Surface Rendering technology[25]. The 3D terrain platform can realize location, paranoma roaming, multi-angle observation, multi-resolution and multi-level detail view.

Real-time data processing calculates three categories of real-time sensing data (meteorological data, geographical data, and monitoring data) stored in the database via the established model, and it processes the calculated data via the parsed mapping algorithm and interpolation method to display current pollution level and range in the targeted area. Recent diffusion trend of chemical pollution can be predicted via relevant parameters setting and historical data analysis. And the pollution level and range can be displayed via paranoma roaming.

6 Conclusions

In this paper, we propose a 3D visualization system for atmospheric chemical pollutant analysis. The main idea of the system is summarized as follows.

1. Build a distributed system for massive data storage and parallel computing.
2. Estimate gas concentrations around 3D space according to optimized diffusion model, terrain data, and real-time sensor data.
3. Display real-time distribution and situation forecast efficiently in 3D view.

We will research the massive unstructured data access and optimize the diffusion model. In addition, web-based 3D visualization will also be an important object to research.

Acknowledgement. This work was financially supported by the National Key Technology Research and Development Program (No. 2011BAK16B00) in China.

References

1. Yu, J.J.Q., Li, V.O.K., Lam, A.Y.S.: Sensor Deployment for Air Pollution Monitoring Using Public Transportation System. In: 2012 IEEE World Congress on Computational Intelligence, WCCI 2012, pp. 1–7 (2012)
2. Xiao, J., Ying, W.: Monitoring System of Regional Environmental Pollution Index Based on Telecommunication Network. In: 2011 International Conference on Electrical and Control Engineering, ICECE 2011, Yichang, China, pp. 254–256 (2011)
3. Chen, W., Chen, S.: Application of GIS Technology in the Emergency Monitoring of Sudden Air Pollution Accident. In: 2010 2nd International Conference on Information Science and Engineering, ICISE 2010, Hangzhou, China, pp. 3550–3555 (2010)
4. Wang, C., Wan, T.R., Palmer, I.J.: Automatic Reconstruction of 3D Environment Using Real Terrain Data and Satellite Images. Intelligent Automation and Soft Computing 18(1), 49–63 (2012)
5. Narashid, R.H., Mohd, W.M.N.W.: Air Quality Monitoring Using Remote Sensing and GIS Technologies. In: 2010 International Conference on Science and Social Research, CSSR 2010, Kuala Lumpur, Malaysia, pp. 1186–1191 (2010)
6. Sequeira, V., Wolfart, E., Bovisio, E., et al.: Hybrid 3D Reconstruction and Image-Based Rendering Techniques for Reality Modeling. In: Conference on Videometrics and Optical Methods for 3D Shape Measurement, San Jose, CA. SPIE, vol. 4309, pp. 126–136 (2001)
7. Hou, H.-D., Zhang, J.-F.: Research on Real-Time Visualization of Large-scale 3D Terrain. In: 2012 International Workshop on Information and Electronics Engineering, IWIEE 2012, Harbin, China, vol. 29, pp. 1702–1706 (2012)
8. Hu, Y., Li, D., He, X., Sun, T., Han, Y.: The Implementation of Wireless Sensor Network Visualization Platform based on Wetland Monitoring. In: 2009 Second International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2009, Tianjin, China, pp. 224–227 (2009)
9. Kamiński, L., Kulawiak, M., Ciżmowski, W., Chybicki, A.: Web-based GIS dedicated for marine environment surveillance and monitoring. In: IEEE Oceans 2009-Europe, Oceans, Bremen, Germany, vol. 1-2, pp. 1247–1253 (2009)

10. Ebrahimi, M., Jahangirian, A.: New Analytical Formulations for Calculation of Dispersion Parameters of Gaussian Model Using Parallel CFD. Environmental Fluid Mechanics 13(2), 125–144 (2013)
11. Leroy, C., Maro, D., Hebert, D., et al.: A study of the atmospheric dispersion of a high release of krypton-85 abovea complex coastal terrain, comparison with the predictions of Gaussian models (Briggs, Doury, ADMS4). Journal of Environmental Radioactivity, 937–944 (2010)
12. Roberti, D.R., Souto, R.P., de Campos Velho, H.F., et al.: Parallel Implementation of a Lagrangian Stochastic Model for Pollution Dispersion. In: Proceedings of the 16th Symposium on Computer Architecture and High Performance Computing, Foz dolguacu, PR, Brazil, pp. 142–149 (2004)
13. Hao, Y., Yu, Q., Qu, J.: Application of ATSTEP in Decision Support System for Nuclear Emergency Management. Nuclear Power Engineering 23(4), 102–107 (2002)
14. Yao, R., Hao, H., Hu, E., Cao, J.: Comparison of Two Kinds of Atmospheric Dispersion Model Chains in RODOS. Radiation Protection 23(30), 146–155 (2003)
15. Bo, X., Ding, F., Xu, H., Li, S.-B.: Review of Atmospheric Diffusion Spersion Model CALPUFF Technology. Environmental Monitoring Management and Technology 21(3), 9–13 (2009)
16. Cai, X., Chen, J., Kang, L.: An Atmospheric Diffusion Model for Conditions of Nuclear Accident. Radiation Protection 23(5), 293–299 (2003)
17. Yang, B., Gu, X.-M., Zhang, F., Zhang, F.: Development of Simulated Evaluation and Decision Support System of Atmospheric Pollution Dispersion. Science of Surveying and Mapping 36(3), 147–149 (2011)
18. Cheng, Y.: Gaussian Model Used Gas Pipeline after the Gas Leak Spread. Petroleum Chemical Industry of Inner Mongolia (14), 49–51 (2010)
19. Li, H., Deng, J., Wang, X., Zhang, L.: Calculation of Diffusion of Radionuclides Cloud in Atmospheric Using the Gaussian Model. Radiation Protection 24(2), 92–99 (2004)
20. Dvorak, R., Zboril, F., Kapoun, M., Masek, I.: Modeling of Atmospheric Dispersion from Point Source. In: Second UKSIM European Symposium on Computer Modeling and Simulation, Liverpool, England, pp. 46–51 (2008)
21. Briant, R., Seigneur, C., Gadrat, M., Bugajny, C.: Evaluation of roadway Gaussian plume models with large-scale measurement campaigns. Geoscientific Model Development 6(2), 445–446 (2013)
22. Tong, X., Ben, J., Zhang, Y.: Design and Quick Display of Global Multi-Resolution Spatial Data Model. Science of Surveying and Mapping 31(1), 72–74 (2006)
23. Zhang, L., Tang, L.-W.: Study of Organizing Seamless Great Capacity Spatial Data Based on Quad-tree. Computer Technology and Development 21(1), 77–80 (2011)
24. Hoppe, H.: Smooth View-dependent Level-Of-Detail Control and its Application to Terrain Rendering. In: 9th Annual IEEE Conference on Visualization, VIS 1998, Research Triangle Park, NC, USA, pp. 35–42 (1998)
25. John, D.: A Novel Technique for Visualizing High-Resolution 3-D Terrain Maps. In: Conference on Stereoscopic Displays and Virtual Reality Systems XIV, San Jose, CA. SPIE, vol. 6490, pp. 49003–49013 (2007)

Detection Method for Credibility Defect of Simulation Based on Sobol' Method and Orthogonal Design

Zhong Zhang, Ke Fang, Fang Wu, and Ming Yang*

Control and Simulation Center, Harbin Institute of Technology,
150080 Harbin, P.R. China
{zhangzhong0108,yangming_csc}@163.com,
fangke@hit.edu.cn, wufangyz@126.com

Abstract. Against the defect detection problem of simulation credibility, a method based on Sobol' method and orthogonal design is proposed. Firstly, taking acceptable range of simulation credibility as measurement standard, then credible indexes and incredible indexes of simulation system are determined. Secondly, experiment scheme is arranged according to an extended table transformed from orthogonal table, and incredible index's defect rank is judged by experimental result analysis combining with membership function of defect rank. Thirdly, Latin Hypercube sampling is taken from relevant indexes of incredible index, and then sensitivity coefficients of relevant indexes are calculated using Sobol' method. Finally an example is given to validate the effectiveness of the proposed method.

Keywords: simulation credibility, defect detection, Sobol' method, orthogonal design, sampling.

1 Introduction

Simulation technology is a comprehensive technology [1] with the advantages of economy, safety, no restrictions on weather condition and location space, which is widely used in many fields [2-3]. Simulation system is the application form of simulation technology. Whether simulation system has sufficient credibility relates directly to the success or failure of simulation system's application. Therefore, How to evaluate simulation credibility has become a hot topic in simulation community.

Carrying out simulation credibility evaluation has two main purposes: one is to measure simulation credibility from overall and partial aspects, and the other is to find out simulation credibility defect which can provide guidance for simulation system's improvement.

Almost all the existing methods for simulation credibility evaluation are proposed against the first evaluation purpose [4-7], and the second evaluation purpose is ignored despite reflecting better the value of evaluation work. Simulation defects cannot be simply interpreted as low-credibility indexes, we should consider the difference

* Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 421–428, 2013.

between different indexes in their influence on simulation credibility. Because the improvement works of simulation systems usually have limitation on time and funds, which make it impossible to improve all indexes of simulation system. Which index should be promoted firstly? Whether the improvement of an index has a negative influence on other indexes and how is the influence degree? All of the above are core problems that defect detection of simulation credibility needs to solve. In this field, it has not yet seen the study. Against these problems, a detection method for simulation credibility defect, which is based on Sobol' method [8] and orthogonal design, is proposed in this paper.

2 Problem Description

Simulation system is a complex composed by multiple interacting elements, which is generally decomposed into several subsystems. Subsystem and element are all components of simulation system. Their difference is that subsystem has the decomposability and it usually contains multiple elements. In consideration of clear expression, below we only use element sets to represent simulation system and its components, and no longer specify correlation between elements.

Now suppose a simulation system $S = \{e_1, e_2,\ldots, e_m\}$, where e_i ($i = 1, 2,\ldots, m$) are elements of S. $P = \{p_1, p_2,\ldots, p_n\}$ is index set of S. Index p_i describes the credibility of component s_i, where $s_i = \{e_{m_{1,i}}, e_{m_{2,i}},\ldots, e_{m_{n_i,i}}\}$, $s_i \subseteq S$, $s_i \neq \varnothing$, $i = 1,2,\ldots,n$. Note that the same component usually needs different indexes to describe its credibility from different angles, in other words, there generally exists p_i and p_j which satisfy $i \neq j$, such that $s_i = s_j$. In conclusion, the credibility evaluation model of simulation system S can be expressed as

$$v(S) = f\left(v_{p_1}(s_1), v_{p_2}(s_2),\ldots, v_{p_n}(s_n)\right) \tag{1}$$

Where $v_{p_i}(s_i)$ represents measurement value of index p_i, which reflects the credibility of component s_i, $0 \le v_{p_i}(s_i) \le 100$, $i = 1,2,\ldots,n$; $f(\bullet)$ represents simulation credibility function, the analytical form of which is difficult to obtain and usually expressed in the form of index system and data fusion function.

The problem this paper is to solve, suppose $f(\bullet)$ and $v_{p_i}(s_i)$ have been obtained, how to measure the negative influence of incredible index p_i on $v(S)$ and the sensitivity of relevant indexes when we improve p_i's credibility.

3 Detection Algorithm for Credibility Defect

We propose a detection algorithm for simulation credibility defect based on Sobol' method and orthogonal design, which includes the following steps:

Step 1. According to preliminary evaluation result of simulation credibility, index set P is divided into two subsets: incredible index set $P_1 = \{p_i \in P \mid 0 \le v_{p_i}(s_i) < \delta\}$ and credible index set $P_2 = \{p_i \in P \mid \delta \le v_{p_i}(s_i) \le 100\}$, where δ represents acceptable level of index's credibility. We generally select δ = 60 in engineering applications. Without loss of generality that $P_1 = \{p_1, p_2, \ldots, p_{n_1}\}$, $P_2 = \{p_{n_1+1}, p_{n_1+2}, \ldots, p_n\}$, $1 \le n_1 \le n$;

Step 2. Judge incredible index's defect rank. Incredible index has a negative influence on simulation system. This is because incredible index's credibility cannot reach its corresponding acceptable level, and then lower simulation credibility. Therefore, we can judge incredible index's defect rank based on the degree of variability in simulation credibility when incredible index is promoted to credible index. Take incredible index p_1 for example, its defect rank is judged in the following steps:

Step 2.1. Choose suitable two-level orthogonal table $L_a(2^c)$ satisfying $c \ge n_1$-1 according to incredible index $p_2, p_3, \ldots, p_{n_1}$;

Step 2.2. Select n_1-1 columns from orthogonal table $L_a(2^c)$ arbitrarily, and then construct an extended table

$$L_{2a}^{\mathrm{T}}\left(2^{n_1}\right) = \begin{pmatrix} (1)_{a\times 1} & L_a\left(2^{n_1-1}\right) \\ (2)_{a\times 1} & L_a\left(2^{n_1-1}\right) \end{pmatrix} \tag{2}$$

Extended table has three features as follows:

(1) The times of different numerals appear in the same column are equal;
(2) In any two columns, when two numerals in the same row are regarded as an ordered pair, the times that each ordered pair appears are equal;
(3) Numeral in ith row, jth column is equal to numeral in (i+a)th row, jth column, i = 1, 2,…, a, j = 2, 3,..., n_1;
These characteristics make extended table have better uniformity and orthogonality than original orthogonal table for the first factor;

Step 2.3. Select two levels of incredible index p_i as $v_{p_i}(s_i)$ and δ, i = 1, 2,...,n_1. Then design 2a experiment schemes according to extended table $L_{2a}^{\mathrm{T}}(2^{n_1})$. Incredible index p_1 observed primarily is arranged in the first column of extended table;

Step 2.4. Use credibility evaluation model (1) to calculate simulation credibility corresponding to different experiment schemes, we get $v_1(S)$, $v_2(S)$,…, $v_{2a}(S)$;

Step 2.5. In experiment scheme i and i+a, values of index p_j are equal, i = 1, 2,…, a, j = 2, 3,...,n_1. Thus, $|v_{i+a}(S) - v_i(S)|$ accurately reflects the change of simulation credibility when p_1 is promoted. Adopt normal membership cloud to determine p_1's membership to defect rank as shown in Fig. 1. The parameters of normal membership clouds corresponding to defect ranks are shown in Table 1. Then a five-dimensional membership vector is calculated based on $|v_{i+a}(S) - v_i(S)|$, we get $R_i = (r_{i,1}, r_{i,2}, r_{i,3}, r_{i,4}, r_{i,5})$, $r_{i,j} \ge 0$, i = 1, 2,…,a, j = 1, 2,…, 5;

Step 2.6. Calculate average membership vector:

$$\overline{R} = \left(\sum_{i=1}^{a} r_{i,1}/a, \sum_{i=1}^{a} r_{i,2}/a, \ldots, \sum_{i=1}^{a} r_{i,5}/a \right) \tag{3}$$

Then defect rank of incredible index p_1 is determined according to the principle of maximum membership degree;

Step 3. Calculate sensitivity coefficients of relevant indexes of incredible index. To promote evaluation value of incredible index, we need to modify its corresponding component, which possibly makes a negative influence on other indexes. This requires us to carry out sensitivity analysis of relevant indexes. If a relevant index's sensitivity is high, we need to avoid adverse impact on it when promoting incredible index. Otherwise, the loss of simulation improvement work is likely to exceed its gain. Take incredible index p_1 for example, sensitivity coefficients of its relevant indexes are calculated in the following steps:

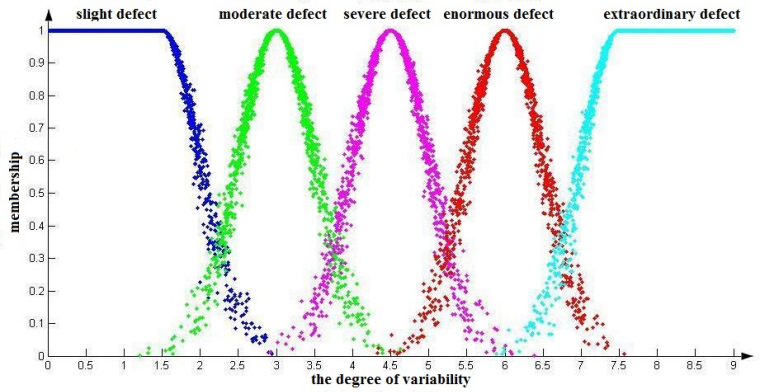


Fig. 1. Normal membership clouds of different defect ranks

Table 1. Parameters of different normal membership clouds

defect rank	expected value	entropy	hyper entropy
slight defect	1.5	0.5	0.05
moderate defect	3	0.5	0.05
severe defect	4.5	0.5	0.05
enormous defect	6	0.5	0.05
extraordinary defect	7.5	0.5	0.05

Step 3.1. Determine relevant index set of incredible index p_1: $M_{p_1} = \{ p_i \in P \mid s_i \bigcap s_1 \neq \varnothing \}$;

Step 3.2. Set the sampling interval of relevant index p_j: $[(1-\tau)v_{p_j}(s_j),(1+\tau)v_{p_j}(s_j)]$, $p_j \in M_{p_1}$, where τ represents the amplitude of sampling, we generally take $\tau = 10\%$ in engineering applications;

Step 3.3. Use Latin Hypercube sampling to construct sample set in the sampling interval of relevant index. To ensure the accuracy of sensitivity analysis, the number of samples is generally 200 times more than the number of relevant indexes;

Step 3.4. Use credibility evaluation model (1) to calculate simulation credibility corresponding to different samples, and then calculate sensitivity coefficient of each relevant index using Sobol' method.

4 Application Example

Take the credibility defect detection of a guidance simulation system for example, we validate the effectiveness of the proposed method.

The basic elements of the guidance simulation system are aerodynamic model (e_1), motion model (e_2), guidance system model (e_3), navigation system model (e_4), atmosphere model (e_5) and earth model (e_6). This example is only used to validate the effectiveness of detection algorithm for credibility defect, so we will not introduce the detail of credibility evaluation process and directly give the index system and index values as shown in Fig. 2 and Table 2. According to all the index values, we calculate the credibility of the guidance simulation system using credibility evaluation model:

$$\begin{aligned} &v(s_1) = f_1\left(f_2\left(v(s_4), v(s_5), v(s_6), v(s_7), v_{p_{13}}, v_{p_{14}}, \ldots, v_{p_{17}}\right), f_3\left(v(s_8), v(s_9)\right)\right) \\ &v(s_4) = f_4\left(v_{p_1}, v_{p_2}\right), v(s_5) = f_5\left(v_{p_3}, v_{p_4}, \ldots, v_{p_8}\right), v(s_6) = f_6\left(v_{p_9}, v_{p_{10}}, v_{p_{11}}, v_{p_{12}}\right), \\ &v(s_7) = f_7\left(v_{p_{18}}, v_{p_{19}}\right), v(s_8) = f_8\left(v_{p_{20}}, v_{p_{21}}, v_{p_{22}}, v_{p_{23}}\right), v(s_9) = f_9\left(v_{p_{24}}, v_{p_{25}}, v_{p_{26}}, v_{p_{27}}\right) \end{aligned} \tag{4}$$

Where $f_1, f_2, f_4, f_5, f_6, f_7$ are OWG operators; f_3, f_8, f_9 are OWA operators. Then we get $v(s_1) = 68.50$.

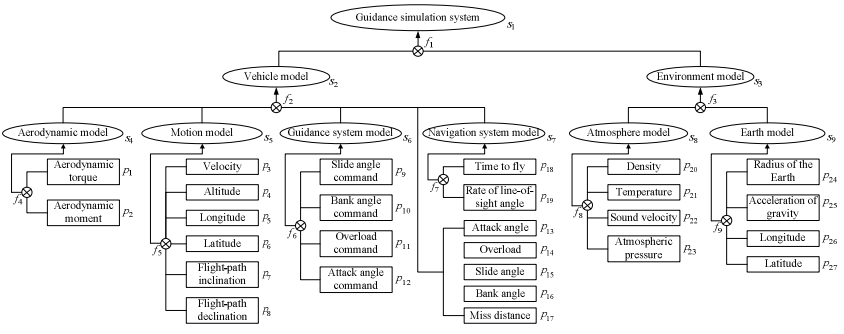


Fig. 2. The index system for credibility evaluation of guidance simulation system

Table 2. The evaluation value of index

component	index	value	component	index	value	component	index	value
s_4	p_1	40.31	s_6	p_{10}	95.73	s_7	p_{19}	67.06
	p_2	36.45		p_{11}	94.12		p_{20}	63.15
s_5	p_3	85.09		p_{12}	96.61	s_8	p_{21}	65.96
	p_4	87.79		p_{13}	62.94		p_{22}	70.36
	p_5	84.77	s_2	p_{14}	51.98		p_{23}	52.47
	p_6	83.98		p_{15}	69.43		p_{24}	70.92
	p_7	80.33		p_{16}	68.72	s_9	p_{25}	64.31
	p_8	80.82		p_{17}	44.95		p_{26}	85.71
s_6	p_9	93.15	s_7	p_{18}	73.14		p_{27}	83.93

Use the algorithm proposed in this paper to detect the credibility defect of the guidance simulation system. The concrete steps are in the following:

Firstly, determine credible index set P_1 and incredible index set P_2. We select δ=60 , then P_1={p_1, p_2, p_{14}, p_{17}, p_{23}}, P_2={p_3,…, p_{13}, p_{15}, p_{16}, p_{18},…, p_{22}, p_{24},…, p_{27}}.

Next, judge the defect rank of incredible index p_1. Select two-level orthogonal table $L_8(2^7)$, and then construct an extended table $L_{16}^{\mathrm{T}}(2^5)$. According to $L_{16}^{\mathrm{T}}(2^5)$, 16 experiment schemes are arranged as shown in Table 3.

Table 3. Experiment scheme

No.	p_1	p_2	p_{14}	p_{17}	p_{23}	No.	p_1	p_2	p_{14}	p_{17}	p_{23}
1	40.31	36.45	51.98	44.95	52.47	9	60.00	36.45	51.98	44.95	52.47
2	40.31	36.45	51.98	44.95	60.00	10	60.00	36.45	51.98	44.95	60.00
3	40.31	36.45	60.00	60.00	52.47	11	60.00	36.45	60.00	60.00	52.47
4	40.31	36.45	60.00	60.00	60.00	12	60.00	36.45	60.00	60.00	60.00
5	40.31	60.00	51.98	60.00	60.00	13	60.00	60.00	51.98	60.00	60.00
6	40.31	60.00	51.98	60.00	52.47	14	60.00	60.00	51.98	60.00	52.47
7	40.31	60.00	60.00	44.95	60.00	15	60.00	60.00	60.00	44.95	60.00
8	40.31	60.00	60.00	44.95	52.47	16	60.00	60.00	60.00	44.95	52.47

Use credibility evaluation model (4) to calculate guidance simulation credibility corresponding to each experiment scheme, we get: ($v_1(s_1)$, $v_2(s_1)$,…, $v_{16}(s_1)$)=(68.50, 68.74, 69.10, 69.34, 69.48, 69.24, 69.28, 69.04, 71.00, 71.25, 71.62, 71.86, 72.02, 71.77, 71.81, 71.56). The average membership vector of p_1 is $\overline{R}$ = (0.319, 0.681, 0, 0, 0), and then we judge p_1 is a moderate defect according to the principle of maximum membership degree. Similarly, p_2, p_{14}, p_{17} and p_{23} are judged slight defects.

Finally, calculate sensitivity coefficients of relevant indexes of incredible index p_1. It can be seen from Table 2 that p_1~p_2 describe the credibility of component s_4 and p_{13}~p_{17} describe the credibility of component s_2. Because of $s_2 \bigcap s_4 = s_2 \neq \varnothing$, the relevant index set of p_1 is $M_{p_1} = \{p_2, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}\}$. Here we select sampling

amplitude τ=10%, then the sampling interval of relevant indexes is [32.81,40.09]× [46.78,57.18]×[62.49,76.37]×[61.85,75.59]×[40.45,49.45]. Sampling was conducted in this interval by Latin Hypercube sampling and a total of 20000 sampling points is recorded. Using credibility evaluation model (4) to calculate guidance simulation credibility corresponding to each sampling point, and then calculate sensitivity coefficient of each relevant index using Sobol' method. The analysis result is shown in Table 4. According to the analysis result, when the aerodynamic model is modified to promote the credibility of p_1, we should sequentially avoid to make negative influence on p_{13}, p_{17}, p_{16}, p_{15}, p_{14}, p_2.

Similarly, we can calculate sensitivity coefficients of relevant indexes of incredible index p_2, p_{14}, p_{17}, p_{23} using Sobol' method. The analysis result is not listed due to space constrains.

In conclusion, in the improvement work of the guidance simulation system, we should pay more resources to promote the credibility of p_1 firstly. After that the credibility of p_2, p_{14}, p_{17}, p_{23} are promoted respectively. Although p_2's credibility is lower than p_1's credibility at present, aerodynamic torque is much more important than aerodynamic moment for a guidance simulation system. Therefore, the credibility of aerodynamic force is considered preferentially in the case of limited resources, which accords with our conclusion.

Table 4. Result of sensitivity analysis

No.	relevant index	the first order sensitivity	the total sensitivity
1	p_2	0.0439	0.0726
2	p_{13}	0.1880	0.2185
3	p_{14}	0.1757	0.1933
4	p_{15}	0.1751	0.2008
5	p_{16}	0.1803	0.2031
6	p_{17}	0.1910	0.2086

5 Conclusions

Against the defect detection problem of simulation credibility, a defect detection method based on Sobol' method and orthogonal design is proposed. According to preliminary evaluation result of simulation credibility, the importance and sensitivity of different indexes are analyzed by designing suitable experiment schemes. The proposed method can accurately find out the bottleneck restricting simulation credibility, which will lay the foundation for simulation credibility optimization.

References

1. Carson, J.S.: Introduction to Modeling and Simulation. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (eds.) Proceedings of the 2004 Winter Simulation Conference, pp. 7–13. IEEE Press, Orlando (2003)

2. Bazilevs, Y., Hsu, M.C., Zhang, Y.: A Fully-Coupled Fluid-Structure Interaction Simulation of Cerebral Aneurysms. Computational Mechanics 46(1), 3–16 (2010)
3. Li, W., Joos, G., Belanger, J.: Real-Time Simulation of a Wind Turbine Generator Coupled with a Battery Supercapacitor Energy Storage System. IEEE Transactions on Industrial Electronics 57(4), 1137–1145 (2010)
4. Fang, K., Wang, Z.C.: CES Net Method for Simulation System's Credibility Evaluation. Control and Decision 26(5), 737–742 (2011)
5. Parker, M.W., Shoop, S.A., Coutermarsh, B.A., Wesson, K.D., Stanley, J.M.: Verification and Validation of a Winter Driving Simulator. Journal of Terramechanics 46, 127–139 (2009)
6. Robinson, S., Brooks, R.J.: Independent Verification and Validation of an Industrial Simulation Model. Simulation 86(7), 405–416 (2010)
7. Liu, F., Yang, M.: An Optimal Design Method for Simulation Verification, Validation and Accreditation Schemes. Simulation 85(6), 375–386 (2009)
8. Sobol, I.M.: On Sensitivity Estimates for Nonlinear Mathematical Models. Matem Modelirovanie 2(1), 112–118 (1990)

The Distribution Characteristics Analysis of Advanced RES Feature

Bin Zhu[1,2,*], Weidong Jin[2], Zhibin Yu[2], and Jianqu Zhu[2]

[1] School of Electron Engineering, Yangtze Normal University, Chongqing, China
[2] School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China
zb8132002@163.com, wdjin@home.swjtu.edu.cn

Abstract. For the further study of advanced radar emitter signals (RES) feature, the distribution of advanced RES features was deeply analysis. Normality test method of RES feature distribution was proposed based on the Mahalanobis distance. First, it realized normality test of one-dimensional feature through the calculation of skewness and kurtosis, and then, through the calculation of Mahalanobis distance, it realized the normality test of RES multi-dimensional feature vectors. Simulation results show that, the distribution of RES feature vector doesn't have the standard normal characteristics. This conclusion can provide the theoretical support for the further analysis of RES features.

Keywords: Radar emitter signal, Feature analysis, Mahalanobis distance, Normality test.

1 Introduction

The recognition of advanced RES is an important part of electronic intelligence systems, electronic support measure systems, and radar warning receiver systems in electronic warfare. The mining researches of new RES feature are becoming research focus [1]. Many scholars have done a lot of in-depth researches in RES feature extraction, but there are little characteristics studies of these features. Some studies make a decision based on the histogram distribution [2]. This method is relatively rough and is not systematic. To this end, Normality test method of RES feature distribution characteristics is proposed based on the Mahalanobis distance. The method first analysis the distribution characteristics of single one-dimensional RES signal characteristics, on this basis, the distribution characteristics of the multi-dimensional RES feature vector were analysis. This research has a positive reference value on RES features evaluation and the further study of advanced RES features.

2 The Distribution Analysis of Advanced RES Feature

The normal distribution is the most important distributed in nature, it can describe a number of random phenomena and act as a true overall model. It created many

* Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 429–434, 2013.

statistical methods has also been widely used based on the assumption that the data generated from a multivariate normal distribution. In practical applications, if the single one-dimensional characteristics of radar emitter signals obey a normal distribution, we wouldn't be able to get the conclusion that the distribution of multidimensional feature vector of RES obeys a normal distribution, so, study on the distribution of RES multidimensional feature vector is very necessary.

2.1 One-Dimensional Normal Distribution Test

Test methods of one-dimensional normal sample substantially are Karl Pearson moment method, Swilk test methods, *P-P* diagram method, *Q-Q* diagram method, Kolmogorov-Smirnov method, Chi-square goodness of fit test methods, etc [3]. Normal distribution characteristics of one-dimensional RES features are tested by using moment method. Moment method test normal distribution by calculating the skewness and kurtosis, which was used the third-moment and fourth-moment principle. When the sample is greater than 200, the test efficacy of this method is high. The skewness coefficient indicates the degree of deviation between distribution and average. It is a measure of asymmetries in the distribution. The kurtosis coefficient indicates the convex and flat degree of the distribution pattern graphics peak. It is a measure of the graphical slope. When kurtosis and skewness are zero, the variable distribution is ideal normal distribution.

For the time sequence $X_i(i=1,2,\cdots,N)$, N is the number of samples. The of skewness coefficients b_s and kurtosis coefficients b_k are defined as follow.

$$b_s = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{x_i-\bar{x}}{s}\right]^3 \tag{1}$$

$$b_k = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{x_i-\bar{x}}{s}\right]^4 - 3 \tag{2}$$

Where, $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ it is the mean value of the time sequence. $s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i-\bar{x}\right)^2}$, it is the standard deviation of the time series. s_{var} is the variance of the time sequence. When the number of samples is large enough, and if the samples follow a normal distribution, the coefficient of skewness and kurtosis follow a normal distribution, and the mathematical expectation is zero, the variance are as follows.

$$s_{bs} = \sqrt{\frac{6(N-2)}{(N+1)(N+3)}} \tag{3}$$

$$s_{bk} = \sqrt{\frac{24N(N-2)(N-3)}{(N+1)^2(N+3)(N+5)}} \tag{4}$$

Assuming that the sequences follow a normal distribution, the confidence level a=0.05, if $|b_s|>1.96s_{bs}$ and $|b_k|>1.96s_{bk}$, reject the hypothesis. This indicates that the sequence does not follow a normal distribution. Otherwise, the assumption holds that the sequences follow a normal distribution.

2.2 Multidimensional Data Normality Test

If a variable set obey the multivariate joint normal distribution, each variable must obey the one-dimensional normal, but the converse is not established. Therefore, for the multi-dimensional characteristic variables, not only each one-dimensional feature would be doing unvaried normality test, but also the joint multi-dimensional characteristic variables would be doing normality test. The theory system of one-dimensional normality test has been well basically established, and so far, the multi-dimensional normality test has not a comprehensive theoretical system [4].

By the nature of the normal distribution, Under H_0, the Mahalanobis distance D^2, which is from the sample X to the center μ, obey the chi-square distribution, the freedom degrees of the chi-square distribution is p. D_i^2 is calculated based on the sample X (i), sort D_i^2 , $D_1^2 \le D_2^2 \le \cdots \le D_N^2$.

Empirical distribution function of the statistics D^2 is taken as.

$$F_N(D_i^2) = \frac{i-0.5}{N} = p_i \approx H(D_i^2 \mid p) \tag{5}$$

Where, $H(D_i^2 \mid p)$ represents the values of the distribution function D_i^2 of $\chi^2(p)$.

Assume that the p_i quintile of chi-square distribution is χ_i^2 , then $H(\chi_i^2 \mid p) = p_i$, that is. $\chi_i^2 = H^{-1}(p_i \mid p)$. Then the quantile p_i of the sample is obtained by the empirical distribution. $D_i^2 = F_N^{-1}(p_i)$. If $H(x \mid p) \approx F_N(x)$, then $D_i^2 = \chi_i^2$, draw a scatter plot of point (D_i^2, χ_i^2) , you can get *Q-Q* diagram of a normal distribution, Specific steps are as follows.

Step 1 Calculate the sample mean $\bar{X}$ and the covariance matrix S of p-dimensional sample points $X_i (i = 1,2,\cdots,n)$. The number of sampling points is n.

$$S = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})' \tag{6}$$

Step 2 Calculate the Mahalanobis distances D_i^2 , which were got from sample point $X_i (i = 1,2,\cdots,n)$ to $\bar{X}$.

$$D_i^2 = (X_i - \bar{X})S^{-1}(X_i - \bar{X})' \quad (i = 1,2,\cdots,n) \tag{7}$$

Step 3 The Mahalanobis distances D_i^2 are arranged in ascending order. $D_1^2 \le D_2^2 \le \cdots \le D_n^2$.

Step 4 Calculate p_i and χ_i^2 . $p_i = \frac{i-0.5}{n}, (i = 1,2,\cdots,n)$, $H(\chi_i^2 \mid p) = p_i$.

Step 5 The coordinate system is established, D_i^2 is the abscissa, and χ_i^2 is the vertical coordinate. These points (D_i^2, χ_i^2) are plotted. The *Q-Q* plot is obtained

Step 6 Analysis of the distribution of the scattered points, if these points are spread in a straight line, which passes through the origin and the slope is 1, and then the original hypothesis is accepted. These data can be considered to obey the multivariate normal distribution, otherwise, reject the original hypothesis.

3 Calculation and Analysis

In order to study the space distribution characteristics of advanced RES features, CR1, CR2 characteristics of resemblance coefficient feature [5] are selected, and M1, M4 characteristics of wavelet gray moments feature [6] are selected. They will be used as a numerical example. The calculation and analysis is completed on the platform of Lenovo E46L. It selected five typical radar emitter signals in BPSK, QPSK, LFM, NLFM, and FSK, added 25db noise to all RES. 1000 feature vectors were extracted by using the aforementioned feature extraction methods, these feature vectors were used to analysis of the calculation example. The distribution parameter calculation results of single one-dimensional RES are shown in table 1 - table 4 below.

Table 1. The feature distribution parameter of resemblance coefficient CR1

RES	$\overline{x}$	s	s_{var}	b_s	b_k	s_{bs}	s_{bk}
BPSK	0.3666	0.0008	0	0.0387	-0.1144	0.0773	0.1538
QPSK	0.0584	0.0003	0	-0.1256	0.0755	0.0773	0.1538
LFM	0.3924	0.0016	0	0.0625	-0.1778	0.0773	0.1538
NLFM	0.2101	0.0005	0	-0.1067	-0.1108	0.0773	0.1538
FSK	0.2188	0.0004	0	0.0507	-0.0730	0.0773	0.1538

Table 2. The feature distribution parameter of resemblance coefficient CR2

RES	$\overline{x}$	s	s_{var}	b_s	b_k	s_{bs}	s_{bk}
BPSK	0.6351	0.0015	0	0.0502	-0.0601	0.0773	0.1538
QPSK	0.1012	0.0006	0	-0.1359	0.0678	0.0773	0.1538
LFM	0.6801	0.0026	0	0.0350	-0.1518	0.0773	0.1538
NLFM	0.3640	0.0009	0	-0.0927	-0.0903	0.0773	0.1538
FSK	0.3792	0.0007	0	0.0433	0.0393	0.0773	0.1538

Table 3. The feature distribution parameter of wavelet gray moment M1

RES	$\overline{x}$	S	s_{var}	b_s	b_k	s_{bs}	s_{bk}
BPSK	67.3700	2.3505	5.5249	0.0799	0.1102	0.0773	0.1538
QPSK	36.6095	1.8420	3.3931	0.3597	0.1422	0.0773	0.1538
LFM	73.1585	2.1936	4.812	0.2574	0.1914	0.0773	0.1538
NLFM	35.2103	1.8506	3.4248	0.2669	0.0807	0.0773	0.1538
FSK	56.6898	2.3591	5.5652	0.3095	0.1504	0.0773	0.1538

Table 4. The feature distribution parameter of wavelet gray moment M4

RES	$\overline{x}$	s	s_{var}	b_s	b_k	s_{bs}	s_{bk}
BPSK	366.2078	6.0736	36.8891	-0.211	0.0074	0.0773	0.1538
QPSK	115.4873	1.853	3.4337	0.0066	0.1315	0.0773	0.1538
LFM	313.1696	7.8114	61.0187	0.1064	-0.0153	0.0773	0.1538
NLFM	94.2184	1.5072	2.2715	0.1578	-0.0372	0.0773	0.1538
FSK	205.9587	3.3526	11.2397	0.0066	-0.2275	0.0773	0.1538

It can be seen from table 1- table 4, if the confidence level a = 0.05, $|b_s| < 1.96 s_{bs}$ and $|b_k| < 1.96 s_{bk}$, then the assumption holds that the sequences follow a normal distribution. This shows that a single RES features are subject to normal.

Mahalanobis distance D_i^2 is horizontal, the quantile χ_i^2 is the vertical axis, and the Q-Q plot of scatter points, as shown in figure 1-figure 2 below, is drawn.

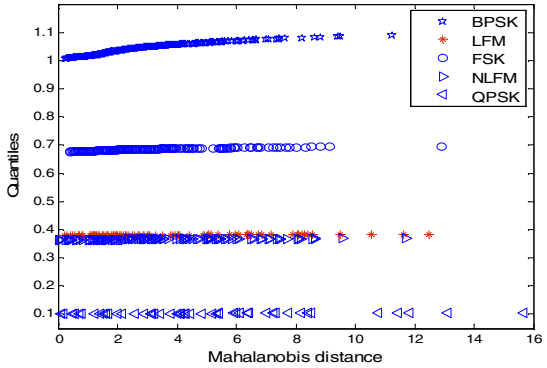


Fig. 1. The Q-Q plot of RES resemblance coefficient feature vectors

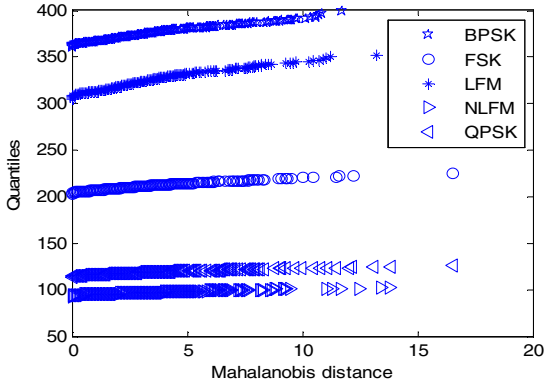


Fig. 2. The Q-Q plot of RES wavelet gray moments feature vectors

As can be seen from figure 1 and figure 2, scattered point (D_i^2, χ_i^2) of Q-Q figure is not dispersed in one straight line, which passes through the origin, and the slope is 1. In accordance with the principles of the Q-Q diagram test methods of chi-square statistic, we can assume that these data are not from multivariate normal population, In other words, the spatial distribution of these RES eigenvectors does not have a normal distribution.

4 Conclusions

The distribution of RES features in the feature space is critical to the radar emitter signal recognition, if some characteristics of different radar emitter signals in the feature space overlap less or no overlap, then the feature is more favorable to the recognition of radar emitter signals. Experimental results show that the single one-dimensional resemblance coefficient feature and wavelet gray moment feature have a normal distribution, but the multi-dimensional resemblance coefficient feature vector and wavelet gray moment feature vector are not necessarily normal distribution. The research group will be based on the research results, and make an in-depth study on RES feature distribution by using new method.

Acknowledgments. This work is supported by the national natural science foundation of China (no. 60971103). This research is funded by Chongqing natural science foundation, and the project no. is cstc2013jcyjA70010. This work is also supported by the science research project of Chongqing education commission (no. KJ121322) and science & technology project of Fuling district of Chongqing city (no. FLKJ, 2012ABA1037). The authors thank the editors and reviewers for their valuable comments and good suggestions.

References

1. Zhu, B., Jin, W.-D.: Feature Extraction of Radar Emitter Signal Based on Wavelet Packet and EMD. Lecture Notes in Electrical Engineering 19, 1408–1415 (2011)
2. Yang, Z.-X.: The Evaluation of Features for Advanced Radar Emitter Signals. Southwest Jiaotong University, Cheng Du (2012) (in Chinese)
3. Fang, J.-G., Mao, M.-C., Cheng, X.-X.: The analysis of statistics distribution characteristics on precipitation in Shanxi. Journal of Northwest University 39, 1408–1415 (2009)
4. Pan, Z.-Y., Zhao, N.-Q.: A New Test on Random Distribution and Multivariate Normal Distribution. Journal of Mathematical Medicine 19, 53–55 (2006)
5. Zhang, G.-X.: Intelligent Recognition Methods for Radar Emitter Signals. Southwest Jiaotong University, Cheng Du (2005) (in Chinese)
6. Zhu, B., Jin, W.-D., Yu, Z.-B.: Feature extraction for advanced radar emitter signals based on CWT and gray moment. International Journal of Advancements in Computing Technology 18, 373–380 (2012)

IMC-PID Design Based on an Approximation of Polynomial Equations

Kimikazu Kawaguchi, Hiroki Shibasaki, Ryo Tanaka, Hiromitsu Ogawa, Takahiro Murakami, and Yoshihisa Ishida

School of Science and Technology, Meiji University,
1-1-1, Higashimita, Tama-ku, Kawasaki, Kanagawa, 214-8571, Japan
kimikawaguchi21@gmail.com

Abstract. In this study, we propose a method of designing IMC-PID based on approximation of polynomial equations. In the industrial field, IMC-PID is used by many engineers, because it has a good robustness and a good performance. However, the design of the conventional IMC-PID controller is complex. To solve this problem, in this paper, we design IMC-PID controller by using polynomial equations approximation. Our simulation results show that the proposed method has a better performance than that of the conventional IMC-PID.

Keywords: Approximation of Polynomial equations, All-pole Approximation, Time Delay, IMC-PID.

1 Introduction

In the industrial field, Internal model control (IMC) is widely used by many engineers, because it has a good robustness and a good control performance. However, at the present day, there is necessary to convert IMC to PID type, because the controller of PID type is widely used. To convert the IMC controller to PID type, it is approximation of the time delay is essential. Rivera *et al.* [1] used zero order Pade approximation and first order Pade approximation to design the IMC-PID controller. Gong *et al.* [2] designed the IMC-PID controller by using second order symmetric Pade approximation. Panda *et al.* [3] and Lee *et al.* [4] designed the IMC-PID by using Maclaurin's series. Design of using Maclaurin's series is known as IMC-Mac. Jin *et al.* [5] used all-pole approximation to approximate the time delay, because they found a problem that it causes zeros to the control system in approximation of using Pade approximation. However, a design of the controller is complex in these conventional methods.

In this study, we design the IMC-PID controller by using an approximation of polynomial equations. By using this approximation, the calculation of the controller becomes simple. In addition, the proposed IMC-PID has a control performance better than the conventional methods. In simulation study, we confirm the effectiveness of the proposed method.

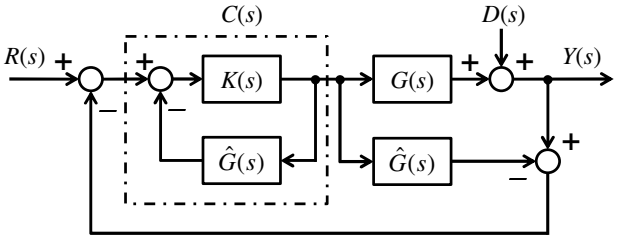
2 Design of IMC System

In this paper, the controlled plant $G(s)$ is expressed as follows.

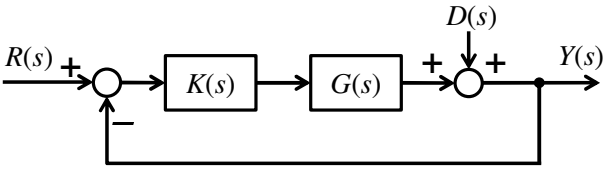
$$G(s) \cong \hat{G}(s) = \hat{G}_{+}(s)\hat{G}_{-}(s) \tag{1}$$

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 435–439, 2013.

In equation (1), $\hat{G}(s)$ is the plant model, $\hat{G}_{+}(s)$ is contains all the time delay and right-half-plane zeros, and $\hat{G}_{-}(s)$ is the transfer function with minimum phase characteristic and contains no predictive item. Then, the IMC control system is illustrated in Fig. 1(a). In Fig.1(a), $K(s)$ is a general controller, $C(s)$ is an IMC controller, $R(s)$ is a reference signal, $Y(s)$ is a plant output, and $D(s)$ is a disturbance.



(a) Block diagram of IMC principle



(b) An equivalent circuit of IMC system

Fig. 1. Block diagram of IMC system

An IMC controller $C(s)$ is obtained by equation (2). It can be rewritten as equation (3). Then, an IMC controller is expressed as equation (4), and IMC filter $f(s)$ is low-pass filter and is a reference model. From equations (3) and (4), $K(s)$ is obtained as equation (5). And Fig.1(a) can be changed into an equivalent circuit as Fig.1(b).

$$C(s) = \frac{K(s)}{1 + K(s)\hat{G}(s)} \tag{2}$$

$$K(s) = \frac{C(s)}{1 - \hat{G}(s)C(s)} \tag{3}$$

$$C(s) = \hat{G}_{-}^{-1}(s) f(s) \tag{4}$$

$$K(s) = \frac{\hat{G}_{-}^{-1}(s)}{f^{-1}(s) - \hat{G}_{+}(s)} \tag{5}$$

3 Proposed IMC-PID

In this section, we convert an IMC controller into PID type by using an approximation of polynomial equations. First, we rewrite a controller $K(s)$ as equation (6) so that it can be converted into PID type.

$$K(s) = \frac{k_0 + k_1 s + k_2 s^2 + \cdots}{s} \cong K_p + \frac{K_i}{s} + K_d s \tag{6}$$

We give a controlled plant $G(s)$ and an IMC filter $f(s)$ as follows, respectively, to explain how to design the IMC-PID controller.

$$G(s) = \frac{b}{1+as} e^{-Ls}, \text{where} \hat{G}_+(s) = e^{-Ls}, \hat{G}_-(s) = \frac{b}{1+as} \tag{7}$$

$$f(s) = \frac{1}{1+\lambda s} \tag{8}$$

In this paper, the time delay e^{-Ls} is approximated as equation (9). Then, we approximate by using all-pole approximation [5]. Equation (10) is given by substituting equation (7) and (8) in equation (5).

$$e^{-Ls} \cong \frac{1}{0.5L^2 s^2 + Ls + 1} \tag{9}$$

$$k_0 + k_1 s + k_2 s^2 + \cdots = \frac{0.5aL^2 s^3 + (aL + 0.5L^2)s^2 + (a+L)s + 1}{b[0.5L^2 \lambda s^2 + (L\lambda + 0.5L^2)s + (\lambda + L)]} \tag{10}$$

We rewrite equation (10) by equation (11). The controller of PID type as equation (12) is obtained by applying an approximation of polynomial equations to (11).

$$\begin{aligned} &(\lambda + L)k_0 + [(L\lambda + 0.5L^2)k_0 + (\lambda + L)k_1]s \\ &+[0.5L^2 \lambda k_0 + (L\lambda + 0.5L^2)k_1 + (\lambda + L)k_2]s^2 \\ &+[0.5L^2 \lambda k_1 + (L\lambda + 0.5L^2)k_2]s^3 = \frac{1 + (a+L)s + (aL + 0.5L^2)s^2 + 0.5aL^2 s^3}{b} \end{aligned} \tag{11}$$

$$\begin{bmatrix} k_0 \\ k_1 \\ k_2 \end{bmatrix} = \frac{1}{b} \begin{bmatrix} \lambda + L & 0 & 0 \\ L\lambda + 0.5L^2 & \lambda + L & 0 \\ 0.5L^2 \lambda & L\lambda + 0.5L^2 & \lambda + L \\ 0 & 0.5L^2 \lambda & L\lambda + 0.5L^2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ a + L \\ aL + 0.5L^2 \\ 0.5aL^2 \end{bmatrix} \tag{12}$$

As described above, the general formula of the IMC-PID controller can be obtained, and it can be design simply by choosing the filter constant λ. Moreover, calculation of controller becomes easy by using algebra calculation.

4 Simulation Results

In this section, in order to confirm the effectiveness of the proposed IMC-PID, we compare our method with the conventional method. These methods are IMC-Mac and IMC-PID by Jin *et al*. In this simulation, we assume that the controlled plant is equation (13). Unit step input is introduced at t = 0 [s], and disturbance signal of -0.1/s is introduced at t = 20 [s]. Finally, the filter constant λ is set to 1. Fig.2 shows a unit step response and Fig.3 shows bode diagram.

From Fig.2, we confirm that the response of the proposed method is similar to the reference model than that of IMC-Mac and IMC-PID by Jin *et al*. This means that the proposed method is closer to the IMC ideal structure than the conventional methods. Furthermore, From Fig.3, the phase curve of the proposed method is similar to the reference model than that of IMC-Mac and IMC-PID by Jin et al. This means that transfer function of the proposed method system is closer to reference model than that of conventional methods.

$$G(s) = \frac{1}{1+1.5s} e^{-1.5s} \tag{13}$$

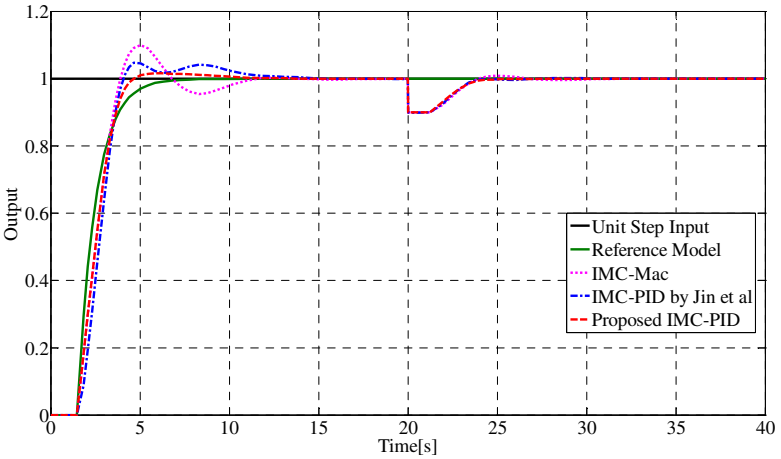


Fig. 2. Step response of Simulation

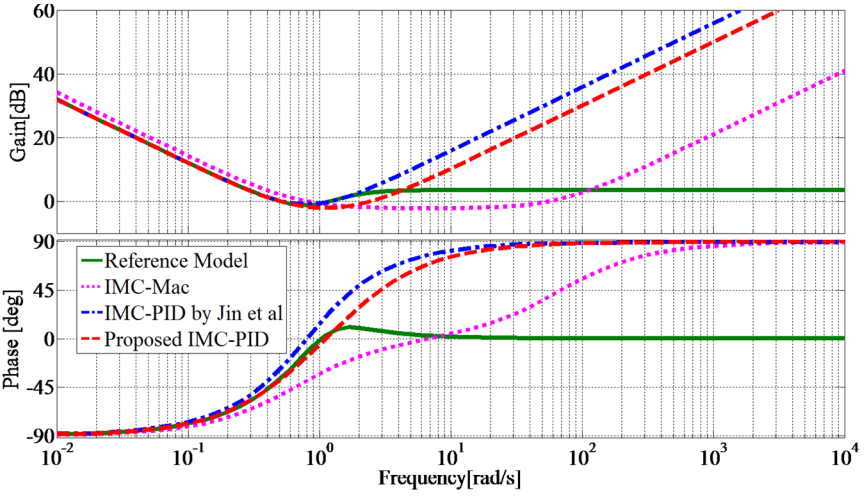


Fig. 3. Bode diagram of Simulation

5 Conclusions

In this paper, we have proposed a method of designing a new IMC-PID based on approximation of polynomial equations. Design of IMC-PID controller has become easy by using algebra calculation, and the general formula of IMC-PID controller has been obtained simple by choosing the filter constant λ. According to simulation results Fig.2, we can confirm that the proposed method has a control performance better than the conventional method. Furthermore, from simulation results Fig.3, our method also shows that it is significantly similar to the reference model in phase curve than other method. That is to say, the proposed method is a design method more similar to IMC principle than conventional methods, moreover, it is a design method which can be easily constructed.

References

1. Rivera, D.E., Morarl, M., Skogestad, S.: Internal model control. 4. PID controller design. Industrial & Engineering Chemistry Research 25, 252–265 (1986)
2. Gong, X., Gao, J., Zhou, C.: Extension of IMC Tuning of PID Controller Parameter. Control and Decision 13(4), 337–341 (1998)
3. Panda, R.C., Yu, C.C., Huang, H.P.: PID tuning rules for SOPDT systems. ISA Transactions 43(2), 283–295 (2004)
4. Lee, Y., Park, S.: PID controller tuning for desired closed-loop responses for SISO systems. AIChE Journal 44(1), 106–115 (1998)
5. Jin, Q., Liu, Q., Wang, Q., Tian, Y., Wang, Y.: PID Controller Design Based on the Time Domain Information of Robust IMC Controller Using Maximum Sensitivity. Chinese Journal of Chemical Engineering 21(5), 529–536 (2013)

Digital Redesign Method for Plants with Input Time Delay

Hikaru Simonmaxwell Karasaki, Hiroki Shibasaki, Ryo Tanaka, Kimikazu Kawaguchi, Hiromitsu Ogawa, and Yoshihisa Ishida

School of Science and Technology, Meiji University,
1-1-1, Higasimita, Tama-ku, Kawasaki, 214-8571 Japan
ce21023@meiji.ac.jp

Abstract. We propose a digital redesign method for plants with input time delay. When a network system is controlled, it is necessary to take into account the time-delay, and to keep the system stable. In this paper, we use the digital redesign and Internal Model Control (IMC). The system is able to compensate input time delay, input side disturbance, and to maintain the stability by combining IMC and input time delay using digital redesign. We confirmed superior performance by simulation study.

Keywords: Digital Redesign, Input Time Delay, Time Delay Compensation Controller, Internal Model Control, Smith compensator.

1 Introduction

The system of the industry is controlled on the network, which is possible to reduce the cost for the wiring, installation, and construct the system in a small space. Furthermore, it is an advantage that by constructing the system on a network, the system maintenance can be performed easily. However, when move the system network, an unavoidable effect of disturbance and unpredictable delays come out, the system becomes unstable. Therefore, there is a need to develop a control method that allows to remove the factors that destabilizes the system.

The purpose of this paper is to design a system that compensates the input time delay using digital redesign [1] [2]. However, steady-state error occurs when we use only the time delay compensation controller designed using the digital redesign. So, we improve performance using the Smith compensator [3] and Internal Model Control [4] [5]. Furthermore, it is possible to maintain higher stability by changing the value of the filter controller of the Internal Model Control (IMC) [4] [5]. We confirmed superior performance by simulation study.

2 Proposed Method

In this section, we describe our proposed method. Our method consists of Input Time Delay [1] [2], IMC [4] [5], and Smith compensator [3]. A block diagram of Input Time Delay Compensation Controller is shown in Fig.1.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 440–444, 2013.

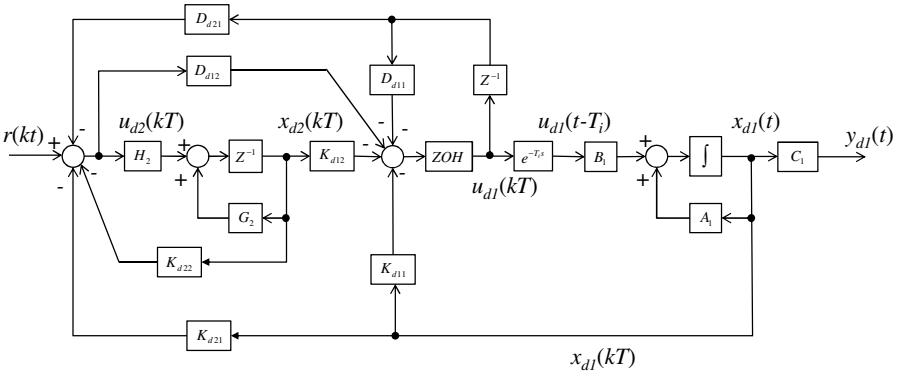


Fig. 1. Block diagram of Input Time Delay Compensation Controller

Input Time Delay Compensation Controller can be written as follows.

$$u_1(t_v) = -K_1 x_1(t_v + T_i) - K_2 x_2(t_v + T_i) \tag{1}$$

The predicted state $x_{d1}(t_v + T_i)$ for the plant input time delay can be written as follows:

$$x_{d1}(t_v + T_i) = G_1^{(v+r)} x_{d1}(kT) + H_1^{(v+r)} u_{d1}(kT - T) + H_0^{(v+r)} u_{d1}(kT) \tag{2}$$

where

$$G_1^{(v+r)} = e^{A_1(v+r)T} \quad , \quad H_1^{(v+r)} = G_1^{(v)}\left[G_1^{(r)} - I_{n1}\right]A_1^{-1}B_1 \quad , \quad H_0^{(v+r)} = \left[G_1^{(r)} - I_{n1}\right]A_1^{-1}B_1$$

The plant with delay-free predicted state $x_{d2}(t_v + rT)$ can be written as follows

$$x_{d2}(t_v + rT) = G_2^{(v+r)} x_{d2}(kT) + H_2^{(v+r)} u_{d2}(kT) \tag{3}$$

Substituting (2), (3) into (1), we obtain the following equation:

$$u_{d1}(kT) = -K_{d11} x_{d1}(kT) - K_{d12} x_{d2}(kT) - D_{d11} u_{d1}(kT - T) - D_{d12} u_{d2}(kT) \tag{4}$$

where

$$K_{d11} = \left(I + K_1 H_0^{(v+r)}\right)^{-1} K_1 G_1^{(v+r)} \quad , \quad K_{d12} = \left(I + K_1 H_0^{(v+r)}\right)^{-1} K_2 G_2^{(v+r)}$$

$$D_{d11} = \left(I + K_1 H_0^{(v+r)}\right)^{-1} K_1 G_1^{(v+r)} \quad , \quad D_{d12} = \left(I + K_1 H_0^{(v+r)}\right)^{-1} K_2 G_2^{(v+r)}$$

A block diagram of the proposed method is shown in Fig.2. The proposed method is designed by using the Input Time delay compensation controller [1] [2], Internal Model Controller (IMC) [4] [5], and Smith compensator [3].

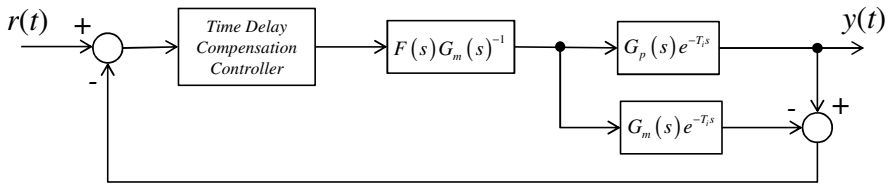


Fig. 2. Block diagram of the proposed method

3 Simulation Results

In this section, we show the simulation of the proposed method. In each example, a unit step set point is introduced at time $t = 0[\mathrm{s}]$, and an input side load disturbance $D_{in}(s) = -0.1/s$ is introduced at time $t = 10[\mathrm{s}]$. The plant is discretized by the zero-order hold method with sampling time $t = 0.01[\mathrm{s}]$.

3.1 Example. 1

Equations (5) and (6) are the plant and IMC filter, respectively. Feedback gains are $K_1 = [1.0836 \quad 7.5403]$, $K_2 = [-1]$.

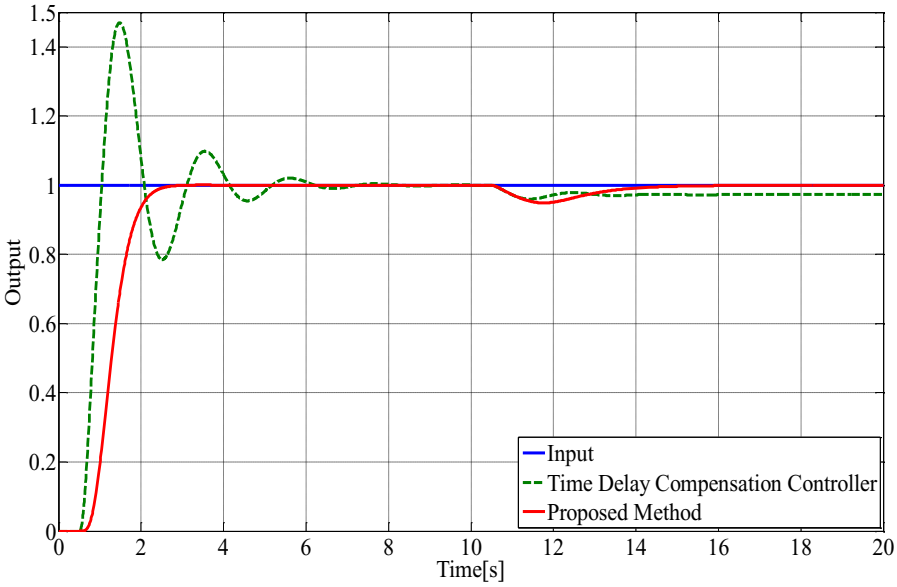


Fig. 3. Simulation Results (Example.3-1)

$$G_p(s)e^{-T_i s} = \frac{0.81}{(0.97s+1)(0.1s+1)}e^{-0.5s} \tag{5}$$

$$F(s) = \frac{1}{0.05s^2 + 0.4s + 1} \tag{6}$$

Fig.3 shows the output response.

Fig.3 is the simulation of the secondary plant with time delay. Convergence time is faster and overshoot is smaller than those of the method controlled only by Time Delay Compensation Controller. In addition, we found that steady-state error did not occur in the proposed method.

3.2 Example. 2

Equations (7) and (8) are the plant and IMC filter, respectively. Feedback gains are $K_1 = [1]$, $K_2 = [-1]$.

$$G_p(s)e^{-T_i s} = \frac{0.79}{0.09s+1}e^{-0.5s} \tag{7}$$

$$F(s) = \frac{1}{0.05s^2 + 0.4s + 1} \tag{8}$$

Fig.4 shows the output response.

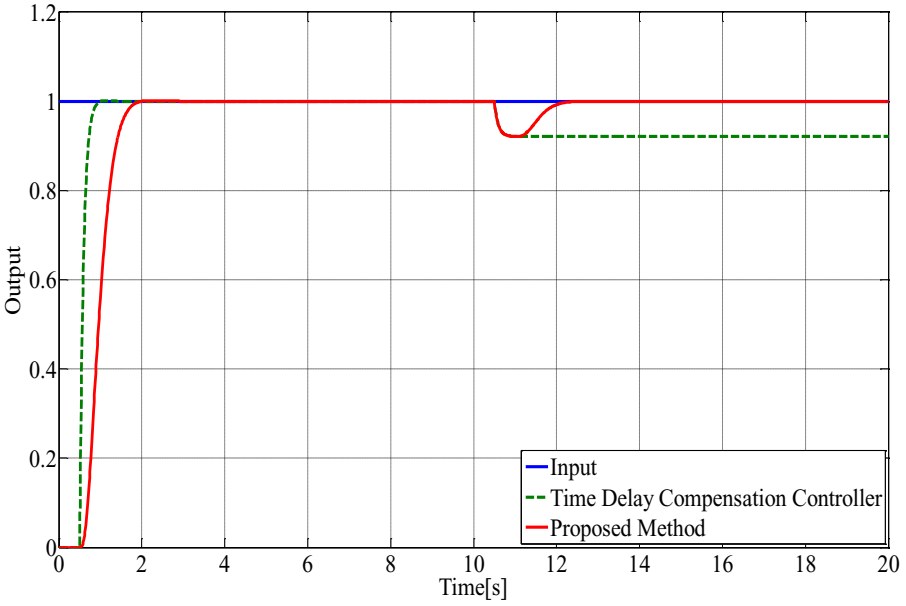


Fig. 4. Simulation Results (Example.3-2)

Fig.4 is the simulation of the plant primary delay system. As shown in simulation results Example.3-2, we have succeeded in eliminating the steady state error that has occurred when it was controlled only by Time Delay Compensation Controller.

The proposed method shows that it has robustness and better tracking.

4 Conclusion

In this paper, we have proposed the digital redesign method for the plant with input time delay. The application of the time delay compensation controller, Smith compensator [3] and IMC [4] [5] makes it possible to eliminate the steady-state error. Furthermore, it is possible to maintain stability by changing the value of the filter in the IMC [4] [5] controller. By applying the proposed method to simulation, the effectiveness of the method has been confirmed.

References

1. Zhang, Y., Akujobi, C.M., Ajuzie, A., Xia, C.: Input Time Delay Compensation with Digital Redesigned A/D Conversion. IEEE Robotics, Automation and Mechatronics, 410–414 (2008)
2. Zhang, Y., Xia, C., Zhang, J., Guo, L.: Inverter Synchronization Control with Internal Model Principle. In: IEEE Electric Machines & Drives Conference (IEMDC), pp. 1265–1269 (2011)
3. Smith, O.J.: A Controller to Overcome Dead Time. ISA Journal 6, 28–33 (1959)
4. Morari, M., Zafiriou, E.: Robust Process Control. Prentice Hall, New Jersey (1989)
5. Shibasaki, H., Endo, J., Hikichi, Y., Tanaka, R., Kawaguchi, K., Ishida, Y.: A Modified Internal Model Control for an Unstable Plant with an Integrator in Continuous-Time System. International Journal of Information and Electronics Engineering 3(4) (July 2013)

DNS–PDF Simulation of Turbulent Mixing in a Reactive Planar Jet

Tomoaki Watanabe[1], Yasuhiko Sakai[2], Kouji Nagata[2], Yasumasa Ito[2], Osamu Terashima[2], and Toshiyuki Hayase[3]

[1] Nagoya University, Nagoya, Japan
Research Fellow of Japan Society for the Promotion of Science
[2] Nagoya University, Nagoya, Japan
[3] Tohoku University, Sendai, Japan
watanabe.tomoaki@nagoya-u.jp

Abstract. Probability density function (PDF) method is implemented in direct numerical simulation (DNS) to simulate turbulent reactive flows (DNS–PDF method). In the DNS–PDF method, a flow field and a non reactive scalar are predicted by the DNS, whereas reactive scalars are predicted by the Lagrangian PDF method, in which a transport equation of joint PDF of reactive scalars is solved by using a large number of notional particles. A mixing time scale for a mixing model used in the PDF method is directly estimated from the DNS result. In the present model for the mixing time scale, the effect of distance between notional particles is implicitly taken into account. The DNS–PDF method is applied to a planar jet with a second-order chemical reaction. The results show that the DNS–PDF method can accurately predict the rms value of mixture fraction fluctuation, and the present model for the mixing time scale is valid. It is also found that the DNS–PDF method can accurately predict mean concentrations of reactive species.

Keywords: Numerical Simulation, Mixing, Chemical Reaction, Turbulent Flow.

1 Introduction

Turbulent mixing of reactive species can be seen in various flows in engineering equipments and the environment. Because the characteristics of turbulent mixing and chemical reactions largely affect the efficiency of engineering equipments, numerical methods for the reactive flows are needed to design engineering equipments and to predict diffusions of pollutants in the environment. Conventional approaches such as moment closure methods [1,2] and large eddy simulations [3,4] face difficulty in modeling chemical source terms when they are applied to the reactive flows [5]. On the other hand, probability density function (PDF) methods [6,7] can strictly treat the chemical source terms without using any models for the chemical source terms. In the Lagrangian PDF method [6], a transport equation of joint PDF of reactive scalars is numerically solved by using a large number of notional particles. The PDF method requires a model for

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 445–452, 2013.

a effect of molecular diffusion (mixing model), in which a mixing time scale has to be specified. The mixing time scale largely affects the numerical results of PDF method. Therefore, the estimation of mixing time scale is important in the PDF method.

In this study, we develop a numerical method to predict reactive flows by combining the PDF method with a direct numerical simulation (DNS). This numerical method is briefly referred to as the DNS–PDF method. In the DNS–PDF method, a flow field and a non reactive scalar are predicted by the DNS, and reactive scalars are predicted by the PDF method. The DNS results of non reactive scalar are used to estimate the mixing time scale, which is required in the mixing model used in the PDF method. The DNS–PDF method is applied to a planar jet with a second-order chemical reaction, and is verified by comparing the results of DNS–PDF method with the DNS results of Watanabe et al [8].

2 DNS–PDF Method

In this section, we describe the DNS–PDF method for reactive flows. In the DNS–PDF method, a flow field and a mixture fraction, which is a non reactive scalar, are predicted by the DNS. In the DNS, an instantaneous velocity U_i and the mixture fraction ξ^* are calculated by numerically solving the continuity equation, the incompressible Navier–Stokes equations, and the transport equation of mixture fraction;

$$\frac{\partial U_j}{\partial x_j} = 0, \tag{1}$$

$$\frac{\partial U_i}{\partial t} + \frac{\partial U_i U_j}{\partial x_j} + \frac{\partial p}{\partial x_i} - \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} = 0, \tag{2}$$

$$\frac{\partial \xi^*}{\partial t} + \frac{\partial U_j \xi^*}{\partial x_j} - D \frac{\partial^2 \xi^*}{\partial x_j \partial x_j} = 0. \tag{3}$$

Here, p is an instantaneous pressure, ν is a kinematic viscosity, and D is a diffusivity coefficient. The Superscript * denotes the scalar quantity calculated by the DNS, and this notation is used to distinguish scalar quantities obtained by the DNS from those by the PDF method.

A joint probability density function $f_{\mathbf{\Phi}}(\mathbf{\Psi}; \mathbf{x}, t)$ of scalar quantities $\mathbf{\Phi} = (\phi_1, ..., \phi_\alpha, ...)$ evolves according to the following equation [9];

$$\frac{\partial f_{\mathbf{\Phi}}}{\partial t} + \frac{\partial}{\partial x_i} \left[\langle U_i \mid \mathbf{\Psi} \rangle f_{\mathbf{\Phi}} \right] = -\frac{\partial}{\partial \psi_\alpha} \left[\left\langle D_\alpha \frac{\partial^2 \phi_\alpha}{\partial x_j \partial x_j} \mid \mathbf{\Psi} \right\rangle f_{\mathbf{\Phi}} + S_\alpha(\mathbf{\Psi}) f_{\mathbf{\Phi}} \right]. \tag{4}$$

Here, $\mathbf{\Psi} = (\psi_1, ..., \psi_\alpha, ...)$ is a sample space vector for $\mathbf{\Phi}$, D_α is a diffusivity coefficient for ϕ_α, S_α is a production rate of ϕ_α by chemical reactions, and $\langle * \mid \mathbf{\Psi} \rangle$ denotes a conditional mean value conditioned on $\mathbf{\Phi} = \mathbf{\Psi}$. In the Lagrangian PDF method, Eq. (4) is numerically solved by using a large number of notional particles. A state of notional particle n is represented by the position $\mathbf{x}^{(n)}$ and

the scalar quantities $\mathbf{\Phi}^{(n)} = (\phi_1^{(n)}, ..., \phi_\alpha^{(n)}, ...)$. The notional particle n evolves according to

$$\frac{\partial \mathbf{x}^{(n)}}{\partial t} = \mathbf{U}(\mathbf{x}^{(n)}, t), \tag{5}$$

$$\frac{\partial \phi_\alpha^{(n)}}{\partial t} = \left[D_\alpha \frac{\partial^2 \phi_\alpha}{\partial x_j \partial x_j} \right]^{(n)} + S_\alpha(\mathbf{\Phi}^{(n)}). \tag{6}$$

The joint PDF $f_{\mathbf{\Phi}}(\mathbf{\Psi}; \mathbf{x}, t)$ is evaluated from notional particles. $\mathbf{U}(\mathbf{x}^{(n)}, t)$ can be obtained from $\mathbf{U}(\mathbf{x}, t)$ predicted by solving Eqs. (1) and (2), and $S_\alpha(\mathbf{\Phi}^{(n)})$ can be calculated without using any models. The first term in Eq. (6) represents the effect of molecular diffusion, and has to be modeled by a mixing model.

One of the simplest mixing models is the Curl's model [10]. In the Curl's model, the effect of molecular diffusion is modeled by the change in the scalar quantities of two selected particles. The Curl's model calculates the change in $\phi_\alpha^{(n)}$ during the time interval dt by the molecular diffusion as follows;

$$\left[d\phi_\alpha^{(n)} \right]_{\text{mix}} = \beta \left[\langle \phi_\alpha \rangle_{(n,m)} - \phi_\alpha^{(n)} \right], \tag{7}$$

$$\left[d\phi_\alpha^{(m)} \right]_{\text{mix}} = \beta \left[\langle \phi_\alpha \rangle_{(m,n)} - \phi_\alpha^{(m)} \right]. \tag{8}$$

Here, $\left[d\phi_\alpha^{(n)} \right]_{\text{mix}}$ is the change in $\phi_\alpha^{(n)}$ by the molecular diffusion modeled by the mixing model, and $\langle * \rangle_{(n,m)}$ denotes the average of two particles n and m. In the Curl's model, $\beta = 1$ with probability P, and $\beta = 0$ with probability $(1 - P)$. The probability P is given by $P = 2dt/\tau_L$, where τ_L is the mixing time scale. τ_L depends on the characteristics of scalar field and the distance between two particles [11]. In this study, the following expression for τ_L is used to implicitly take into account the effect of distance between two particles;

$$\tau_{\text{L}} = \frac{\langle \xi''^2 \rangle_{(n,m)}}{\langle N^* \rangle_{(n,m)}}, \tag{9}$$

$$\xi''^{(n)} \equiv \xi^{*(n)} - \langle \xi^* \rangle_{(n,m)}. \tag{10}$$

Here, $N^* = D(\partial \xi^* / \partial x_j)^2$ is a scalar dissipation rate. $\xi^{*(n)}$ and $N^{*(n)}$ can be obtained by interpolating the DNS results of mixture fraction. According to Cleary et al. [11], the particle which is the closest to the particle n in the reference space $(\mathbf{x}, \xi^*)$ is selected as the mixing particle m. In three-dimensional flows, a distance between the particles n and m in the reference space is defined by

$$d^2_{(n,m)} = \frac{1}{2} \left[\sum_{i=1}^{3} \left(\frac{x_i^{(n)} - x_i^{(m)}}{l_x} \right)^2 + \left(\xi^{*(n)} - \xi^{*(m)} \right)^2 \right]. \tag{11}$$

Here, l_x is a characteristics length scale in the physical space $\mathbf{x}$.

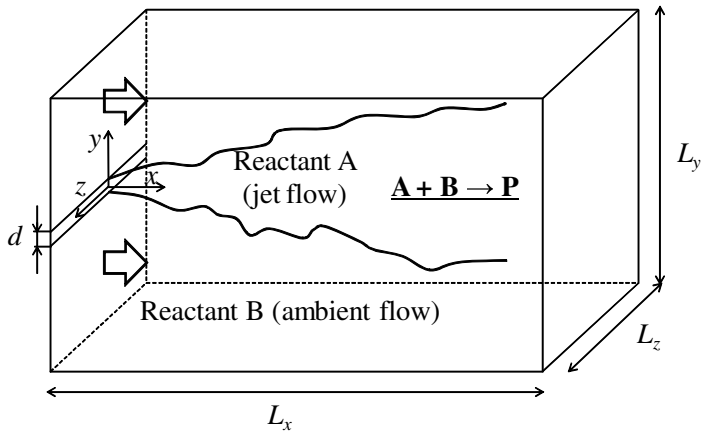


Fig. 1. A planar jet with a second-order chemical reaction

3 DNS–PDF Simulation of Reactive Planar Jet

The DNS–PDF simulation is performed for a planar jet with a second-order chemical reaction A + B → P shown in Fig. 1. The same reactive planar jet was previously investigated using the DNS by the authors [8]. The jet flow containing the reactant A is ejected into the ambient flow containing the other reactant B through the rectangular nozzle with the width of d. The two reactants are mixed in the jet flow, and the product P is produced by the second-order chemical reaction. The ratio of the streamwise mean velocity in the ambient flow, U_{M}, to the streamwise velocity averaged in the jet exit, U_{J}, is $U_{\mathrm{M}}/U_{\mathrm{J}} = 0.056$. The Reynolds number based on U_{J} and d is $Re = U_{\mathrm{J}}d/\nu = 2{,}200$. The Schmidt number ($Sc = \nu/D_\alpha$, $\alpha = \mathrm{A}, \mathrm{B}$, and P) is set to 1. According to the experiment of reactive planar liquid jet [12], the initial concentrations of reactant A and B are set to $\Gamma_{\mathrm{A0}} = 0.4$ mol/m^3 and $\Gamma_{\mathrm{B0}} = 0.2$ mol/m^3, respectively. The mixture fraction ξ is defined by $\xi = (\Gamma_{\mathrm{A}} - \Gamma_{\mathrm{B}} + \Gamma_{\mathrm{B0}})/(\Gamma_{\mathrm{A0}} + \Gamma_{\mathrm{B0}})$, where Γ_α is an instantaneous concentration of species α. The stoichiometric value of mixture fraction is given by $\xi_S = \Gamma_{\mathrm{B0}}/(\Gamma_{\mathrm{A0}} + \Gamma_{\mathrm{B0}})$. The maximum concentration of product P in the stoichiometric condition is $\Gamma_{\mathrm{P0}} = \Gamma_{\mathrm{A0}}\Gamma_{\mathrm{B0}}/(\Gamma_{\mathrm{A0}} + \Gamma_{\mathrm{B0}})$. A production rate of reactive species α by the chemical reaction, S_α, is given by

$$S_{\mathrm{P}} = -S_{\mathrm{A}} = -S_{\mathrm{B}} = k\Gamma_{\mathrm{A}}\Gamma_{\mathrm{B}}. \tag{12}$$

Here, k is a reaction rate constant. The production rate of product P normalized by U_{J}, d, and Γ_{P0} is written by $\hat{S}_{\mathrm{P}} = Da\hat{\Gamma}_{\mathrm{A}}\hat{\Gamma}_{\mathrm{B}}$, where $\hat{\Gamma}_\alpha = \Gamma_\alpha/\Gamma_{\alpha 0}$ and Da is the Damköhler number defined by $Da = k(\Gamma_{\mathrm{A0}} + \Gamma_{\mathrm{B0}})d/U_{\mathrm{J}}$. The DNS–PDF simulation is performed for the three different chemical reactions which satisfy $Da = 0.1$, 1, and 10. The size of computational domain is $L_x = 9.5\pi d$ in the streamwise (x) direction, $L_y = 7.7\pi d$ in the lateral (y) direction, and $L_z = 2.6\pi d$ in the spanwise (z) direction.

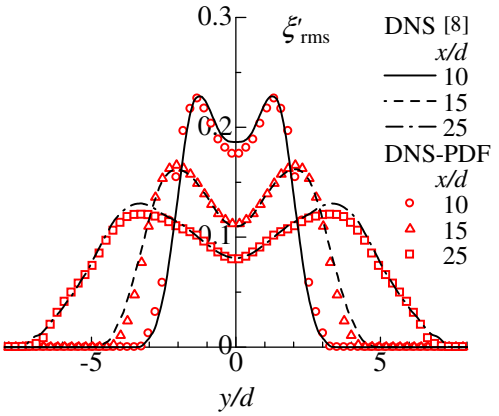


Fig. 2. Lateral profiles of rms value of mixture fraction fluctuation

In the DNS, the fractional step method is used to solve the governing equations which are spatially discretized by using central differences, and the Poisson equation is solved by a conjugate gradient method. For time integrals, the Crank–Nicolson method is used for the viscous term in the y direction, and the third-order Runge–Kutta method is used for the other terms. The convective outflow condition [13] is applied to the y–z plane at $x = L_x$. Random fluctuations are used to generate the inflow velocity which satisfies the mean velocity profile and the variance of streamwise velocity fluctuation measured at the jet nozzle in the experimental apparatus [12]. The free slip boundary condition is applied to the lateral boundaries, and the periodic boundary condition is applied to the homogeneous (z) direction. The computational domain is resolved by $454 \times 250 \times 74$ ($x \times y \times z$) grid points. The equidistant grids are used for the x and z directions. In the y direction, the grid is stretched near the lateral boundaries, and the fine grid is used near the jet centerline.

In the PDF methods, $\mathbf{\Phi} = (\xi, \Gamma_\mathrm{P})$ is calculated by solving Eqs. (5) and (6), and then the instantaneous concentrations of reactants A and B are calculated from a mass conservation law;

$$\Gamma_\mathrm{A} = \Gamma_{\mathrm{A}0}\xi - \Gamma_\mathrm{P}, \tag{13}$$

$$\Gamma_\mathrm{B} = \Gamma_{\mathrm{B}0}(1-\xi) - \Gamma_\mathrm{P}. \tag{14}$$

Equation (6) is solved by using the fractional step methods. 77,760 particles are introduced into the computational domain. The mean distance between the closest particles is $0.42d$, and lies between the Taylor microscale and the Batchelor scale, which is the smallest scale of scalar fluctuation. The effect of molecular diffusion is modeled by the Curl's model. In Eq. (11), the characteristics length scale l_x is set to the nozzle width d. For the notional particles, the inflow boundary condition is applied to the y–z plane at $x = 0$ and the x–z planes at $y = \pm L_y/2$, and the outflow boundary condition is applied to the y–z plane at $x = L_x$. The notional particle which leaves the computational domain across

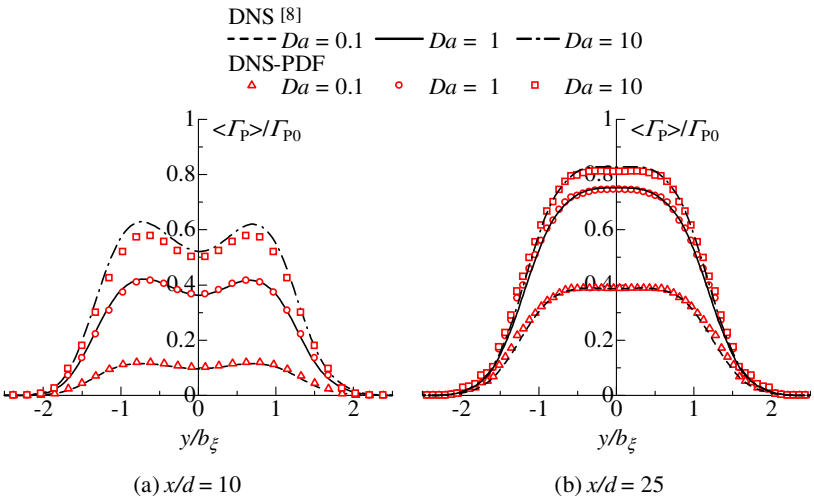


Fig. 3. Lateral profiles of mean concentration of product P at (a) $x/d = 10$ and (b) $x/d = 25$

the inflow and outflow boundaries is replaced at the inflow boundary. In order to keep the local number density of particles statistically constant, the position where the particle is placed is determined according to an instantaneous flow rate normal to the inflow boundary. When the particle is placed at the inflow boundary, $\xi^{(n)} = 1$ and $\Gamma_{\mathrm{P}}^{(n)} = 0$ are imposed for the particle placed in the jet nozzle, and $\xi^{(n)} = 0$ and $\Gamma_{\mathrm{P}}^{(n)} = 0$ are imposed for the particle placed in the ambient flow. The periodic boundary condition is applied for the notional particles in the z direction. To evaluate the statistics of reactive scalars from the notional particles, the computational domain is divided into 100×80 $(x \times y)$ small regions, and the statistics at each region are estimated from the particles in the region.

In DNS of reactive flows, reactive scalar transport equations are solved in DNS as well as the gorverning equations of velocity field. Therefore, the computational cost of DNS is extensive when multiple reactive species are involved in the flow field. In contrast, in the DNS-PDF method, reactive scalar transport equations do not have to be solved in DNS, because the reactive scalars are simulated in the PDF method. The computaitonal cost of PDF method is low compared with the DNS. Using the PDF method instead of DNS of reactive scalar field enables us to reduce the computational costs of numerical simulation of reactive flows.

4 Results and Discussions

In this section, the results of DNS–PDF method are compared with the previous DNS results [8]. Figure 2 shows the lateral profiles of rms value of mixture fraction fluctuation ξ'_{rms}. The profile of ξ'_{rms} calculated by the PDF method

depends on the mixing time scale, and can be used to varify the model for the mixing time scale. In Fig. 2, it is found that ξ'_{rms} obtained by the DNS–PDF method agrees well with the DNS result. This agreement implies that the model for the mixing time scale (Eq. (9)) is valid.

Figure 3 shows that the lateral profiles of mean concentration of product P, $\langle \mathit{\Gamma}_{\mathrm{P}} \rangle / \mathit{\Gamma}_{\mathrm{P0}}$, for $Da = 0.1$, 1, and 10 at $x/d = 10$ and 25. In Fig. 3, the lateral position y is normalized by the half width of mean mixture fraction b_ξ. At $x/d = 10$, the lateral profile of $\langle \mathit{\Gamma}_{\mathrm{P}} \rangle$ has peak values at $y/b_\xi \approx 0.7$, whereas $\langle \mathit{\Gamma}_{\mathrm{P}} \rangle$ does not depend on the lateral position near the jet centerline at $x/d = 25$. It is found that similar profiles of $\langle \mathit{\Gamma}_{\mathrm{P}} \rangle$ can be obtained by the DNS–PDF method. Figure 3 shows that the DNS–PDF method can accurately predict the mean concentrations of reactive species.

5 Conclusions

The DNS–PDF method is developed to simulate turbulent reactive flows. In the DNS–PDF method, a flow field and a non reactive scalar are predicted by the DNS, whereas reactive scalars are predicted by the PDF method. In the DNS–PDF method, the mixing time scale for the mixing model is directly estimated from the DNS result. The DNS–PDF method is applied to the planar jet with a second-order chemical reaction. It is found that the rms value of mixture fraction fluctuation obtained by the DNS-PDF method agrees well with the DNS result. This agreement implies that the models for the mixing time scale used in this study is valid. It is also found that the DNS–PDF method can accurately predicts mean concentrations of reactive species.

Acknowledgments. Part of the work was carried out under the Collaborative Research Project of the Institute of Fluid Science, Tohoku University. This work was supported by JSPS KAKENHI Grant Number 25002531 and MEXT KAKENHI Grant Numbers 25289030, 25289031, and 2563005.

References

1. Wang, D.M., Tarbell, J.M.: Closure Models for Turbulent Reacting Flows with a Nonhomogeneous Concentration Field. Chem. Eng. Sci. 48, 3907–3920 (1993)
2. Veynante, D., Vervisch, L.: Turbulent Combustion Modeling. Prog. Energy Combust. Sci. 28, 193–266 (2002)
3. Schumann, U.: Large-eddy Simulation of Turbulent Diffusion with Chemical Reactions in the Convective Boundary Layer. Atmos. Environ. 23, 1713–1727 (1967)
4. Michioka, T., Komori, S.: Large-eddy Simulation of a Turbulent Reacting Liquid Flow. AIChE J. 50, 2705–2720 (2004)
5. Fox, R.O.: Computational Models for Turbulent Reacting Flows. Cambridge Univ. Pr. (2003)
6. Pope, S.B.: PDF Methods for Turbulent Reactive Flows. Prog. Energy Combust. Sci. 11, 119–192 (1985)

7. Haworth, D.C.: Progress in Probability Density Function Methods for Turbulent Reacting Flows. Prog. Energy Combust. Sci. 36, 168–259 (2010)
8. Watanabe, T., Sakai, Y., Nagata, K., Terashima, O., Suzuki, H., Hayase, T., Ito, Y.: Visualization of Turbulent Reactive Jet by Using Direct Numerical Simulation. Int. J. Model. Simul. Sci. Comput. 4, 1341001 (2013)
9. Pope, S.B.: The Probability Approach to the Modelling of Turbulent Reacting Flows. Combust. Flame 27, 299–312 (1976)
10. Curl, R.L.: Dispersed Phase Mixing: I. Theory and Effects in Simple Reactors. AIChE J. 9, 175–181 (1963)
11. Cleary, M.J., Klimenko, A.Y.: A Generalised Multiple Mapping Conditioning Approach for Turbulent Combustion. Flow Turbul. and Combust. 82, 477–491 (2009)
12. Watanabe, T., Sakai, Y., Nagata, K., Terashima, O., Kubo, T.: Simultaneous Measurements of Reactive Scalar and Velocity in a Planar Liquid Jet with a Second-order Chemical Reaction. Exp. Fluids 53, 1369–1383 (2012)
13. Dai, Y., Kobayashi, T., Taniguchi, N.: Large Eddy Simulation of Plane Turbulent Jet Flow Using a New Outflow Velocity Boundary Condition. JSME Int. J. 37, 242–253 (1994)

On the Wave Spectrum Selection in Ocean Wave Scene Simulation of the Maritime Simulator

Li-ning Chen, Yi-cheng Jin, Yong Yin, and Hong-xiang Ren

Key Laboratory of Marine Dynamic Simulation & Control for Ministry of Communications, Dalian Maritime University, Dalian, China
lnchen1981@163.com

Abstract. The Phillips spectrum is widely used in the real-time scene simulation of ocean waves. Structural analysis has shown the Phillips spectrum to be an instantaneous spatial spectrum. Its frequency spectrum agrees with the spectrum form raised by Neumann, and is close to the P-M spectrum. Its directional distribution function is in the form recommended by the International Towing Tank Conference. However, the Phillips spectrum has two problems in application. First, neither the value nor the calculation method of the spectral constant A is given. Second, the height above ocean surface of the spectral wind speed is not provided. The spectral constant A is calculated on the hypothesis that the wave energy per unit area of the Phillips spectrum equals that of the P-M spectrum, and the height is specified with reference to the P-M spectrum. Apart from the Phillips spectrum, other spectrum should be considered, so we use the JONSWAP-Poisson spectrum. The spectrum has the JONSWAP spectrum as its frequency spectrum, and its distribution function is the Poisson form. The two spectra are applied for ocean wave scene simulation. A comparison of the simulation results of the two spectra shows that wind speed and fetch length can affect the shape of the generated ocean wave. The simulated ocean waves of the Phillips spectrum can reflect the wind speed and direction influence on the wave, but they cannot reflect the fetch length effect. The simulated waves of the JONSWAP-Poisson spectrum can embody the effect of the wind speed, wind direction, and fetch length on the wave. The frame rates of the two spectra are equal. To sum up, for wave spectrum selection of wave scene simulation in the maritime simulator, the wind parameter of the spectrum should contain the wind speed, wind direction, and fetch length, and the wave generated by the spectrum should be in $(-\pi,\pi]$ of the wind direction. Thus, the JONSWAP-Poisson spectrum is more suitable than the Phillips spectrum. The method has been used in the maritime simulator.

Keywords: ocean wave scene simulation, ocean wave spectrum, directional distribution function, wind speed, fetch length.

1 Preface

The real-time scene simulation of the ocean is widely applied in the field of the computer graphics and virtual reality. The simulation of ocean wave scene is one of hot topics of the computer graphics.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 453–465, 2013.

Mihalef et al. suggest that there are three scales of ocean wave scene simulation: the small scale with spray and foam, the middle scale with water ripple, and the large scale with rolling-over and breaking wave [1]. The ocean wave can be described with Computational Fluid Dynamics (CFD) model or the surface wave model. So we classify the scene simulation of ocean waves into two types, the scene simulation with CFD model and that with the surface wave model.

When CFD method is used for simulation, the liquid viscidity is considered, and it needs to solve the governing equations of the viscid liquid [2]. The governing equations include the Navier-Stokes (N-S) equations [3-5] and Lattice Boltzmann Method (LBM) shallow water equations [6]. By solving the equations, the properties of fluids, such as density, pressure, temperature etc, can be calculated. Additionally, the interaction and flowing law between the fluid and its adjacent solid can be obtained. So such simulation is fit for free-surface, mixed-flow, multiphase-flow, and fluid-solid coupling. However, the governing equation solving consumes a variety of the computer resource. Thus the simulation fits the small and middle scale ocean wave. For the large scale scene, the real-time capability is not good.

For another simulation type, the ocean wave is deemed as the surface wave between the sea water and air. This way discriminates the ocean wave into a variety of component waves. Compared with the former type, its calculation amount is smaller, and with better real-time capability. It fits large scale ocean waves. This type of the scene simulation contains two methods. One is the method based on the regular wave [7-10], and the other one is the method based on the irregular [11-14]. The method based on the regular wave mainly applies the Gerstner wave to describe the wave motion, and the generated wave is regular. For the method based on the irregular wave, the wave is generated by 2-dimension (2D) Inverse Fast Fourier Transform (IFFT) for a random spectral function. The generated wave is irregular. Both methods use the wave spectrum, which makes the generated wave can reflect the wind influence on the ocean surface.

The proper method of simulating the ocean wave scene is decided by the application object and purpose. In this paper, the simulated ocean wave is applied in the maritime simulator. The maritime simulator is used for the training of seafarers [15,16]. In the training, the seafarer observes the simulated ocean wave scene in the simulator and infers the wind attributes. Based on this inference, the seafarer takes the appropriate ship-handling method. The method used in the maritime simulator meets the following requirements: (1) a large-scale ocean wave scene is simulated, and the scene simulation is real-time; and (2) the simulation embodies the wind effect on wave. A large-scale ocean wave scene indicates that the simulated sea area is large (sometimes several square kilometers). The word “real-time” means that the scene simulation has a certain degree of frame rate, usually above 30f/s [16]. The wind effect on wave is mainly manifested in the wave height, the propagation direction, and the wavelength. In addition to the wind speed and direction, the fetch length also affects the wave. The fetch length, also called the fetch, is the length of water over which a given wind has blown. According to the No. 2.14 standard of Det Norske Veritas (DNV) [16], “the simulator shall provide at least two different wave spectra, variable in direction, height and period” for class A and B maritime simulator.

Therefore, the scene simulation with the surface wave model is more appreciate for the maritime simulator. The actual ocean wave can be deemed as a random process [17]. Thus in the scene simulation with the surface wave model, the method based on the irregular wave is more close to the actual condition than the method based on the regular wave. So the method based on the irregular wave is adopted in this paper.

2 Algorithm Introduction

In the world coordinate xyz, the plane xy parallels with ocean surface with the axis z perpendicular to the ocean surface. $\mathbf{x} = (x, y)$. And ω is the circular frequency. For the 2D ocean surface, if λ is the wavelength, the wave number $k = 2\pi / \lambda$. Further, for the 3D ocean surface, the wave number vector $\mathbf{k} = (k_x, k_y)$, k_x is the component of $\mathbf{k}$ on x axis and k_y is that of $\mathbf{k}$ on y axis. Let θ the angle between $\mathbf{k}$ and the positive x axis, and $k = |\mathbf{k}|$.. Let the wind speed vector $\mathbf{u} = (u_x, u_y) = (U\cos\alpha, U\sin\alpha)$. U is the wind speed, and α is the angle between the wind speed and the positive x axis. In oceanography and navigation, the wind direction is the direction from which it originates [18]. Let α' the wind direction. The relation between α' and α is as:

$$\alpha' = \begin{cases} \alpha + \pi, & 0 \le \alpha < \pi \\ \alpha - \pi, & \pi \le \alpha < 2\pi. \end{cases}$$

Thus the height field is expressed as:

$$\xi(\mathbf{x}, t) = \mathrm{Re}[\sum_{kx} \sum_{ky} \tilde{\xi}(\mathbf{k}, t) \exp(\mathrm{i}\mathbf{k} \cdot \mathbf{x})]. \tag{1}$$

where Re is to get the real part; $\tilde{\xi}(\mathbf{k}, t)$ is the random spectral function which decides the structure of the ocean surface; i is the imaginary unit; if $M \times N$ points are sampled on the area $L_x \times L_y$ of the ocean surface, $k_x = 2\pi m / L_x$, $k_y = 2\pi n / L_y$, $\mathbf{x} = (x, y) = (mL_x / M, nL_y / N)$, both m and n are positive integers, and $-M/2 \le m < M/2$, $-N/2 \le n < N/2$. Expression (2) is to do 2D IFFT for $\tilde{\xi}(\mathbf{k}, t)$ to calculate $\xi(\mathbf{x}, t)$. By IFFT, the height field is transformed from the $\mathbf{k}$ domain to the $\mathbf{x}$ domain. $\tilde{\xi}(\mathbf{k}, t)$ is expressed as:

$$\tilde{\xi}(\mathbf{k}, t) = \tilde{\xi}_0(\mathbf{k}) \exp[\mathrm{i}\omega(\mathbf{k})t] + \tilde{\xi}_0^*(-\mathbf{k}) \exp[-\mathrm{i}\omega(\mathbf{k})t]. \tag{2}$$

where $*$ means to get the conjugate complex. According to the dispersion relation, in deep water, the relation between the wave number and circular frequency is as:

$$\omega^2(k) = \mathrm{g}k,\ k(\omega) = \omega^2 / \mathrm{g}. \tag{3}$$

$\tilde{\xi}_0(\mathbf{k})$ is expressed as:

$$\tilde{\xi}_0(\mathbf{k}) = (\varepsilon_r + \mathrm{i}\varepsilon_i)\sqrt{\Psi(\mathbf{k}) / 2}. \tag{4}$$

where ε_r and ε_i are independent Gaussian random with mean value 1 and variance 0; $\Psi(\mathbf{k})$ is the instantaneous spatial spectrum that will be introduced in Section 3. The Gaussian random is applied in Expression (4), so under the same initial condition, the generated ocean wave of each time is not repeated, and the generated wave cannot be predicted. In other words, the wave is irregular. The mostly common used $\Psi(\mathbf{k})$ is the Phillips spectrum, denoted as $\Psi_{Ph}(\mathbf{k})$. $\Psi_{Ph}(\mathbf{k})$ is expressed as:

$$\Psi_{Ph}(\mathbf{k}) = \Psi_{Ph}(k,\theta) = Ak^{-4} \exp\left[-(kL)^{-2}\right] \cos^2(\theta - \alpha). \tag{5}$$

where A is a constant and $L = U^2 / \mathrm{g}$.

If the ocean height field is obtained merely by the above-mentioned steps, then the shape of the generated ocean surface would be geometrically symmetric. Even in good weather, however, actual ocean waves are asymmetric, which may be sharply peaked at their tops and flattened at the bottoms. To achieve that effect, choppy wave is introduced. The choppy wave model originates from the visual effect of the linear wave. It is analytically tractable, numerically efficient, and robust [19]. $\tilde{h}(\mathbf{k},t)$ is used to generate a 2D displacement vector of choppy wave, expressed as:

$$\mathbf{D}(\mathbf{x},t) = \mathrm{Im}[\sum_{kx}\sum_{ky} \frac{\mathbf{k}}{k} \tilde{\xi}(\mathbf{k},t) \exp(\mathbf{k}\cdot\mathbf{x})]. \tag{6}$$

where Im is to get the imaginary part. The horizontal position of a wave grid point is $\mathbf{x} + \lambda\mathbf{D}(\mathbf{x},t)$, where λ is a constant and a convenient method of scaling the importance of the displacement vector. Choppy wave does not affect wave heights directly but warps the horizontal positions of surface point according to the spatial structure of the height field. In this way, the peak is sharpened and the valley is broadened, making the simulation more realistic.

Although the Phillips spectrum has been widely used in ocean wave scene simulation [11-14], some problems still exist in its application. Firstly, neither the value nor the computing method of the spectral constant A is clear. Secondly, the

height above ocean surface of the spectral wind speed is not provided. Wind speed changes as the height over sea varies, so the height above the ocean surface of the spectral wind speed is usually specified. For example, the height above the ocean surface is 7.5 m for the Neumann spectrum wind speed, 19.5 m for the P-M spectrum, and 10 m for the JONSWAP spectrum.

3 Phillips Spectrum Structure

Let $\xi(\mathbf{x},t)$ be the surface displacement. The symbol $\overline{\ }$ is to get the mean value, ρ is the density of the sea, g is the gravity acceleration, and $\rho \mathrm{g}\overline{\xi^2}/2$ is the average potential of the ocean wave per unit area. So $\overline{\xi^2}$ can represent the ocean wave energy, and be expressed as:

$$\overline{\xi^2} = \int_{\omega} S(\omega)\mathrm{d}\omega = \iint_{\mathbf{k}} \Psi(\mathbf{k})\mathrm{d}\mathbf{k} = \int_{\omega}\int_{\theta} E(\omega,\theta)\mathrm{d}\omega\mathrm{d}\theta.$$

where $S(\omega)$ is the frequency spectrum, $\Psi(\mathbf{k})$ is the instantaneous spatial spectrum, and $E(\omega,\theta)$ is the directional spectrum [17]. $S(\omega)$, $\Psi(\mathbf{k})$ and $E(\omega,\theta)$ are density functions of ocean wave energy. We notice that the Phillips spectrum is an instantaneous spatial spectrum. The relation of $S(\omega)$, $\Psi(\mathbf{k})$ and $E(\omega,\theta)$ can be expressed as:

$$E(\omega,\theta) = S(\omega)D(\theta) = \Psi\left[k(\omega),\theta\right]k(\omega)\mathrm{d}k(\omega)/\mathrm{d}\omega.$$

where $D(\theta)$ is the directional distribution function.. It can be seen that for the majority of $D(\theta)$, $\theta-\alpha\in(-\pi,\pi]$, and only in a few $D(\theta)$, such as $\cos-2n$ form, is $\theta-\alpha\in[-\pi/2,\pi/2]$ [17]. Let r be the integer and $r\geq 0$, m_r the r th order spectral moment of $S(\omega)$, m_r is expressed as:

$$m_r[S(\omega)] = \int_{\omega} \omega^r S(\omega)\mathrm{d}\omega.$$

Particularly, $m_0[S(\omega)] = \overline{\xi^2}$. The spectral width, denoted as υ, is calculated as $\upsilon = (m_0 m_2 / m_1^2 - 1)^{1/2}$ and $\upsilon\in(0,1)$. It represents the spectrum frequency range and energy dispersion.

Let $E_{Ph}(\omega,\theta)$, $D_{Ph}(\theta)$, and $S_{Ph}(\omega)$ be the directional spectrum, frequency spectrum, and directional distribution function of $\Psi_{Ph}(\mathbf{k})$ respectively. The following results can then be achieved:

$$\begin{aligned} &E_{Ph}(\omega,\theta)=S_{Ph}(\omega)D_{Ph}(\theta),\\ &S_{Ph}(\omega)=A\pi \mathrm{g}^2\omega^{-5}\exp\left(-\mathrm{g}^4\omega^{-4}U^{-4}\right),\\ &D_{Ph}(\theta)=\begin{cases}(2/\pi)\cos^2(\theta-\alpha), & \text{if } \theta-\alpha\in[-\pi/2,\pi/2]\\ 0, & \text{other}\end{cases}. \end{aligned} \tag{7}$$

So $D_{Ph}(\theta)$ is the $\cos-2n$ form distribution with $n=1$ [17], and is the directional spreading function recommended by the International Towing Tank Conference (ITTC).

The P-M spectrum is commonly used, and its expression is:

$$S_{PM}(\omega)=\alpha \mathrm{g}^2\omega^{-5}\exp\left\{-\beta\left[\mathrm{g}(U_{19.5}\omega)^{-1}\right]^4\right\}.$$

where both α and β are constants, and $\alpha=8.1\times10^{-3}$ and $\beta=0.74$. Furthermore, $U_{19.5}$ is the wind speed at the height of 19.5 m above the ocean surface. When $S_{Ph}(\omega)$ and $S_{PM}(\omega)$ are compared, some similarities are found, as follows:

(1) The two spectra coincide with the spectrum formulation proposed by Neumann. The Neumann spectrum formulation is

$$S(\omega)=A\omega^{-p}\exp\left(-B\omega^{-q}\right).$$

where A and B contain parameters related to wind or wave. So far, many wave spectra are in accordance with the Neumann form. For $S_{Ph}(\omega)$, $p=5$ and $q=4$; for $S_{PM}(\omega)$, $p=5$ and $q=4$.

(2) Wind speed is the only parameter of wind, and neither the fetch length nor the wind duration is included.

(3) The two spectra have the same spectral width. We can obtain $\upsilon_{Ph}=\upsilon_{PM}=0.42$, so the Phillips and P-M spectra have the same energy dispersion.

$\overline{\xi^2}$ of the Phillips and P-M spectra can be expressed as:

$$\overline{\xi^2}_{Ph}=A\pi U^4/(4\mathrm{g}^2),\ \overline{\xi^2}_{PM}=2.84\times10^{-5}U_{19.5}^4.$$

Because the S_{Ph} and the P-M spectra are capable of having much in common, it is assumed that the wind speed of the Phillips spectrum is the same as that of the P-M spectrum, namely $U_{19.5}$. Moreover, the P-M spectrum is gained through long-time observation, and its data are fairly sufficient. It is easy to use, and it can be directly integrated. For these reasons, the P-M spectrum is referred to in computing the constant A. Let the Phillips spectrum have the same wave energy of component waves per unit area as the P-M spectrum, that is, $\overline{\xi^2}_{Ph} = \overline{\xi^2}_{PM}$. We then can get $A = 3.48 \times 10^{-3}$.

If $\overline{\xi^2}_{Ph} = \overline{\xi^2}_{PM}$, obviously, two conclusions can be drawn, as follows:

(1) The Phillips spectrum and the P-M spectrum are capable of having the same mean of the largest part of wave heights. The mean is denoted by $H_{1/a}$ ($a > 1$). $H_{1/a}$ is the mean of the $1/a$ largest of the wave height observed in a wave field. $H_{1/3}$ (when $a = 3$) and $H_{1/10}$ (when $a = 10$) are the most commonly used, and , $H_{1/3}$ is called the significant wave height. $H_{1/a}$ can reflect the important characteristics of the ocean wave, and is given attention in the fields of navigation and port engineering. It can be calculated as

$$H_{1/a} = (8m_0 \ln a)^{1/2} + a(2\pi m_0)^{1/2}\{1 - \mathrm{erf}[(\ln a)^{1/2}]\}.$$

where $\mathrm{erf(t)}=1.1283\int_0^t e^{-t^2} \mathrm{d}\, t$. Therefore, it can be concluded from this expression that the values of $H_{1/a}$ of the two spectra are equal when $\overline{\xi^2}_{Ph} = \overline{\xi^2}_{PM}$.

(2) The values of the mean wave height of both the Phillips spectrum and the P-M spectrum are equal. This conclusion can be obtained from the formula of mean wave height as follows:

$$\bar{H} = H_{1/a}\left\{[(4/\pi)\ln a]^{1/2} + a - a\mathrm{erf}[(\ln a)^{1/2}\right\}^{-1}..$$

When the two spectra share the same value of $H_{1/a}$, the $\bar{H}$ of the two spectra are equal.

If the two spectra adopt the same value of $U_{19.5}$, then m_0, $H_{1/a}$, and $\bar{H}$ of the two spectra are equal. Thus, it is reasonable to assume that the wind speed of the Phillips spectrum is $U_{19.5}$, that is, the wind speed at 19.5 m above the sea surface. As the wind speed $U_{19.5}$ rises, m_0, $H_{1/a}$, and $\bar{H}$ increase, which means that the ocean wave gets choppier as the wind speed rises. This is in accordance with the actual condition.

Figure 1 shows the simulated ocean wave of the Phillips spectrum.

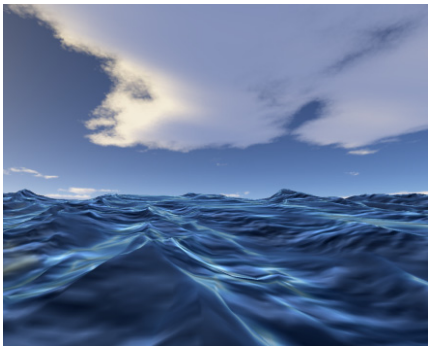


Figure 1(a) $U_{19.5} = 6.0m/s$, $\alpha' = 180^{\circ}$ Figure 1(b) $U_{19.5} = 9.6m/s$, $\alpha' = 180^{\circ}$

Fig. 1. Simulated ocean wave of the Phillips spectrum

4 Simulating with JONSWAP-Poisson Spectrum

The wind parameter of the Phillips spectrum only contains the wind speed, and does not contain the fetch length. The ocean wave observation shows that the fetch length affects the ocean wave shape along with the wind speed. Therefore, another instantaneous spatial spectrum the JONSWAP-Poisson spectrum, denoted as $\Psi_{J-Po}(\mathbf{k})$, is produced. The frequency spectrum of $\Psi_{J-Po}(\mathbf{k})$ is the JONSWAP spectrum, and the wind parameter of the JONSWAP spectrum contains the wind speed and fetch length. The directional distribution function of $\Psi_{J-Po}(\mathbf{k})$ is the Poisson form. $\Psi_{J-Po}(\mathbf{k})$ is expressed as:

$$\Psi_{J\text{-}Po}(\mathbf{k}) = \Psi_{J-Po}(k,\theta) = \frac{\alpha}{4\pi}\frac{1}{k^4}\exp\left(-\frac{5}{4}\frac{\omega_0^4}{\mathrm{g}^2k^2}\right) \times \gamma^{\exp\left[-(\sqrt{\mathrm{g}k}-\omega_0)^2/(2\sigma^2\omega_0^2)\right]}(1-x^2)[1-2x\cos(\theta-\alpha)+x^2]^{-1}. \tag{8}$$

where $\alpha = 0.076\left(\overline{X}\right)^{-0.22}$, $\overline{X}$ is the dimensionless fetch, $\overline{X} = \mathrm{g}X/U_{10}$, X is the fetch, U_{10} is the wind speed at the height of 10 m above the ocean surface, ω_0 is the spectral peak frequency, $\omega_0 = 22\left(\mathrm{g}/U_{10}\right)\left(\overline{X}\right)^{-0.33}$ (when $\omega \le \omega_0$, $\sigma = 0.07$, and when $\omega > \omega_0$, $\sigma = 0.09$), γ is the peak-shape parameter,

$\gamma \in [1,7]$ with average 3.3, $\theta - \alpha \in (-\pi, \pi]$ and $x \in (0,1)$. Further, $\overline{\xi^2}_{J-Po}$ is as:

$$\overline{\xi^2}_{J-Po} = \left\{5.42\times10^{-9} X^{1.1} \gamma / \ln[(1+\gamma)/1.043]\right\} \times U_{10}^{1.8}. \quad (9)$$

Thus, the wind parameter of $\Psi_{J-Po}(\mathbf{k})$ includes the wind speed and fetch length. The expression (9) shows that for the JONSWAP-Poisson spectrum, when γ and the wind speed are fixed, the wave energy per unit area, $\overline{\xi^2}_{J-Po}$, rises as the fetch length X increases. Figure 2 shows the simulated ocean wave of $\Psi_{J-Po}(\mathbf{k})$.

Figure 2(a) $X = 50km$

Figure 2(b) $X = 100km$

Fig. 2. Simulated ocean wave of the JONSWAP-Poisson spectrum with $x = 0.6$, $U_{10} = 8.0m/s$, $\alpha' = 180^{\circ}$ and $\gamma = 3.3$

5 Simulation Result Comparison

The wind speed of the Phillips spectrum and P-M spectrum is $U_{19.5}$ and that of the JONSWAP spectrum is U_{10}; hence, to facilitate the comparison, $U_{19.5}$ and U_{10} should be interchanged. In engineering, the following expression can be used for interchanging: $U_{19.5}/U_{10} = (19.5/10.0)^{a}$, where a is the Hellman exponent, and for stable air above open water surface, $a = 0.27$. Therefore, the relation between $U_{19.5}$ and U_{10} is $U_{19.5}/U_{10} = 1.20$.

Thus, $m_0[S_J(\omega)]$ and $\overline{\xi^2}_J$ can be expressed with $U_{19.5}$ as:

$$m_0[S_J(\omega)] = \overline{\xi^2}_J = \left\{3.76\times10^{-9} X^{1.1} \gamma / \ln[(1+\gamma)/1.043]\right\} U_{19.5}^{1.8}.$$

Let $\omega_{0,Ph}$, $\omega_{0,PM}$, and $\omega_{0,J}$ be the peak frequency for the Phillips, P-M, and JONSWAP spectra, respectively. They can be expressed as:

$$\omega_{0,Ph} = 0.95\mathrm{g}U_{19.5}^{-1},\ \omega_{0,PM} = 0.88\mathrm{g}U_{19.5}^{-1},\ \omega_{0,J} = 23.41\mathrm{g}^{0.67}X^{-0.33}U_{19.5}^{-0.34}.$$

As mentioned earlier, $\upsilon_{Ph} = \upsilon_{PM} = 0.42$. The spectral width of the JONSWAP spectrum υ_J is calculated as:

$$\upsilon_J = \left[(5+\gamma)(10.89+\gamma)(6.77+\gamma)^{-2} - 1\right]^{1/2}.$$

If $\gamma = 3.3$, $\upsilon_J = 0.40$, υ_J approaches υ_{Ph} and υ_{PM}, and the energy accumulation degrees of the three spectra are very close. As the υ of the three spectra are close, m_0 and ω_0 affect the shape of the produced ocean wave. The $\overline{\xi^2}$ and ω_0 of the Phillips and P-M spectra are affected by the wind speed, and not by the fetch length; however, $\overline{\xi^2}$ and ω_0 of $\Psi_{J-Po}(\mathbf{k})$ are impacted by both the wind speed and the fetch length.

The wind speed presented in Figure 1(b) is higher than that given in Figure 1(a), and as the $\overline{\xi^2}$ of Figure 1(b) is bigger, so the wave of Figure 1(b) is bigger than that of Figure 1(a). Additionally, The wave of Figure 1(b) looks “longer” than that of Figure 1(a). This is caused by the peak frequency. We notice that $\omega_{0,Ph}$ varies inversely as the wind speed $U_{19.5}$. The wind speed in Figure 1(b) is larger than that of Figure 1(a), and $\omega_{0,Ph}$ in Figure 1(b) is smaller than that of Figure 1(a). This makes short-length waves (high frequencies) become more obvious in Figure 1(b), and the wave in Figure 1(b) is “longer” visually. In Figure 2, we notice that when the wind speed is constant, the larger the fetch length is, the bigger the generated wave is. This coincides with the expression (9). According to the relation between $U_{19.5}$ and U_{10}, we notice that the wind speed of Figure 1(b) equals to that of Figure 2, but $\overline{\xi^2}$ of Figure 1(b) is bigger than that of Figure 2, so the wave size of Figure 1(b) is larger than that of Figure 2.

The authors tested the frame rate using the two spectra. The test is carried out on a PC with medium configuration as follows: CPU Intel E7400, internal storage 2GB, and graphics card NVIDIA GeForce GT 430. When $M = N = 256$, the frame rates of the two spectra are equal, 61f/s. At $M = N = 512$, the frame rates of the two are still equal, but decreased to 26f/s. The test shows that the values of M and N are the leading cause of the frame rate. The bigger the M and N, the more the calculation amount of IFFT and the lower the frame rate. To sum up, under the

condition of the same computer hardware, if M and N are fixed, changing the ocean wave spectrum cannot affect the frame rate.

In the author's opinion, as to spectra selection for ocean wave scene simulation in the maritime simulator, two aspects need be considered. First, the wind parameter of the spectrum should include the wind speed, wind direction, and fetch length, so that the simulation result can show the fetch length influence on the ocean wave. Second, the generated wave of most $D(\theta)$ distributes $(-\pi,\pi]$ of the wind direction, and only a few with $[-\pi/2,\pi/2]$ [17], so $D(\theta)$ with $\theta\text{-}\alpha\in(-\pi,\pi]$ is recommended. Therefore, compared with the Phillips spectrum, $\Psi_{J-Po}(\mathbf{k})$ meets the two aspects and is more comfortable.

6 Conclusions

The problems existing in the application of the Phillips spectrum widely used in ocean wave scene simulation are indicated here. First, neither the value nor the calculation principle of the spectrum constant A is given. Second, for the spectral wind speed, its height above the ocean surface is not clarified. The two problems can affect whether the result of the scene simulation can accurately show the influence of wind. If the Phillips spectrum is selected for ocean wave simulation, the two problems need to be settled.

Thus, the Phillips spectrum is analyzed first. We found the Phillips spectrum to be an instantaneous spatial spectrum. Its frequency spectrum $S_{Ph}(\omega)$ is in accordance with Neumann spectrum formulation, with $p=5$, $q=4$, and its directional distribution function $D_{Ph}(\theta)$ is the type recommended by the ITTC. The wind parameter of $S_{Ph}(\omega)$ only includes the wind speed, which is similar to that of the P-M spectrum, and the two spectra have the same spectral width. Thus, by assuming that the wave energy per unit area of the Phillips spectrum is equal to that of the P-M spectrum, the author calculates the constant A of the Phillips spectrum, and clarifies the height over ocean surface of the spectral wind speed in reference with the P-M spectrum.

In addition to the Phillips spectrum, the JONSWAP-Poisson spectrum, $\Psi_{J-Po}(\mathbf{k})$, is employed for ocean wave scene simulation, and the simulation results of the used spectra are compared to select which kind of the ocean wave spectrum is proper for the maritime simulator. The frequency spectrum for $\Psi_{J-Po}(\mathbf{k})$ is the JONSWAP spectrum, and its directional distribution function is the Poisson form.

The simulated waves of the two spectra are compared. The simulation results of the Phillips spectrum can reflect the influence of wind speed and direction on ocean wave, but cannot show the effect of fetch length. However, when $\Psi_{J-Po}(\mathbf{k})$ is used, the simulated wave can embody the effect of wind speed, wind direction, and

fetch length. Additionally, under the same condition, the frame rates of the two are the same. Under the same computer hardware condition, if M and N are fixed, the frame rates of the two spectra are equal.

Based on the above discussion, it is deemed that such a spectrum can be selected for the maritime simulator: the spectral wind parameter contains the wind speed, wind direction, and fetch length, so the simulation results can more completely show the wind effect on the simulated wave. Therefore, $\Psi_{J-Po}(\mathbf{k})$ is more suitable for the maritime simulator than the Phillips spectrum.

Future work should continue to improve the simulation results of ocean wave scene, making it more realistic. The marine radar wave extractor equipped on the training ship "Yukun" of Dalian Maritime University may be used to observe ocean wave to obtain a more realistic wave spectrum for simulating an ocean wave scene.

Acknowledgement. This work is supported by 973 Program of P. R. China (No. 2009CB320805) and the Fundamental Research Funds for the Central Universities (No. 3132013302).

References

1. Mihalef, V., Metaxas, D., Sussman, M.: Animation and Control of Breaking Waves. In: 2004 ACM SIGGRAPH / Eurographics Symposium on Computer Animation, pp. 315–324. ACM Press, New York (2004)
2. Pritchard, P.: Fluid Mechanics, SI Version, 8th edn. John Wiley & Sons (Asia) Ptc Ltd., Singapore (2011)
3. Muller, M., Solenthaler, B., Keiser, R., et al.: Particle-based Fluid-fluid Interaction. In: 2005 EUROGRAPHICS ACM SIGGRAPH Symposium on Computer Animation, pp. 237–244. ACM Press, New York (2005)
4. Yan, K., Wang, Z., Liao, B., et al.: Physically Based Realistic Modeling and Rendering of Ocean Waves. Computer-aid Design & Computer Graphics 20(9), 1117–1124 (2008)
5. Chen, X., He, J., Yan, K., et al.: An Integrated Algorithm of Real-time Fluid Simulation on GPU. Computer-aid Design & Computer Graphics 22(3), 396–405 (2010)
6. Ojeda, J., Susín, A.: Enhanced Lattice Boltzmann Shallow Waters for Real-time Fluid Simulations. In: 2013 ACM SIGGRAPH / EUROGRAPHICS Symposium on Computer Animation, pp. 25–28. ACM Press, New York (2013)
7. Fourier, A., Reeves, W.: A Simple Model of Ocean Waves. In: The 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1986), pp. 75–84. ACM Press, New York (1986)
8. Yin, Y., Ren, H., Liu, X.: Research on Real Time Algorithm of Viewing Scene in Distributed Navigational Simulation System. In: 2008 Asia Simulation Conference - the 7th International Conference on System Simulation and Science Computing, pp. 65–69. IEEE Press, New York (2008)
9. Feng, K., Liu, T.: Research on Wave Real-time Simulation Based on 3D Gerstner Wave. Engineering Graphics 30(5), 53–57 (2009) (in Chinese)
10. Yuksel, C.: Wave Particles. In: The 34th International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH 2007), pp. 1–8. ACM Press, New York (2007)

11. Tessendorf, J.: Simulating Ocean Water. In: The 26th International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH 1999), Course Notes 26 (1999)
12. Ren, H., Jin, Y., Chen, L.: Real-time Rendering of Ocean in Marine Simulator. In: 2008 Asia Simulation Conference - The 7th International Conference on System Simulation and Scientific Computing, pp. 1133–1136. IEEE Press, New York (2008)
13. Miandji, E., Sargazi Moghadam, M., Samavati, F., et al.: Real-time Multi-band Synthesis of Ocean Water with New Iterative Up-sampling Technique. Visual Computer 25(5-7), 697–705 (2009)
14. Dupuy, J., Bruneton, E.: Real-time Animation and Rendering of Ocean Whitecaps. In: SIGGRAPH Asia 2012 Technical Briefs, Article No.: a15. ACM Press, New York (2012)
15. International Maritime Organization (IMO): The Manila Amendments to the International Convention on Standards of Training, Certification and Watchkeeping for Seafarers. Dalian Maritime University Press, Dalian (2010) (in Chinese)
16. Det Norske Veritas (DNV): Standard for Certification Maritime Simulator Systems No.2.14. DNV Publications, Oslo (2011)
17. COST 714 Working Group 3: Measuring and Analyzing the Directional Spectrum of Ocean Waves. EU Publications Office, Brussels (2005)
18. Liu, D., Leng, M.: Meteorology and Oceanography for Mariners. Dalian Maritime University Press, Dalian (2011)
19. Nouguier, F., Guérin, C., Chapron, B.: "Choppy Wave". Model for Nonlinear Gravity Waves. Geophysical Research 114(C09012), 1–16 (2009)

Contribution of NaP-Channels to the Property of Subthreshold Resonance Oscillation

Babak Vazifehkhah Ghaffari, Shahrum Shah bin Abdullah, Mojgan Kouhnavard, and Tatsuo Kitajima*

Malaysia-Japan International Institute of Technology,
University Teknologi Malaysia Kuala Lumpur
Jalan Semarak, 54100 Kuala Lumpur, Malaysia
{Babakghaffary,mjgn_kouhnavard}@yahoo.com,
{shahrum,kitajima}@ic.utm.my

Abstract. Subthreshold resonance oscillations are observed in many excitatory/inhibitory neurons in our brain. Although they have been thought to play an important role in behavioral or perceptual states in animals, detail properties of these phenomena have not been clarified, yet. It is necessary to understand first these oscillatory features to clarify the contribution of these rhythmic oscillations to higher brain function, such as short-term memory, the working memory, long-term potentiation and long-term depression. Among various voltage-dependent channels thought to be involved in the generation of these oscillations, hyperpolarization-activated potassium channel (h channel) and persistent inactivating sodium channel (NaP) are considered, because these two voltage-dependent channels are closely related to the sustained oscillatory activity observed in Entorhinal cortex and other Neocortex regions. This feature article considers a compartmental neuron model with an h-channel and a NaP-channel. The NaP-channel contribution to the property of subthreshold resonance oscillation is examined by computer simulation of this neuron model.

Keywords: hyperpolarization-activated potassium channel, persistent inactivating sodium channel, subthreshold resonance.

1 Introduction

Various neural oscillations are observed in our brain. Nowadays, these oscillations are considered that they are closely related to higher information processing functions of the brain, such as learning and memory. More details studies are required to clarify their mechanisms from not only experimental but also theoretical standing points of view. Many experiments have shown that, if the input current, whose frequency increases or decreases with time, is given to neurons, their membrane potentials can take the maximum at some specific input frequency, especially at the level of subthreshold. This phenomenon is called as subthreshold resonance. Many excitatory and inhibitory neurons in the brain show such subthreshold resonance oscillations. They

* Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 466–474, 2013.

were first observed in squid giant axon by Mauro et al. [1] and Koch, et al. [2] in the 1980s. Since then, such resonance phenomena have been found in various brain regions, such as inferior olive neurons [3], thalamic neurons [4, 5, 6], cortical neurons [7, 8, 9, 10], and hippocampal CA1 pyramidal neurons and interneurons [11]. Subthreshold resonance oscillations observed in the brain mean that neurons have the characteristics of frequency selectivity, that is, the band pass filter property. According to the electrical circuit theory, this fact indicates that some inductive elements may be associated with subthreshold resonance phenomena. As experimental techniques advance, it has been clarified that various voltage-dependent ion channels have been involved in the generation of these oscillations, such as hyperpolarization-activated potassium channels (h-channel) [8, 12, 13], fast persistent inactivating sodium channels (NaP-channel) [14], and slow non-inactivating potassium channels (Krs-channel) [9]. Furthermore, voltage-dependent calcium channels (T-channel), and calcium-dependent potassium channels (I_C-channel) have been reported to be also involved in subthreshold resonance phenomena [4]. Recently, Narayanan et al. [15] reported that the current through h-channel would be closely related to the synaptic plasticity in the hippocampal CA1 area, which will be thought as one of fundamental phenomena of learning and memory function in the brain.

It has been shown that both an h-channel and a NaP are closely related to the sustained oscillatory activity observed in Entorhinal cortex and other neocortex resions [14]. Recently, one of authors showed that a compartment model with an h-channel and a NaP-channel can be transformed in to the equivalent RLC electrical circuit at the level of subthreshold and showed that basic property of subthreshold resonance phenomena [16]. In this study, one compartmental neuron model with two transmembrane ionic channels, h-channel and NaP-channel, is considered. For this neuron model with Hodgkin-Huxley type dynamics, we examine how the amplitude of a NaP-channel conductance affects to the property of the subthreshold resonance oscillation by computer simulations. We also study how the change of equilibrium potential to the property of subthreshold resonance phenomena.

2 The Dynamics of Voltage-Dependent Channels

2.1 Hyperpolarization-Activated Potassium Channel (h-channel)

A conductance based h-channel (hyperpolarization-activated potassium channel) model is shown in Fig. 1.

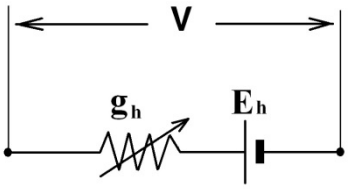


Fig. 1. A conductance based *h*-channel model: V is the membrane potential (mV). E_h is the equilibrium potential for *h*-channel *(mV)*. g_h is the conductance for *h*-channel (mS/cm^2). I_h is the current through *h*-channel ($\mu A/cm^2$).

The dynamics of h-channels can be gives as follows [17]:

$$I_h = g_h(V).[V - E_h], \tag{1}$$

$$g_h(V) = \bar{g}_h.\left[k.m_{hf} + (1-k).m_{hs}\right], \tag{2}$$

Where, $\bar{g}_h$ is the maximum amplitude of h-channel and k is the constant. $m_{hf}(t)$ and $m_{hs}(t)$ are dimensionless variables satisfy the following first order differential equations:

$$dm_{hf}(t)/dt = \left(m_{hf\infty} - m_{hf}(t)\right)/\tau_{mhf} \tag{3}$$

$$dm_{hs}(t)/dt = (m_{hs\infty} - m_{hs}(t))/\tau_{mhs} \tag{4}$$

Parameters $m_{hf\infty}$ and $m_{hs\infty}$ represent voltage dependent steady states and τ_{hf} and τ_{hs} are time constants of $\mathrm{m_{hf}}(t)$ and $\mathrm{m_{hs}}(t)$, respectively, they are given by:

$$m_{hf\infty}(V) = 1/1 + exp((V + 79.2)/9.78) \tag{5}$$

$$\tau_{mhf}(V) = 1 + 0.51/((exp\,((V - 1.7)/10 + exp\,(-(V + 340)/52)) \tag{6}$$

$$m_{hs\infty}(V) = 1/(1 + exp\,((V + 71.3)/7.9) \tag{7}$$

$$\tau_{mhs}(V) = 1 + 5.6/(exp((V - 1.7)/14) + exp\,((V + 260)/43) \tag{8}$$

The initial conditions at the resting potential (V=-65) are given by:

$$m_{hf0} = m_{hf\infty}(-65) \tag{9}$$

$$m_{hs0} = m_{hs\infty}(-65) \tag{10}$$

The individual h-channel ionic current (I_h) obtained by stepwise process called Runge-Kutta 4th order (RK4), which will be discussed later.

2.2 Persistent Inactivating Sodium Channel (NaP-channel)

A conductance based NaP-channel (persistent-inactivating sodium channel) model is shown in Fig. 2.

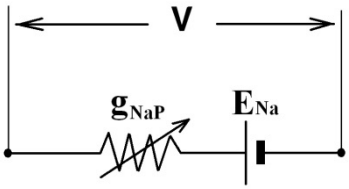


Fig. 2. A conductance based *NaP*-channel model: *V* is the membrane potential (*mV*), E_{Na} is the equilibrium potential for *NaP*-channel *(mV)*. g_{NaP} is the conductance for *NaP*-channel (mS/cm^2). $\mathrm{I_{NaP}}$ is the current through NaP-channel ($\mu A/cm^2$).

The dynamics of NaP-channels defined as follow [17]:

$$I_{NaP} = g_{NaP}(V).(V - E_{Na}), \tag{11}$$

$$g_{NaP}(V) = \bar{g}_{NaP}.m_{NaP}, \tag{12}$$

Where, $\bar{g}_{NaP}$ is the maximum amplitude of h-channel. m_{NaP}, is dimensionless variable that satisfy the following first order differential equation:

$$dm_{NaP}(t)/dt = \alpha_{NaP}(1 - m_{NaP}(t)) - \beta_{NaP}m_{NaP}(t) \tag{13}$$

Where, α_{NaP} and β_{NaP} are voltage dependant rate constant, which obtained as:

$$\alpha_{NaP}(V) = 1/[0.15 \times (1 + exp(-(V + 38)/6.5))] \tag{14}$$

$$\beta_{NaP}(V) = exp(-(V + 38)/6.5)/[0.15 \times (1 + exp(-(V + 38)/6.5))] \tag{15}$$

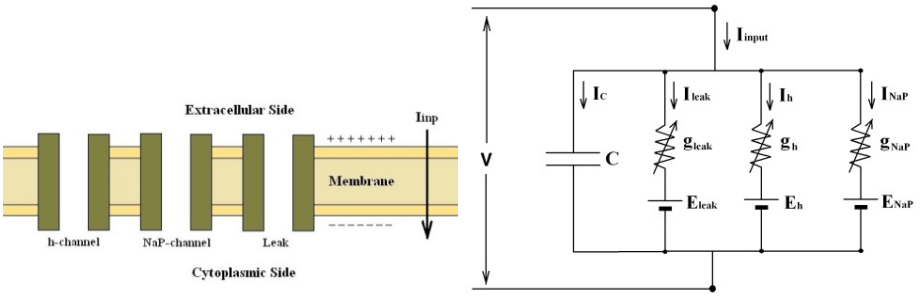
The initial conditions at the resting potential (V=-65) are calculated as:

$$m_{NaP0} = \alpha_{NaP}(V)/\alpha_{NaP}(V) + \beta_{NaP}(V) \tag{16}$$

The individual NaP-channel ionic current (I_{NaP}) obtained by RK4 stepwise process.

3 Model Neuron with h-channel and NaP Channel

Here we consider a compartmental neuron model with h-channel, and Nap-channel, which is known in Fig. 3.



a) b)

Fig. 3. Compartmental neuron model: a) Schematic model of a single neuron. b) Electrical circuit model of a neuron. *V* is the membrane potential, E_{Na} and E_h are equilibrium potentials for *NaP* and *h* channels *(mV)*, respectively. g_{leak} is the leak conductance (mS/cm^2), g_{NaP} and g_h are the conductances for *NaP and h* channels (mS/cm^2). C is the membrane capacitance $\mu F/cm^2$, I_{input} is the synaptic input current $\mu A/cm^2$ and E_{leak} is the reversal potentials *(mV), C* is the membrane capacitance $\mu F/cm^2$.

From Fig. 3, the following current relation is obtained:

$$I_{input}(t) = I_c(t) - I_{leak}(t) - I_h(t) - I_{NaP}(t) \tag{17}$$

Where, $\mathrm{I_C}(t)$ is the current through membrane capacitance C, $\mathrm{I_{leak}}(t)$ the leak current, $\mathrm{I_h}(t)$ the current through an h-channel, and $\mathrm{I_{NaP}}(t)$ the current through an NaP-channel. $\mathrm{I_C}(t)$ and $\mathrm{I_{leak}}(t)$ are given as follow:

$$I_c(t) = C.dV(t)/dt \tag{18}$$

$$I_{leak}(t) = \bar{g}_{leak}.(V(t) - E_{leak}) \tag{19}$$

Where, $\bar{\mathrm{g}}_{\mathrm{leak}}$ is the maximum leak conductance. $\mathrm{I_h}(t)$ and $\mathrm{I_{NaP}}(t)$ are given by equations (1) and (11), respectively.

4 Computer Simulation

In order to evaluate the property of the subthreshold resonance phenomena of a neuron, the following input current is used:

$$I_{input} = I_0.sin(2\pi f(t).t) \tag{20}$$

$$f(t) = f_0 + (f_m + f_0).t/T \tag{21}$$

Where, the frequency *f(t)* is increasing from f_0 to f_m in simulation period *T*. the MATLAB R2012a (64-bit) was used to solve the dynamics. The values of the constants and parameters used in computer simulations are showed in Table. 1.

4.1 Effects of the Amplitude of NaP-channel Conductance

To study how the amplitude of NaP-channel affects the subthreshold resonance, the value of the amplitude of NaP-channel conductance was altered. Fig. 4 (a), (b), and (c) show the membrane potential and FFT amplitude for $\bar{g}_{NaP} = 0.3\ mS/cm^2$, *0.5* mS/cm^2, *0.7* mS/cm^2, respectively. As the value of $\bar{g}_{NaP}$ increases, the excitatory property caused by Na^+ through a NaP-channel becomes stronger, so the membrane potential is increased resulting in increased FFT amplitude, and also 90% width of frequency is narrowed, as shown in Table. 2. These results show that the resonance property in neurons is enhanced by the activation of NaP-channel. However, the resonance frequency is changed because of the persistent activity of NaP-channel.

Table 1. Values of the constants and parameters used in computer simulations

C	$1.5\mu F/cm^2$
g_l	$0.5ms/cm^2$
g_h	$1.5ms/cm^2$
E_h	*-20mV*
g_{NaP}	$0.5ms/cm^2$
E_{Na}	*55 mV*
I_0	*-2.25μm*
f_m	*40 Hz*

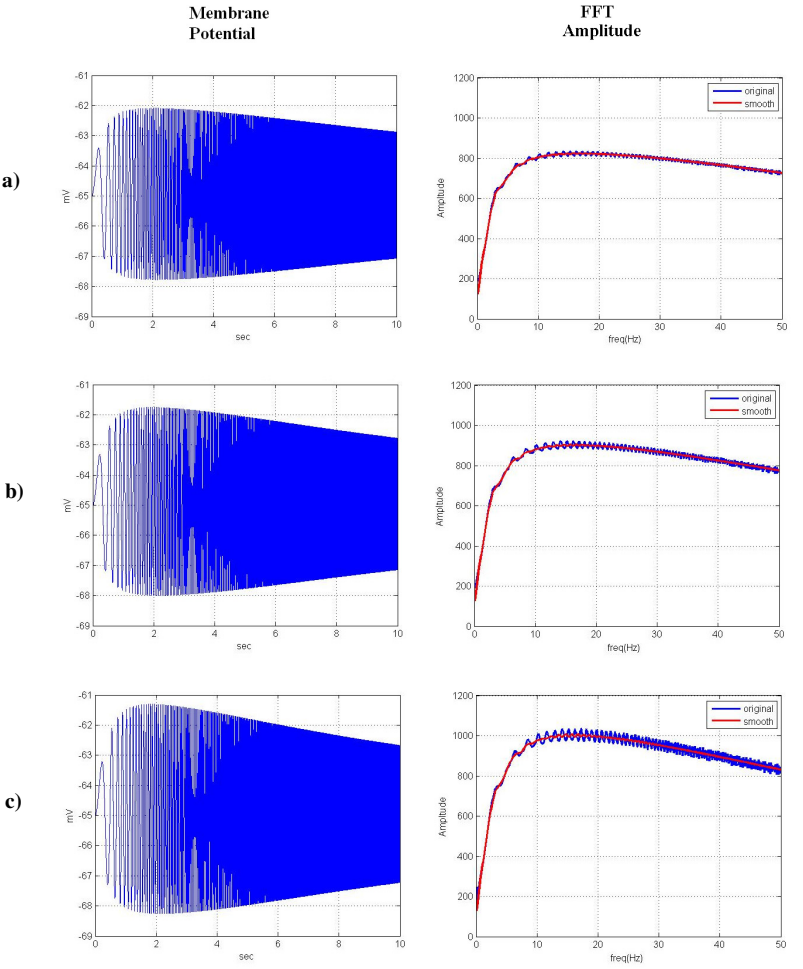


Fig. 4. Simulation results by changing the value of the amplitude of NaP-channel conductance: a) $\bar{g}_{NaP} = 0.25\ mS/cm^2$, b) $\bar{g}_{NaP} = 0.5\ mS/cm^2$, c) $\bar{g}_{NaP} = 0.75\ mS/cm^2$.

Table 2. Maximum value of simulation result in time and frequency domain

	$\bar{g}_{NaP} = 0.25\ ms/cm^2$	$\bar{g}_{NaP} = 0.5\ ms/cm^2$	$\bar{g}_{NaP} = 0.75\ ms/cm^2$
Maximum Potential (mV)	**-62.09**	**-61.75**	**-61.31**
Amplitude of FFT	**824**	**920**	**1007**
Resonance Frequency	**16.8**	**15.8**	**15.2**
90% Width of frequency	**405**	**362**	**318**

5 Conclusion

In this paper, effects of a NaP-channel conductance on a compartmental neuron model (includes h-channel and leak channel) is simulated. The results show that a NaP-channel contributes not only to increase the amplitude of membrane potentials, but also to clarify the resonance property, that is, the more the amplitude of NaP-channel conductance increases, the less the width of 90%-peak of FFT amplitude becomes.

Although the frequency selectivity of neurons should play an important role in behavioral or perceptual brain functions in animals, their practical roles have not yet been clarified. Furthermore, as a neurons can generate action potentials when its somatic membrane potential over the threshold potential, only small change of membrane potential by Subthreshold Resonance phenomena would cause a drastic change of neural oscillation. These problems will be very interesting and suggestive in order to develop new innovative, bio-mimetic technologies and applications in various engineering fields. In order to treat these problems, it will be necessary to study in more detail by using both experimental approach and theoretical approach by using modeling and computer simulation.

Acknowledgement. B.V.Ghaffari and M.Kouhnavard would like to thank Malaysia-Japan International Institute of Technology (MJIIT) for financial support and also B.V.Ghaffari would like to thank head of Biologically Inspired System and Technology Laboratory (Bio-iST Lab) for his research support.

References

1. Mauro, A., Conto, F., Dodge, F., Shoe, R.: Subthreshold Behavior and Phenomenological Impedance of the Squid Giant Axon. Gen. Physiol. 55, 497–523 (1970)
2. Narayanan, R., Johnston, D.: The *h* Channel Mediate Location Dependence and Plasticity of Intrinsic Phase Response in Rat Hippocampal Neurons. Neuroscience 28, 5846–5860 (2008)
3. Lampl, I., Yarom, Y.: Subthreshold Oscillations of the Membrane Potential: A Functional Synchronizing and Timing Device. Neurophysiology 70, 2181–2186 (1993)
4. Pedroarena, C., Llinas, R.: Dendritic Calcium Conductances Generate High-Frequency Oscillation in Thalamocortical Neurons. Proceedings of National Academy of Sciences of the United States of America 94, 724–728 (1997)
5. Hutcheon, B., Miura, R.M., Yarom, Y., Puil, E.: Low-Threshold Calcium Current and Resonance in Thalamic Neurons; A Model of Frequency Preference. Neurophysiology 71, 583–594 (1994)
6. Puil, E., Meiri, H., Yarom, Y.: Resonant behavior and frequency preference of thalamic neurons. Neurophysiology 71, 575–582 (1994)
7. Hutcheon, B., Miura, R.M., Puil, E.: Subthreshold Membrane Resonance in Neocortical Neurons. Neurophysiology 76, 683–697 (1996a)
8. Hutcheon, B., Miura, R.M., Puil, E.: Models of Subthreshold Membrane Resonance in Neocortical Neurons. Neurophysiology 76, 698–714 (1996b)
9. Gutfreund, Y., Yarom, Y., Segev, I.: Subthreshold Oscillations and Resonant Frequency in Guinea-Pig Cortical Neurons Physiology and Modelling. Physiology London 483, 621–640 (1995)
10. Llinas, R.R., Grace, A.A., Yarom, Y.: In Vitro Neurons in Mammalian Cortical Layer 4 Exhibit Intrinsic Oscillatory Activity in the 10-50-Hz Frequency Range. Proceedings of National Academy of Sciences of the United States of America 88, 897–901 (1991)
11. Cobb, S.R., Buhl, E.H., Halasy, K., Paulsen, O., Somogyl, P.: Synchronization of Neuronal Activity in Hippocampus by Individual GABAergic Interneurons. Nature 378, 75–78 (1995)
12. Acker, C.D., Kopell, N., White, J.A.: Synchronization of Strongly Coupled Excitatory Neurons: Relating Network Behavior to Biophysics. Computational Neuroscience 15, 71–90 (2003)
13. Brunel, N.: Firing-rate Resonance in a Generalized Integrate-and-Fire Neuron with Subthreshold Resonance. Phys. Rev. 67, 051916 (2003)
14. Dickson, C.T.: Oscillatory Activity in Entorhinal Neurons and Circuits 911, 127–150 (2000)
15. Stafstrom, C.E.: Persistent Sodium Current and Its Role in Epilepsy. Epilepsy Currents 7, 15–22 (2007)
16. Kitajima, T.: The property of Subthreshold Resonance Observed in Voltage-Dependent Ion Channel. In: Word Research and innovation Conversion on Engineering and Technology 2012, keynote address 3. Kual Lumpur (2012)
17. Moser, E.I., Moser, M.B.: A metric for space. Hippocampus 18, 1142–1156 (2008)
18. Dwyer, J., Lee, H., Martell, A., Drongelen, W.: Resonance in Neocortical Neurons and Networks. Neuroscience 36, 3698–3708 (2012)
19. Izhikevich, E.M.: Resonance and Selective Communication via Bursts in Neurons having Subthreshold Oscillations. BioSystems 67, 95–102 (2002)

20. Shay, C.F., Boardman, I.S., James, N.M., Hasselmo, M.E.: Voltage Dependence of Subthreshold Resonance Frequency in Layer II of Medial Entorhinal Cortex. Hippocampus 22, 1733–1749 (2012)
21. Zemankovics, R., Kali, S., Paulsen, O., Freund, T.F., Hajos, N.: Differences in Subthreshold Resonance of Hippocampal Pyramidal Cells and Interneurons: The Role of h-Current and Passive Membrane Characteristics. Physiol. 12, 2109–2132 (2010)
22. Fransen, E.: Hippocampal Microcircuits: Entorhinal Cortex Cells. In: Graham, V.B., Cobb, S., Vida, I. (eds.) Cutsuridis. Springer Series in Computational Neuroscience, vol. 5, pp. 375–398. Springer, NewYork (2010)
23. Steffanach, H.A., Witter, M., Moser, M.B., Moser, E.I.: Spatial Memory in the Rat Requires the Dorsolateral Band of the Entorhinal Cortex. Neuron. 45, 301–313 (2005)
24. Klink, R., Alonso, A.: Ionic Mechanisms for the Subthreshold Oscillations and Differential Electro-Responsiveness of Medial Entorhinal Cortex Layer II Neurons. Neurophysiol 70, 144–157 (1993)
25. White, J.A., Budde, T., Kay, A.R.: A Bifurcation Analysis of Neuronal Subthreshold Oscillations. Biophys 69, 1203–1217 (1995)
26. Hodgkin, A.L., Huxley, A.F.: A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve. Physiol. 117(4), 500–544 (1952)
27. Yue, C., Remy, S., Su, H., Beck, H., Yaari, Y.: Proximal Persistent Na+ Channels Drive Spike Afterdepolarizations and Associated Bursting in Adult CA1 Pyramidal Cells. Neurosci. 25, 9704–9720 (2005)
28. Stafstrom, C.E., Schwindt, P.C., Crill, W.E.: Repetitive Firing in Layer V Neurons from Cat Neocortex in Vitro. Neurophysiol 52, 264–277 (1984)
29. French, C.R., Sah, P., Buckett, K.J., Gage, P.W.: A Voltage-Dependent Persistent Sodium Current in Mammalian Hippocampal Neurons. Gen. Physiol. 95, 1139–1157
30. Koch, C.: Cable Theory in Neurons with Active, Linearized Membranes. Biol. Cybern. 50, 153 (1984)

An OM Mapping Based Federation Development and Execution Process for Overlapped Federation Executions

Bin Xiao, Hongbo Sun, and Tianyuan Xiao

Department of Automation, Tsinghua University, Beijing 100084, China
xiaobin07@mails.tsinghua.edu.cn,
{sunhongbo02,xty-dau}@tsinghua.edu.cn

Abstract. This paper presents an OM mapping based federation development and execution process for overlapped federation executions. The proposed process includes four main stages: Scenario Analysis, Requirements Analysis, Collaboration Preparation and Federation Execution. It uses mapped OMs to represent collaboration knowledge for overlapped federations in a simple and flexible way. This OM construction and manipulating mechanism improves efficiency and flexibility of overlapped HLA federations and reduces workload without the loss of accuracy and consistency. This paper also discusses some mapping based OM modeling and manipulating issues under the situation of overlapped federation executions, which includes OM mapping template and OM mapping architecture. In the end, the implementation is discussed on the basis of TH_RTI, a RTI version developed by National CIMS ERC, Tsinghua University, China. The proposed approach has great potential to improve efficiency of FEDEP for overlapped federation executions, reduce the work load for dynamic collaborations among overlapped federations, and enhance the applicability and flexibility of HLA systems.

Keywords: OM Mapping, HLA, FEDEP, Overlapped Federation Execution.

1 Introduction

As a well-known modeling and integration standard of distributed simulation, HLA (High level Architecture [1]) has been successfully adopted in various simulation systems and been extended to some other research areas [2]. Since simulation is an important part of collaborative product development, some CPD (Collaborative Product Development) systems use HLA as their basic architecture [3,4].

However, when applying HLA in this research area, there raise some new challenges. Within these challenges, multi-federation is the one which has close relationship with multi-disciplinary nature of CPD, which means several overlapped or bridged federations will execute and collaborate at the same time [5].

Generally speaking, the problem of multi-federation can be solved by several means, such as federation merging, federation bridging, hierarchical federation and overlapped federation.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 475–484, 2013.

Federation merging integrates separated federations which need to be collaborated in future into one federation, and the FED (Federation Execution Description) files also integrate all collaboration requirements within and among separated federations. This method is the easiest way to resolve multi-federation problem and solve the problem in a static way (This means all collaboration, within or among separated federations are all preset before federation execution starts). When separated federations focus on different topics, such as collaborative design, simulation and optimization, the efficiency of this method cannot be guaranteed. Furthermore, this method cannot deal with the situation that separated federations cannot be integrated together for some reasons like security, performance, environments, etc.

Federation bridging establishes an intermediate node between two federations, and this node is called a federation bridge that will be in charge of communications between two separated federations [6]. This method isolates separated federations into different control domain and communication domain, and keeps their independency well. However, the communication protocol and mechanism of the federation bridge need to be especially designed.

Hierarchical federation organizes separated federations like a tree [7]. Parent federation is in charge of communication and collaboration among its direct children federations and leaf nodes only focus on their own collaboration topic. This method can be deemed as a special overlapped federation but the communication and collaboration of separated federations are dealt by their parent federations.

Overlapped federation uses some federates of separated federations as federation bridges. Separate federations own their separated control domains and communication domains, and some federates will belong to different federations at the same time. Therefore, some mechanism needs to be introduced to solve the problem of different FED files for different federations.

This paper proposes an OM (Object Model) mapping based FEDEP (Federation Development and Execution Process) for overlapped federation executions to give an outline of using OM mapping to deal with the problem of different FED files for overlapped federations. And this paper is organized as follows: the motivation of this research is described in Section 2; OM mapping based FEDEP is proposed in Section 3; discussions and conclusions are given in the last section.

2 Motivation

HLA-based federation development can be divided into 11 steps: Analyze Scenario, Generate Federation Conceptual Model, Analyze Collaboration Requirements, Generate FOM, Negotiate Federation Agreements, Realize Federate, Realize RTI, Create Federation, Start Federation Execution, Execute Federation and Terminate Federation Execution, As Figure 1 shows [8], which is constructed according to IDEF0, box presents a certain function, arrow at left presents inputs of the function, right arrow means outputs, top arrow present controls of the function, and bottom arrow is mechanisms for the execution of function.

In this process, the input is federate SOMs (Simulation Object Models), Existing Conceptual Models and collaboration Aim. The output is Scenario Instances, Federate Modification, Supporting Database, RTI (Run-Time Infrastructure) & its initial data, interaction/object class instance, class subscriber, class publisher, synchronize point and results. Resources involved include Scenario Lib, Domain Resources, Data Dictionary, OM Lib and Other Resources. Scenario Lib supports Analyze Scenario by Existing Scenarios. Domain Resources provides Domain Knowledge to analyze scenario and generate collaboration conceptual model. Data Dictionary stores Meta OM models for Generate FOM (Federation Object Model). When Generate FOM, related Existing FOMs and BOMs (Base Object Models) also need to be retrieved from OM Lib. Other Resources include tools and knowledge related to Generate FOM.

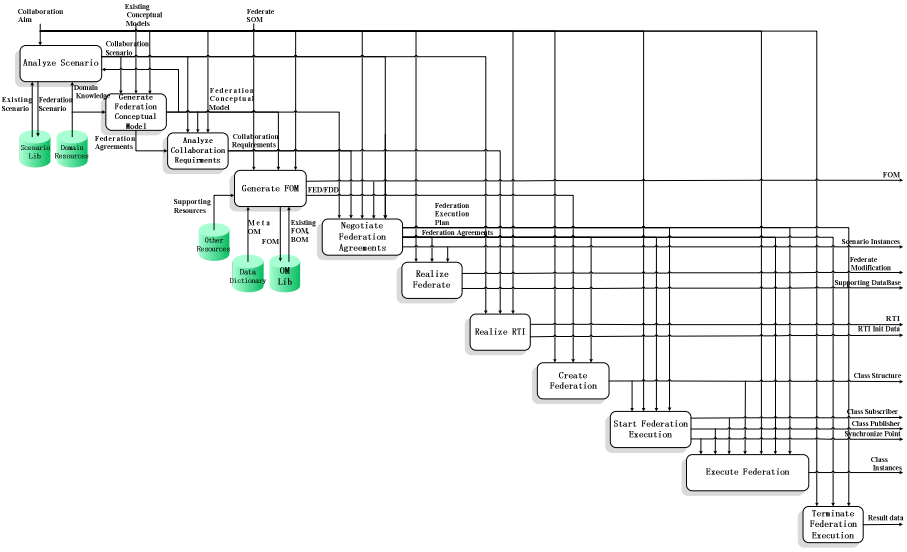


Fig. 1. Collaboration knowledge modeling and usage in FEDEP

Collaboration knowledge represents itself as Collaboration Scenario, Federation Conceptual Model, Federation Execution Plan, FED/FDD (FOM Document Data), FOM, Collaboration Requirements, Class Subscriber, Class Publisher, and Class Structure in this process. Collaboration Scenario describes environment and desired process of collaborations. Federation Conceptual Model is the description of concepts and their relations in collaborations. Federation Execution Plan gives operation order of participant federates. FOM is the standard description of federation concepts. FED/FDD stores FOM information. Collaboration Requirements describes Federation Execution Plan and collaboration requirements of federates. Interaction/object Class Structure is inheritance relations among these classes. Class Subscriber is federates that subscribe interaction or object classes. Class Publisher is federates that publish interaction or object classes, which supervise data distribution during collaborations. As one form of collaboration knowledge, OM (includes SOM, FOM and FED/FDD files) is one of the key entities which have closely relationship with the coming collaborations.

From the view of overlapped federations, OMs have several shortcomings which limit its usage and efficiency.

First, it is a time-consuming and cost task to construct OMs according to different federations, and semantic accuracy and consistency are also hard to maintain.

Second, SOM generation is not mentioned in FEDEP, which is the basis for following steps.

Third, class RootObject is the only one form of OM which can be used during collaboration executions. However, it is loaded only once before Start Federation Execution, so it cannot satisfy collaboration requirement changing in the process of federation execution.

In this paper, an OM mapping based FEDEP is proposed to give a flexible and consistent way of OM representation and manipulating so that collaboration knowledge can be used in an easier, more consistent, flexible and efficient way for overlapped federations.

3 OM Mapping Based FEDEP

In overlapped federation, there are two kinds of federates, one is owned by one federation, and the other is shared by several federations. During the whole federation execution, the shared federates will exchange data with several RTI belonging to different sub-federation respectively, and so all sub-federations belonging to the collaboration federation can work together coordinately.

In OM Mapping Based FEDEP, one important work is identifying and defining OM Mapping relationship among given federates based on the collaboration requirements, includes identifying and defining source and target object/interaction class, as well as their mapping rules. The other work is building OM mapping models according to the OM mapping relationship to support federation execution. Defining OM mapping relationship and building OM Mapping models are important identification and data exchange basis of given federation.

The process of modeling and utilizing OM mapping relationship can be separated into four stages: Scenario Analysis, Requirements Analysis, Collaboration Preparation and Federation Execution (Figure 2).

In stage Scenario Analysis, scenario is converted to federation collaboration conceptual under the control of Collaboration Aim. Collaboration Requirement is provided and used as the input of Requirements Analysis.

In stage Requirements Analysis, SOM of each federate, and sharing relationship of each federate among sub-federations are identified and defined under the control of collaboration aim. SOM is represented by OMT (Object Model Template); sharing relationship is represented by XML (Extensible Markup Language). Moreover, the mapping relationships among object/interaction class are defined. Requirements are used as the input of collaboration preparation.

In stage Collaboration Preparation, FOM is generated according to federate SOMs and the OM mapping relationships, and OM mapping models are constructed according to the FOM and mapping relationship.

In stage Federation Execution, OM mapping execution models are constructed according to OM mapping models, and collaboration is executed according to OM mapping execution models, collaboration requirements and collaboration aim.

3.1 Requirements Analysis

The input of Requirement Analysis is Collaboration Conceptual model, the output is Federate Collaboration architecture Described by UML Sequence Diagram and Communication Diagram, and SOM of each federate described by OMT, and the mapping relationship among SOMs. The resources involved include SOMs Lib, OM mapping rules Lib. SOMs Lib supports collaboration requirements analysis by existing federate SOMs. OM mapping rules Lib provides existing rules for OM mapping. This stage can be divided into two activities: Analyze federation architecture, Analyze OM mapping relationship (Figure 3).

Analyze federation architecture generates federate and sub-federation collaboration structure and collaboration mechanism described by UML Sequence Diagram and Communication Diagram according to existing federates and federations under the control of collaboration target.

Analyze OM mapping relationship Analyzes the relationship among existing federates SOMs defined in Analyze federation architecture activity, and provides OM mapping relationship which includes the source and target object/interaction class, as well as their mapping rules for future use.

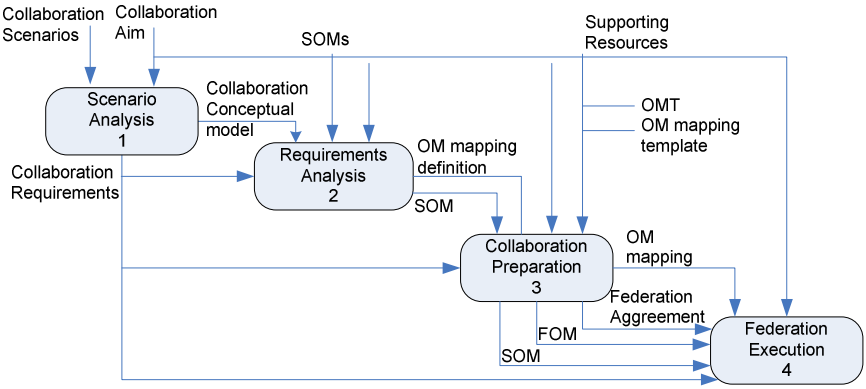


Fig. 2. Top-Level View

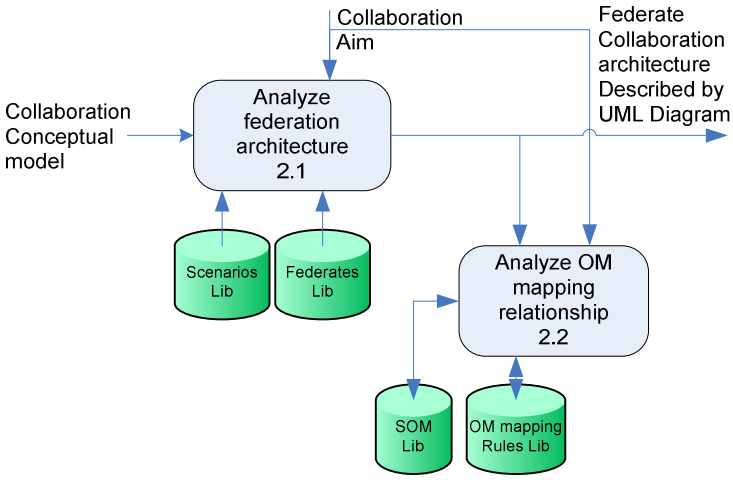


Fig. 3. Requirements Analysis

3.2 Collaboration Preparation

In this stage, the input is Federate Collaboration architecture Described by UML Communication Diagram, SOM, OM mapping relationship. The output is Federation Collaboration FOM, Federation Agreements, Federate Modifications, Supporting Database, TH_RTI (extended by adding an adaptive OM mapping layer, Figure 4), and RTI initial Data. And the reusable resources involve SOMs Lib, OM mapping rules Lib (Figure 5).

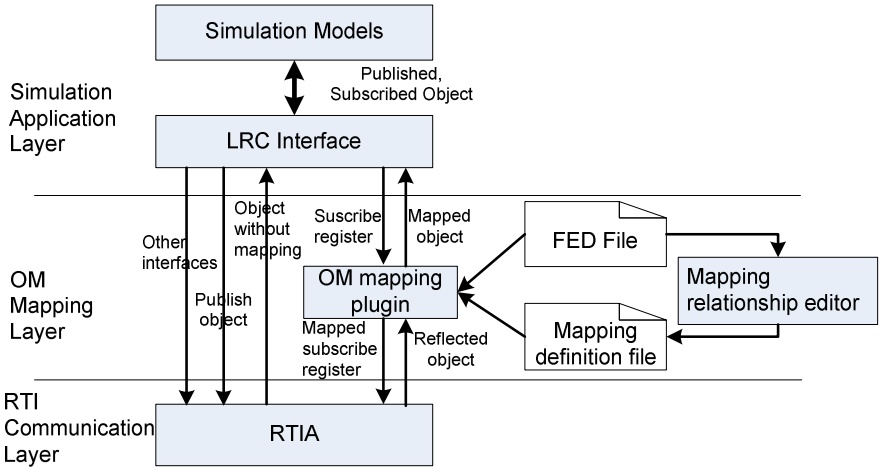


Fig. 4. OM mapping layer in TH_RTI

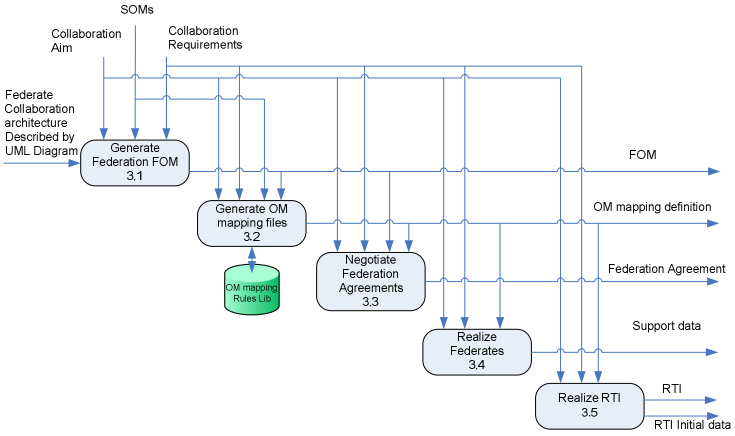


Fig. 5. Collaboration Preparation

In OM mapping based FEDEP, FOM can be established by either automatic transformation from SOM files or FCA-like method with the help of Extended UML Sequence Diagram and Communication Diagram.

The process of Collaboration Preparation can be separated into five activities: Generate Federation FOM, Generate OM mapping files, Negotiate Federation Agreements, Realize Federates, and Realize RTI.

SOM is generated by reference existing SOM for reuse federate, or by building new SOM on the basis of OMT according to the Requirements Analysis. Based on SOM and federate Collaboration architecture, the FOM is constructed under the control of Collaborative Aim.

Based on SOM and FOM, OM mapping files are generated according to the mapping relationship with the help of mapping relationship editor. OM mapping files are XML formatted file, and a mapping file record all the mapping relationship (mapping object and mapping rules) in connection with a federate (Figure 6).

```
<MsgMap SOM="RadarSim.fed" FOM="HST.fed" Date="2013-04-03">
    <FedClass Name="AirCraft" Type="ObjClass">
        <ClassItem Name="Long" DataType="double" />
        <ClassItem Name="Lat" DataType="double" />
        <ClassItem Name="Alt" DataType="double" />
        <ClassItem Name="VelX" DataType="double" />
        <ClassItem Name="VelY" DataType="double" />
        <ClassItem Name="VelZ" DataType="double" />
    </FedClass>
    <ClassMap SOMCLASS="Plane">
        <ItemMap Name="Name" DataType="string" OperExp="AirCraft.Name"
defaultVal="Boeing"/>
        <ItemMap Name="Longitude" DataType="double"
OperExp="AirCraft.Long" defaultVal="0.0"/>
        <ItemMap Name="Latitude" DataType="double" OperExp="AirCraft.Lat"
defaultVal="0.0"/>
        <ItemMap Name="Altitude" DataType="double" OperExp="AirCraft.Alt"
defaultVal="0.0"/>
        <ItemMap Name="Speed" DataType="double"
OperExp="(Pow(AirCraft.VelX,2)
        +Pow(AirCraft.VelY,2)
        +Pow(AirCraft.VelZ,2))*3.6" defaultVal="0.0"/>
    </ClassMap>
</MsgMap>
```

Fig. 6. The example of OM mapping file

According to federation collaboration ontology and collaboration requirement, federation agreements are generated under the guidance of Collaboration Aim.

Under the control of Collaboration Aim and Requirements, according to Federation Agreements, SOM and OM mapping relationship, federates and RTI can be realized properly.

3.3 Federation Execution

The input of Federation Execution includes the Collaboration Aim, SOM, FOM, OM mapping files, Federation Agreements, and Collaboration Requirements. The output consists of concept subscriber, publisher, synchronize points, concept instances and result data. In this process, OM mapping model used to instance OM mapping relationship will be constructed dynamically in adaptive OM mapping layer of TH_RTI according to the OM mapping files. TH_RTI will determine which object/interaction classes should be mapped, and which ones should not be mapped. When OM mapping relationship is changed, OM mapping files should be redefined and Federation execution should be restarted. This process involves five activities: Create Federation, Start federation Execution, Execute Federation, Terminate Federation Execution, and Evaluate Federation Execution (Figure 7).

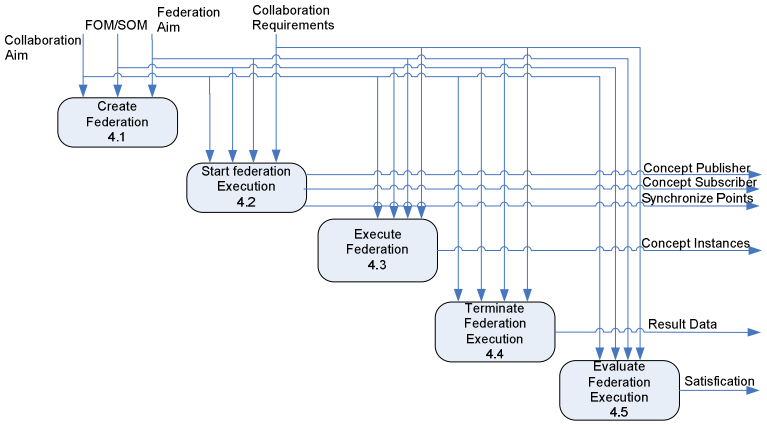


Fig. 7. Federation Execution

Federation is created by one federate and other federates will join in one by one. The main work is to specify federation name, federation execution name, related federates and register the OM mapping relationship with RTI, as well as construct OM mapping model instances.

The purpose of Start Federation Execution is to create the conceptual instances of synchronization points, register publishers and subscribers of concept instances which correspond to interaction and object classes.

In Execute Federation, data distribution is controlled by instances of publishers and subscribers. Execution process is guided by execution plan, federation agreements

and collaborative target. The outputs in this stage are concept instances, update and reflection of property values.

In the process of the federation execution, for those reflected object/interaction classes needed to be mapped, RTI will call corresponding OM mapping model instance to transform reflection of property values into a specified form according to the SOM of federate, so federate can understand and process reflected values correctly. For example, definition of subscribed class in SOM of subscriber is an object class named place and with three properties named longitude, latitude and altitude respectively. In federation execution, RTI will transmit object class named Location with properties named x, y, and z to subscriber according the registered OM mapping relationship. In order to understand and process the reflection of Location class instance, OM mapping model must transform reflection into corresponding format defined in SOM.

Terminating Federation Execution stops execution of given federation and complete results collection according to Federation Execution Plan, Federation Agreements and Collaboration Aim.

Evaluate Federation Execution analyze whether federation execution satisfies collaboration aim according to federation execution results, federation execution plan and federation agreements.

From the processed described above, OM mapping based FEDEP can be described as Figure 8. In this process, OM mapping establishes consistent understanding of object/interaction class with different definition among publisher and subscriber, and mechanism of construct OM mapping model instance dynamically according to OM mapping files reduces workload of reuse federate and improves federate reusability.

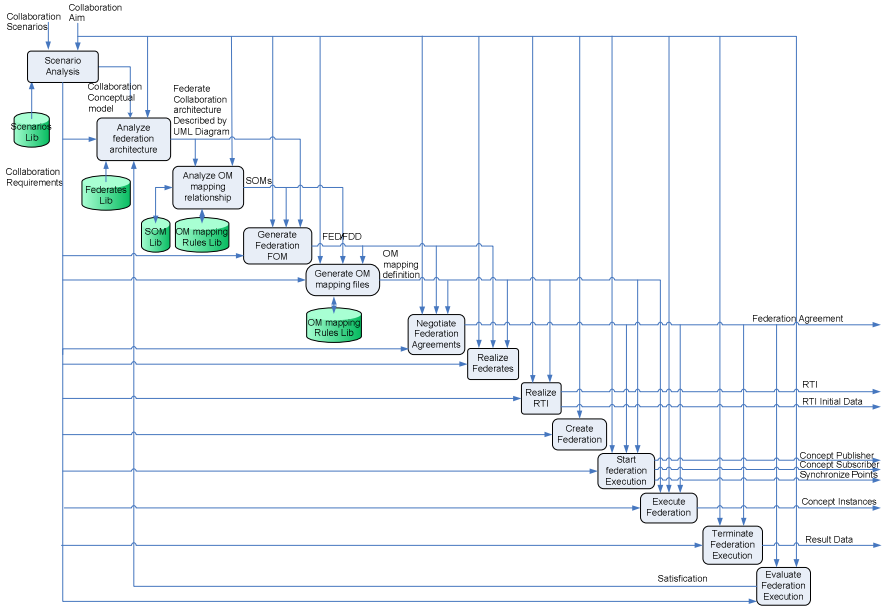


Fig. 8. Global view of OM mapping based FEDEP

4 Conclusion

As one of the key entities of collaboration knowledge in HLA, OM needs to have some novel mechanism to support overlapped federation executions. In this paper, OM mapping mechanism is introduced to deal with the problem of bridge federates needs different FED files for different federations in a flexible, simple and efficient way.

Other than overlapped SOM may be established ambiguously and inconsistently according to the requirements from separated federations which is time-consuming and cost much, this paper introduces OM mapping template to help SOM construction of overlapped federations, which could improve collaboration modeling efficiency.

Comparing with HLA keeping a blind eye on dynamic collaboration among overlapped federation executions, this paper proposes a novel OM mapping mechanism which adds an OM mapping layer between LRC and RTIA. It can support dynamic collaboration of overlapped federations in a flexible way.

In order to verify the feasibility of OM mapping, this research also has implemented the mechanism in TH_RTI. The result shows that the OM mapping based federation execution and development process can support overlapped federation executions in an efficient and flexible way.

Acknowledgment. This work is supported in part by the Chinese National Science and Technology Support Program (Grant NO. 2012BAH35F01).

References

1. Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules, The Institute of Electrical and Electronics Engineers, Inc., New York (2000)
2. Zhang, H., Wang, H., Chen, D.: Integrating web services technology to HLA-based mul-tidisciplinary collaborative simulation system for complex product development. In: 12th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2008, Xi'an, China, pp. 420–426 (April 2008)
3. Xu, Y., Xiao, T., Liang, C., Zhang, L.: Federated integration of networked manufacturing service platforms. Advanced Engineering Informatics 22, 317–327 (2008)
4. Tang, S., Xiao, T., Fan, W.: A collaborative platform for complex product design with an extended HLA integration architecture. Simulation Modelling Practice and Theory 18(8), 1048–1068 (2010)
5. Hao, J., Zhao, X., Huang, J., Huang, K.: On Interconnection Technology of High Level Architecture Multi-Federations. Journal of System Simulation 14(6), 714–717 (2002)
6. Han, C., Hao, J., Huang, J., Huang, K.: Research on Architecture of HLA Multiple Federations Interconnected by Bridge Federate. Journal of System Simulation 18(S2), 318–322 (2006)
7. Sun, H., Fan, W., Shen, W., Xiao, T., Hao, Q.: A Hierarchical Federated Integration Architecture for Collaborative Product Development. International Journal of Computer Integrated Manufacturing 25(10), 901–913 (2012)
8. IEEE Computer Society. IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP, IEEE Std 1516.3-2003). The Institute of Electrical and Engineers, New York (2003)

BMRSS: BOM-Based Multi-Resolution Simulation System Using Components

Gongzhuang Peng, Huachao Mao, and Heming Zhang

State CIMS Engineering Research Centre, Department of Automation, Tsinghua University, Beijing 100084, P.R. China
pgz12@mails.tsinghua.edu.cn

Abstract. The Base Object Model, BOM, is specially identified as a reusable and composable model component for quick development of simulation models, which is very helpful for multi-resolution system.This paper proposes a framework named BMRSS including the process of components developement, management and simulation. BOMs library and components library are seen as special cases of web services, which further support the models reuse. MDA and XSLT technology are applied for codes auto-generation, and simulation components are generated directly from model documents. The key part of the framework is a 3-level resolution control mechanism: Resolution state chart is used to define the global resolution state, attribute dependency graph captures relationships among attributes between neighboring resolutions, connection BOM defines the entities and interplays related to resolutions. To support the Multi-Resolution component simulation, a dual-engine simulation is designed with an Internal Exchange Service Server (IESS) for each federate and the bottom supporting RTI. An air-to-air attack and defense scenarios which is built including red-side federates and blue-side federates demonstrates the effectiveness of the approach and corresponding tools.

Keywords: BOM-based, multi-resolution, component, resoultion control mechanism, simulation engine, RTI.

1 Introduction

Modeling and simulation are playing key roles in the engineering, academic and military fields. With the size and complexity of models required to be solved increasing fast, an exorbitant amount of computational hardware and network bandwidth is in great demand. However, when the magnitude of interacting entities in a large-scale simulation comes to millions which is common in the joint warfare community, normal computers cannot perform well any more. Besides improving capabilities of computer hardware, we have to deal with issues on modeling methods to effectively take advantage of such capabilities. A fairly standard technique now is to represent a model hierarchically, with various levels of resolution and depth of aggregation to be able to interact with one another, which is the so-called multi-resolution modeling (MRM).

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 485–496, 2013.

MRM is able to balance the relationship between simulation requirements and resource constraints by aggregating and disaggregating properly. The process of developing simulation models in the context of constructing large-scale distributed simulation systems is still time consuming though. By developing reusable model components with different resolutions, we can reduce the costs of development and validation of simulation models significantly. As simulation components can also be seen as a special case of web services, developers of different federates are allowed to utilize this service, which has further supported the models reuse.

The federate can be constructed by composing these reusable simulation model components. But how does one describe the components containing enough information about their internal structure and external interface? BOMs have been specifically identified as a potential facilitator for providing a specification to describe these components, which are selected and assembled, in a meaningful way through the various exposed metadata. A BOM can be thought of as a reusable package of information representing a pattern of simulation interplay.

In this paper, BOM-Oriented Multi-Resolution Simulation System (BMRSS) which involves the process of conceptual modeling, components developing, components assembling and federates simulation is introduced. The remainder of this paper is organized as follows: Section 2 gives an overview of work related to what is presented in this paper. Section 3 explains our approach in detail of developing multi-resolution model components based on BOMs. Then these components are simulated on our Dual Engine Simulation (DES) system in Section 4. A battlefield simulation is conducted to validate the feasibility and availability of BMRSS in section 5. Our conclusions and directions for future research appear in Section 6.

2 Related Research

In this section a brief overview of some of the closely related research activities is presented. A variety of representative MRM methods are firstly introduced, including the formalism and key technologies of MRM. Then we present the applications of BOM for MRM and component-based simulation.

2.1 Research on MRM

Natrajan [1] develops a MRM framework, UNIFY, which consists of techniques such as Multiple Representation Entities (MREs), Attribute Dependency Graphs (ADGs) and taxonomy of interactions. Unlike traditional approaches to MRM that simulate only one (or one kind of) model at any given time, such as aggregation-disaggregation (AD) and selective viewing (SV), a MRE incorporates concurrent representations of multiple models. Consistency within a MRE is maintained by using ADGs and mapping functions reflecting relationship between attributes across representations. While this approach satisfies MRM requirements effectively and solves the problem of maintaining consistency across multiple models on some degree, UNIFY has many disadvantages. In order to maintain the real-time consistency, models of all resolutions are required to run during the simulation lifetime, which incurs the highest

resource usage cost. The simulation resource is restricted and it may cause a deadly failure to the large-scale distributed simulation as a result.

Liu [2] proposes a formal multi-resolution modeling specification (MRMS) for MRM based on DEVS (Discrete Event Specification). To support MRM's character of changing structure dynamically, MRMRS extends internal elements of DEVS and develops a model family (MF) composed of multiple models. Compared with another representative formal description of MRM-SES/MB, which is proposed by Zeigler [3], MRMS describes the static structures as well as the interactive behaviors of models. This approach provides a guidance to develop a relatively universal description of MRM. Yet models constructed by DEVS are complicated and lack of information about resolution control.

In this paper, a MRM framework is developed based on BOM, which contains enough information for a model and can be used for the rapid constructions and modifications of simulations. In addition to standard BOMs that are used to represent model components, resolution-related BOMs are applied to describe the consistency and resolution control information.

2.2 BOM Applications

Since BOM was proposed by the Simulation Interoperability Standards Organization (SISO) in 2003, it has widely been used to support conceptual modeling and object modeling for Modeling and Simulation (M&S) and System Engineering projects [13].

Chase[4] from SimVentions analyzes the application of Encapsulated (ECAP) BOM which includes behavioral information for modeling a specific class and contains additional meta-data to support composability. Load balancing and federation optimization are examined to be benefits of applying MRMs by a demo federate. Moradi [5, 6] from the Swedish Defence Research Agency (FOI) use BOMs for component-based simulation development and utilize Web Service (WS) technology for further supporting of reuse and composability. National University of Defense Technology has developed a federate development environment named KD-SmartSim based on BOM and HLA simulation system. KD-SmartSim achieves parallel computing of the component-based simulation system on RTI. Junjie Huang [7] develops the multi-granularity modeling framework of a large-scale system with UML, while its detail design is completed with BOM. Yuan Li extends standard BOM with MRM-related information: consistency information and resolution control to support his MRM framework.

3 BMRSS Architecture

As is shown in Fig. 1. Architecture of BMRSS, BOM-based Multi-Resolution Simulation System (BMRSS) framework is made up of BOM model development, BCI development based on automatic code generator, components management and simulation management.

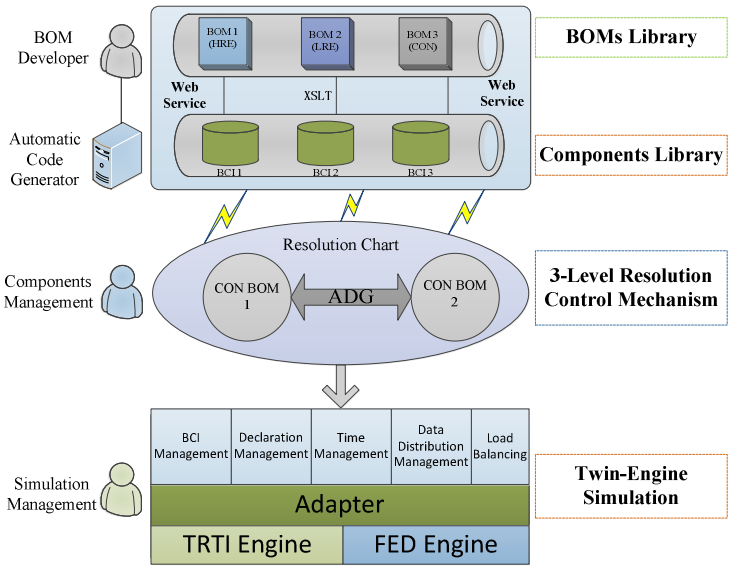


Fig. 1. Architecture of BMRSS

Entity BOMs of different resolutions and Con BOMs (Connection BOM) are developed in the beginning. In the process of code generation, MDA (Model Driven Architecture) which separates the logical behavior models from the implemental platform is introduced into the development of BOM-based simulation model components. The XSLT is applied for codes auto-generation, and the XSLT templates and the architecture of the component framework are designed. Both the BOM models and the components are developed as cases of Web services, which mean that they can be composed together and aggregated to deliver functionalities according to user requirements. To manage the components of multiple resolutions, a 3-level resolution control mechanism is proposed: Resolution State Chart is used to define the global resolution state, Attribute Dependency Graph (ADG) captures relationships among attributes in concurrent representations between neighboring resolutions, Connection BOM (Con BOM) defines the variables and states related to high resolution entity (HRE) and low resolution entity (LRE). Dual Engine Simulation (DES) has been designed to support the Multi-Resolution component simulation with an Internal Exchange Service Server (IESS) for each federate and the bottom supporting Run Time Infrastructure (RTI).

3.1 Code Auto-generation of BOM Components Based on Web Service

Structure of BOM Components

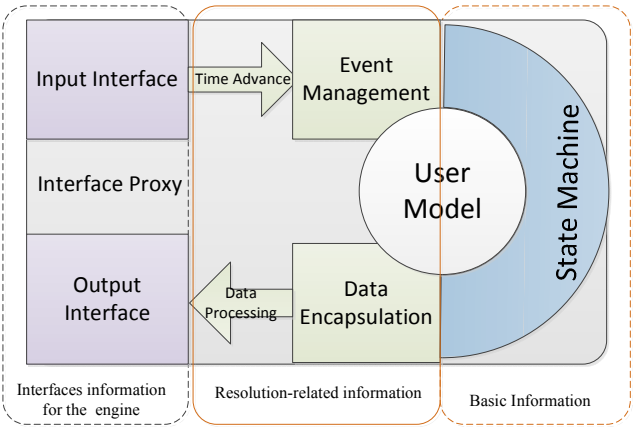


Fig. 2. Structure of a BOM Component

A simulation component is a software element of a simulation model with well-defined functionalities and behaviors that can be well identified in the process of software reuse [6]. Functionally, components support the system to select and assemble reusable simulation components in various combinations into simulation systems to meet user requirements. Fig. 2 shows the structure of a BOM component, which includes three parts: basic information of the model, interfaces information for the simulation engine and resolution-related information.

The basic information of a component is indeed a description of model, which contains metadata and elements of the component. Metadata shows the users of general information about the component itself – what it simulates and how it can be used, which makes the component easy to be understood and convenient to be reused by developers. As for the elements, the static structure of a component such as entities and attributes is defined by the HLA Object Model, while the dynamic behavior is defined by the patterns of interplay and the state machine. The static attribute information presents the description and data information which is necessary for model initialization and running. The behavior information represents the changing process of the model with advance of the simulation time by using the changing conditions, results and algorithms of the model that are defined in the The behavior information represents the changing process of the model with advance of the simulation time by using the changing conditions, results and algorithms of the model that are defined in the resolution-related information of a component. The resolution-related information also includes the process of event management, data processing and data Encapsulation. While the basic information and resolution-related information are platform independent, interfaces information for the simulation engine is implementation specific to a platform and programming language. It includes the input and output interfaces of the component which can be used to run on the simulation engine, and provides the function of publishing and subscribing.

Implementation of Code Generation

A BOM is fundamentally an XML document that contains the model description information and model data information for code generation of the component model. An XML document is defined into a well-formed and flexible text that satisfies a list of syntax rules. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web. In order to generate executable codes from a BOM XML document, XSLT is applied which is a formal language for transforming XML documents into other XML documents to extract useful information for the component. Fig. 3 illustrates the mapping rules of a BOM document to a component document. Generally, each element of the BOM is transformed into a class in component codes. Model Identification that describes the contact and reuse information corresponds to the Component Management Class that deals with assembling component. Patterns Description is converted into Engine Scheduling Class because of the state-change condition in this part. However, multiple elements of BOM might be turned into the same element of component by applying the rules. For example, Object Class includes elements from both the Entity Type and the HLA Object Classes. Users can customize and modify their own template files by using XSLT template provided by the code generator, thus they can control the output of the object code conveniently and flexibly.

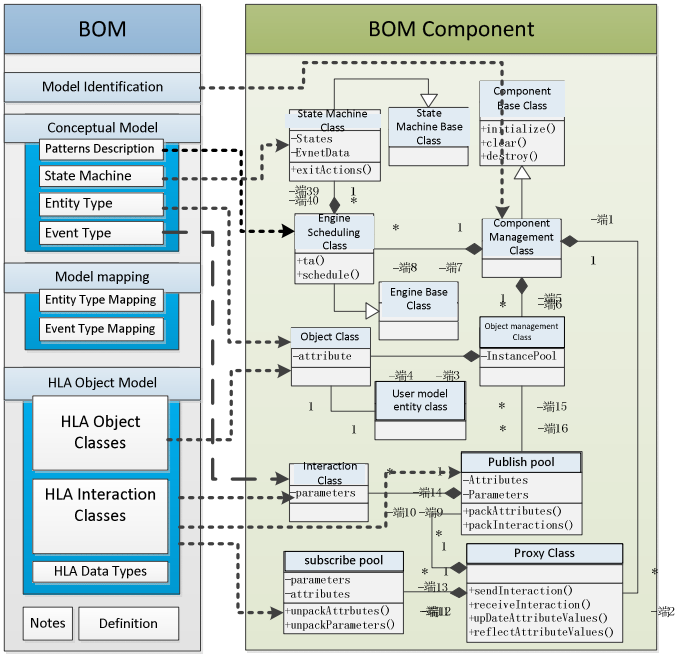


Fig. 3. Implementation of code generation

After the components are generated, they will be tested by component testing module including static and dynamic testing. Static testing checks out whether the construction and content of components are complete and tests interfaces of publishing and subscribing, while dynamic testing tests the load of components with different resolutions.

A simulation component is a software element of a simulation model with well-defined functionalities and behaviors as is the case with Web Services (WSs). And the platform and language independent interfaces of WSs allow the easy integration of heterogeneous models. So the code generator is developed based on the .NET technology, which offers an integrated support for Web pages efficiently. Developers who have the access to components library can search for components on the Web. A new component is created by filling in some parameters, so that the simulation modeler needs not be a sophisticated programmer any more.

3.2 3-level Resolution Control Mechanism for Components

A good resolution control method helps the simulation to achieve the best mix of computational and analysis resource, which makes MRM more effective and powerful [8]. In this paper, a 3-level Resolution Control Mechanism for Components is proposed to maintain consistency among the concurrent representations of models.

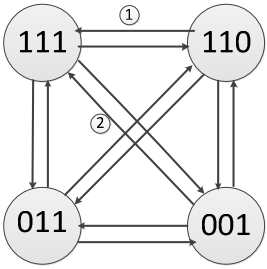


Fig. 4. First level - Resolution State Chart

The first level of the resolution control is named Resolution State Chart (RSC), which uses a binary vector to represent the resolution of current entities in simulation.

$$\left\{(a_1\, a_2\, a_3\, \cdots\, a_n)\,\middle|\, a_i = 0\, or\, 1, i = 1, 2, \cdots, n\right\}$$

Where n is the amount of resolution layers, and $a_i = 1$ means that the ith resolution component is running in the moment. Fig. 4 is a part of the RSC in a joint warfare community. As shown in Fig.4, {111} means that all the three levels are running. There are two types of transaction conditions labeled in the chart: ① is related to the trigger events that are set by the model developer, ② is determined by the simulation resources which are restricted by hardware condition.

The second level of resolution control is Attribute Dependency Graph (ADG) [2]. ADG captures relationships of attributes in concurrent representations among neighboring resolutions. In Fig. 5. Second level - Attribute Dependency Graph, we show all the attributes of entities as nodes labeled with appropriately-subscripted names.

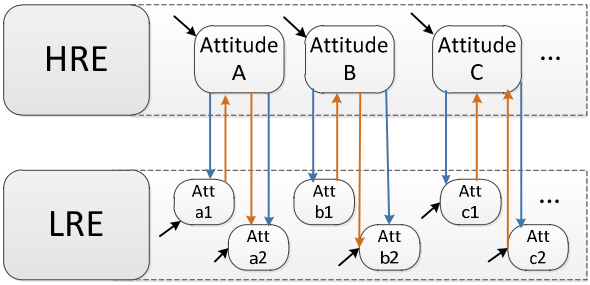


Fig. 5. Second level - Attribute Dependency Graph

The third level of the resolution control is Connection BOM (Con BOM) defining the variables and states related to high resolution BOM and low resolution BOM. Components are internally consistent and interact at multiple representation levels concurrently. **Table 1** below introduces the details of Con BOM.

Table 1. Contents of Con BOM

Name	BOM Element	Function
MapEntity	Entity Type	To record auxiliary properties for mappings
ResChange	Entity Type	To describe transaction conditions of resolution change
Aggregation	Pattern of interplay	To describe the sender, receiver, and the trigger event of aggregation action
Disaggregation	Pattern of interplay	To describe the sender, receiver, and the trigger event of disaggregation action

3.3 Component-Oriented Twin-Engine Simulation Architecture

Twin-Engine Simulation is designed to support the Multi-Resolution model simulation with an Internal Exchange Service Server (IESS) for each federate and the Run Time Infrastructure (RTI) to communicate and interact among federates, as is shown in Fig. 6.

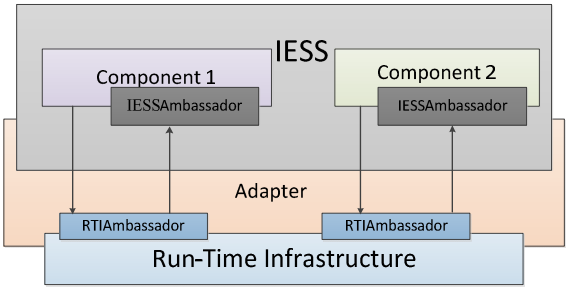


Fig. 6. Structure of Twin-Engine Simulation

IESS provide six services including the Resolution Control Management, Time Management, Components Management, Engine Management, Data Management and Statement Management. Resolution Control Management takes advantage of the 3-level Resolution Control Mechanism introduced above. Time Management guarantees the event time order transfer among the components according to the objective order. Components Management is responsible for destruction and initiation of components as soon as it receives Aggregation/Disaggregation request from a resolution-related instance. Components within a federate are assigned to different threads and Engine Managements will deal with resource assignment and load balance. Data Management transfers all the data and events among the components and output them to the adapter. State Management manages the input and output data of respective components statement.

RTI of the engine is developed by CIMS laboratory of Tsinghua University. It conforms to the IEEE 1516 API specifications and includes these services: federation management, declaration management, object management, ownership management, time management, data distribution management and support services.

In order to ensure accurate and efficient simulation of the Twin-Engine system, optimistic time advance mechanism and conservative resolution control algorithm are taken into consideration. Critical Time Synchronization algorithm is proposed to meet the dynamic changes of multiple models as well as time synchronization between two simulation engines. The critical time is a time during the execution of a simulation where a decision might be made, or the time at which component might change its resolution or behavior. If this decision can be made at any time during an interval, it is the latest such time. As is known, time management is essentially the computation of the Lower Bound Time Stamp (LBTS) across federates in a distributed simulation.

Fig. 7 shows a way to reduce the cost of LBTS query, in which LBTS is used only at critical time. It ensures that events are processed in time stamp order and simulation in the twin-engine is real-time.

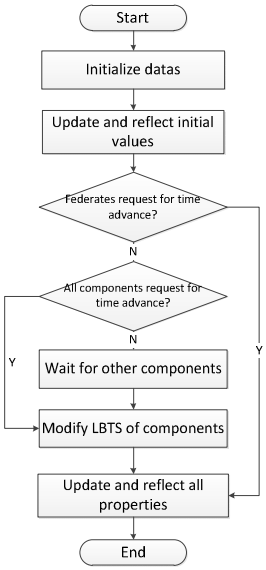


Fig. 7. Flowchart of Critical Time Synchronization algorithm

4 Example Realized by Our Tools

Tools for BMRSS are implemented after the previous analysis and design, which integrate all the functions of the mentioned modules. To demonstrate the application of the tools, an air-to-air attack and defense scenarios is built including red-side federates and blue-side federates. As is shown in Fig.8, blue-side federate includes a multi-resolution plane fleet and a multi-resolution sense system, while the red-side federate is composed of a multi-resolution enemy plane fleet. A blue-side federate represents the low resolution model and a plane fleet represent the middle resolution model which can disaggregate into several high resolution planes – wing plane, lead plane and so on. A same structure is applied to the red-side federate.

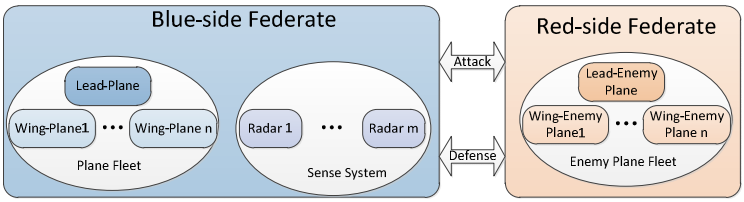


Fig. 8. Air-to-air attack and defense

In the simulation process, the blue-side planes search for the invading red-side plane in the mode of fleet. In this state all the components are running at a low resolution, since individual entity interacting is not required. When an enemy plane appears in the field of blue-side represented by a map in Fig. 9, the plane fleet of blue-side receives a

disaggregate order. Then both the plane fleet and the sense system run at a high resolution, during which the lead plane is responsible for the task of communicating with radar and sending attacking orders to wing planes. The simulation will end with the destroyed or expelled of red-side plane. To reduce the simulation cost maximally, components aggregate into low resolution ones. The tools implemented are proved to be very helpful for component modeling and effective for resolution controlling.

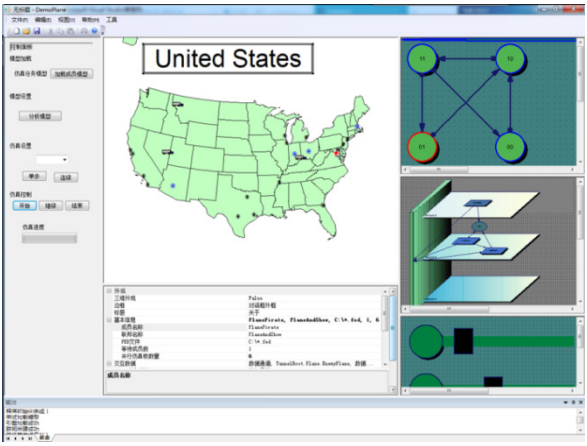


Fig. 9. Example realized by BMRSS

5 Conclusions

As a result of these efforts, the BOM component has shown effectiveness for solving many of the problems within large-scale joint simulation exercises which require multi-resolution modeling. BMRSS proposed in the paper can help the developers to build reusable component and executable codes, which makes the development convenient and flexible. The resolution control mechanism that includes RSC, ADG and Con BOM makes MRM more accurate and powerful. Dual Engine Simulation architecture makes it possible for components to interact and maintain consistency inside federate within IESS.

Acknowledgements. This work was supported by the National Key Technology R&D Program under Grant # 2012BAF15G00.

References

1. Natrajan, A.: Consistency maintenance in concurrent representations. University of Virginia (2000)
2. Baohong, L., Kedi, H.: A formal description specification for multi-resolution modeling (MRM) based on DEVS formalism. In: Kim, T.G. (ed.) AIS 2004. LNCS (LNAI), vol. 3397, pp. 285–294. Springer, Heidelberg (2005)

3. Zeigler, B.P., Sarjoughian, H.S.: Introduction to DEVS Modeling and Simulation with JAVA: Developing Component-Based Simulation Models, Arizona Center for Integrative Modeling and Simulation. University of Arizona and Arizona State University, Tucson, Arizona (2005)
4. Chase, T., Gustavson, P.: The Application of Base Object Models (BOMs) for Enabling Multi-Resolution Modeling. In: 2004 Fall Simulation Interoperability Workshop (2004)
5. Moradi, F., Nordvaller, P., Ayani, R.: Simulation model composition using BOMs. In: 10th IEEE International Symposium on Distributed Simulation and Real-Time Applications, pp. 242–252 (2006)
6. Moradi, F.: Component-based simulation model development using BOMs and web services. In: 1st Asia International Conference on Modeling & Simulation, pp. 238–246 (2007)
7. Huang., J., Zhang, H.: Multi-Granularity Modeling of virtual prototyping in collaborative product design. In: 12th International Conference on Computer Supported Cooperative Work in Design, pp. 710–715. Xi'an (2008)
8. Powell, D.R.: Control of entity interactions in a hierarchical variable resolution simulation. Los Alamos National Lab., NM (1997)
9. Mei, Y., Ying, C., Jian, H., Peng, Z.: Research on atomic component model development in BOM-based HLA simulation. In: 2nd International Conference on Software Engineering and Service Science, pp. 905–910 (2011)
10. Gustavson, P., Chase, T.: Using XML and BOMS to rapidly compose simulations and simulation environments. In: Simulation Conference Proceedings, vol. 2, pp. 1467–1475 (2004)
11. Qiang, H., Yong, P., Ming-xin, Z.: Parallelization of simulation engine for BOM component on multi-core. In: 3rd International Conference on Communication Software and Networks (ICCSN), pp. 250–254 (2011)
12. Xiaocheng, L., Bin, C., Ke, Z., Kedi, H.: Execution management of the BOM-based simulation system. In: Asia Simulation Conference-7th International Conference on System Simulation and Scientific Computing, pp. 287–290 (2008)
13. Chunguang, P., Qiang, H., Xiaocheng, L., Xinye, Z.: Component Scheduling in the Distributed Simulation based on BOM. In: Second International Conference on Computer Modeling and Simulation, pp. 98–102 (2010)
14. Base Object Models, `http://www.boms.info/`

VTF Gateway: A Solution for Interconnecting VTF and HLA

Xinlong Tan[1,2], Yiping Yao[1,2], Laibin Yan[2], Tengfei Hu[1,2], and Feihua Cai[3]

[1] College of Computer, National University of Defense Technology, Changsha, Hunan, China
wo8351txl@163.com, ypyao@nudt.edu.cn
[2] College of Information System and Management, National University of Defense Technology, Changsha, Hunan, China
lbyan@nudt.edu.cn, hutengfei-2007@163.com
[3] China Academy of Launch Vehicle Technology, Beijing, China
caifeihua@gmail.com

Abstract. VTF (Virtual Test Framework) is a useful infrastruction for virtual-testing applications. The existed models and applications are mostly built upon HLA specification. In order to make use of these accumulated achievements, it is necessary to interconnect VTF applications and HLA simulation systems using gateway. Since the application interfaces of VTF models is generated according to VFL (VTF Language) file, the gateway should be dynamically re-generated when interconnecting to a new VTF application. The architecture interfaces or time management of exsited gateway is not entirely suitable for VTF. To address these problems, a VTF Gateway generated method is proposed in the paper. After setting mapping relations between VTF data and HLA data by users, VGCT (VTF Gateway Generation and configuration Tool) can generate Runtime Gateway automatically. We also propose a solution for synchronizing the two equivalent heterogeneous systems. Experimental results demonstrate that VTF Gateway is valid and correct.

Keywords: VTF, HLA, heterogeneous interconnection, VFL/FOM mapping, code generation, time management.

1 Introduction

VTF (Virtual Test Framework) is a kind of typical general support framework with the aim of solving the problems caused by heterogeneity during developing and testing of military products [1]. It has been used to promote the synchronous development ability of military industrial products designing, virtual test and evaluation. It also provides support for the standardization in the process of test and evaluation. Users can define simulation models with VFL (VTF Language) and develop applications quickly with VTF application framework provided by VTF. In this way, development time can be reduced and models can be isolated from data.

Bridging heterogeneous simulation systems via gateway to build large-scale complex simulation systems becomes one of the hottest spots in simulation field. To date,

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 497–507, 2013.

many models based upon VTF have been coming out. In order to make use of existed resources, it is necessary to research the technologies to interconnect VTF application systems and HLA [2-4] simulation systems. The existing research on interconnection of heterogenous systems is mainly concentrated on the interoperability between HLA federations or HLA federation and other simulation systems. As far as we know, there are four main solutions [5]: federate gateway, bridge federate, RTI broker and RTI interoperation protocol. Benoit Breholee et al. identified the problem categories of federate gateway and put forward solutions to them. As a result, a kind of federate gateway was designed and implemented [6]. Dingel et al. conducted in-depth research on bridge technology of HLA [7] [8], using bridge federate implemented multiple HLA federal interconnection. Zhang compared HLA with TENA (Test and Training Enabling Architecture) and found the best way to achieve seamless interconnection for TENA is the TENA gateway [9]. Reference [10-12] also made some research to interconnect multiple simulation systems.

Federate gateway interconnects federates in different federations directly. That is to say, federates communicate directly through the gateway. The gateway does not support the HLA interface specification; it just transforms data and plays as a controler rule. Therefore, the workload of this gateway is always large. Meanwhile, the lack of generality is also a serious problem. RTI broker interconnects RTIs directly and communicates with them. It can provide efficient and convenient support for users to realize simulation of large-scale, but the premise is that the standard APIs to communicate with RTI should be provided. In order to realize the RTI interoperability protocol, standard interoperability protocol between RTI and its components should be defined.

Bridge federate syncretizes two federations as a member federate at the same time. However, for every simulation application, it should implement a new federate agent that will increase the user's burden. In this paper, we adopt and improve the architecture of bridge federate. Based upon this kind of improved architecture, we propose the solution for heterogeneous interconnection: VTF Gateway.

The paper is organized as follows. In section 2, we introduce the architecture of VTF Gateway in detail. The key technologies are illustrated in section 3. We validate VTF Gateway in section 4. In the last section, we draw a conclusion to VTF Gateway.

2 Architecture of VTF Gateway

VTF Gateway includes VGCT (VTF Gateway Configuration and generation Tool) and Runtime Gateway.

2.1 VGCT Architecture

In order to solve the problem brought by repeated development of Runtime Gateway, we create VGCT that is responsible for two tasks: Firstly, it provides visual interface for users to assign mapping relations between VTF models and HLA models; secondly, it generates VTF Gateway according to these mapping relations.

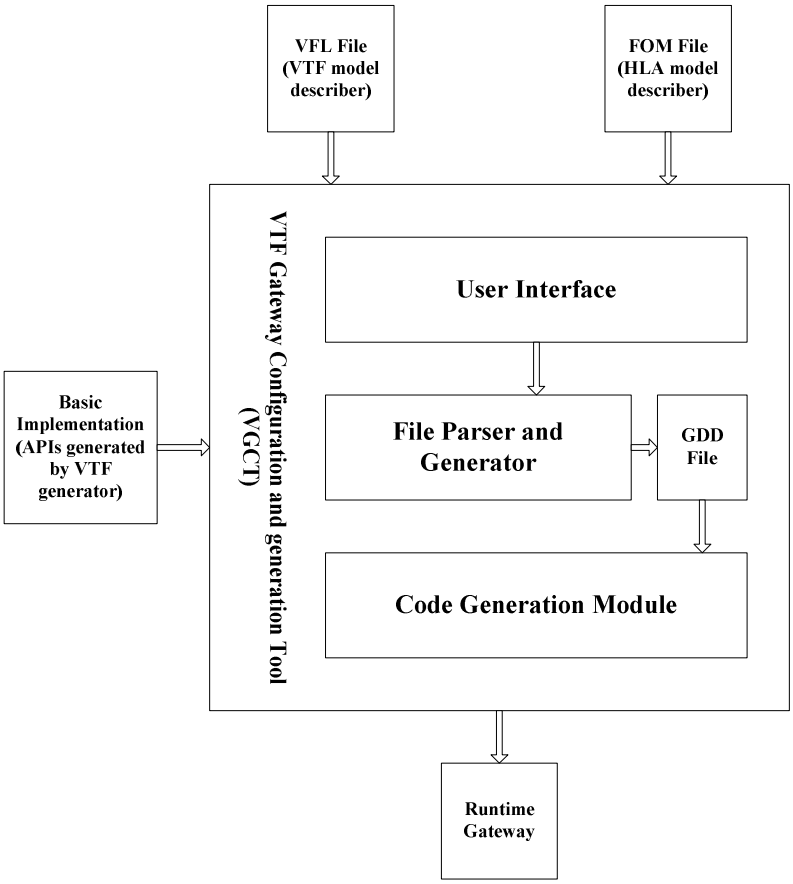


Fig. 1. VGCT architecture

Fig. 1 illustrates the architecture of VGCT. VGCT comprises three parts: user interface, file parser and generator, code generation module.

User interface accepts VFL file and FOM file as inputs and provides interface for users to assign mapping relations between VTF models and HLA models.

File parser and generator can parse VTF models and HLA models onto user interface. According to the parsed information, users configure mapping relations. On the other hand, file parser and generator generates GDD (Gateway Data Description) file to store the mapping relations.

Using GDD file, basic implementation and the code framework as inputs, code generation module generates Runtime Gateway.

2.2 Runtime Gateway Architecture

Runtime Gateway consists of three components: VTF agent module, HLA agent module and control module. See Fig. 2.

VTF agent will connect to VTF middleware as a VOM (VTF Object Model). Its main function is to get the data that are subscribed by HLA applications, from VTF applications and transform them to HLA agent. HLA agent will send the data to RTI finally.

Similarly, HLA agent joins in HLA federation by connecting to RTI. Its main function is to get the data from HLA applications and transform them to VTF agent. VTF agent will send the data to VTF at last.

The job for control module is to assist in initializing VTF agent and HLA agent, coping with data-transmitting threads, realizing synchronization between VTF agent and HLA agent. On the other hand, it will wait and receive user's input command.

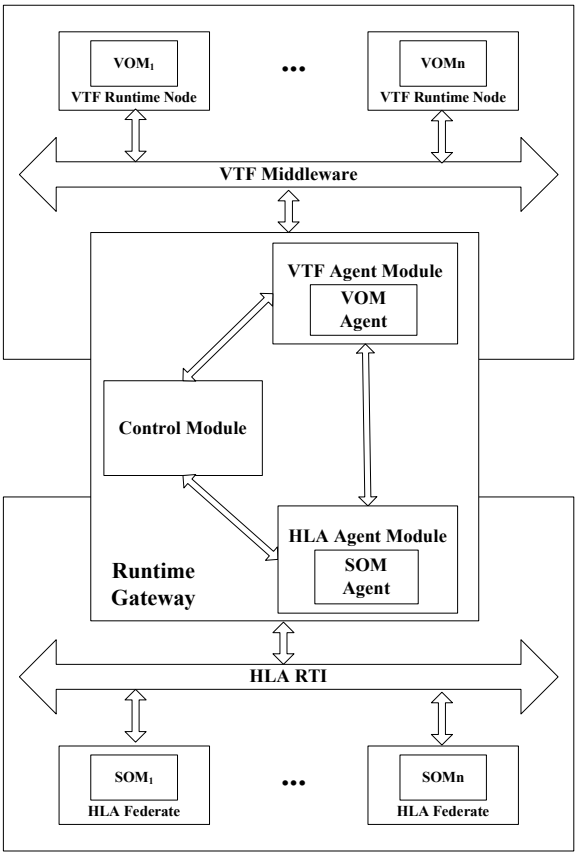


Fig. 2. Runtime gateway architecture

Fig. 2 illustrates the architecture of Runtime Gateway clearly. In this kind of architecture, VTF or RTI just need to exchange data with the agent of the respective modules. The control module, which manages and coordinates VTF agent and HLA agent, can promote the efficency of Runtime Gateway.

To sum up, the gateway is not only a VOM node but also a HLA federate. It has three main features:

- It is able to map VFL/FOM models, realizing data transmitting and filtering between the applications from the two sides.
- It can be generated by VGCT, thus can interconnect VTF and RTI from various manufacturers. On the other hand, the quality of code and efficiency of development are ensured.
- It provides time management between VTF application system and HLA simulation system, maintaining sequence of cause and effect relationship during simulation.

3 Key Technologies

3.1 VFL/FOM Mapping

VFL is a language used to describe attributes as well as operations of models that participate in the test on VTF. The core part of VFL meta-model is VTO, named Virtual Test Object (because of the function of publishing state, it is also named State Publishing Object), represented by "Class" key word in meta-model. Local Class and Message are also important metamodel, they are similar to "Struct" in C language. Other auxiliary meta-model includes Vector, Interface, VTO Pointer, Operation and Fundamental Type.

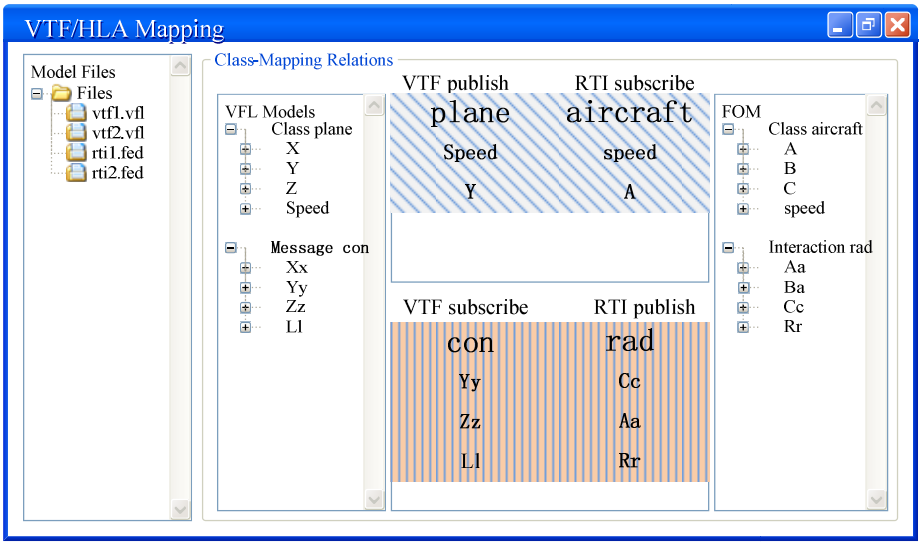


Fig. 3. A conceptual figure of configuring mapping relations in windows

In order to transmit and filter the data between VTF and HLA applications, VTF Gateway needs to know how to switch data from RTI to VTF or the other way around. Hence, VFL and FOM models mapping relations are necessary. When users assign the mapping relations using VGCT, VTOs in VTF and objects in HLA are mapped to each other; messages in VTF and interactions in HLA are mapped to each other in the same way (Fig.3 shows a conceptual figure of mapping). The mapping relations are refined to the layer of attributes or parameters. VGCT generates GDD file to retain the mapping relations.

GDD file is a file stores metadata of mapping relations. It includes five descriptions: SourceFileInfo, RuntimeEnvInfo, VtfPublishInfo, VtfSubscribeInfo and VtfMappingInfo. We use Vtf2RtiClass and Rti2VtfClass, mapping data type, to store class-mapping relations. Algorithm to map VFL/FOM is written as follows.

```
Begin
Load relations into Vtf2RtiClass and Rti2VtfClass.
Write VFL file name and FOM file name into SourceFileInfo
description.
Write VTF domain name and RTI feration name into Runti-
meEnvInfo description.
Foreach class mapping relation in Rti2VtfClass:
  Create a new VtfSubscribe node.
  Set vtfSubClassName and rtiPubClassName.
  Set bPublish = false.
  Create a VtfMappingInfo child node.
  Foreach attribute mapping relation:
      Create an AttrMapping child node in VtfMappingInfo.
      Set vtfAttrName and rtiAttrName.
  Endfor
Endfor
Foreach class mapping relation in Vtf2RtiClass:
  Create a new VtfPublish node.
  Set vtfPubClassName and rtiSubClassName.
  Set bPublish = true.
  Create a VtfMappingInfo child node.
  Foreach attribute mapping relation:
      Create an AttrMapping node in VtfMappingInfo.
      Set vtfAttrName and rtiAttrName.
  Endfor
Endfor
End.
```

3.2 Code Generation

Because of variability of VTF APIs and repeated development of bridge federate agent, VTF Gateway should be re-implemented each time when interconnecting to a

new VTF application system. Motivated by lightening the burden of users, ensuring quality as well as efficiency, we need to generate VTF Gateway automatically. In [13], we have discussed the details of code generation. Because of the complexity of code structure, we cannot use the method of writing information into code templates simply. Instead, we use GDD file and the code framework in the generation process. The way to generate VTF Gateway is shown in Fig. 4.

The code framework consists of a series of structured classes. For each class mapping relation, VGCT class generator will generate a manager class to manage its behavior, such as, declaration and destruction of publisher or subscriber, data updating et al.

Basic implementation, generated by VTF code generator, is the description and implementation of operations of VTF models. Only by calling these operations, we can update or obtain data from each model. In a word, basic implementation is the paragramming interface for applications to communicate with VTF. Using generated manager classes, metadata from GDD file and basic implementation as inputs, VGCT code generator generates all code files for VTF Gateway.

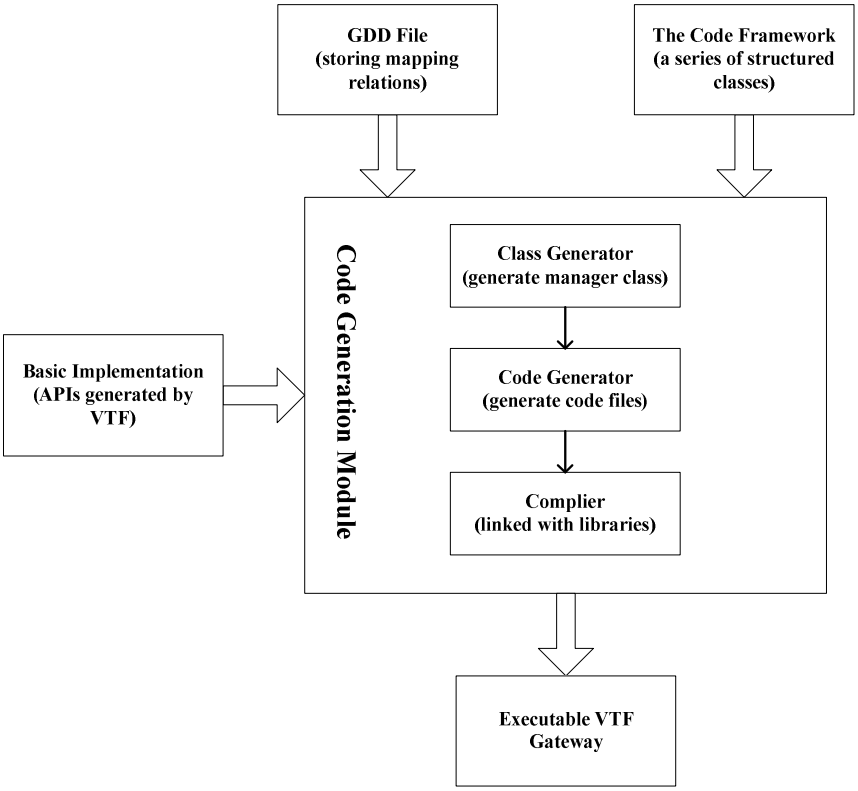


Fig. 4. VGCT Generates VTF Gateway

3.3 Time Management

When VTF application system and HLA simulation system are using time management on both sides, time management is also necessary for VTF Gateway. In order to ensure that time of heterogeneous systems advances correctly and orderly, we have conducted some research on time management for VTF Gateway [14].

Under the rules of the conservative time advance mode in HLA, the federate is forbidden to send any message with the timestamp less than T + L(L represents the value of lookahead) after making a request to advance to time T by calling TimeAdvanceRequest(T) function. At the same time, RTI needs to send all of the messages, the time stamp of which is less than T, to this federate. If VTF application system and HLA simulation system both follow this rule, a problem arises. Before VTF Gateway making a request to advance time T to RTI (step ② in Fig. 5), VTF Gateway should send all of messages with timestamp less than T to RTI (step ①). In order to get all of these messages from VTF (step C), VTF Gateway should make a request to advance to time T to VTF first (step B). On the other side, before requesting to VTF, VTF Gateway should send all of messages from RTI to VTF (step A). The dilemma arises: who should make the advance request first?

The deadlock is presented in Fig. 5 and Fig. 6.

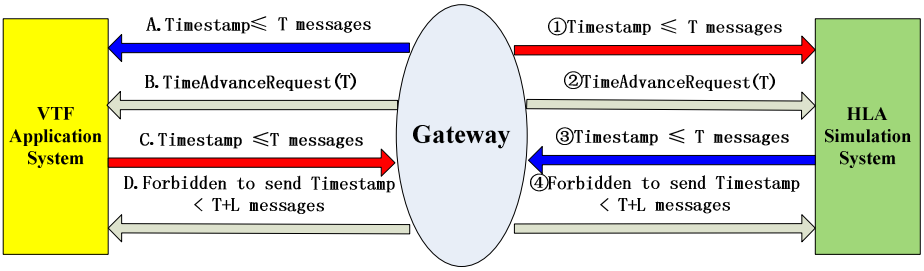


Fig. 5. Deadlock of time advancement

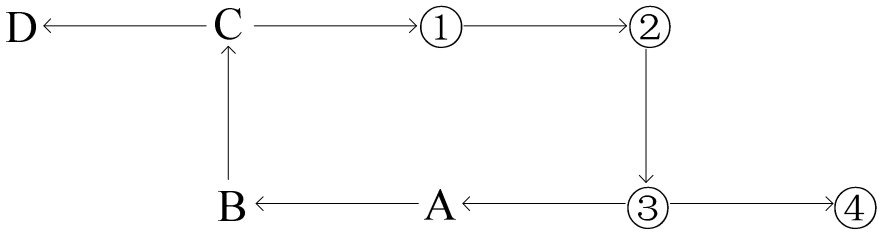


Fig. 6. Order of time advancing

In order to find a way out of this dilemma, we must break the equivalent situation of HLA and VTF system. That is to say, either HLA system should be considered as a VOM application of VTF system or VTF system should be considered as a federate of HLA system. If we considered HLA system as a VOM application, it is necessary but seems impossible to modify the structure of HLA run-time infrastructure. In the case

of considering VTF application system as a federate of HLA simulation system, when all of the nodes in VTF application system have made a time request to advance time within a time step, VTF will make a request to VTF Gateway (step ② in Fig. 7). Then VTF Gateway figures out the time value to advance and makes a request to RTI (step ④). After receiving the request, RTI will decide whether to meet this request or not by calculating. If yes, RTI sends all the messages with timestamp lesser than T to gateway (step ⑤). TimeAdvanceGrant (T) will be sent to gateway afterwards (step ⑥). At last, gateway will send these messages (step ⑦) before sending GateWayTimeAdvancegGrant (T) to VTF (step ⑧). The deadlock is broken in this way. Time synchronization between VTF and HLA is realized!

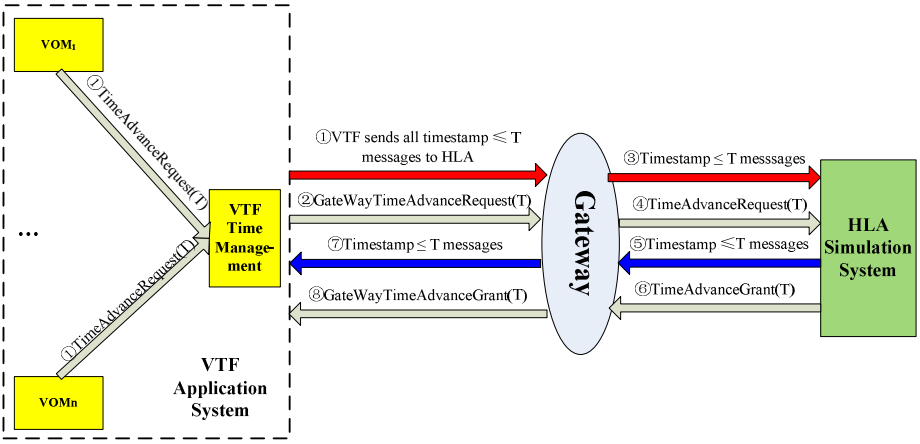


Fig. 7. Time advancement of "VTF-System" federate

4 Validation

VTF Gateway interconnects applications of Mak RTI [15] and VTF. Under 10 senders-40 receivers and 20 senders-30 receivers working conditions, every application updates a data per 10ms.

Table 1. Experiment environments

Operating system	Window xp
Programming language	C++
Network	100Mbps ethernet
RTI	Mak RTI, version4.1.1
VTF	VTF, v2.0a
Mapping relations(Publish→Subscribe):	
Object.BaseEntity(RTI)	OMsample::Platform(VTF VTO)
OMsample::LocalMessage(VTF Message)	Interaction.WeaponFire(RTI)

Fig. 8 presents a screenshot of started gateway.

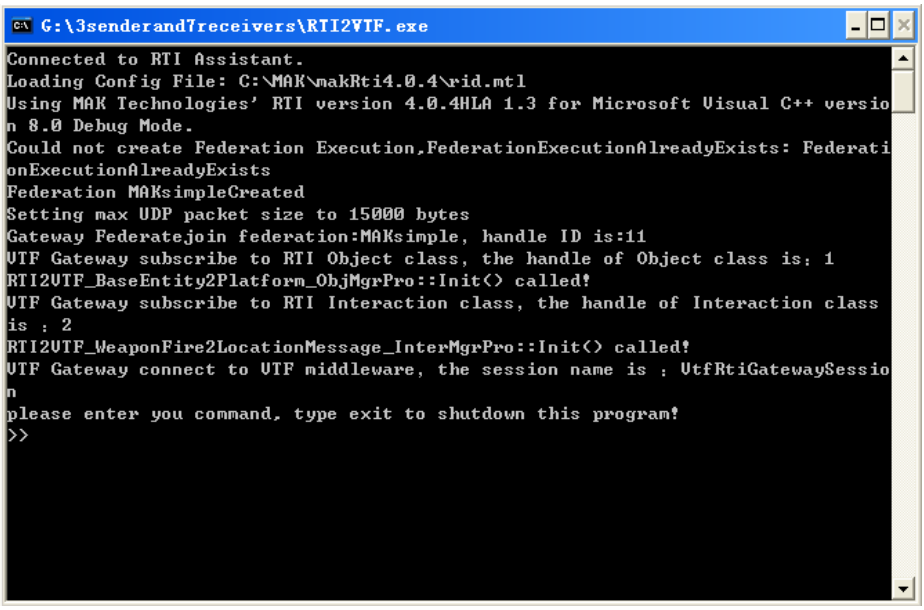


Fig. 8. Screenshot of started gateway

The experiment demonstrates that data are transmitted and filtered correctly in two working conditions, indicating that classes and attributes are mapped correctly. VTF Gateway can be generated rightly and be able to support the interconnection of VTF and heterogenous HLA simulation system.

5 Conclusion

In this paper, we have presented the VTF Gateway used as architectual glue for composing multiple heterogeneous simulation system. Firstly, we introduced VTF and existed researches on gateway technologies. Sequently, we presented the overall architecture and key technologies of VTF Gateway. Lastly, we conducted an experiment to validate VTF Gateway.

VTF Gateway architecture comprises VGCT and VTF Gateway. VGCT can map VTF models and HLA models and then generate VTF gateway, tackling the repeated development problem, lightening the burden of users as well as ensuring quality and efficiency. The time management solution we proposed adressed the dilemma of synchronizing equivalent heterogeneous systems.

As the practice shows, VTF Gateway is valid, feasible, and correct. It can be generated by VGCT quickly and completely automatically. In a certain scale range, it can map and transmit data rapidly and correctly.

Acknowledgment. We appreciate the support from National Science Foundation (NSF) of China grant (No.61170048) and the Ph.D. Programs Foundation of Ministry of Education of China (No. 20124307110017).

References

1. Zhao, W., Peng, J.: Research and Development on Technology of Virtual Test and Rvaluation of Military Industrial Products. Computer Measurement and Contrl 19(6), 1257–1277 (2011) (复杂军工产品虚拟试验验证技术研究与发展)
2. IEEE 2000a. IEEE 1516 Standard for Modeling and Simulation (M&S). High Level Architecture (HLA)—Framework and Rules (2000)
3. IEEE 2000b. Standard for Modeling and Simulation (M&S). High Level Architecture (HLA)—Federate Interface Specification, IEEE 1516.1 (2000)
4. IEEE 2000c. Standard for Modeling and Simulation (M&S). High Level Architecture (HLA)—Object Model Template Specification, IEEE 1516.2 (2000)
5. Myjak, M.D., Clark, D., Lake, T.: RTI Interoperability Study Group Final Report. In: Proceedings of the Simulation Interoperability Workshop (1999)
6. Breholee, B., Siron, P.: Design and Implementation of a HLA Inter-federation Bridge. In: EURO SIW (2003)
7. Dingel, J., Garlan, D., Damon, C.A.: A Feasibility Study of the HLA Bridge. School of Computer Science, Carnegie-Mellon Univ. Pittsburgh PA (2001)
8. Dingel, J., Garlan, D., Damon, C.A.: Bridging the HLA: Problems and Solutions. In: Sixth IEEE International Workshop on Distributed Simulation and Real-Time Applications, pp. 33–42. IEEE (2002)
9. Zhang, X., Zhai, Z., Feng, P.: Research on Technology of Distributed Gateway fro Heterogeneous Range Simulation. Computer Measurement and Contrl 19(9), 2263–2265 (2011) (靶场仿真中分布式异构网关技术研究)
10. Boer, C.A., Verbraeck, A.: Connecting High Level Distributed Simulation Architecture: An Approach for a FAMAS-HLA Bridge. In: Proceedings of the 14th European Simulation Symposium, pp. 23–26 (2002)
11. Cai, W., Turner, S.J., Gan, B.P.: Hierarchical Federations: An Architecture for Information Hiding. In: Proceedings of the 15th Workshop on Parallel and Distributed Simulation, pp. 67–74 (2001)
12. Gan, B.P., Liu, L., Turner, S.J., Cai, W., Hsu, W.: Distributed Supply Chain Simulation Across Enterprise Boundaries. In: Proceedings of 2000 Winter Simulation Conference, pp. 1245–1251. IEEE, Orlando (2000)
13. Tan, X., Yao, Y., Yan, L., Cai, F.: Code Generation for VTF Gateway. In: 2013 International Conference on Information System and Engineering Management, pp. 468–473. IEEE, Changsha (2013)
14. Hu, T., Yao, Y., Jiang, Z., Cai, F.: Research on Technology of Time Synchronization between Heterogeneous Systems HLA and VITA. In: 2013 International Conference on Information System and Engineering Management, pp. 478–482. IEEE, Changsha (2013)
15. Mak RTI 4.1.1 ReferenceManual, `http://www.mak.com/`

The Stratospheric Airship Non-forming Descent Path Simulation and Analysis

Zi-long Cong, Hui Ning, and Fan Yang

Northwest Institute of Nuclear Technology, Xi'an 710024, Shanxi, China

Abstract. In recent years, with the development of aviation technology, the airship has many advantages compared to conventional aircraft by the concern in many countries. United States, Germany, Russia, China and many other countries show great interests in the potential applications of the airship. Because the airships volume is very huge, so the landing process will inevitably cause a threat to the security of airline and ground facilities, especially the use of non-forming recovery method of stratospheric airship, the landing of security control is more important. This paper analyzes the forces on airship, modeling the airship non-forming decline model, and establishing a drop in direct sunlight, ground reflection, infrared radiation, convection exchange thermal model. According to the parameters of a test platform, the paper simulates the airship track of non-forming declining. Use the experiment data to analyze the simulation results and verify the validity of the model.

Keywords: Stratospheric airship, track simulation, non-forming descent model.

1 Introduction

In recent years, with the development of aviation technology, airship has many advantages compared to conventional aircraft, so it has been concerned by many countries. United States, Germany, Russia, China and many other countries show great interests in the potential applications of the airship. Because the airship's volume is very huge, so the landing process will inevitably cause a threat to the security of airline and ground facilities, especially the use of non-forming recovery method of stratospheric airship, and the landing of security control is more important. If the airship non-molding landing track can be predicted, it is possible to plan ahead based on the predicted results of safe area, improve the safety factor of airship recovery.

Wang [1], Wang [3] etc., who applied theories like statics, dynamic to build up the airship model in 6 degree, and using linear process for non-linear model by Linear perturbation theory, study on the airship modeling, especially on the controlling problem when the airship shape is the same. Shi[2] simulated the rising process of stratospheric airship, but neither carried out simulation verification, nor simulated the mode of non-forming descent process. Farley. Roberto Palumbo has carried out considerable modeling in the aspects of statics, dynamic, thermodynamics during the

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 508–518, 2013.

blastoff and residency process of balloon. This paper analyzes the airship non-forming descent process, builds up the non-forming descent model, implements the simulation software according to the model, carries out the simulation using a particular platform parameters, compares the simulated data and the experimental data, and test the correctness of the model.

2 Non-forming Descent Model

When the airship is non-forming descending, we first open the helium valve (it is normally located in the top of airship, which is easy to descend and exhaust). Because the internal airship has a certain pressure differences with the external environment, helium gas will exhaust from the valve. In the initial stage of exhaust, airship is in the state of overpressure, the volume stays the same, so the buoyancy keeps still, while the gravity decreases after the helium exhaust, which leads to the rising of airship. When the overpressure is finished, the airship volume starts to decrease, which leads to the buoyancy of the airship decreases, and then starts to descend.

During the modeling of analysis the force in the airship descent process, we have the following assumptions:

(1) Because of the non-forming descent strategy, the distance between mass center and buoyancy center increases, this leads to the airship hard to turn over during the descent process, so we can assume that the gravity and buoyancy are in the same line during the airship descent process.

(2) Because of the non-forming descent strategy, the airship does not have the pneumatic shape, so this paper computes the shape model as sphere to make it more simplify.

In the descent process, the external force of the airship includes aerodynamic, fluid inertia force, gravity, and buoyancy, etc. The gravity of airship equals to the sum of the gravity of airship internal gas, capsule, and accessories. The computation equation is as 1.

$$G = (V_{he}\rho_{he} + V_{air}\rho_{air})g + G_0 \tag{1}$$

Where V_{he} is the volume of helium capsule, ρ_{he} is the density of helium, V_{air} is the volume of air capsule, ρ_{air} is the density of air, and G_0 is the gravity of airship accessories.

The buoyancy equation is as follows according to the Archimedes Law.

$$B = V_{he}\rho_{air}g \tag{2}$$

Reference [8] gives the computation mode when fluid inertia force is considered as the additional mass, shown as Eqs. 3-4.

$$m_{xy} = 0.5\rho_{air}V_{gas} \tag{3}$$

$$m_z = 0.5\rho_{air}V_{gas} \tag{4}$$

2.1 Aerodynamic

2.1.1 Vertical Resistance

After simplification, the descent process of airship can be simplified by the descent of a sphere with radius R when it is has a certain volume, the air resistance computation equation is shown as Eq.5.

$$F_{zc} = \frac{1}{2}C_z\rho_{air}S_zV_z^2 \tag{5}$$

Where C_z is the resistance coefficient of airship, ρ_{air} is the air density, S_z is the front face area, V_z is the vertical speed.

The volume of Airship gradually decreases during the descent process, while the speed in the vertical direction will gradually increases, so the bottom of the airship will appear a concave, shown as Fig. 1.

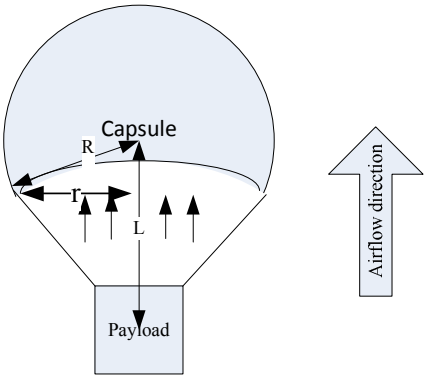


Fig. 1. Airship descent process

After a period of descent, the internal gas of the airship only has a little left, the external capsule eventually forms a thing like parachute. References [9,10] studied on the simulation model of parachute, the air resistance model of the parachute is similar to Eq. 5 during the descent process. Where the radius equation of the formed parachute is shown as Eq. 6.

$$r = R^2/L \tag{6}$$

The resistance coefficient is 0.9.

2.1.2 Lateral Thrust

Airship is affected by the upper wind during the descent process, while the upper wind is affected by measurement, so it is necessary to assume that the speed $\vec{V}_{wind}$ is in the horizontal plane, so the lateral thrust of non-forming airship can be expressed by Eq. 7.

$$\vec{F}_{xyc} = \frac{1}{2} C_{xy} \rho_{air} S_{xy} \vec{V}_{air}^2 \tag{7}$$

Where C_{xy} is the lateral thrust coefficient of airship, ρ_{air} is the air density, S_{xy} is the front face area, $\vec{V}_{air}$ is the relative air speed , so $\vec{V}_{air} = \vec{V}_g - \vec{V}_{wind}$.

2.2 Gas Exhaust Model

In the descent process, airship opens the helium capsule valve in its top, and the valve exhaust speed is subjected to the pressure difference between the gas in the two sides of valve and the density of internal gas. Reference [7] provides the speed when gas goes through the valve, shown as Eq. 8.

$$V_{flow} = C_{pai} \cdot \sqrt{2 \frac{\Delta P}{\rho_{he}}} \tag{8}$$

Where C_{pai} is the exhaust coefficient of valve, ΔP is the pressure difference of the two types of gas, ρ_{he} is the exhaust helium density, so the gas mass changing equation is shown as Eq. 9.

$$dM_{he} = V_{flow} \cdot A_{valve} \cdot \rho_{he} \cdot dt = A_{valve} \cdot C_{pai} \cdot \sqrt{2 \rho_{he} \Delta P} \cdot dt \tag{9}$$

2.3 Descent Process Dynamics Equation

According to Newton's second law, the airship motion equations in the vertical and horizontal direction are as Eq. 10-11.

$$B - G - F_{zc} = m \frac{dV_z}{dt} \tag{10}$$

$$\vec{F}_{xyc} = m \frac{d\vec{V}_{air}}{dt} \tag{11}$$

3 Airship Thermodynamics Model

The descent process of airship contains complication heat transfer process. Different researches show that the "extra-thermal" affects the descent process, so it is necessary to study the heat transfer modeling between the descent process and environment and its internal, based on the analysis modeling of dynamic and statics. References [7, 8] show that in the process of aerostat blastoff, residency, and return, the energy exchange mainly includes direct solar radiation, cloud reflection solar radiation, ground reflection solar radiation, ground infrared radiation, atmospheric infrared radiation, capsule self-radiation, forced convection cooling, capsule internal, and natural convection cooling energy. Shown as Fig.2.

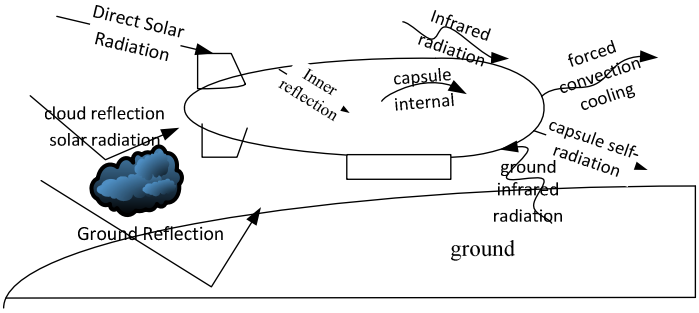


Fig. 2. Energy exchange diagram of airship in the air

According to reference [7], direct solar energy is expressed by Eq. 12.

$$Q_{Sun} = a \bullet A_{project} \bullet I_{Sun} \bullet \tau_{atm} \bullet \left[1 + \tau \bullet \left(1 + r_{effective} \right) \right] \tag{12}$$

Ground reflection energy can be expressed by Eq. 13.

$$Q_{Albedo} = a \bullet q_{albedo} \bullet A_{surf} \bullet \tau_{ViewFactor} \bullet \left[1 + \tau \bullet \left(1 + r_{effective} \right) \right] \tag{13}$$

Infrared Environment can be expressed by Eq. 14.

$$Q_{IREarth} = a_{IR} \bullet q_{IREarth} \bullet A_{surf} \bullet \tau_{ViewFactor} \bullet \left[1 + \tau_{IR} \bullet \left(1 + r_{effective} \right) \right] \tag{14}$$

The convection heat transfer between airship and atmosphere contains the convection heat transfer of the external surface and atmosphere and the convection heat transfer of internal surface and buoyancy gas. The external convection heat transfer contains the natural convection and forced convection. The heat transfer mainly relates to the transfer area, temperature difference, and transfer rate of the two transfer mass, where the convection heat transfer between external surface and atmosphere environment is shown as Eq. 15.

$$Q_{ConvExt} = H_{external} \bullet A_{effective} \bullet \left(T_{air} - T_{film} \right) \tag{15}$$

The convection heat transfer between the internal surface and buoyancy gas is shown as Eq.16.

$$Q_{ConvInt} = H_{internal} \bullet A_{effective} \bullet \left(T_{film} - T_{gas} \right) \tag{16}$$

According to the first law of thermodynamics, the energy conservation equation of airship internal gas can be expressed by Eq. 17.

$$\frac{m_{he}}{M_{mol}} C_{v \cdot he} \frac{dT_{he}}{dt} = dQ - p_{he} \frac{dV_{he}}{dt} \tag{17}$$

Where m_{he} is the helium mass, C_{phe} is the helium specific heat under constant pressure, T_{he} is the temperature, p_{he} is pressure intensity, dQ is the heat transfer between airbag and helium. Eq. 17 can be expressed by Eq. 18 after simplification.

$$\frac{dT_{he}}{dt} = \frac{Q_{ConvInt}}{c_v \bullet m_{he}} + (\gamma - 1) \bullet \frac{T_{he}}{\rho_{he}} \bullet \frac{d\rho_{he}}{dt} \tag{18}$$

The capsule temperature can be expressed by Eq. 19

$$\frac{dT_{film}}{dt} = \frac{\left(Q_{Sun} + Q_{Albedo} + Q_{IREarth} + Q_{atms} - Q_{IRfilm} + Q_{ConvExt} - Q_{ConvInt} \right)}{c_f \bullet M_{film}} \tag{19}$$

Where Q_{IRfilm} is the amount of infrared emission of airship capsule.

4 Simulation Software Design

According to the model in the previous two sections, we apply C++ to implement the simulation program of the airship non-forming descent process, and the program interface is shown as Fig. 3. The figure shows that the software is composed of four areas. The left part is the input area, the upper part is the important simulation parameter display area, the bottom part is the simulation controlling area, and the middle part is the related parameter graph manipulation display area.

Parameter setting area is used to set the related simulation initial parameters. This figure shows the important parameters in the simulation, and more detailed parameter inputs are shown in the last page, e.g. ground reflection rate, valve vent rate, etc.

The display area mainly shows the current simulation time, the distance between a certain point, the latitude and longitude of current simulation point. The controlling area mainly controls the start and suspend of simulation, which is also able to control the interval of each step.

Fig. 3. Simulation software interface

The graph display area applies OpenGL to display the estimated 2D track, and applies curves to show the changes of height, temperature, pressure difference, density, upper wind and the speed that changes with time.

5 Simulation Result Comparison Analysis

In order to test the validation of the stratospheric airship non-forming descent model, the simulation results were compared with a particular flight experimental data.

5.1 Speed Data Comparison

Figs 4-5 is the east direction speed comparison curve and the north direction speed comparison curve.

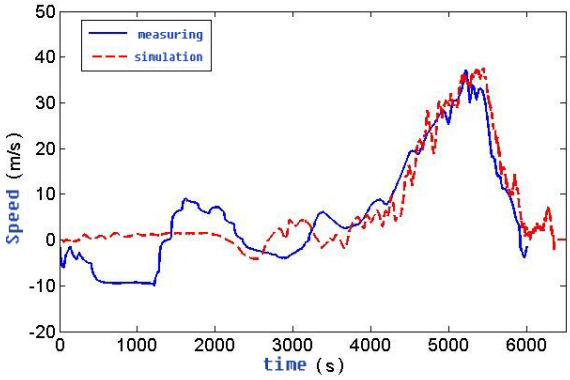


Fig. 4. East speed data comparison diagram

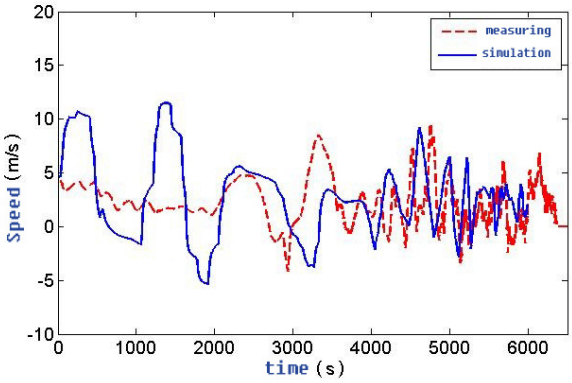


Fig. 5. North speed data comparison diagram

The above two figures show that the east direction speed curve is fitted well in the upper part, but the simulation data and measurement data have large deviations before 3000 seconds.

Fig 6 is the simulation data and measurement data comparison diagram in the vertical direction.

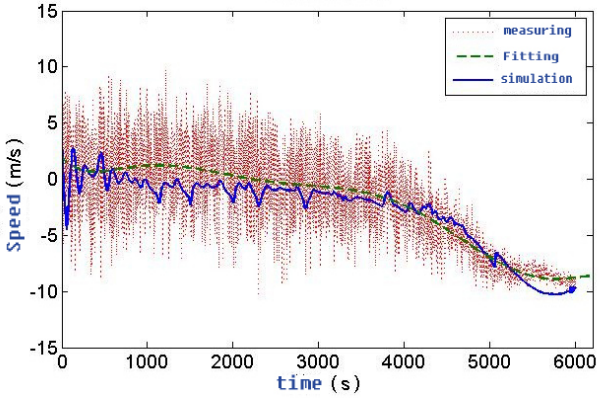


Fig. 6. Vertical speed comparison curve

The figure shows that the green fitting curve fits well with the blue simulation curve.

5.2 Displacement Data Comparison

Fig.7 is the displacement data comparison diagram in the west-east direction.The figure shows that the airship mainly flies to the east in the descent process, because displacement is the integral of the speed, it is same with the west-east speed comparison, the entire curves fit well, the largest error appears in the descent stage.

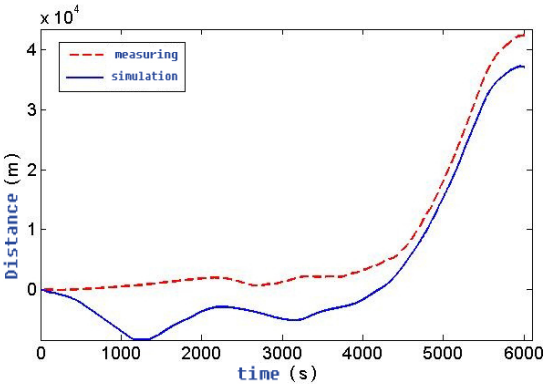


Fig. 7. West-east displacement comparison diagram

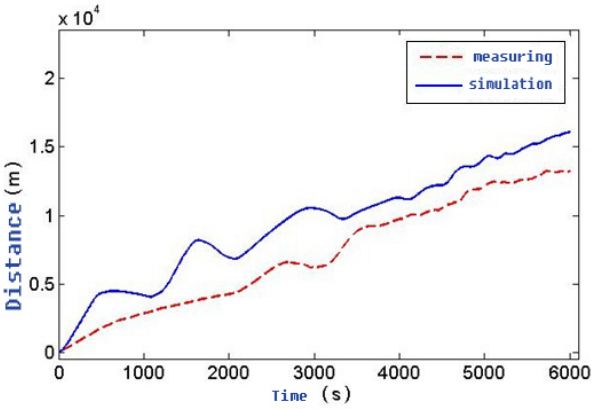


Fig. 8. North-south displacement comparison diagram

Fig.8 is the displacement data comparison diagram in the north-south direction. The figure shows that the airship mainly flies to the north in the descent process. Because the displacement is the integral of speed, it is the same with the north-south speed comparison, the errors mainly appear in the first 4000 seconds, and the placement difference is about 1.7km.

Fig. 9 is the height comparison curve. The figure shows that the vertical speed gradually increase when the platform starts to exhaust, shown as Fig.9, the height curve has larger errors from 1500 seconds to 4000 seconds, and the largest error is about 1km.

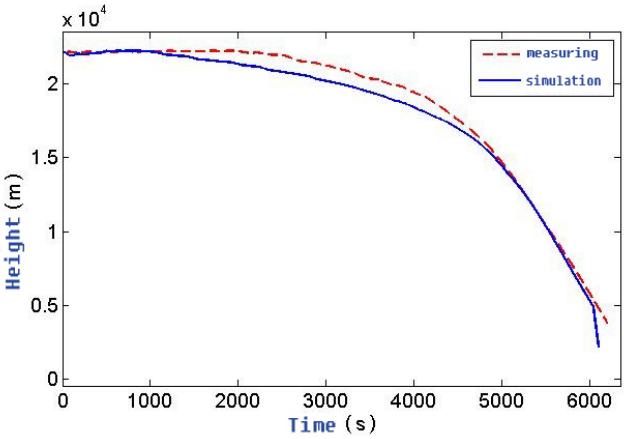


Fig. 9. Height comparison

6 Conclusion

This paper builds up the thermodynamics and dynamics combined airship non-forming descent model, and makes data simulation on the particular platform descent process, compares the simulation data and real measurement data. The data show that the stratospheric airship non-forming descent model simulation result fits well with experimental measured data, which reflects the stratospheric using non-forming descent mode, as well as the changes of airship platform's different physical parameters.

References

1. Wang, R.-P.: Stratospheric Airship Modeling and Decoupling. Journal of Projectiles Rocket Missiles and Guidance (2005)
2. Hong, S.: Numerical Simulation of a Stratospheric Airship During Its Ascending. Missile and Space Vehcile (3) (2008), Sum No.295
3. Wang, M.-J.: Modeling and Simulation Analysis of Stratospheric Airship Platform. Computer Simulate 5 (2008)
4. Lee, S.-J., Kim, S.-P., Kim, H.-K., Lee, H.-C.: Development of Autonomous Flight Control System for 50m Unmanned Airship. In: Proceedings of the 2004 IEEE, pp. 457–462 (2004) ISSNIP2004
5. Kim, D.-M., et al.: Korea Stratospheric Airship Program and Current Results. In: AIAA's 3rd Annual Aviation Technology, Intergration, and Operations(ATIO) Tech, November 17-19, Denver, Colorado (2003)
6. Schuch, E.M.: Towfish Design,Simulation and Control. Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirement for the degree of Masters of Science in Aerospace Engineering 6 (2003)

7. Farley, R.E.: Balloon ascent: 2-D simulation tool for the ascent and float of high altitude balloons. AIAA 2004-7412 (2005)
8. Roberto Palumbo:Analysis Code for High-Altitude Balloons. AIAA 2007-6642
9. Natke, H.G.: Die korrektur des rechenmodells eines elasomechanischen systems mittels gemessener erzwungener schwingungen. Internation Journal of Mechanical Science 32(3), 191–203 (1980)
10. Ljung, J.: System Identification: Theory for the Uses. Prentice Hall PTR, Englewood Cliffs (1987)

Simulation-Based Control for CPS Application to Aircraft Flexible Assembly

Hui Li, Linxuan Zhang, Tianyuan Xiao, and Jun Liu

State CIMS Engineering Research Center
Tsinghua University
Beijing, 100084, P.R. China
lihui08@mails.tsinghua.edu.cn

Abstract. Based on the ideology of cyber physical systems and combining the requirements of aircraft flexible assembly, this paper puts forward ACPS (Assembly-oriented Cyber Physical Systems) and demonstrates its architecture via contrasting between ACPS and traditional assembly system. By taking aircraft components docking as an example, this paper also introduces ACPS's software and hardware requirements, system composition, functional module division of assembly simulation control system, as well as key issues to resolve. ACPS then will be actualized to improve the level of aircraft assembly as well as provide theoretical and experimental support for next generation smart assembly systems.

Keywords: Flexible Assembly, CPS, Virtual Simulation, Real-time Control.

1 Introduction

Aircraft assembly is an essential part of aircraft manufacturing. In accordance with the principle of size coordination, aircraft assembly combines and connects aircraft parts and components according to the design and technique requirements by adapting assembly fixtures and equipment, thus forming higher level assembly parts or overall units. Aircraft assembly technique has so far experienced the following developing processes of manual assembly, semi-mechanical/semi-automated assembly, mechanical/ automated assembly, flexible assembly[1]. The application of 3D digital assembly simulation technique can solve the ignored problems during process of traditional assembly design, and avoid the losses of products, periods, manpower, costs caused by the false assembly methods. Currently, the major aerospace companies at abroad widely adopt digital tools to design products, processes and tools, especially simulate 3D digital assembly process under virtual environment, and put the results of design and simulation into actual products' assembly; the scholars at home also carry out many related research work, such as AVIC(Aviation Industry Corporation of China) design having realized automatic connecting system in aircraft assembly[2]; Some scholars detail the application of assembly simulation technique to wing-body docking[3]; Some use OGRE(Object-Oriented Graphics Rendering Engine) and

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 519–527, 2013.

PhysX(physical engine) as the development platform instead of customizing commercial CAD/CAM platforms or implementing the system through OpenGL directly[4]; Some plans automatic assembly for complex products such as aircraft components and an assembly planning and simulation system called AutoAssem has been developed[5]; Others propose that automatic measurement method in aircraft assembly contributes to improve assembly efficiency[6]; Some hold the view of building the integrative framework supporting design, simulation and optimization to manage the aircraft production cycle[7]. Although there are many relevant studies, the simulation world and the physical world in actual assembly are still short of extensive internetworking, interconnecting and interoperation, and sufficient coordination and optimization have not been achieved.

CPS (Cyber Physical Systems) integrates information cells and physical elements by 3C (Computation, Communication, Control) techniques, thereby realizing the advanced technology to accurately perceive and effectively control physical systems[8]. Currently, the study of CPS application to aircraft flexible assembly has not been unfolded in China, hence, by combining concepts of CPS and characters of aircraft flexible assembly, this paper proposes a basic framework of building ACPS (Assembly-oriented Cyber Physical Systems) including the following aspects—sensing, communication, computing, simulation, control, and application. The research group of this paper has conducted the study of CPS architecture for assembly[9], thus, on the foundation of former study, and under the background of an ongoing project, the paper further discuss the design and achievement of ACPS. This paper also explains the above aspects' corresponding functions, and offers expected applications of aircraft assembly, thus providing theoretical support for the development of smart assembly systems.

2 ACPS Architecture

The core of solving assembly problems by deeply integrating aircraft assembly cyber systems and assembly physical systems is that assembly physical objects and system state are transmitted into cyber system, furthermore, based on 3C techniques, and integrating information cells with physical elements, advantages of cyber technique in sensing, transmission, storage, analysis mining, and optimizing control will be taken. What's more, through the interaction and feedback between cyber system and physical system, cyber communication, system coordinating, optimizing decision control of assembly system will be realized, which will become an important research direction for CPS application and the development of next generation smart assembly systems. Thus, the study of ACPS architecture has important academic value and practical significance either in the view of fundamental research or engineering application.

Combining concepts of CPS and characters of aircraft flexible assembly, this part presents the basic framework of ACPS as below, including computing system, network system and physical system.

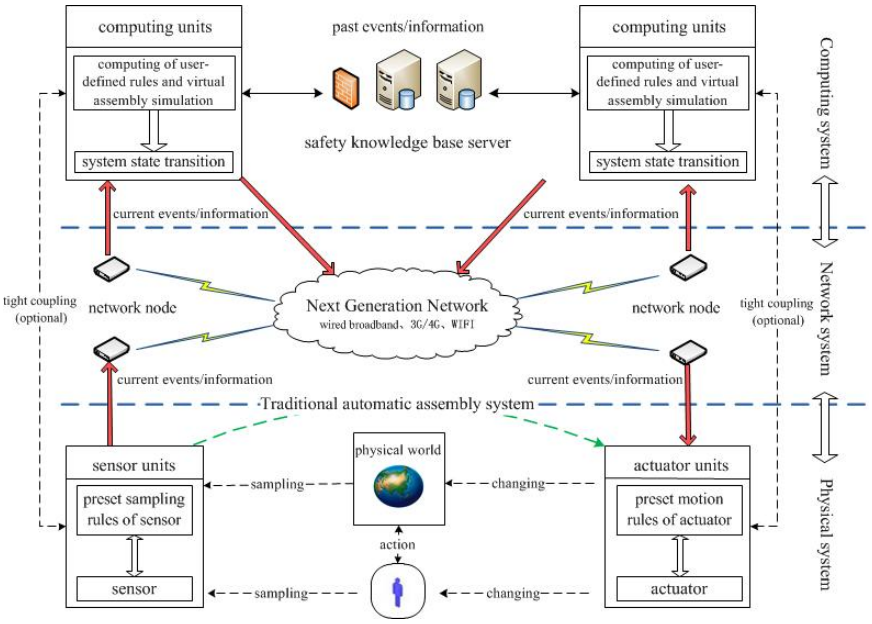


Fig. 1. ACPS architecture

From Fig.1, ACPS is mainly composed of sensor units (physical sensing equipment, such as laser/infrared tracker, force sensor), actuator units (numerically controlled equipment) and computing units (data server, simulation control computer, etc.), which are connected by network. The ACPS subsystems are presented below:

(1) Physical system

Sensor units can acquire necessary events/information data by preset sampling rules of sensor, and the sampling objects include some physical properties of concerned physical devices and human operation; Physical world generally refers to all kinds of ready-to-assembly tools; Actuator units operate actuator by preset motion rules of actuator, thus finally changing the condition of physical world and people.

(2) Network system

Network system, composed of communication base stations and network nodes, is responsible for the reliable transmission of current events/information data. Communication modes that can be adopted in aircraft assembly include wired network, wireless network, Wi-Fi, Bluetooth, GSM, etc. Aircraft assembly cyber system needs to obtain and process large amounts of information, such as devices' location, position and force situation; hence, the key issue in network system is to safeguard the high reliability and effective transmission of the information.

(3) Computing system

Different from traditional CPS, computing units only achieves computing of user-defined rules, while the computing units in ACPS integrate virtual assembly simulation to realize simulation based computing and control. The operating principles of

computing units are presented in Fig.2: assembly system is first modeled, and the data of current events/information can be obtained from communication network. Then obtained data are put into simulation model to compute. By comparing with the data of past events/information, whether the system state is transformed will be judged. If it is, the current events/information will be sent to performing units by network nodes; If not, the simulation computing process is repeated. Virtual assembly simulation is a noticeable feature of this part, including assembly sequences planning, assembly path planning/motion trajectory planning, collision and interference checking, and assembly resources integrating management, etc. In addition, integrating with CPS, virtual simulation can track the assembly progress in real-time, and timely simulation computing can be carried out to revise assembly scheme, thus tightly combining simulation and control to improve assembly efficiency.

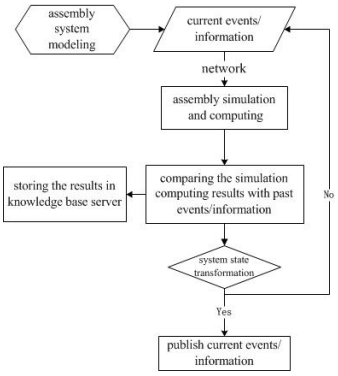


Fig. 2. Workflow of computing units

3 ACPS Composition of Aircraft Assembly

The production period of aircraft assembly is long for its various tools, complex procedures, and high precision. ACPS application to aircraft assembly can realize real-time simulation, computing and control during assembly process, and find out problems of assembly in time to formulate modification plans, thus shortening production period of aircraft assembly. Based on the introduction of ACPS architecture in section 2, this section carries out the overall design of ACPS on the background of actual assembly (e.g. aircraft components docking).

3.1 Hardware and Software Environment

(1) Hardware Environment
More than 2 wired/Wireless network card PCs/servers; network router; one or more of sensing devices, such as six-degree of freedom 3D space position tracker, laser tracker, iGPS, force sensor; aircraft assembly simulator; programmable motion control card, TV, grating ruler, or commercial NC locator, etc.

(2) Software Environment
Operating system: Windows 7, Windows XP, Windows 2003 and above
Software: Visual C++ 6.0 and above, CATIA, ACIS, MATLAB

3.2 ACPS Composition

As shown in Fig.3, measuring equipment, such as laser tracker, hand-held photographic measuring equipment, electric theodolite, transfer the measured data to simulation console by network. Then simulation console conducts data analysis and processing, offline/online assembly simulation, visual simulation state display, control decision, etc. Last, simulation console sends the control data to numerical control performing equipment by network.

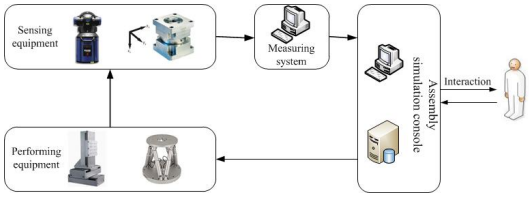


Fig. 3. Sketch of ACPS Composition

3.3 Functional Module Partition

The above analysis arrives at that the focal point is to build assembly simulation console serving as a link between measuring system and performing devices. Major functions of assembly simulation control system are explained in Fig.4.

(1) Measured Data Fusion
During aircraft assembly, there are many measuring points, assembly objects and tools, and the coordinates of different measuring equipment are different. Besides, the forms of measuring data acquired by different measuring equipment are also different, hence it is important to unify the form and coordinate measuring data. What's more, the measuring errors are analyzed to find out error compensation methods, thus guaranteeing measuring accuracy to its largest extent.
(2) Assembly Object Modeling
The model referred in this part means all of the entity information of work-pieces, tools, and devices during the process of aircraft assembly. It can be divided into geometrical information (e.g. point, line, surface, spatial location, size, and position, etc.) and physical information (e.g. material, roughness, rigidity, viscosity, and color, etc.).

Assembly information description includes cooperating connection information, designing constraint information, auxiliary semantic information. Cooperating connection information refers to the correlated information between all work-pieces constituting the assembly, including the constraint relations of 3D geometric assembly and the topological relations[10].

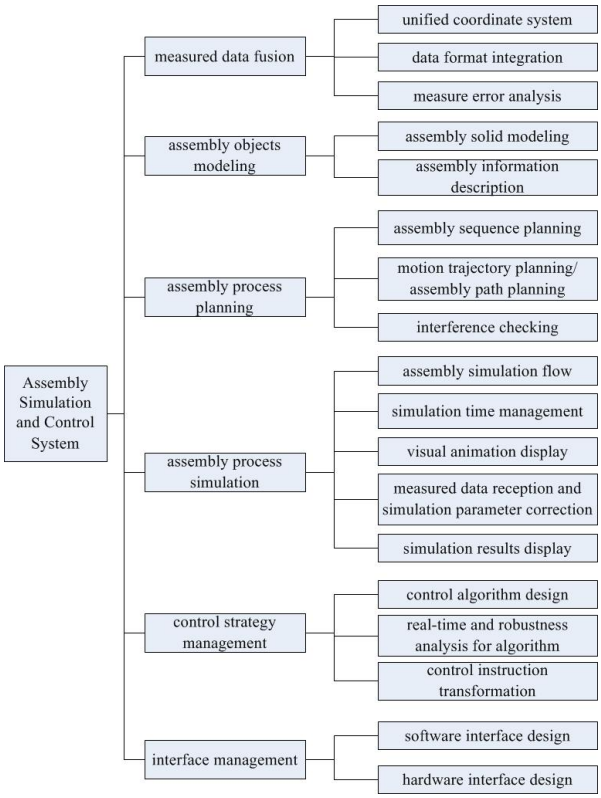


Fig. 4. Functional Partitioning Map

(3) Assembly Process Planning

Planning the assembly sequence of work-piece hasn't been involved in aircraft components docking. During components docking, one end is fixed while another end moves slowly to accurately connect with fixed part. The design of motion trajectory influences the accuracy of docking to a large extent. Computing methods designed in this paper should focus on the following four aspects: target position calculation, trajectory planning method design under assembly constraints[11], motion trajectory effectiveness evaluation, and motion trajectory optimization. And what's more, work-pieces and assembly fixtures may crash, contact or interact with surroundings, hence, dynamic and static interference checking should be conducted by using graded progressive algorithm to avoid the phenomenon that don't adapt to the objective reality, such as interpenetration and overlap among objects.

(4) Assembly Process Simulation

Simulation flow is designed based on the planning results providing the interface of flow design. Simulation can be operated once after the users filling in parameters. And by realizing visualizations during simulation process, users can monitor the simulation process at all times.

Then simulation time management is important. Offline simulation mainly coordinates the simulation time of modules. Besides that, online simulation also covers the coordination between measured data time and simulation time, namely, it achieves time synchronization via the timestamp of measured data, or carries out time compensation when the time is not in synch.

Measured Data Reception and Simulation Parameter Correction. The measured data are used to check the consistency between simulation process and actual assembly, while the data receiving module is applied to transform the measured data into the needed format in simulation comparison. If not, the correction will be conducted by necessary online parameters to ensure the smooth operation of assembly.

Finally it's simulation results display. After simulation process, the conditions of motion trajectory, driving force, and acceleration between work-pieces and NC devices should be presented in the form of figures, tables and equations. In addition, the above conditions should be stored to provide data support for other simulations.

(5) Control Strategy Management

Control Algorithm Design. It is necessary to design control algorithm for actual assembly units after simulation. The issue of multi-axis coordinated control is involved in the components docking. The improved algorithm will be proposed by comparing the strengths of heredity, ambiguity, neural network, self-adaptive control, generalized predictive compensation algorithm, feedback predictive compensation algorithm, sliding mode control algorithm, thus achieving synchronous control between the main and the auxiliary axes.

Real-time and Robust Analysis for Algorithm. Influenced by network stability, the reception of measured data will be delayed, in addition, computing and simulation also requires a certain time. After the delayed and computing time, the conditions of actual assembly have already changed. Considering situation of delay and signal interference, it is necessary to build delay model, introduce feedforward compensation, and improve algorithms to ensure the control of real-timeness and robustness.

Control Instruction Transformation. According to the designed control algorithm, control instructions can be automatically transformed into the code needed by the terminal, hence driving the operation of NC devices.

(6) Interface Management

To build simulation platform based on ACIS which can provide CATIA, MATLAB interfaces. Thereby, users just properly call the interfaces when programming.

Hardware Interface Design has two situations: one is to link the programmable motion control cards, and corresponding hardware codes are needed; the other is to use the open interfaces of commercial NC devices, and relative corresponding software codes are also needed.

3.4 Key Issues to Resolve

(1) Integration of Simulation, Computing, and Control

Most existing researches are done under the separation between simulation and control, i.e. offline preassembly simulation. Namely, assembly scheme will be received at

first after modeling, computing, and simulating the assembly objects under virtual environment, and then the assembly scheme will be artificially removed to actual assembly environment. Though this kind of assembly simulation can analyze and verify the realizability of aircraft parts' assembly motions, the consistency of simulated effects and physical assembly can't be ensured since the differences between simulation model and practical system, thus constrained the application of research findings. Some other researches are real-time monitoring semi-physical simulation system, only providing some indexes to assist in determining assembly quality without timely visual assembly process; furthermore, such problems as the internetworking, interconnecting and interoperation between actual assembly objects and simulation system have not been well solved. Combining idea of CPS, this paper builds ACPS based on simulation. ACPS plans assembly scheme in simulation system, receives measured data in real-time on system state judgment to revise simulation process, as well as control performer to finish assembly, thus achieving the integration of simulation, computing, and control.

(2) Optimization of Motion Trajectory Planning

During aircraft components docking, the design of motion trajectory directly influences the accuracy of docking, which has been a heated topic for scholars both at home and abroad. ACPS concerns offline/online simulation, hence, higher requirements are requested for motion trajectory planning algorithm. Therefore, this paper adopts simulation based optimal trajectory planning approach under assembly constraints. Namely, computing module of trajectory planning is put into simulation system, and by comparing various motion trajectory equation, optimization solution will be obtained under the constraints of the shortest time constraint, the minimum driving force constraint, the least position transformation, as a result, the optimal motion trajectory will be acquired to guarantee the pose accuracy.

(3) Real-time Problems

Based on effective information measured by sensors, assembly simulation control system carries out real-time visualized aircraft assembly procedures, tracks devices' positions, design appropriate control algorithm, process and analyze the data transferred from computing layer, thus achieving accurate control of aircraft assembly physical system. During this process, it is important to ensure the issue of real-time. Such procedures as network transmission, calculation of assembly simulation planning, option of control decisions, and artificial intervention are time-consuming. Therefore, the system state may be changed at the time of sending control instructions, which will cause operation failure. Consequently, how to guarantee ACPS real-time is the key issue to achieve effective control.

3.5 Summary

By taking aircraft components docking as an example, section 3 introduces ACPS's software and hardware requirements, system composition and functional modules of assembly simulation control system, and it also explains the mission of each functional modules and the key issues to resolve. On this basis, prototype system will be built next, with focusing on motion trajectory planning, virtual assembly simulation, and real-time control under the unstable network. Building prototype system can be realized by programming in software ACIS and MATLAB.

4 Conclusion

Based on the ideology of cyber physical systems and combining the development requirements of aircraft flexible assembly, this paper puts forward ACPS (Assembly-oriented Cyber Physical Systems) including the following aspects—sensing, communication, computing, simulation control, and application. This paper also explains the above aspects' corresponding functions. Then, according to the aircraft assembly process, this paper proposes the idea of building prototype system, carries on the application analysis from software/hardware environment, overall system composition, functional module division and key issues, and offers expected applications of aircraft assembly, thus providing theoretical support for the aircraft flexible assembly and the development of smart assembly systems. Next, based on the software function structure in part three, we will begin to design and realize the functions referred in part three, with the focus on designing and realizing algorithm of motion trajectory optimization and real-time control during aircraft docking, and the above two algorithms will be checked in physical and simulating environment.

Acknowledgements. This research is supported in part by National Science and Technology Support Program (Grant 2012BAF15G01).

References

1. Guo, E.: Foreign aircraft flexible assembly technology. Aeronautical Manufacturing Technology 9, 28–32 (2005)
2. Zou, F., Zhang, S.: Automatic Alignment & Butt System for Aircraft Final Assembly. Aeronautical Manufacturing Technology 7, 32–36 (2008)
3. Wu, X.: Application of Assembly Simulation Technology in Aircraft Wing-body Docking. Journal of Slianghai University of Electric Power 3, 277–280 (2012)
4. Wang, D., Zhang, L., Wang, M., Xiao, T., Hou, Z., Zou, F.: A Simulation System Based on OGRE and PhysX for Flexible Aircraft Assembly. In: Proceedings of the Workshop on Principles of Advanced and Distributed Simulation (2012)
5. Xu, L.D., Wang, C., Bi, Z., Yu, J.: AutoAssem: An Automated Assembly Planning System for Complex Products. IEEE Transactions on Industrial Informatics 3 (2012)
6. Jamshidi, J., Kayani, A., Iravani, P., Maropoulos, P.G., Summers, M.D.: Manufacturing and assembly automation by integrated metrology systems for aircraft wing fabrication. Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture 1 (2010)
7. Bres, A., Monsarrat, B., Dubourg, L., Birglen, L., Perron, C., Jahazi, M., Baron, L.: Simulation of friction stir welding using industrial robots. Industrial Robot 1 (2010)
8. Liu, X., Liu, X., Wang, Z., Cheng, W., Li, J.: A Cyber-Physical System. Shangdong Science 3, 56–61 (2010)
9. Dong, J., Xiao, T., Zhang, L.: A Prototype Architecture for Assembly-Oriented Cyber-Physical Systems. Communications in Computer and Information Science 323, 199–204 (2012)
10. Li, Y.: A Study on the T-CPS-Oriented Microscopic Traffic Cognitive Techniques and Related Key Problems. Chongqing University, Chongqing China (2012)
11. Zhu, Y., Huang, X., Li, S., Fang, W.: Automation Adjustment and Tracking Measurement of Fuselage Position and Pose. Mechanical Science and Technology for Aerospace Engineering 7, 1121–1127 (2012)

Lateral Carrier Landing Performance Affecting Factors of Small Carrier-Based UAV

Fengying Zheng[1], Huajun Gong[2], Ju Jiang[2], and Ziyang Zhen[2]

[1] Academy of Frontier Science, Nanjing University of Aeronautics & Astronautics, Nanjing, 210016, China
zhfy@nuaa.edu.cn
[2] College of Automation Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing, 210016, China
{ghj301,jiangju,zhenziyang}@nuaa.edu.cn

Abstract. Research on factors affecting carrier landing performance of small UAV has vital significance to ensure the safe landing of UAV. The influences of external environmental disturbance and UAV lateral attributes change are studied. First, Small UAV lateral carrier landing system is constructed. Comprehensive computer models, which include ship dynamics, airwake, navigational error, and airplane dynamics and kinematics, dynamic airplane control structure based on improved LQR with frontal compensator, are then designed to simulate unmanned carrier landings. Second, aircraft lateral attributes are varied and landing performance statistics are recorded for each configuration. Finally, the aircraft preliminary design limits are generated. Simulation results show that severe sea condition has the greatest impact on landing dispersion, and in lateral aerodynamic attributes, side force coefficient with sideslip angle which is generally negative is the most important factor. Increasing its absolute value with physical constraints will significantly improve landing performance.

Keywords: carrier UAV, flight control, landing performance, disturbance sources, lateral aerodynamic attributes.

1 Introduction

Unmanned systems are becoming increasingly important to our military, recent naval wars show that unmanned air vehicle (UAV) has a vital role in the future of the navy [1]. Due to carrier-based UAV is just getting started [2], less of the research results publicly report in the UAV carrier landing.

The Automatic carrier landing system has been in used for years [3], but it is neither flight critical nor attempts at severe sea-state because the pilot provide for reversion to manual control [4,5]. Going to a UAV raises the automated system to the status of flight critical and permits greater design freedom, no longer constrained by the limitation of the human operator. This research is intended to begin to know the design constraints imposed by fully autonomous carrier landing.

Miniaturization is the development trend of the carrier-based UAV [6], however, because of its small size, light weight, when landing, the small UAV is more

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 528–540, 2013.

vulnerable to the impact of its aerodynamic attributes and outside disturbances. We must take account of these unfavorable aspects when we make research. Studies on carrier landing performance affecting factors are mainly concentrated on large or medium-sized UAV. Ref [7] uses dynamic inversion technology to analysis landing performance of an unmanned combat aerial vehicle. Ref [8] lists a variety of carrier aircraft landing specification. Ref [9] gives a medium-sized UAV design specification, especially focus on longitudinal performance. All of above references do not put much attention on the lateral system.

Aircraft Lateral dynamics is a more complex problem because of the coupling of the airplane's lateral and directional axes, plus the coupling of the ship's lateral and directional axes[9]. In this paper, we take an active small carrier UAV as model, develop its lateral dynamics and kinematics model, complete lateral flight control design, and analysis lateral landing performance include environment disturbance sources and attributes change. Landing performance statistics are recorded for each configuration. The influence of each attribute is extracted from the landing statistics. Finally, the aircraft preliminary design limits are generated. The controller design and simulation are conducted in the MATLAB/Simulink environment.

2 Small UAV Lateral Automatic Carrier Landing System

Main lateral landing performance of carrier aircraft includes mean and standard deviation of lateral position error, lateral deviation range, lateral boarding rate, etc, as shown in Table 1[10-11].

Table 1. Lateral touchdown dispersion index

Performance index	Target Performance	Allowable performance
Lateral Mean	0.61m	1.22 m
Lateral Std Deviation	0.91m	1.52 m
Lateral Deviation range	-1.52~1.52m	-3.05~3.05m
Boarding rate	75%	65%

For assessment of small UAV carrier landing performance under different conditions, the UAV lateral landing simulation system is constructed, as indicated in Figure 1.

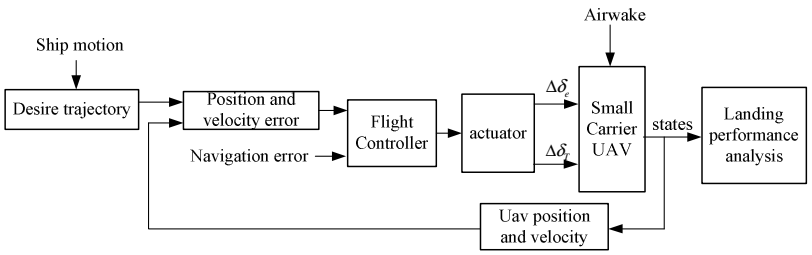


Fig. 1. Structural allocation of small carrier UAV lateral landing system

Flight control system uses LQR controller with frontal compensator. The ship motion model and navigation error model are provided by the Naval Air Systems Command [10,11]. Airwake model references MIL-F-8785C military specification given specific atmospheric turbulence mathematical model for landing [12]. A ten-millisecond time step (0.01 sec) is selected to match models.

3 Lateral Control System

3.1 Small Carrier-Based UAV Modelling

This paper refers to small carrier-based unmanned aerial vehicle aerodynamic parameters [5]. First, establish the full nonlinear dynamics and kinematics model of UAV, then complete trim and linearization when given steady straight-line flight status as a reference point. finally get the linearized small perturbation equations.
UAV state vectors are as follows:

$$x=[V\ \alpha\ \beta\ p\ q\ r\ \theta\ \phi\ \psi\ x_g\ \ y_g\ \ h] \tag{1}$$

Where V is flight speed, α is angle of attack, β is sideslip angle. p,q,r are airplane angular velocity components about body-fixed axes. θ,ψ,ϕ are airplane pitch angle, bank angle, and heading angle. x_g, y_g, h are the components of airplane centre of mass along earth-fixed axes.
According to the force equations in airflow axes, moment and kinematic equations in the body-fixed axes, we can establish the 12-order nonlinear coupled differential equations.

$$\begin{cases}
\dot{V} = (T\cos\alpha\cos\beta - D + G_{xa})/m \\
\dot{\alpha} = (-T\sin\alpha - L + G_{za} + mV(-p\cos\alpha\sin\beta + q\cos\beta - r\sin\alpha\sin\beta))/mV\cos\beta \\
\dot{\beta} = (-T\cos\alpha\sin\beta + Y + G_{ya} - mV(-p\sin\alpha + r\cos\alpha))/mV \\
\dot{p} = (c_1 r + c_2 p)q + c_3\bar{L} + c_4 N \\
\dot{q} = c_5 pr - c_6(p^2 - r^2) + c_7 M \\
\dot{r} = (c_8 p - c_2 r)q + c_4\bar{L} + c_9 N \\
\dot{\phi} = p + (r\cos\phi + q\sin\phi)\tan\theta \\
\dot{\theta} = q\cos\phi - r\sin\phi \\
\dot{\psi} = (r\cos\phi + q\sin\phi)/\cos\theta \\
\dot{x}_g = u\cos\theta\cos\psi + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) + w(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) \\
\dot{y}_g = u\cos\theta\sin\psi + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) + w(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) \\
\dot{h} = u\sin\theta - v\sin\phi\cos\theta - w\cos\phi\cos\theta
\end{cases} \tag{2}$$

Subsequently, calculate trimmed variables when given benchmark height $h_0 = 100$ m, benchmark velocity $V_0 = 30\text{m/s}$, and selected performance index:

$$J = \dot{V}^2 + \dot{\alpha}^2 + \dot{q}^2 \tag{3}$$

The variables needed to trim are the angle of attack α, elevator deflection δ_e, throttle deflection δ_T. Give the initial values of variables, use optimization algorithm, let J infinite close to zero, and obtain the corresponding optimal trimmed variables, as shown in figure 2, $\alpha_0 = 2.4219^\circ$, $\delta_{e_0} = 0.573^\circ$, $\delta_{T_0} = 1.869N$.

In accordance with the principle of small perturbations, Lateral aircraft dynamics can be characterized by a system of four linear differential equations, in a state-space formulation as follows.

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.238 & 0.0422 & -0.999 & 0.4666 \\ -78.985 & -27.83 & 6.794 & 0 \\ 22.504 & -0.946 & -1.020 & 0 \\ 0 & 1 & 0.0423 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} 0 & 0.116 \\ 350.02 & 8.453 \\ -11.85 & -15.10 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{4}$$

Additionally, the kinematic equations are available:

$$\begin{cases} \Delta\dot{\psi} = \Delta r / \cos\theta_0 \\ \Delta\chi = \Delta\psi + (\Delta\beta - \Delta\phi \sin\alpha_0) / \cos\gamma_0 \\ \Delta\dot{y} = U_0 \cos\gamma_0 \cdot \Delta\chi \end{cases} \tag{5}$$

The eigenvalues can be determined by finding eigenvalues of the matrix A:

$$|sI - A| = 0 \tag{6}$$

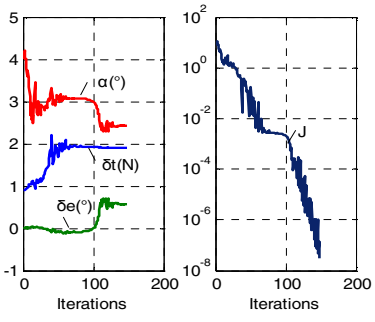
The solution of the characteristic equation yields the eigenvalues:

$$s_1 = -27.6250; \quad s_{2,3} = -0.7577 \pm 5.1402i; \quad s_4 = 0.0452 \tag{7}$$

Correspond to the aircraft's roll mode, Dutch roll mode and spiral mode .The damping ratio and undamped natural frequency for Dutch roll mode are as follows:

$$\zeta_{Dr} = 0.1468, \omega_{Dr} = 5.196 \tag{8}$$

Figure 3 shows a cross-lateral open-loop aircraft response when input the initial roll angle.



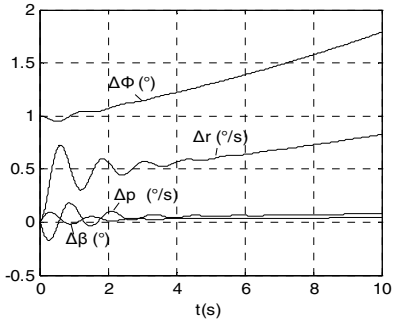


Fig. 2. trimmed variables and performance index **Fig. 3.** Aircraft Response to unit roll angle

From the response curve, we can see that sideslip angle and roll angle rate gradually stabilize, aond the roll angle and yaw angle rate gradually increases, the aircraft cannot be restored to original flying condition

3.2 Improved LQR Flight Controller Design

Consider small UAV's vulnerability and poor flight stability, the LQR controller with strong robustness is used to design lateral flight control system. In order to minimize the observed tracking error without excessive control activity, we add frontal compensator to offset the closed-loop poles.

In this case the outputs of interest are lateral velocity perturbation and lateral deviation. The two values we selected to track are Δv and Δy. The controller is required to simultaneously maintain the desired lateral velocity and line-up as precisely as possible. To accomplish this, integral tracking on Δv and Δy is introduced to achieve zero steady-state error in response to a pure ramp input (yielded by the angle between the flight deck axis and ship axis, when ship motion). As a result, the establishment of the UAV augmented state equation as follow:

$$\dot{X}_c = \mathbf{A_c} X + \mathbf{B_c} u \tag{9}$$

Where:

$$X_c = [\Delta v \ \ \Delta p \ \ \Delta r \ \ \Delta\phi \ \ \Delta\psi \ \ \Delta y \ \ \frac{\Delta y}{s} \ \ \frac{\Delta v}{s}]^T$$

$$\mathbf{A_c} = \begin{bmatrix} \mathbf{A} & zeros(4,4) \\ \begin{bmatrix} 1 & 0 & 0 & -u_0 \cdot \sin\alpha_0 & u_0 \cdot \cos\gamma_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \end{bmatrix} \quad \mathbf{B_c} = \begin{bmatrix} \mathbf{B} \\ zeros(4,2) \end{bmatrix}$$

The functional form of the performance index can be provided as a quadratic index:

$$J_c = \int_0^\infty (\overline{y}^T \mathbf{Q} y + u^T \mathbf{R} u) dt \tag{10}$$

The matrix $\mathbf{Q}$ weights the output states, and $\mathbf{R}$ weights the control inputs. The output $\overline{y}$ is composed of the elements given in equation (11) below.

$$\overline{y} = \begin{bmatrix} (s+2+\frac{2}{s})\Delta h \\ (1+\frac{0.5}{s})\Delta u \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -u_0 \cdot \sin\alpha_0 & u_0 \cdot \cos\gamma_0 & 2 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix} \cdot X_c \tag{11}$$

The effect is to place two zeros in the open-loop transfer function, which attract the closed-loop poles. Zero locations are determined by trial, attempting to minimize the observed tracking error without excessive control activity.

Introduce optimal control law, provided as follow:

$$u = \begin{bmatrix} \Delta\delta_e \\ \Delta\delta_T \end{bmatrix} = \mathbf{K} X_c \tag{12}$$

Where $\mathbf{K}$ is constant state feedback gain matrix determined by solving the steady-state matrix Riccati equation:

$$\mathbf{PA} + \mathbf{A}^{\mathrm{T}}\mathbf{P} - \mathbf{PBR}^{-1}\mathbf{BP} + \mathbf{Q} = \mathbf{0}, \quad \mathbf{K} = -\mathbf{R}^{-1}\mathbf{BP} \tag{13}$$

In order to better control, it is needed to adjust the weighting matrix $\mathbf{Q}$ and $\mathbf{R}$. And the choice of $\mathbf{Q}$ and $\mathbf{R}$ are compromised between adjustment speed and control capability of state, the larger $\mathbf{Q}$ can get a faster adjustment speed, and the lager $\mathbf{R}$ can make the necessary control capability reduced. For our problem, the weight values of $\mathbf{Q}$ and $\mathbf{R}$ are determined using the method suggested by Nelson[13]. Finally, have:

$$\mathbf{K} = \begin{bmatrix} -0.247 & -0.009 & -0.721 & -1.03 & -14.03 & -0.507 & -0.352 & 0.085 \\ 0.53 & 0.0312 & -0.245 & 1.12 & 8.234 & 0.29 & 0.183 & 0.177 \end{bmatrix} \tag{14}$$

However, Note that the implementation system and the system for gain calculations are not the same. The difference is the gains are calculated for y and y/s making it a regulator problem, but the implementation used y_{error} and y_{error}/s transforming it to a tracking problem. Provided as control law:

$$u = \begin{bmatrix} \Delta\delta_e \\ \Delta\delta_T \end{bmatrix} = \mathbf{K}X_c', \quad X_c' = [\Delta v \;\; \Delta p \;\; \Delta r \;\; \Delta\phi \;\; \Delta\psi \;\; \Delta y - \Delta y_{cmd} \;\; \frac{\Delta y - \Delta y_{cmd}}{s} \;\; \frac{\Delta v}{s}]^T \tag{15}$$

Where Δy_{cmd} is line-up command added deck motion.

Figure 4 shows the response diagram of the flight control system while input unit step lateral deviation signal.Figure5 shows the response while input unit ramp signal superimposed amplitude 1m, frequency 0.5 rad / s sinusoidal signal.

It can be seen from the figure, the flight control system designed can quickly and accurately track command. Tracking unit step signal, steady-state error is zero, and control surfaces restore fast, tracking slope superimposed sinusoidal signal, steady state error does not exceed 0.5m. So the improved LQR controller designed can be a general method for small UAV flight control law design.

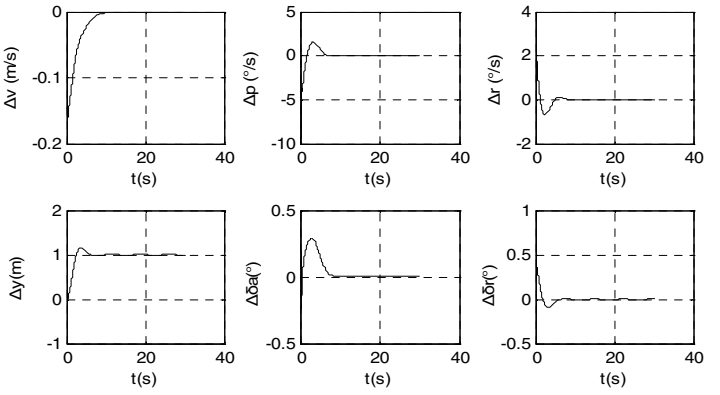


Fig. 3. Flight control system response to unit step command

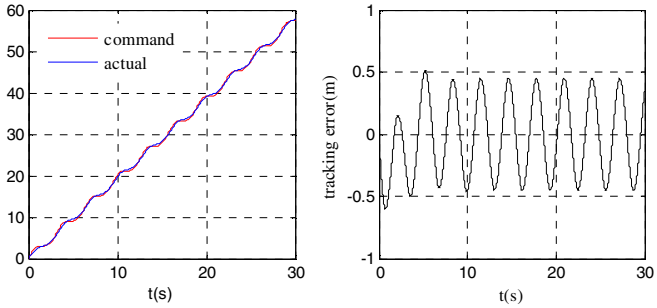


Fig. 4. Flight control system response to given command

4 Performance Analysis with Disturbance Sources

4.1 Ship Dynamic

The dynamics and dimensions of CVN 65, the U.S.S. Enterprise, are employed throughout the research. The distances from the ship's center of motion to the landing area in feet are 68m aft, 19.5m up, and 3m left. Carrier geometry is shown as figure 6. The ship dynamics model provided generate six degree-of-freedom time histories of ship motion for a sea-state chosen by the user. Note that all ship displacements are referenced to the ship center of motion. Since the desired touchdown point(DTP) is displaced a lateral distance from the center of motion, the DTP's translational displacement is dependent upon both the translational and angular displacements of the ship. These angles had to be converted to distances Y to be useful to the landing task. Equations (16) given below are the exact lateral relationships.

$$Y_{DTP} = 19.5\sin(\phi_s) + 3\cos(\theta_s)\cdot\cos(\psi_s) - 3 + 68\sin(\psi_s) + y_s \tag{16}$$

Where, θ_s, ϕ_s, ψ_s are pitch, roll and yaw angle of ship, y_s is lateral translational displacements. For the purposes of this research, sea-state 3, 4, and 5 are modeled. Table 2 provides the relevant RMS amplitudes.

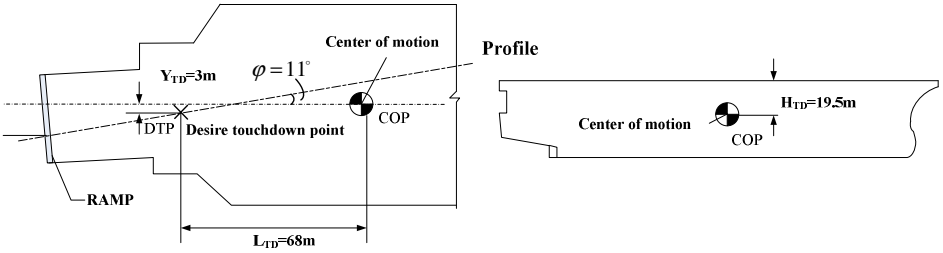


Fig. 5. CVN 65 Carrier Geometry

Table 2. RMS Amplitudes for Modeled Sea-states

Sea-state	θ_s	ϕ_s	ψ_s	x_s	y_s	h_s
3	0.763	0.21	0.12	0.838	0.53	2.11
4	1.22	0.33	0.30	1.4	0.84	3.81
5	1.83	0.49	0.29	2.1	1.26	5.06

The total influence of ship motion for all three modeled sea-states is presented in Figure 7. The landing area lateral displacement is graphed in meter versus time.

4.2 Airwake

Due to the special atmospheric conditions when carrier landing, the MIL-F-8785C military specification gives specific airwake model. The lateral component is composed of two parts: the free atmosphere turbulence component and wake random component. The detailed calculation process can see reference [14] .Take sea state 3 as an example, Set UAV speed $V_0 = 30m/s$, wind of deck V_{wfd} =15m/s , initial distance $D_0 = 1800m$, glide slop angle $\gamma_0 = 3.5^\circ$, and Get total lateral airwake as shown in figure 8.

4.3 Navigation Error

The Joint Precision Automated Landing System (JPALS) serves as the guidance system on first generation shipboard UAV. JPALS operates using differential-GPS data, blended with inertial navigation on both the airplane and the carrier. The proximity of the ship and airplane during approach and landing cause both GPS receivers to experience the same atmospheric disturbance, resulting in very tight error bounds. NAVAIR provides one and a half hours of JPALS flight test data sampled at 50 Hz. with both the measured and the true position. Figure 9 shows navigation error.

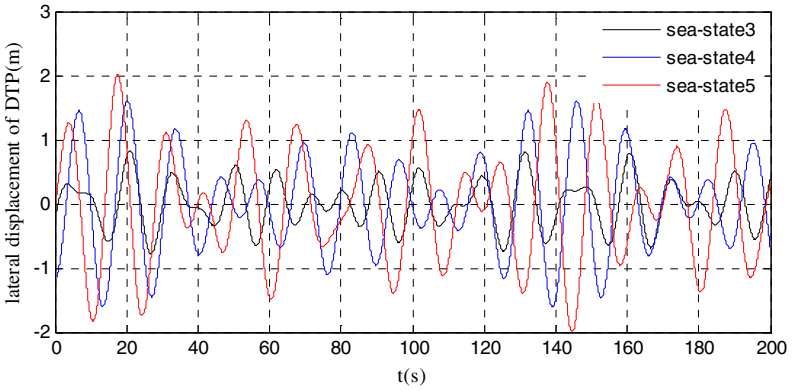


Fig. 6. Lateral displacement of DTP due to ship motion

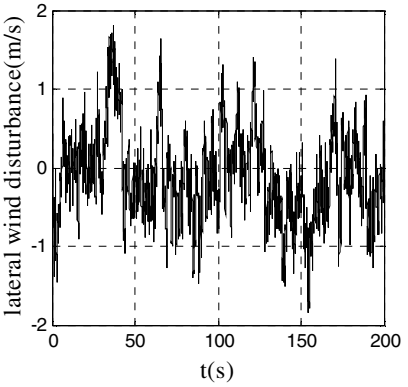


Fig. 7. Lateral airwake simulation

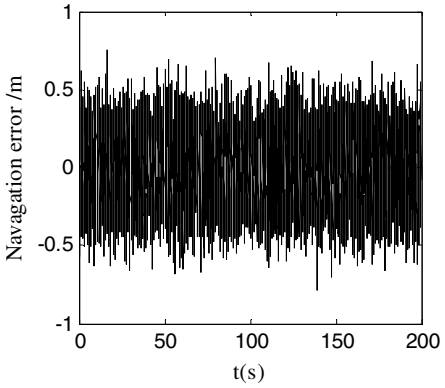


Fig. 8. navigation error simulation

4.4 Performance Analysis

It is important to understand the effects of disturbance sources on landing performance. To do this, one thousand simulations are run with each source of noise individually. All other simulations are run with all three sources of error turned on. Table 3 shows these simulation results. Where, 'Nav' denotes navigation error added. Note that these boarding rates account only for landing position, do not account for the possibility of extreme touchdown attitudes at which arrestment would be unsuccessful. In addition, there are many unpredictable factors in the actual UAV landing not taken into account in the simulation. Thus, simulations narrow landing deviation of the allowable range, if deviation exceeds ±2 m, the landing will be assumed unsuccessful.

Table 3. Combination Effects of Disturbance Sources

Case	Condition	Lateral Mean(m)	Lateral Std.Dev(m)	Lateral Deviation(m)	Boarding rate(%)
1	Nav	-0.0246	0.1350	-0.3381~0.4262	100
2	Airwake(15*)	-0.0672	0.3261	-0.8554~0.9661	100
3	Airwake(15), nav	-0.0731	0.3864	-0.9321~1.0342	100
4	Airwake(20), nav	-0.0921	0.3709	-1.0432~0.9774	100
5	Seastate3	0.1832	0.4207	-1.2610~1.3827	100
6	Seastate3, Airwake(15), nav	0.2005	0.4516	-1.4832~1.6528	100
7	Seastate3, Airwake(20), nav	-0.2752	0.4842	-1.3400~1.8978	100
8	Seastate4	0.4185	0.5905	-1.8232~2.678618	93
9	Seastate4, Airwake(15), nav	0.4231	0.6010	-2.0927~2.9137	92
10	Seastate4, Airwake(20), nav	0.4312	0.5872	-2.1346~2.8033	92
11	Seastate5	0.4890	0.7321	-4.8764~4.3919	86
12	Seastate5, Airwake(15), nav	0.5021	0.7138	-5.0739~5.2138	84
13	Seastate5, Airwake(20), nav	0.4976	0.7872	-5.0655~5.8321	85

*the value of WOD(wind of deck), for example: Airwake(15) denotes the value of airwake when WOD=15m/s.

Each interference would impact on landing performance. Navigation error has negligible influence, on the case of navigation error only, landing mean deviation and standard deviation are very small. However, ship motion and airwake have a great influence on landing performance.

From the data, we can see that ship motion is the dominant source of landing error. The landing performance of sea-state 4 only is worse than Sea-state 3 added air wake and navigation error. Sea-state 5 is the most demanding case because of the magnitude of the required flight path changes. The deck moves as much as 1.5 meter laterally in ten seconds, challenging the system's ability to track a command.

Additionally, simulations show that on all conditions the lateral mean and standard deviation can meet the requirements. On sea-state 3, landing deviation range can also meet requirement. Sea-state 4 and 5 landing deviation ranges exceed the target, and have lower success rate.

5 Performance Analysis with Lateral Attributes Change

Study the impact of the stability derivatives Cy_β、Cl_β、Cn_β on small UAV's landing performance. Note that no attempt is made to optimize LQR controller performance by tuning Q and R. The purpose is to ensure that the effects of airframe attributes are not masked by variances in controllers.

5.1 Influence of Side Force Coefficient with Sideslip Angle and Rolling Moment Coefficient with Sideslip Angle

The combined influence of side force coefficient with sideslip angle Cy_β and rolling moment coefficient with sideslip angle Cl_β is evaluated by a combined total of three hundred thousand simulation runs. The Cy_β is varied from -0. 4 rad^{-1} to -0.04 rad^{-1} in increments of 0.04 rad^{-1}. The Cl_β is varied from -0.1 to -0.01 in increments of 0.01. At each combination, one thousand simulated landings are performed for each sea-state, and at each simulation, airwake and navigation error are added.

Finally, simulation indicates that the boarding rates of sea-state 3 for all modelled configurations are more than 65% . This means that if landing position is the only criterion, the autonomous system would have an acceptable boarding rate for all conditions up to and including sea-state3.

The boarding rate for sea-states 4 and 5 are plotted below in Figures10, each has Cy_β increasing up the y-axis and Cl_β increasing along the x-axis.

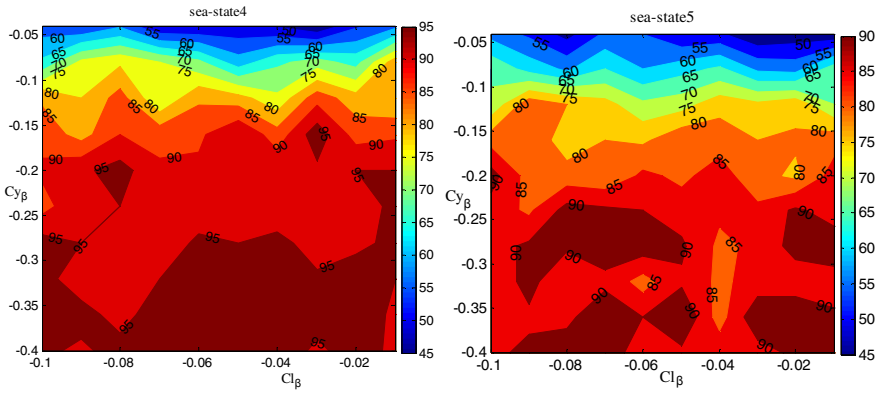


Fig. 9. Sea-state 4 and sea-state 5 boarding rate contour

Figure 10 illustrates the boarding rate ranges depend on the aircraft aerodynamic parameters. In general, Cy_β is negative, and higher $|Cy_\beta|$ provides higher boarding rates, A $|Cy_\beta|$ of 0.15 rad^{-1} or greater is required for target performance. While Cl_β has little effect.

Boarding rates are strongly related to the landing dispersion. Figure 11 depicts the landing dispersions for the sea-state4 and sea-state 5. Recall the target performance presented in Table 1 is a standard deviation of 0.9m or less. Figure11 demonstrates the minimum acceptable $|Cy_\beta|$ is consistent with the above requirement imposed by boarding rate.

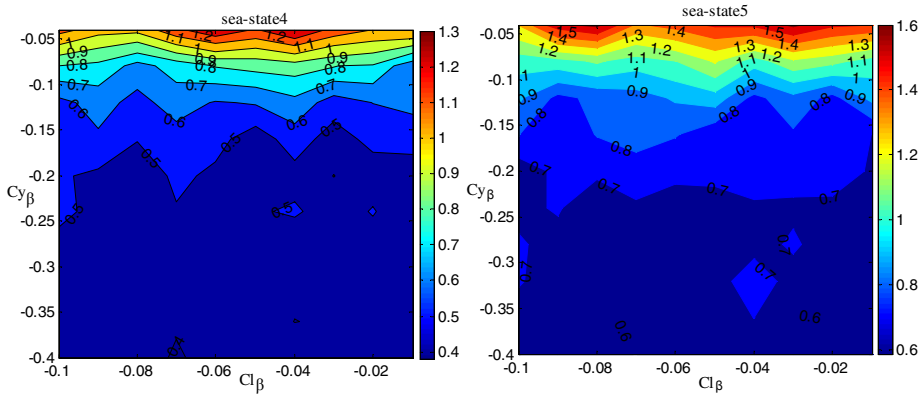


Fig. 10. sea-state4 and sea-state 5 landing position error standard deviation contour

5.2 Influence of Side Force Coefficient with Sideslip Angle and Yawing Moment Coefficient with Sideslip Angle

The effect of yawing moment coefficient with sideslip angle coefficient Cn_β on landing performance is evaluated from 0.01rad^{-1} to 0.1rad^{-1} in increments of 0.01 rad^{-1}. The Cy_β is varied in exactly the same manner as for the previous section, but Cl_β is held constant ($Cl_\beta = -0.0598$ rad^{-1}). Again, one thousand simulations are conducted for each aircraft configuration.

Figure 12 depicts the resultant landing dispersions of sea-state5. As in the previous section, boarding rates are above 75% for all configurations at sea-state3. Landing performance is again highly dependent on Cy_β, which improves for increasing $|Cy_\beta|$. Varying Cn_β has minimal effects on landing performance. But, form the figure, we can find that the landing performance has an worse trend with increasing Cn_β.

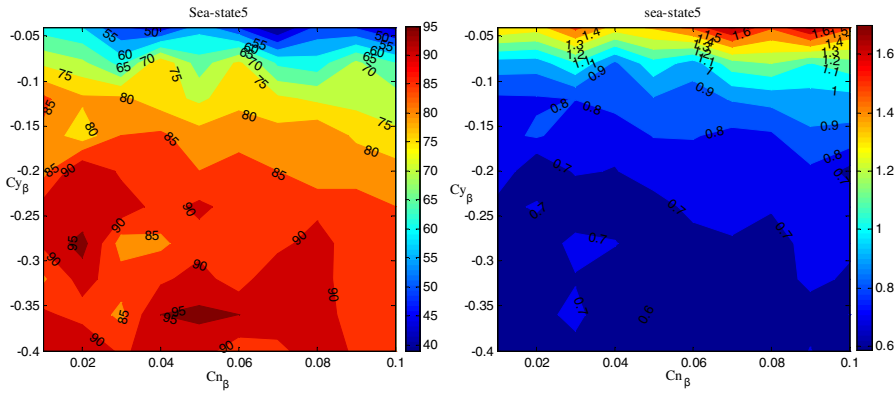


Fig. 11. Sea-state 5 boarding rate and landing position error standard deviation contour

6 Conclusions

Factors affecting lateral carrier landing performance of small UAV are studied. The results indicate that: Ship motion is the dominant cause of landing errors. Landing dispersion and boarding rates are highly dependent on sea-state. Rolling moment and yawing moment coefficient with sideslip angle coefficient have negligible influence on landing precision. Side force coefficient with sideslip angle is the dominant aircraft attribute for all landing performance criteria and each improves with increasing side force coefficient with sideslip angle. The minimum absolute value of side force coefficient with sideslip for suitable lateral landing performance is 0.15 per radian.

References

1. Sierra Nevada Corporation. UAV common automatic recovery system (UCARS), http://www.sncorp.com/prod/cnsatm/uav/uav1.html (accessed April 25, 2006)
2. Zhao, T.: Development of the Shipborne UAVS. Ship Electronic Engineering 30(4), 21–24 (2010)
3. Steinberg, M.: Development and simulation of an F/A-18 fuzzy logic automatic carrier landing system. In: IEEE International Conference on Fuzzy Systems Proceedings, vol. 2, pp. 797–802 (1993)
4. Sousa, P., Wellons, L., Colby, G., et al.: Test Results of an F/A-18 Automatic Carrier Landing Using Shipboard Relative Global Positioning System, Report No. NAWCADPAX /RTR-2003/122, Naval Air Warfare Center Aircraft Division, Patuxent River, MD, 9 (2003)
5. Lizarraga, M.I.: Autonomous landing system for a UAV. Naval Postgraduate School, Monterey (2004)
6. Liu, Q., Yuan, S.Z.: Longitudinal carrier landing system design for UAV based on Journal of Anhui University (Natural Science Edition) 35(1), 47–51 (2011)
7. Nicholas, A.: Denison. Automated Carrier Landing of an Unmanned Combat Aerial Vehicle Using Dynamic Inversion. Department of The Air Force University, Ensign (2007)
8. Yang, Y.D.: Translation set of carrier aircraft landing technology. National Defence Industry Press (2003)
9. Sweger, M.J.F.: Design specifications development for unmanned aircraft carrier landings. United States Naval Academy Annapolis, United States (2003)
10. Rudowsky et al.: Review of the Carrier Approach Crite ria for Carrier-Based Aircraft Phase I; Final Report. Report number: NAWCADPAX/TR-2002/71. Naval Air Systems Command (2002)
11. Yang, Y.D., Yu, J.Y.: Guidance and control of carrier aircraft aircraft landing. National Defence Industry Press (2007)
12. Flying Qualities of Piloted Aircraft. Military Specification MIL-F-8785C. United States Department of Defence (November 1980)
13. Nelson, R.C.: Flight Stability and Automatic Control. McGraw-Hill Book Company (1998)
14. Jiao, X.: Research on carrier-based aircraft landing condition and wave-off decision technology. Nanjing University of Aeronautics and Astronautics (2011)

Automatic Calibration of Field Mill Using Virtual Instrument

Yong Cui, Haiwen Yuan, Qiusheng Wang, and Xiao Song

School of Automation Science and Electrical Engineering
Beihang University
Beijing,China
cuiyong@buaa.edu.cn

Abstract. Field mill instruments are widely employed for the measurement of atmospheric electric fields, dc voltages, and electric fields around electric transmission line. The key point of an electric field sensor is calibration. There is some shortcoming in traditional manual calibration. A set of automatic calibration system is simulated and designed herein based on the virtual instrument technology and applied to improve the calibration efficiency and accuracy of the field mill.

Keywords: Field Mill, Virtual Instrument, Automatic Calibration, HVDC.

1 Introduction

Field mill is widely used in many fields to measure atmospheric electric field [1-6], high voltage [7], electric field around electric transmission line [8, 9], etc. The construction of HVDC transmission line as the important national strategic policy realizes its rapid development. Audible noise, radio interference, total electric field and other physical parameters are applied to evaluate the level of electromagnetism and the field mill is used to measure the total electric field at ground level for the overhead electric transmission lines.

BATEMAN and Fort Ada briefly introduced the calibration principle and process of the field mill[3], [5].There was neither detail introduction of various components of the calibration platform nor the specially designed software to make further analysis of the features of the sensor. Peter Tant and Hang Bo made a quite detail introduction of calibration method applied to the total electric field measurement at ground level for HVDC transmission lines. They focused on the analysis of electric field characteristics of the calibration device itself rather than the automatic calibration of the sensor[8],[9]. Manual calibration was still applied to calibrate the sensor with one or several sensor objects all the time in the mentioned references. The output voltage of the power supply shall be manually adjusted for the manual calibration system and the output of the sensor under this voltage will be recorded by hand. In addition, theoretical value of electric field strength might be calculated on the basis of the magnitude of supply voltage and the relationship between electric field strength and output voltage of the sensor might be analyzed, which provides the basis for the calculation of static characteristics of the sensor. It is a process in great need of labor that not only wastes

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 541–550, 2013.

time and energy, but introduces artificial error. Specifically, there will be a known shortcoming on manual basis in case of the mass production of sensors (as shown in Figure 1).

Therefore, a kind of automatic calibration device based on the virtual instrument technology is introduced in this study to analyze the static characteristics of the sensor including sensitivity, hysteresis and repetition without manual intervention.

Fig. 1. The mass production of field mills

Here is the structure of this paper: the composition of auto-calibration system of the electric field sensor is first introduced; secondly, calibration device is analyzed on the basis of finite element analysis and design of calibration software which is the key component is discussed in detail; finally, the relevant analysis of experimental results is made.

2 Theory and Method

With the principle of a conductor generating inductive charge in external electric field, electric field is measured by field mill [6-7]. Based on the Gauss theorem, the electric flux of electrostatic field passing any closed surface equals to the algebraic addition of free charges surrounded by the surface dividing by the vacuum dielectric constant in vacuum.

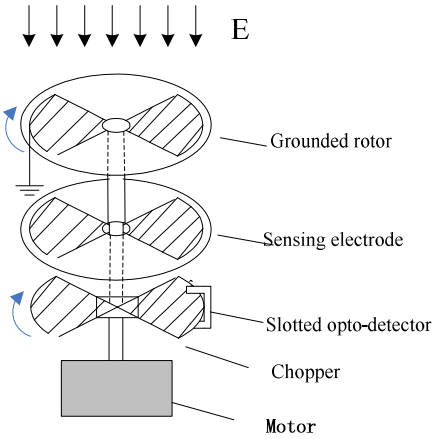


Fig. 2. Mechanical structure of filed mill

The mechanical structure of the field mill is shown in Figure.2. The filed mill mainly consists of two parallel metal electrodes with surface openings (a rotating grounded electrode and a sensing electrode), a chopper, an opto-detector and a motor. The sensing electrode is fixed; moreover, the rotating electrode and the chopper coaxially rotate with the motor. The active area of the sensing electrod cyclically exposed to the incident electric field through the grounded rotor. Consequently, inductive charge with periodic changes generates alternating current signal:

$$i(t) = \frac{dQ(t)}{dt} = \varepsilon_0 E \frac{dS(t)}{dt} \tag{1}$$

where ε_0 is the vacuum dielectric constant, E is the incident electric field strength and $\frac{dS(t)}{dt}$ is the change rate of the area of the sensing electrode exposed to the measured electric field. Therefore, the measured electric field strength might be measured by detecting the magnitude of the induced current *i (t)*. This kind of sensor is widely used to measure the total electric field at ground level for the HVDC overhead transmission lines, characterized by small volume, light weight, good linearity, and simple operation etc.

In calibration, the field mill is placed at the uniform field generated by two parallel plates according to the standard of IEEE Std 1227TM-1990(R2010). Appropriate area of the plates and separation between parallel plates are required to generate uniform electric field and provide space to place the sensor in calibration. The calibration of the sensor is realized by connecting adjustable DC power supply to two parallel plates and changing the electric field strength between parallel plate electrodes by means of changing power supply output voltage.

2.1 Overview of the Automatic Calibration System

Automatic calibration system consists of industrial computer, programmable DC power supply, calibration device and data acquisition card. The calibration device is composed by upper and lower parallel round metal plates, and a special clamp is adopted to fix the sensor on the lower plate with a circular opening.The voltage of the DC power supply is controlled by industrial computer with GPIB bus, which can change the electric field generated by the calibration device.Once the changed electric field is induced by the sensor, it will output analog signal for the data acquisition card which thereafter makes the acquired analog signal associated with the electric field generated by the calibration device for the purpose of obtaining the static characteristics of the sensor. XDC300-20 of Xantrex company used as the programmable DC power supply could generate adjustable voltage ranged from 0 to 300V and maximum output power of 6000W. A USB-powered Handyscope HS4 oscilloscope with four input channels sampling at 50 MHz 16-bit is used as the analogue-to-digital converter needed to digitize the output signal of field mill for

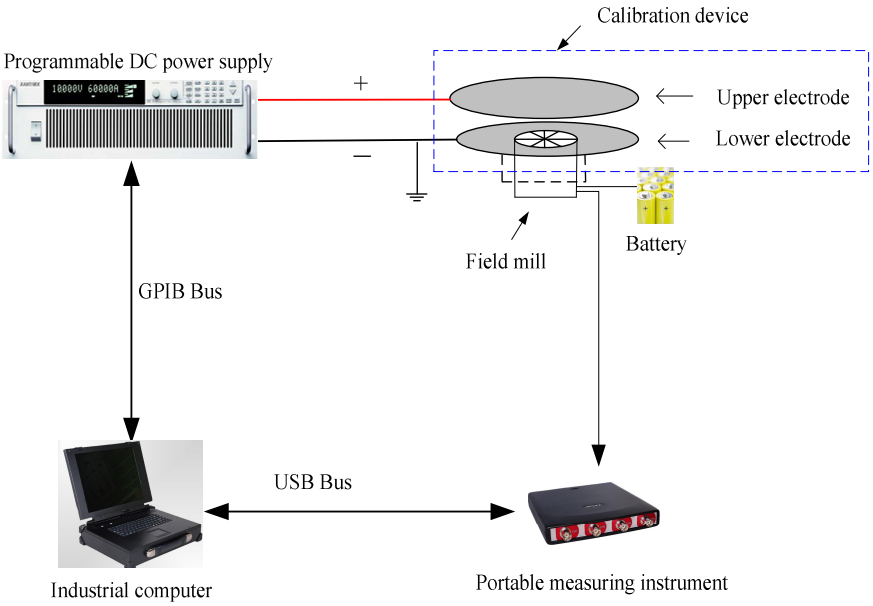


Fig. 3. Diagram of automatic calibration system for field mill

further processing on an industrial computer.The diagram of the automatic calibration system is illustrated in Figure.3.

The design of the calibration device and the development of PC-based calibration software play a critical role in the design of the overall auto-calibration system. The design of the calibration device should meet certain conditions to generate uniform electric field. Moreover, the calibration software controls the whole process of the calibration and calculates the calibration results.

2.2 The Finite Element Analysis for the Calibration Device

As shown in Figure 4, the modeling of mechanical structure of the calibration device is modeled based on the correlation theory of the finite element. As shown in Figure 5, the electric field distribution between electrode plates of the calibration device might be solved by element subdivision and load on the basis of this geometric model. The structure parameters of the calibration device are designed according to the obtained electric field distribution. In the calculation and solution procedure, the voltage applied to two plates is changed to maintain the theoretical value of the electric field between plates in 20kV/m all the time. The diameter of field mill shell in current design is 8cm and the diameter of the sensing electrode is 6cm. In addition, the sensor shell gets in touch with the lower plate of the calibration device.

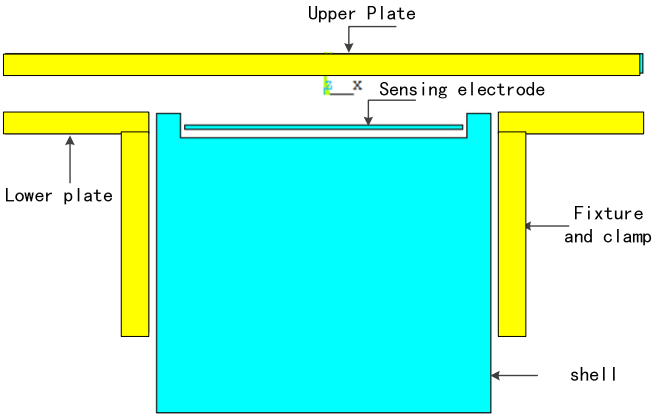
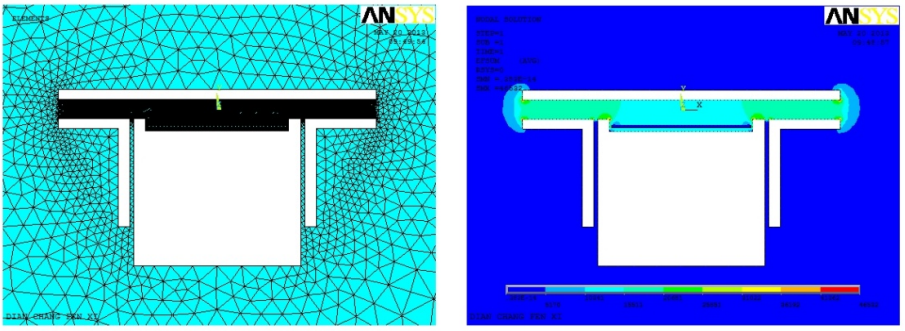


Fig. 4. Simplified geometrical model of field mill and calibration device.（The yellow area is the calibration device and the blue area is filed mill）



(a) (b)

Fig. 5. (a)Mesh generation of the sensor and calibration (b)Electric field vector diagram around the sensor and the calibration device

In the design process of the calibration device, the plate diameter (L=16cm) is invariable because of the dimensional limit of the sensor and the consideration of the portability of the device. Under such circumstances, the influences of the distance between plates (H) on electric field between plates need to be analyzed so as to determine the distance between plates (H). The results in Figure 6 might be got with respective H value of 1 cm, 2 cm ,3 cm, 4 cm and 5 cm in the model built according to the Figure5. The Figure 6 showed that if the size of the plate (L=16cm) and theoretical electric field (20kV/m) are invariable, the real electric field between plates is gradually approximate to the theoretical electric field (20kV/m) with the increasing distance between plates (H); meanwhile, within the sensitivity area of the sensing electode(x<0.03m), electric field is not a constant value and the real electric field realizes the distortion with the increasing center distance of the calibration device.

The distortion of the real electric field between plates is shown in Figure 7. It is found that when H=5cm, the distortion of the electric field is least. With the overall analysis of Figure 6 and Figure 7, it is found that when H=5cm, the real electric field is more approximate to electric field of theoretical calculation; simultaneously, the distortion rate of the electric field is lower.

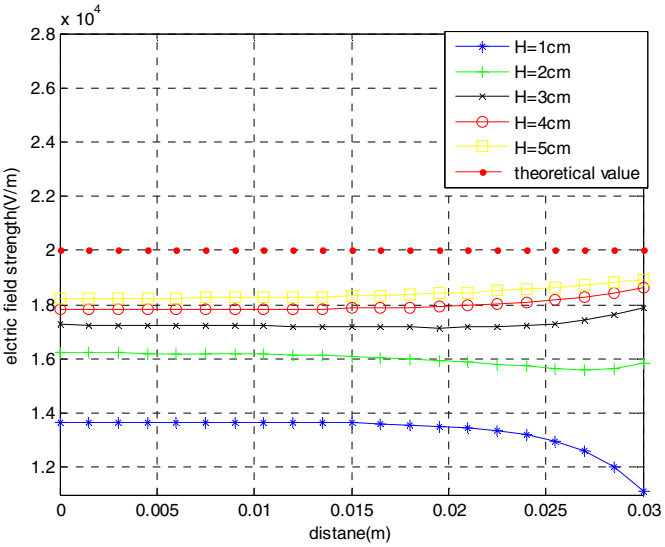


Fig. 6. Transverse electric field distribution diagram at the top of sensing plate with different separation distance between the upper plate and lower plate of the calibration device

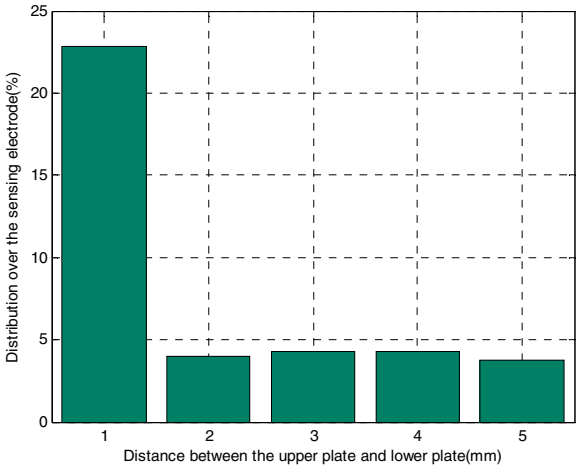


Fig. 7. The relationship between distribution over the sensing electrode and the seperation of two plates

If the distance between the upper plate and lower plate is invariable (H=5cm), the electric field distribution over the sensing electrode of field mill is shown in Figure 8 when the diameter of the plate is 16cm and 32cm respectively, which indicates that the increase of L has a weak influence on the reduction of the electric field distortion between two plates. It seems to be contradictory to the general consideration that the smaller the ratio of H/L, the less electric field distortion between plates, because electric field distortion rate is quite small if H/L is less than 0.5 or less and it might be considered that H/L in this range might not affect the electric field. In the foresaid simulation, H/L is ranged from 0.03125 to 0.15625 (5/16~5/32) which is far less than 0.5. Therefore, the influence of the change of H/L resulted from the change of L can be basically ignored, which explains why two curves are basically coincided after the change of H/L.

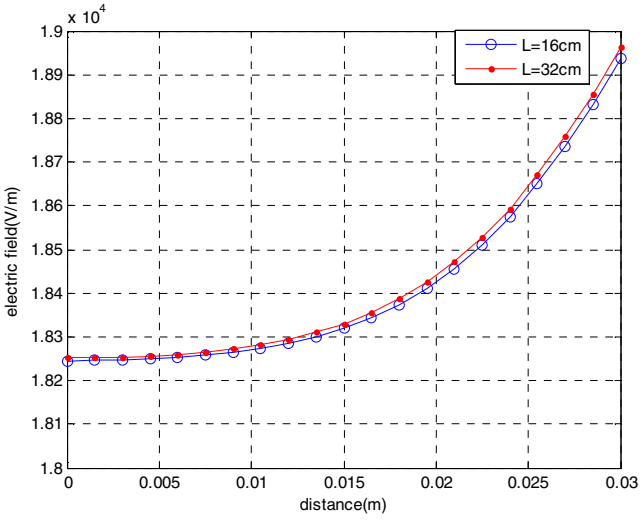


Fig. 8. Transverse electric field distribution diagram over sensing electrode with different diameter of calibration device plate when *H*=5cm

To sum up, when the diameter of the plate is 16cm and the distance between plates is 5cm, the real electric field is most approximate to theoretical electric field and the electric field distortion in sensing area of the sensing electrode of the field mill is least. Simultaneously, if it is guaranteed that the H/L is less than 0.5, the diameter of the plate (L) has extremely less intrusive on uniform electric field.

2.3 Design of Calibration Software

The calibration software is developed with the platform provided by the virtual instrument software, Labwindows/CVI^{TM}. Virtual instrument is the direction of the research and development of the measurement. LabWindows/CVI^{TM} based on C language used as software development platform enormously reduces the difficulty compared with the common object-oriented software programming and improves the software development efficiency and expansibility. Relevant operation with the

build-in GPIB is performed to control the programmable power supply; besides, with the build-in DLL, the control over data acquisition card might be realized; moreover, the signal processing of the acquired sensor datas might be performed by the advanced analysis tools provided by CVI. Measurement and analysis results are stored in ACCESS database or realize report output in the form of Excel. Hardware and labor are replaced by software with the powerful functions of the virtual instrument platform. The overall framework of the software for the total system is shown in Figure 9 (a) and the main interface of the software after login is shown in Figure9 (b).

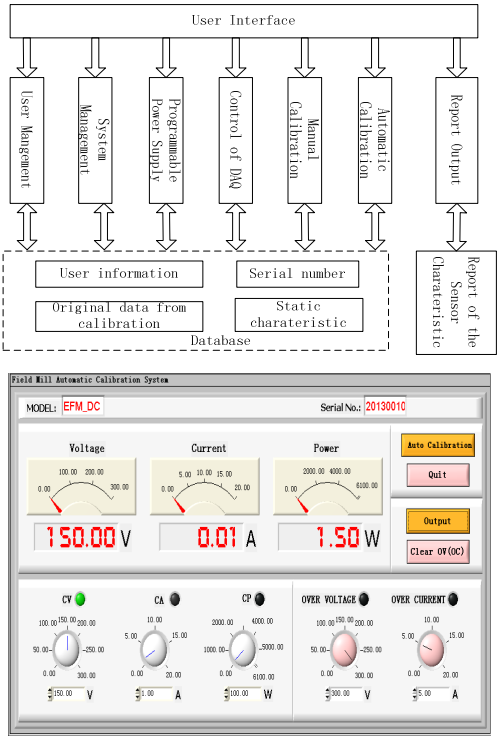


Fig. 9. (a) The block diagram of the automatic calibration software. (b) The main panel of software.

3 Experimental Results

The established automatic calibration system is applied to the sensors with the serial number of 20130010, 20120159, 20120035 and 20120036 and the experimental results are shown in Figure10. It is found that all of these four sensors have better linearity.

As shown in Table 1, the analysis functions of the software is used to obtain the sensitivity, non-linear error and R^2 of these four sensors. In Table 1, it is observed that the maximum non-linear error of these four sensors is at most 0.18% and R^2 is 1 without exception, which indicates terrific linear characteristics of these sensors and simultaneously shows the validity of the design of the calibration system.

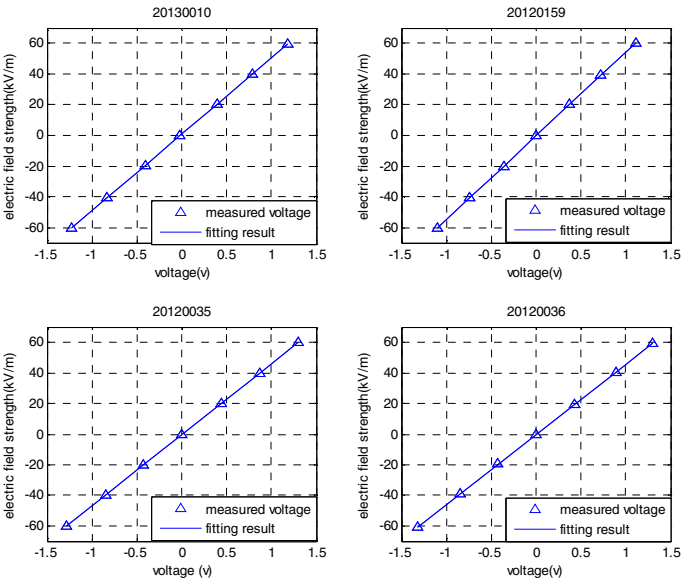


Fig. 10. The calibration results of four filed mills

Table 1. The sensitivity, non-linear error and R^2 of these four sensors

Serial number	Sensitivity	Sensitivity error （R^2）	Nonlinear error （%）
20130010	49.6750	1.000	0.15
20120159	54.4976	1.000	0.12
20120035	46.4296	1.000	0.13
20120036	46.0124	1.000	0.18

4 Conclusion

Automatic calibration system applied to the total electric field sensor at ground level for the HVDC overhead transmission lines is designed in this research; moreover, finite element analysis and designs of the calibration device in the system are made. Then the calibration software based on the virtual instrument is developed to calibrate the sensor made by laboratory. The experimental results show that this system can provide calibration service for the sensor in a better way. With this system, the calibration accuracy and efficiency of the field mill can be improved compared with the manual calibration.

Acknowledgment. This work is supported by the National Nature Science Fund of China under grant 61273125.

References

1. Rao, Y.J., Gnewuch, H., Pannell, C.N., Jackson, D.A.: Electro-optic electric field sensor based on periodically poled LiNbO. Electronics Letters 35(7), 596–597 (1999)
2. Cui, Z., Gong, C., Xia, S.: Design and modelling of a microsensor for atmospheric electric field measurement. In: Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS 2003, pp. 154-158. IEEE (May 2003)
3. Bateman, M.G., Stewart, M.F., Podgorny, S.J., Christian, H.J., Mach, D.M., Blakeslee, R.J., Daskar, D.: A low-noise, microprocessor-controlled, internally digitizing rotating-vane electric field mill for airborne platforms. Journal of Atmospheric and Oceanic Technology 24(7), 1245–1255 (2007)
4. Montanyà, J., Rodríguez, P., Bergas, J., Illa, A., Hermoso, B., Candela, I.: A new electrostatic field measurement method: The coherent-notch field mill. Journal of Electrostatics 65(7), 431–437 (2007)
5. Fort, A., Mugnaini, M., Vignoli, V., Rocchi, S., Perini, F., Monari, J., Fiocchi, F.: Design, Modeling, and Test of a System for Atmospheric Electric Field Measurement. IEEE Transactions on Instrumentation and Measurement 60(8), 2778–2785 (2011)
6. Jiang, H., Zhou, B., Li, Y.: The latest design and development of the field mill used as atmospheric electric field sensor. In: 5th Asia-Pacific Conference on Environmental Electromagnetics, CEEM 2009, pp. 159–163. IEEE (September 2009)
7. Tant, P., Bolsens, B., Sels, T., Van Dommelen, D., Driesen, J., Belmans, R.: Design and application of a field mill as a high-voltage DC meter. IEEE Transactions on Instrumentation and Measurement 56(4), 1459–1464 (2007)
8. Zhang, B., Wang, W., He, J., Zeng, R., Yin, H.: Calibration of field-mill instrument for measuring DC electric field. In: 2012 International Conference on High Voltage Engineering and Application (ICHVE), September 17-20, pp. 455–458 (2012)
9. Cui, Y., Yuan, H., Wang, Q., Tian, C.: Finite Element Analysis and Design on Calibrating Device of Rotary Electric Field Sensor. In: 2012 Second International Conference on InInstrumentation, Measurement, Computer, Communication and Control (IMCCC), pp. 49–52. IEEE (December 2012)

Electrostatic Simulation in Dielectric Composite Material by Extended Element-Free Galerkin Method

Taku Itoh[1,⋆], Ayumu Saitoh[2], Soichiro Ikuno[1], and Atsushi Kamitani[3]

[1] School of Computer Science, Tokyo University of Technology,
1404-1 Katakura-machi, Hachioji, Tokyo 192-0982, Japan
[2] Graduate School of Engineering, University of Hyogo,
2167 Shosha, Himeji, Hyogo 671-2280, Japan
[3] Graduate School of Science and Engineering, Yamagata University,
4-3-16 Johnan, Yonezawa, Yamagata 992-8510, Japan
taku@m.ieice.org

Abstract. To investigate applicability of the eXtended Element-Free Galerkin method (X-EFG) to realistic problems, the X-EFG has been applied to an electrostatic problem in a dielectric composite material. Numerical experiments show that the electrostatic potential obtained by the X-EFG is naturally distributed in the dielectric composite material, and the accuracy of the numerical solution obtained by the X-EFG is almost the same as that of the Standard EFG (S-EFG). In addition, the condition numbers of the coefficient matrices of the linear systems generated by the X-EFG are considerably less than those of the S-EFG.

Keywords: Meshless methods, Collocation, Lagrange multiplier, Electromagnetic analysis, Partial differential equations, Condition number.

1 Introduction

Meshless methods as typified by the Element-Free Galerkin method (EFG) [2], the Meshless Local Petrov-Galerkin method (MLPG) [1], and the meshfree Radial Point Interpolation Method (RPIM) [13] have widely been applied to numerical simulations in a lot of fields, including electromagnetics, and have produced many attractive results [3,10,7,6,12,11]. In the meshless methods, elements of a geometrical structure are no longer required [9].

In meshless methods, boundary conditions of course have to be imposed by using some kinds of methods. In particular, the EFG employs the Lagrange multiplier [2] to impose the essential boundary condition. In the Lagrange multiplier, evaluations of some numerical integrations are usually required.

On the other hand, the EFG has recently been reformulated, and the resulting EFG is called the eXtended EFG (X-EFG) [8]. In the X-EFG, a new imposing method of the essential boundary condition is proposed. By using this method,

⋆ Corresponding author.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 551–560, 2013.

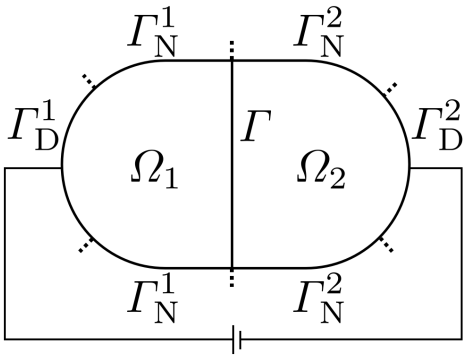


Fig. 1. Schematic view of a dielectric composite material

the essential boundary condition can be imposed without any numerical integrations. Hence, it is expected that, in comparison with the EFG, the X-EFG can be applied to various problems more easily.

The purpose of this study is to investigate applicability of the X-EFG to realistic problems. To this end, the X-EFG is applied to an electrostatic problem in a dielectric composite material, and the performance of the X-EFG is compared with that of the standard EFG.

2 Electrostatic Problem in Dielectric Composite Material

2.1 Governing Equations

A non-dimensional form of an electrostatic problem in a dielectric composite material can be described by

$$-\nabla\cdot(\epsilon_k\nabla\phi_k)=\rho_k \quad \text{in } \Omega_k\ (k=1,2), \tag{1}$$

$$\phi_k=\bar{\phi}_k \quad \text{on } \Gamma_{\mathrm{D}}^k\ (k=1,2), \tag{2}$$

$$\epsilon_k\frac{\partial\phi_k}{\partial n_k}=\bar{q}_k \quad \text{on } \Gamma_{\mathrm{N}}^k\ (k=1,2), \tag{3}$$

where the dielectric composite material is constructed by Ω_1 and Ω_2, and Γ_{D}^k, Γ_{N}^k and Γ are parts of $\partial\Omega_k$ such that $\Gamma_{\mathrm{D}}^k\cup\Gamma_{\mathrm{N}}^k\cup\Gamma=\partial\Omega_k$ and $\Gamma_{\mathrm{D}}^k\cap\Gamma_{\mathrm{N}}^k\cap\Gamma=\emptyset$. Here, Γ is the interface between Ω_1 and Ω_2. A schematic view of the dielectric composite material is shown in Fig. 1. In addition, ϵ_k, ϕ_k and ρ_k are the electric permittivity, the electrostatic potential and the charge density on Ω_k, respectively. Furthermore, $\boldsymbol{n}_k$ indicates an outward unit normal on $\partial\Omega_k$, and $\bar{\phi}_k$ and $\bar{q}_k$ denote prescribed boundary values on Γ_{D}^k and Γ_{N}^k, respectively. Note that interface conditions are assumed as

$$\phi_1=\phi_2,\ \rho_1=\rho_2 \quad \text{on} \quad \Gamma. \tag{4}$$

and

$$\epsilon_1 \frac{\partial \phi_1}{\partial n_1} + \epsilon_2 \frac{\partial \phi_2}{\partial n_2} = 0 \quad \text{on} \quad \Gamma. \tag{5}$$

From (1), (3) and (5), the weak form is derived as

$$^{\forall} w \text{ s.t. } w\big|_{\Gamma_{\mathrm{D}}^k=0} \, (k=1,2) :$$
$$\sum_{k=1}^{2} \epsilon_k \int_{\Omega_k} \nabla w \cdot \nabla \phi_k \mathrm{d}^2 \boldsymbol{x} = \sum_{k=1}^{2} \left(\int_{\Omega_k} w \rho_k \mathrm{d}^2 \boldsymbol{x} + \epsilon_k \int_{\Gamma_{\mathrm{N}}^k} w \bar{q}_k \mathrm{d}s \right). \tag{6}$$

Here, $w(\boldsymbol{x})$ is a test function.

2.2 Discretization by Using the X-EFG

Equation (6) can separately be discretized for each domain. Hence, throughout this subsection, we do not indicate the domain number k.

To discretize (6), the nodes, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ are first placed both in Ω and on $\partial\Omega$, and shape functions $\psi_1(\boldsymbol{x}), \psi_2(\boldsymbol{x}), \ldots, \psi_N(\boldsymbol{x})$ are determined by using the Moving Least-Squares (MLS) approximation [2,1,9]. Here, N is the number of nodes in $\Omega \cup \partial\Omega$. In the following, M denotes the number of nodes on $\partial\Omega$. In addition, the orthonormal system in $\mathbb{R}^N$ and that in $\mathbb{R}^M$ are denoted by $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N\}$ and $\{\bar{\boldsymbol{e}}_1, \bar{\boldsymbol{e}}_2, \ldots, \bar{\boldsymbol{e}}_M\}$, respectively.

Let us first discretize the weak form (6) except the constraint. To this end, we assume that ϕ and w can be expanded with $\psi_i(\boldsymbol{x})$ $(i = 1, 2, \ldots, N)$ as follows:

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{N} \hat{\phi}_i \psi_i(\boldsymbol{x}), \quad w(\boldsymbol{x}) = \sum_{i=1}^{N} \hat{w}_i \psi_i(\boldsymbol{x}). \tag{7}$$

By substituting (7) into (6), we obtain

$$(\hat{\boldsymbol{w}}, A\hat{\boldsymbol{\phi}} - \boldsymbol{f}) = 0, \tag{8}$$

where $\hat{\boldsymbol{w}} = [\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_N]^{\mathrm{T}}$, and $\hat{\boldsymbol{\phi}} = [\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_N]^{\mathrm{T}}$. In addition, A and $\boldsymbol{f}$ are defined as

$$A \equiv \sum_{i=1}^{N} \sum_{j=1}^{N} \epsilon \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j \ \mathrm{d}^2 \boldsymbol{x} \ \boldsymbol{e}_i \boldsymbol{e}_j^{\mathrm{T}}, \tag{9}$$

$$\boldsymbol{f} \equiv \sum_{i=1}^{N} \left(\int_{\Omega} \psi_i \rho \ \mathrm{d}^2 \boldsymbol{x} + \epsilon \int_{\Gamma_{\mathrm{N}}} \psi_i \bar{q} \ \mathrm{d}s \right) \boldsymbol{e}_i \,. \tag{10}$$

Next, the constraint $w|_{\Gamma_{\mathrm{D}}=0}$ in (6) is discretized. To this end, the constraint is rewritten as the equivalent proposition:

$$^{\forall} \beta(s) : \int_{\partial\Omega} \beta(s) w(\boldsymbol{x}(s)) \ \mathrm{d}s = 0, \tag{11}$$

and an arbitrary function $\beta(s)$ is assumed to be contained in span$(N_1, N_2, \ldots, N_M)$, where $N_1(s)$, $N_2(s), \ldots, N_M(s)$ are linearly independent functions on $\partial\Omega$. Here, s is a parameter for representing $\partial\Omega$. By using $N_1(s)$, $N_2(s), \ldots, N_M(s)$, (11) can be discretized as

$$(\hat{\boldsymbol{w}}, \boldsymbol{c}_m) = 0 \ (m = 1, 2, \ldots, M), \tag{12}$$

where $\boldsymbol{c}_m$ $(m = 1, 2, \ldots, M)$ are defined as

$$\boldsymbol{c}_m \equiv \sum_{i=1}^{N} \int_{\partial\Omega} N_m(s)\psi_i(\boldsymbol{x}(s)) \, \mathrm{d}s \ \boldsymbol{e}_i. \tag{13}$$

Note that (12) indicate $\hat{\boldsymbol{w}} \in \mathcal{V}^{\perp}$, where

$$\mathcal{V} = \mathrm{span}(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_M). \tag{14}$$

Hence, the weak form (6) can be discretized as

$$^{\forall}\hat{\boldsymbol{w}} \in \mathcal{V}^{\perp} : (\hat{\boldsymbol{w}}, A\hat{\boldsymbol{\phi}} - \boldsymbol{f}) = 0. \tag{15}$$

Since $(\mathcal{V}^{\perp})^{\perp} = \mathcal{V}$, (15) can be written as

$$A\hat{\boldsymbol{\phi}} - \boldsymbol{f} \in \mathcal{V}. \tag{16}$$

Therefore, there exists $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^M$ such that

$$A\hat{\boldsymbol{\phi}} + C\hat{\boldsymbol{\alpha}} = \boldsymbol{f}, \tag{17}$$

where $C \equiv [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_M]$ and $\hat{\boldsymbol{\alpha}} \equiv [\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_M]^{\mathrm{T}}$.

Finally, the boundary condition (2) is discretized. By the similar procedures for discretizing the constraint $w|_{\Gamma_{\mathrm{D}}=0}$, (2) can be discretized as

$$C^{\mathrm{T}}\hat{\boldsymbol{\phi}} = \boldsymbol{g}, \tag{18}$$

where

$$\boldsymbol{g} \equiv \sum_{j=1}^{M} \int_{\Gamma_{\mathrm{D}}} N_j(s)\bar{\phi}(s) \, \mathrm{d}s \ \bar{\boldsymbol{e}}_j. \tag{19}$$

Equations (17) and (18) can be written in the form,

$$\begin{bmatrix} A & C \\ C^{\mathrm{T}} & O \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\phi}} \\ \hat{\boldsymbol{\alpha}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}. \tag{20}$$

Equation (20) is a discretized form of the electrostatic problem described in Sect. 2.1. Note that (20) is generated for each domain. Hence, two different linear systems have to be solved in this problem.

In the X-EFG, the linearly independent functions $N_1(s), N_2(s), \ldots, N_M(s)$ are required in (13) and (19). Since an arbitrary function $\beta(s)$ is assumed to

be contained in $\mathrm{span}(N_1, N_2, \ldots, N_M)$, $N_1(s), N_2(s), \ldots, N_M(s)$ can be chosen arbitrarily. Here, to impose the essential boundary condition easily, δ-functions defined on $\partial\Omega$ are employed as $N_i (i = 1, 2, \ldots, M)$. The explicit form of $N_i (i = 1, 2, \ldots, M)$ is given as

$$N_i(s) = \delta(s - s_i) \quad (i = 1, 2, \ldots, M). \tag{21}$$

Note that, on $\partial\Omega$, the ith boundary node $\boldsymbol{x}_i$ is represented by s_i, i.e., $\boldsymbol{x}_i = \boldsymbol{x}(s_i)$. By using the δ-functions, C and $\boldsymbol{g}$ can be rewritten as

$$C = \sum_{i=1}^{N} \sum_{j=1}^{M} \psi_i(\boldsymbol{x}(s_j))\ \boldsymbol{e}_i \bar{\boldsymbol{e}}_j{}^{\mathrm{T}}, \quad \boldsymbol{g} = \sum_{j=1}^{M} \bar{\phi}(s_j)\ \bar{\boldsymbol{e}}_j. \tag{22}$$

It must be noted here that, by using the δ-functions as $N_i (i = 1, 2, \ldots, M)$, C and $\boldsymbol{g}$ can easily be evaluated without any integrations.

3 Numerical Experiments

Throughout this section, the domain number k is always set as $k = 1, 2$. In this section, numerical experiments are conducted to evaluate the X-EFG for an electrostatic problem in a dielectric composite material. To this end, the performance of the X-EFG for this problem is compared with that of the Standard EFG (S-EFG), that denotes the EFG with the Lagrange multiplier, in a dielectric composite material whose boundary shape is represented as $f(\boldsymbol{x}) = x^2/9 + y^2/4 - 1 = 0$, and Ω_1 and Ω_2 are defined as $\max(f(\boldsymbol{x}), x) < 0$ and $\max(f(\boldsymbol{x}), -x) < 0$, respectively (see Fig. 2). In addition, Γ_{D}^1 and Γ_{D}^2 are defined as $f(\boldsymbol{x}) = 0$ with $x \leq -2$ and that with $x \geq 2$, respectively. Furthermore, Γ is defined as $f(\boldsymbol{x}) < 0$ with $x = 0$, and $\Gamma_{\mathrm{N}}^k = \partial\Omega_k - \Gamma_{\mathrm{D}}^k - \Gamma$. Here, we set $\rho_1 = \rho_2 = \sin x \sin y$, $\epsilon_1 = 4.5$ and $\epsilon_2 = 9.4$. As boundary conditions, we set $\bar{\phi}_1 = 1$ on Γ_{D}^1, $\bar{\phi}_2 = 3$ on Γ_{D}^2, $\bar{\phi}_1 = \bar{\phi}_2 = 5$ on Γ, and $\bar{q}_k = 0$ on Γ_{N}^k.

The boundary nodes $\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_{M_k}^k$ are uniformly placed on $\partial\Omega_k$, and the nodes $\boldsymbol{x}_{M_k+1}^k, \boldsymbol{x}_{M_k+2}^k, \ldots, \boldsymbol{x}_{N_k}^k$ are also uniformly placed in Ω_k, where M_k and N_k are the numbers of nodes on $\partial\Omega_k$ and in Ω_k, respectively. In addition, the exponential weight function [2],

$$w(r) \equiv \begin{cases} \dfrac{\exp[-(r/c)^2] - \exp[-(R/c)^2]}{1 - \exp[-(R/c)^2]} & (r \leq R), \\ 0 & (r > R), \end{cases} \tag{23}$$

is adopted for the MLS approximation. Here, R denotes a support radius, and c is a user-specified parameter. We set $R = 1.7h$ and $c = h$, where h is the average of the minimum distance between two nodes.

In the MLS approximation, the shape functions $\psi_i^k(\boldsymbol{x})$ $(i = 1, 2, \ldots, N_k)$ can be determined by

$$\psi_i^k(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) B_k^{-1}(\boldsymbol{x}) \boldsymbol{b}_i^k(\boldsymbol{x}), \tag{24}$$

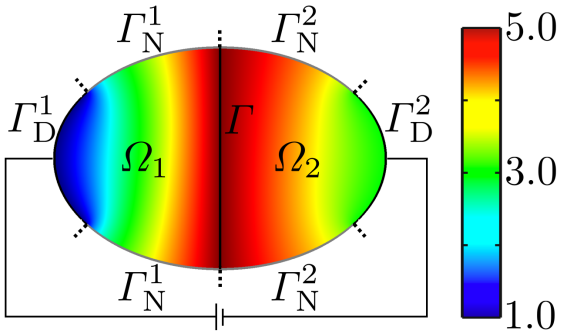


Fig. 2. Distribution of the electrostatic potential obtained by the X-EFG for $N_1+N_2 = 138816$

where $\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) = [1\ x\ y]$. In addition, the matrix $B_k(\boldsymbol{x})$ and the vector $\boldsymbol{b}_i^k(\boldsymbol{x})$ are defined as

$$B_k(\boldsymbol{x}) = \sum_{j=1}^{N} w_j^k(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}_j^k)\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j^k), \tag{25}$$

$$\boldsymbol{b}_i^k(\boldsymbol{x}) = w_i^k(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}_i^k), \tag{26}$$

where $w_i^k(\boldsymbol{x}) = w(|\boldsymbol{x}-\boldsymbol{x}_i^k|)$. For (9), the partial derivatives of $\psi_i^k(\boldsymbol{x})$ with respect to $X(= x$ and $y)$ can be determined as

$$\psi_{i,X}^k(\boldsymbol{x}) = \boldsymbol{p}_X^{\mathrm{T}}(\boldsymbol{x})B_k^{-1}(\boldsymbol{x})\boldsymbol{b}_i^k(\boldsymbol{x}) + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})[B_{k,X}^{-1}(\boldsymbol{x})\boldsymbol{b}_i^k(\boldsymbol{x}) + B_k^{-1}(\boldsymbol{x})\boldsymbol{b}_{i,X}^k(\boldsymbol{x})], \tag{27}$$

where

$$B_{k,X}^{-1}(\boldsymbol{x}) = -B_k^{-1}(\boldsymbol{x})B_{k,X}(\boldsymbol{x})B_k^{-1}(\boldsymbol{x}). \tag{28}$$

For evaluating domain integrals contained in (9) and (10), a square cell structure being independent of the nodes is used [2,9], and the Gauss-Legendre quadrature is employed. The number N_{Q} of quadrature points depends on the number m of nodes in a cell. Throughout this section, N_{Q} is handled on the similar criterion in [2], i.e., $N_{\mathrm{Q}} = n_{\mathrm{Q}} \times n_{\mathrm{Q}}$, where $n_{\mathrm{Q}} = \lfloor\sqrt{m}+0.5\rfloor+2$. Note that, if a cell contains a part of boundary of domain, we set $n_{\mathrm{Q}} = 2(\lfloor\sqrt{m}+0.5\rfloor+2)$. In addition, the number N_{C} of cells is set as $N_{\mathrm{C}} = m_{\mathrm{C}} \times m_{\mathrm{C}}$, where $m_{\mathrm{C}} = \lfloor\sqrt{N}\rfloor$. To evaluate line integrals contained in (10), the Gauss-Legendre quadrature is also employed, and the number of quadrature points is set as 5.

To solve the linear system (20), a LU factorization is employed. In particular, we adopt the sequential SuperLU [5]. In addition, the Column Approximate Minimum Degree Ordering (COLAMD) [4] is employed as an ordering method. This ordering method can easily be used by setting `options.ColPerm = COLAMD` in the SuperLU.

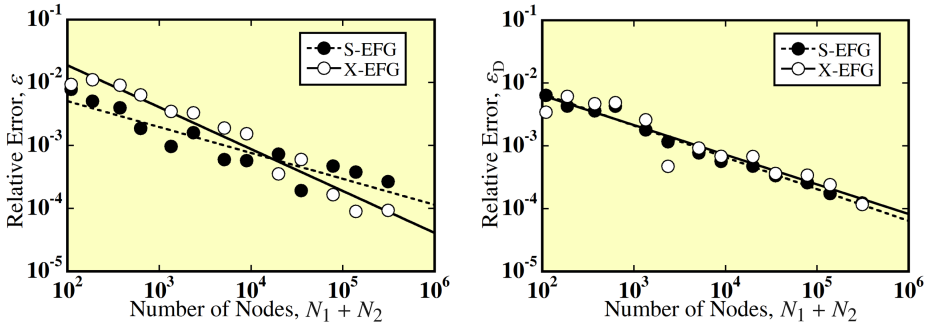


Fig. 3. (a) Dependence of the relative error ε on the number of nodes. Here, N_1 and N_2 are the numbers of nodes in $\Omega_1 \cup \partial\Omega_1$ and $\Omega_2 \cup \partial\Omega_2$, respectively. (b) Dependence of the relative error $\varepsilon_{\rm D}$ on the number of nodes.

Computations were performed on a computer equipped with a 2.66GHz Intel Core i7 920 processor, 24GB RAM, Ubuntu Linux ver. 12.10, and g++ ver. 4.7.2. Note that we only used a single core of this processor in the following experiments. Compiler options were set as "`-O3 -Wall -m64`." In addition, for using SuperLU, the OpenBLAS [14] ver. 0.1 alpha 2.2 was employed for acceleration.

First, the distribution of the electrostatic potential obtained by using the X-EFG is shown in Fig. 2. We see from this figure that the electrostatic potential is naturally changed in the dielectric composite material. For investigating the accuracy of the electrostatic potential, a numerical solution ϕ of X-EFG is compared with that of the S-EFG. To this end, relative errors ε and $\varepsilon_{\rm D}$ are calculated as a function of the number $N_{\rm total}$ of nodes and are shown in Fig. 3(a) and Fig. 3(b), respectively, where $N_{\rm total} = N_1 + N_2$. The relative errors ε and $\varepsilon_{\rm D}$ are defined by

$$\varepsilon \equiv \frac{\sqrt{\displaystyle\int_{\Omega_1\cup\Omega_2} (\phi - \phi_{\rm FEM})^2 {\rm d}^2\boldsymbol{x}}}{\sqrt{\displaystyle\int_{\Omega_1\cup\Omega_2} \phi_{\rm FEM}^2 {\rm d}^2\boldsymbol{x}}}, \text{ and } \varepsilon_{\rm D} \equiv \frac{\sqrt{\displaystyle\int_{\Gamma_{\rm D}^1\cup\Gamma_{\rm D}^2\cup\Gamma} \left(\phi - \bar{\phi}\right)^2 {\rm d}s}}{\sqrt{\displaystyle\int_{\Gamma_{\rm D}^1\cup\Gamma_{\rm D}^2\cup\Gamma} \bar{\phi}^2 \, {\rm d}s}}, \tag{29}$$

where $\phi_{\rm FEM}$ is a numerical solution of the electrostatic potential obtained by the Finite Element Method (FEM). In addition, since (4) is assumed, we represent the numerical solution ϕ as

$$\phi(\boldsymbol{x}) \equiv \begin{cases} \phi_1(\boldsymbol{x}) \text{ for } \boldsymbol{x} \in \Omega_1, \\ \phi_2(\boldsymbol{x}) \text{ for } \boldsymbol{x} \in \Omega_2. \end{cases} \tag{30}$$

Note that, for calculating ε, we always use $\phi_{\rm FEM}$ obtained with $N_{\rm total} = 311346$. Namely, we consider $\phi_{\rm FEM}$ obtained with $N_{\rm total} = 311346$ as an almost exact solution of ϕ. We see from Fig. 3(a) that, for $N_{\rm total} < 10^4$, the accuracy of ϕ

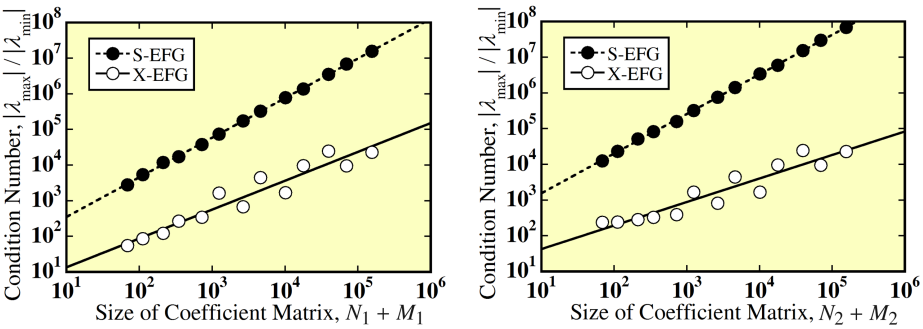


Fig. 4. Dependence of the condition numbers on the size of coefficient matrices for (a) $k = 1$ and (b) $k = 2$

obtained by the S-EFG is better than that of the X-EFG. However, for $N_{\text{total}} \geq 10^4$, the accuracy of ϕ obtained by the X-EFG is superior than that of S-EFG except for $N_{\text{total}} = 35224$. In addition, we see from Fig. 3(b) that, on $\Gamma_{\text{D}}^1 \cup \Gamma_{\text{D}}^2 \cup \Gamma$, there is no obvious difference between accuracy of ϕ obtained by X-EFG and that of S-EFG. From these results, we consider that the accuracy of ϕ obtained by the X-EFG is almost the same as that of S-EFG. In addition, since the difference between X-EFG and S-EFG is the imposing method of the essential boundary condition, we consider that both imposing methods have almost the same performance. Note that, in the X-EFG, the essential boundary condition can be imposed without any integrations. Hence, without losing the accuracy of the numerical solution, a code of the X-EFG can be developed more easily in comparison with that of the S-EFG. In addition, the computational time of the X-EFG is almost equal to that of the S-EFG, e.g. for $N_{\text{total}} = 311346$, the computational time of the X-EFG is about 463.7(s) and that of the S-EFG is about 469.7(s).

Next, to investigate properties of linear systems, the condition numbers of the coefficient matrices for $k = 1, 2$ are calculated as a function of the size of matrix, $N_k + M_k$. The condition number is determined by $|\lambda_{\max}|/|\lambda_{\min}|$, where $|\lambda_{\max}|$ and $|\lambda_{\min}|$ are the maximum and minimum eigenvalues of the coefficient matrix of (20). The calculation results for $k = 1$ and $k = 2$ are shown in Figs. 4(a) and 4(b), respectively. We see from these figures that the condition numbers of the coefficient matrices generated by the X-EFG are considerably less than those of the S-EFG. Hence, it is expected that the X-EFG can be applied to larger scale problems in comparison with the S-EFG, since increase of the condition number can be repressed by using the X-EFG.

4 Conclusion

To investigate applicability of the X-EFG to realistic problems, the X-EFG has been applied to an electrostatic problem. By using the X-EFG, there is a merit

such that the essential boundary conditions can be imposed without any integrations. In the numerical experiments, the performance of the X-EFG has been investigated by applying to an electrostatic problem in an ellipsoidal-shaped dielectric composite material. Conclusions obtained in the present study are summarized as follows:

1. The electrostatic potential obtained by the X-EFG is naturally distributed in the dielectric composite material.
2. Without losing the accuracy of the numerical solution, a code of the X-EFG can be developed more easily in comparison with that of the S-EFG. In addition, the computational time of the X-EFG is almost equal to that of the S-EFG.
3. The condition numbers of the coefficient matrices of the linear systems generated by the X-EFG are considerably less than those of the S-EFG. Hence, it is expected that the X-EFG can be applied to larger scale problems in comparison with the S-EFG.

As future work, the X-EFG will be applied to three-dimensional realistic problems in various fields, since the results have shown that the X-EFG is valid especially for large scale problems.

Acknowledgment. This work was partially supported by JSPS KAKENHI Grant Number 24700053.

References

1. Atluri, S.N., Zhu, T.: A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. Comput. Mech. 22, 117–127 (1998)
2. Belytschko, T., Lu, Y.Y., Gu, L.: Element-free Galerkin methods. Int. J. Numer. Methods Eng. 37, 229–256 (1994)
3. Coppoli, E.H.R., Mesquita, R.C., Silva, R.S.: Induction machines modeling with meshless methods. IEEE Trans. on Magnetics 48(2), 847–850 (2012)
4. Davis, T.A., Gilbert, J.R., Larimore, S., Ng, E.: A column approximate minimum degree ordering algorithm. ACM Trans. Mathematical Software 30, 353–376 (2004)
5. Demmel, J.W., Eisenstat, S.C., Gilbert, J.R., Li, X.S., Liu, J.W.H.: A supernodal approach to sparse partial pivoting. SIAM J. Matrix Analysis and Applications 20, 720–755 (1999)
6. Ikuno, S., Hanawa, T., Takayama, T., Kamitani, A.: Evaluation of parallelized meshless approach: Application to shielding current analysis in HTS. IEEE Trans. on Magn. 44, 1230–1233 (2008)
7. Ikuno, S., Takayama, T., Kamitani, A.: Weight function control of moving least-squares interpolants: Application to axisymmetric shielding current analysis in hts. IEEE Trans. on Magn. 46(8), 3097–3100 (2010)
8. Kamitani, A., Takayama, T., Itoh, T., Nakamura, H.: Extension of meshless Galerkin/Petrov-Galerkin approach without using Lagrange multipliers. Plasma and Fusion Research 6(2401074) (2011)
9. Liu, G.R.: Meshfree Methods: Moving beyond the Finite Element Method, 2nd edn. CRC Press LLC, Boca Raton (2009)

10. Manzin, A., Ansalone, D.P., Bottauscio, O.: Numerical modeling of biomolecular electrostatic properties by the element-free Galerkin method. IEEE Trans. on Magn. 47(5), 1382–1385 (2011)
11. Nakata, S., Takeda, Y., Fujita, N., Ikuno, S.: Parallel algorithm for meshfree radial point interpolation method on graphics hardware. IEEE Trans. on Magn. 47(5), 1206–1209 (2011)
12. Ni, G., Ho, S.L., Yang, S., Ni, P.: Meshless local Petrov-Galerkin method and its application to electromagnetic field computations. International Journal of Applied Electromagnetics and Mechanics 19, 111–117 (2004)
13. Wang, J.G., Liu, G.R.: A point interpolation meshless method based on radial basis functions. Int. J. Numer. Meth. Engng. 54(11), 1623–1648 (2002)
14. Xianyi, Z., Qian, W., Chothia, Z.: Openblas homepage, `http://xianyi.github.com/OpenBLAS/`

Performance Measurement of Parallel Vlasov Code for Space Plasma on Scalar-Type Supercomputer Systems with Large Number of Cores

Takayuki Umeda[1] and Keiichiro Fukazawa[2]

[1] Solar-Terrestrial Environment Laboratory, Nagoya University, Nagoya 464-8601, Japan
umeda@stelab.nagoya-u.ac.jp
[2] Research Institute for Information Technology, Kyushu University, Fukuoka 812-8581, Japan
fukazawa@cc.kyushu-u.ac.jp

Abstract. Space plasma is a collisionless, multi-scale, and highly nonlinear medium. Computer simulations with the first-principle kinetic equation are essential for studying multi-scale processes in space plasma. We develop a parallel Vlasov (collisionless Boltzmann) simulation code for practical use on currently-existing supercomputer systems. The weak-scaling benchmark test shows that our parallel Vlasov code achieves a high performance and a high scalability. The parallel Vlasov code is applied to various physical problems in space plasma, such as cross-scale coupling between fluid scales and particle kinetic scales.

1 Introduction

No less than 99.9% of the matter in the visible Universe is in the plasma state. The plasma is a gas in which a certain portion of the particles are ionized, which is considered to be the "fourth state" of the matter. Our solar system is also filled with plasma particles ejected from the Sun. The stream of plasma is called the solar wind, which also carries the intrinsic magnetic field of the Sun. Neutral gases in the upper atmosphere of the Earth are also ionized by a photoelectric effect due to absorption of energy from sunlight. Since the number density of plasma particles in space is low and the mean-free path (average distance between collisions of plasma particles) is large, the word "space plasma" is generally equivalent to collisionless plasma. The plasma behaves as a dielectric medium with strong nonlinear interactions between plasma particles and electromagnetic fields. That is, the motion of plasma results in an electric current, electric currents modify the surrounding electromagnetic fields, and electromagnetic fields modify the motion of plasma, which results in a new electric current. Thus the computer simulation is an essential approach in studies of space plasma.

There are numerous types of self-consistent computer simulations that treat space plasma according to various approximations. The "global"-scale processes, such as structures and dynamics of solar/stellar winds, solar/stellar flares, and magnetosphere of stars, planets and satellites, are commonly described by magneto-hydro-dynamic (MHD), Hall-MHD and multi-fluid models. On the other hand, electron-scale processes, such as acceleration and heating of electrons, are described by the full kinetic model, i.e., the Maxwell equations and either the Newton-Lorentz equation or the

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 561–569, 2013.

Vlasov (collisionless Boltzmann) equation for both electron and ion particles. Hybrid methods treat ions as particles and electrons as a fluid for ion-scale processes.

Conventionally, MHD simulations have been widely used for numerical modeling of global-scale problems such as magnetosphere of stars and planets. The MHD or fluid equations are derived by taking the zeroth, first, and second moments of the kinetic Vlasov equations with the zeroth, first, and second moments being the conservation laws of the density, momentum, and energy, respectively. Thus, the MHD simulations need several diffusion coefficients, which are essentially due to first-principle kinetic processes that are eliminated in the framework of the fluid approximations. Recent high-resolution in-situ observations have also suggested that fluid scale and particle kinetic scale in space plasma are strongly coupled with each other, which is called cross-scale coupling. To understand the cross-scale coupling in space plasma, it is important to include full kinetics in global-scale simulations, which is a final goal of space plasma physics. However, such a large-scale kinetic simulation requires enormous computing resources.

The full-electromagnetic particle-in-cell (PIC) code, in which plasma particles are treated as individual charged particles by solving the Newton-Lorentz equations of motion together with the Maxwell equations. The standard explicit PIC method is a well-developed numerical technique which has numerical procedures and concepts that are quite simple without approximation in the basic laws of collisionless plasmas, and is very powerful and widely used for studying full kinetics in space plasma. However, this method has a difficulty in load-balancing for massively parallel computers. This is because the PIC method uses both Eulerian variables (field variables that depend on both time and space) and Lagrangian variables (particle positions and velocities that depend only on time), and the number of particles on each computing process often becomes nonuniform. A limitation on the number of particles per cell also gives rise to strong numerical thermal fluctuations in the PIC method.

The Vlasov method is considered to be an alternative to the PIC method. The Vlasov method follows spatial and temporal developments of distribution functions defined in the position-velocity phase space by solving the Vlasov (collisonless Boltzmann) equation together with the Maxwell equations. The Vlasov method allows us to achieve a high scalability on massively parallel computers much easier than the PIC method, because the Vlasov method uses only Eulerian variables. The Vlasov method is also free from any numerical noise. However, the Vlasov method needs a huge computer resource to treat "hyper"-dimensional ($>$3D) distribution functions in position and velocity phase spaces. It is still difficult to treat the 3P3V (three dimensions for position and three dimensions for velocity) phase space even with the recent supercomputers, since a computation with 40^6 grid points requires 160GB memory and only a system with large shared memory allows us to perform 3P3V (6D) Vlasov simulations. Although 2P2V and 2P3V phase spaces are treated on the current supercomputer systems, the massively-parallel computation is essential for large-scale Vlasov simulation.

We develop numerical schemes for Vlasov simulations for practical use on currently-existing supercomputer systems. This paper gives a performance measurement study of our parallel Vlasov code on scalar-type systems with large number of cores toward future Exa-scale computing.

2 Overview of Numerical Schemes

The Vlasov model solves the kinetics equations of space plasma, i.e., the Maxwell equations (1) and the Vlasov (collisionless Boltzmann) equation (2),

$$\left.\begin{aligned} \nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t} \\ \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} \\ \nabla \cdot \mathbf{B} &= 0 \end{aligned}\right\} \tag{1}$$

$$\frac{\partial f_s}{\partial t} + \mathbf{v}\frac{\partial f_s}{\partial \mathbf{r}} + \frac{q_s}{m_s}\left[\mathbf{E} + \mathbf{v} \times \mathbf{B}\right]\frac{\partial f_s}{\partial \mathbf{v}} = 0 \tag{2}$$

where $\mathbf{E}$, $\mathbf{B}$, $\mathbf{J}$, ρ, μ_0, ϵ_0 and c represent electric field, magnetic field, current density, charge density, magnetic permeability, dielectric constant and light speed, respectively. The Vlasov equation (2) describes the development of the distribution functions by the electromagnetic (Lorentz) force, with the collision term in the right hand side set to be zero. The distribution function $f_s(\mathbf{r}, \mathbf{v}, t)$ is defined in position-velocity phase space with the subscript s being the species of singly-charged particles (e.g., $s = i, e$ for ions and electrons, respectively). The Maxwell equations and the Vlasov equation are coupled with each other via the current density $\mathbf{J}$ that satisfies the continuity equation for charge

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0 \tag{3}$$

These equations are regarded as the "first principle" of the collisionless plasma.

It is not easy to solve hyper-dimensional Vlasov equation numerically, in terms of both computational resources and computational accuracy. The Vlasov equation (2) consists of two advection equations with a constant advection velocity and a rotation equation by a centripetal force without diffusion terms. To simplify the numerical time-integration of the Vlasov equation, we adopt a modified version of the operator splitting [1],

$$\frac{\partial f_s}{\partial t} + \mathbf{v}\frac{\partial f_s}{\partial \mathbf{r}} = 0 \tag{4}$$

$$\frac{\partial f_s}{\partial t} + \frac{q_s}{m_s}\mathbf{E}\frac{\partial f_s}{\partial \mathbf{v}} = 0 \tag{5}$$

$$\frac{\partial f_s}{\partial t} + \frac{q_s}{m_s}\left[\mathbf{v} \times \mathbf{B}\right]\frac{\partial f_s}{\partial \mathbf{v}} = 0 \tag{6}$$

Equations (4) and (5) are scalar (linear) advection equations in which $\mathbf{v}$ and $\mathbf{E}$ are independent of $\mathbf{r}$ and $\mathbf{v}$, respectively. We adopt a multidimensional conservative semi-Lagrangian scheme [1] for solving the multidimensional advection equations. In the full electromagnetic method, it is essential to use conservative schemes for satisfying the continuity equation for charge. With the multidimensional conservative semi-Lagrangian scheme, the continuity equation for charge (3) is exactly satisfied. In the

present study, we compute the numerical flux by using the multi-dimensional advection scheme [1] with a positive, non-oscillatory and conservative limiter [2,3] for stable time-integration of advection equations. Equation (6), on the other hand, is a multi-dimensional rotation equation which follows a circular motion of a profile at constant speed by a centripetal force. For stable rotation of the profile on the Cartesian grid system, the "back-substitution" technique [4] is applied. In addition, Maxwell's equations are solved by the implicit Finite Difference Time Domain (FDTD) method.

The velocity distribution function has both configuration-space and velocity-space dimensions, and defined as a hyper-dimensional (>3D) array. There are some additional communications overhead in parallelizing over the velocity-space dimensions since a reduction operation is required to compute the charge and current densities (the zeroth and first moments) at a given point in configuration space. Thus we adopt the "domain decomposition" only in configuration space, where the distribution functions and electromagnetic fields are decomposed over the configuration-space dimensions. This involves the exchange of ghost values for the distribution function and electromagnetic field data along boundaries of each procession element. The non-oscillatory and conservative scheme [2,3] uses six grids for numerical interpolation, and three ghost grids are exchanged by using the "`Mpi_Sendrecv()`" subroutine in the standard message passing interface (MPI) library for simplicity and portability [5]. We also use the "`Mpi_Allreduce()`" subroutine for the convergence check on each iteration of the implicit FDTD method. Note that the code allows thread parallelization over the velocity-space dimensions via OpenMP.

3 Performance Evaluation

We conduct the performance measurement test of our parallel Vlasov code with a phase-space grid of $(N_{v_x}, N_{v_y}, N_{v_z}, N_x, N_y) = (30, 30, 30, 40, 20)$ on one core, which corresponds to a weak-scaling test with 1GB/core. We use the following four systems for the benchmark test: the Fujitsu FX1 at Nagoya University and JAXA, the K computer at the RIKEN AICS, the Fujitsu FX10 at the University of Tokyo and Kyushu University, and the Fujitsu CX400 at Kyushu University. The summary specification of these system is given in Table 1. Note that the FX1, K-computer and FX10 have similar architecture, while the CX400 is a standard PC cluster.

Figure 1 shows the inter-node parallel performance on these systems. Note that the computational performance is measured by using the hardware counter installed on the Fujitsu FX1, FX10, and K-computer systems, and the performance on the Fujitsu CX400 is estimated based on the result on the FX1/FX10/K. Here, it should be noted that we used the so-called flat MPI parallelization for the CX400 and the hybrid parallelization with four threads (i.e., a combination of MPI process and OpenMP thread parallelizations so that *each process has* $30 \times 30 \times 30 \times 80 \times 40$ *grids*) for the FX1, FX10 and K-computer in the present performance measurement.

The peak performances (performance efficiency) of the parallel Vlasov code are 15.43TFlops (12.8%) on the FX1 with 12,032 cores, 160.41TFlops (15.3%) on the K-computer with 65,536 cores, 192.24TFlops (16.9%) on the FX10 with 76,800 cores, and 57.21TFlops (16.2%) on the CX400 with 16,384 cores.

Table 1. Summary specification of computer systems

System	FX1	K-computer	FX10	CX400
Site	Nagoya Univ./JAXA	RIKEN AICS	Kyushu Univ./ Univ. of Tokyo	Kyushu Univ.
CPU	SPARC64 VII	SPARC64 VIIIfx	SPARC64 IXfx	Xeon E5-2680
Clock	2.5GHz	2.0GHz	1.848GHz	2.7GHz
Num. of cores	4	8	16	8
Num. of sockets	1	1	1	2
Memory per node	32GB/16GB	16GB	32GB	128GB
Inter-node	DDR InfiniBand	Tofu 6D mesh/torus	Tofu 6D mesh/torus	FDR Infiniband
Number of nodes	768/3008	82,944	768/4800	1476

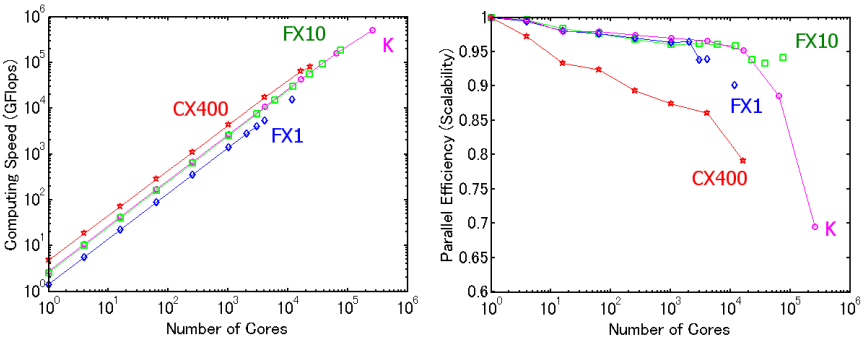


Fig. 1. Performance of different supercomputer systems with scalar processors. (left) Computational speed and (right) scalability as a function of the number of cores. The diamond, square, circle, and star marks represent the results on the FX1, FX10, K, and CX400, respectively.

We obtained a high scalability of ∼90% on the FX1, FX10 and K-computer even with more than 10,000 cores. On the FX1, the scalability becomes worse when the all computational resources (384 nodes: 3072 cores at Nagoya Univ. and 3008 nodes: 12,032 cores at JAXA, respectively) are used. By contrast, the FX10 gives almost constant performance even with the all computational resources (768 nodes: 12,288 cores at Kyushu Univ. and 4800 nodes: 76,800 cores at Univ. of Tokyo) are used. The performance becomes worse with 8192 nodes (65,536 cores) on the K-computer, due to a performance bottleneck of the "`Mpi_Allreduce()`" subroutine used for solving the Maxwell equations. The system gives better performance (performance efficiency >16.5%) with less than 2048 nodes (16,384 cores). It should be noted that jobs with more than 32,768 nodes (262,144 cores) exceeded wall time of the present benchmark test (10 min.) during the initial simulation setup subroutine. This is essentially due to the performance bottleneck of the "`Mpi_Allreduce()`" subroutine which is used for the convergence check on each iteration of the Poisson equation for the field initialization.

The network topology of the CX400 system at Kyushu University is a (non-fat) tree where five groups of 256 nodes and one group of 196 nodes are connected each other

and all the nodes in each group are connected each other with the fat-tree full bisection bandwidth. The system gives better performance (performance efficiency $>$20%) with less than 256 nodes(4096 cores), while the performance becomes worse with $>$256 nodes due to the network bandwidth capacity.

4 Global Vlasov Simulation

Since hyper-dimensional Vlasov simulations need huge amount of computer memory, there have been a small number of hyper-dimensional Vlasov simulations applied for practical plasma physics such as collisionless space plasma [6] and laser plasma [7]. In fact, a large number of papers on numerical schemes for Vlasov simulations have been published but most of them have been benchmarked only with classical (non-practical) problems. The present parallel Vlasov code is a one of the few examples of hyper-dimensional Vlasov simulations which has successfully been applied to studies of the Kelvin-Helmholtz instability [8] and the magnetic reconnection [9], which are "local" processes taking place at boundary layers of magnetosphere. Recently "global" Vlasov simulations are also performed on the interaction between the solar wind and an unmagnetized small astronomical body with a spatial scale of ion gyro radius [10,11]. It should be noted that hyper-dimensional Vlasov simulations are widely performed in magnetically confined plasma, such as tokamak plasma in thermonuclear fusion devices, with the guiding center, drift-kinetic and gyro-kinetic approximations [12]. Gyro motion of charged particles are approximately eliminated in these models. On the other hand, full Vlasov simulations with the gyro motion (Eq.6), such as the present study, are much more difficult to perform.

In the present study, we extend the global Vlasov simulation of an unmagnetized body to a weakly-magnetized body. Figure 2 shows the spatial profile of ion density together with the configuration of magnetic field lines obtained by a 5D (two spatial and three velocity dimensions) global Vlasov simulation in which spatial grids are taken in the two dimensional (2D) $x-y$ plane and velocity grids are taken in the three dimensional (3D) $v_x - v_y - v_z$ space. There exists a dielectric circle (cylinder) at $\mathbf{r} \equiv (x, y) = (0, 0)$, on the surface of which the electric charge accumulates by absorption of plasma particles. The radius of the body is set to be $R_L = 2r_i$, where r_i is the gyro radius of solar-wind ions.

There exists a uniform plasma flow as the solar wind in the simulation box at the initial state. The plasma flow has a shifted Maxwellian velocity distribution and is directed in the x direction with a flow velocity $V_S = 5V_{ti}$. The uniform plasma flow is continuously injected from the left boundary to the right. In order to save computational costs, electrons are assumed to be much heavier than the reality with the mass ratio $m_i/m_e = 25$. Assuming that the ions and electrons have the same temperature ($T_i/T_e = 1.0$), the thermal velocity of electrons becomes $V_{te} = 5V_{ti}$. For the velocity space, we use $N_{v_x} \times N_{v_y} \times N_{v_z} = 60 \times 60 \times 60$ gird points. The range of the velocity space is taken to be $-15V_{ti} \le v_i \le 15V_{ti}$ and $-10V_{te} \le v_e \le 10V_{te}$ for all v_x, v_y and v_z directions, where V_t denotes the thermal velocity.

The solar wind carries an Inter-planetary Magnetic Field (IMF), which is the intrinsic magnetic field of the Sun. In the present study, a uniform ambient magnetic field as the

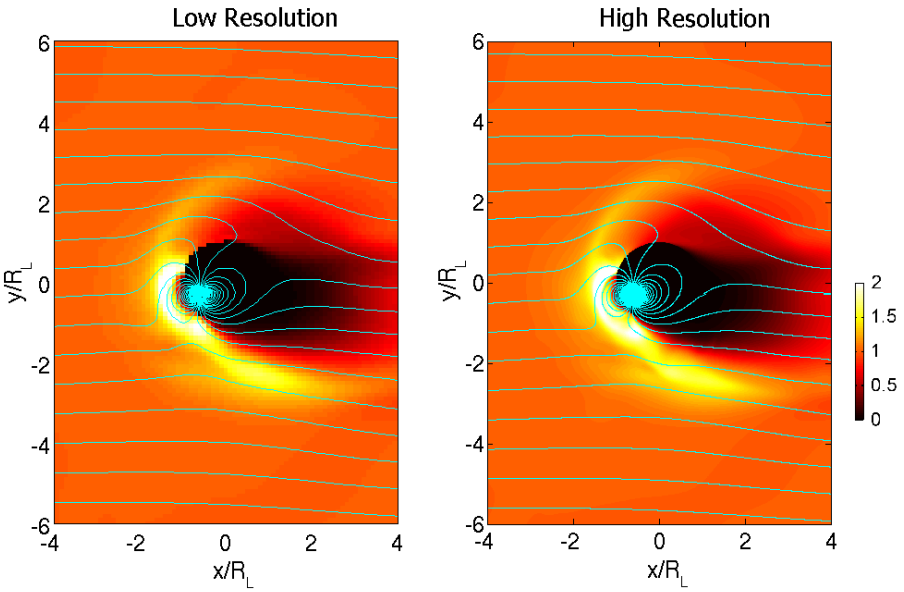


Fig. 2. Spatial profile of ion density together with the configuration of magnetic field lines at $\omega_{ci}t = 1.2$ obtained by a global Vlasov simulation. Structures of bow shock and wake tail formed as a result of the interaction between a weakly-magnetized object and the solar wind are shown. The result in the left and right panels are obtained by using the R815 and the K-computer, respectively.

IMF is taken in the $x - z$ plane in order to include both ion gyro motion and magnetic convection in the $x - y$ plane. The magnitude of the ambient magnetic field is given by $B_0 \equiv \sqrt{B_{x0}^2 + B_{z0}^2} = m_i/q_i\omega_{ci}$ with $\omega_{ci}/\omega_{pi} = 0.02$, where ω_c and ω_p are the cyclotron and plasma angular frequency, respectively. Thus the ion inertial length and electron Debye length become $l_i = 2r_i$ and $\lambda_D = 0.02r_i$, respectively. The IMF is imposed in the 45° with respect to the x axis with $B_{x0} = B_{z0} = B_0/\sqrt{2}$. To carry the ambient magnetic field in the direction of plasma flow (x direction), we need a uniform electromotive (Faraday's inductive) force. A uniform external electric field is applied in the y direction with the magnitude of $E_{y0} = V_S B_{z0}$. The ion beta and Alfven Mach number are $\beta_i = 0.5$ and $M_A = 2.5$, respectively.

There exists an inner boundary at $|\mathbf{r}| = \sqrt{x^2 + y^2} = R_L$. We assume that the electric conductivity of the body is small, like an insulator. Numerically, the spatial numerical flux is set to be zero for $|r| < R_L$. Then, spatial advection in the body is not allowed and the electric charge of plasma particles onto the body accumulates on its surface. Note that we do not use any special treatment for electromagnetic waves at the inner boundary. That is, the zero current condition is applied in the body, and electromagnetic waves propagate freely into the body without conductance.

In the low-resolution run in the left panel, a total of $N_x \times N_y = 160 \times 240$ grid points are used for the configuration space grids. The low-resolution run uses a total of 400GB memory and is performed on the DELL PowerEdge R815 systems at Solar-Terrestrial Environment Laboratory, Nagoya University [13]. The grid size of the low-resolution

run is taken to be $\Delta \equiv \Delta x = \Delta y = 0.1R_L = 0.05r_i = 10\lambda_D$. This means than micro-scale processes such as plasma oscillation and formation of Debye sheath are eliminated. On the other hand, in the high-resolution run in the right panel, a total of $N_x \times N_y = 1920 \times 2560$ grid points are used for the configuration space grids. The high-resolution run uses a total of 60TB memory and is performed on the K-computer. The grid size of the low-resolution run is taken to be $\Delta \equiv \Delta x = \Delta y = 0.01R_L = 0.005r_i = \lambda_D$, and all the physical processes of collisionless space plasma are included.

The preliminary result shows that the structure of the ion density in the vicinity of the body is almost the same in both low- and high-resolution runs, at a very early stage of the simulation run ($\omega_{ci}t = 1.2$). That is, a high-density region is formed on the dayside (left of the body) around the boundary between open and closed magnetic field lines. On the other hand, a low-density region called the wake is going to be formed on the nightside (right of the body). Further computation of the high-resolution run is currently in progress.

5 Conclusion

In this paper, we have made performance measurements of a new Vlasov-Maxwell code on massively-parallel scalar computer systems, which are the the Fujitsu FX1, the Fujitsu FX10, the Fujitsu CX400 and the K-computer systems. The weak-scaling benchmark test shows that our parallel Vlasov code achieves a high scalability ($>90\%$) on more than 10,000 cores with recent inter-node connection devices. Currently we are performing a 5D (2P3V) high-resolution "global" Vlasov simulation of a small magnetosphere of a small astronomical body by using K-computer. It should be noted that a 6D (3P3V) simulation is essential for full understanding on the global structure and dynamics of magnetosphere. However, this requires numerous computing resources with a total of more than 50PB memory as well as a shared memory of more than 200GB, and is left as a (far) future study.

Acknowledgement. This work was supported by MEXT/JSPS under Grant-in-Aid for Young Scientists (B) No.23740367 and Challenging Exploratory Research No.25610144. The K computer at the RIKEN Advanced Institute for Computational Science is used as a HPCI Systems Research Project (#hp120092). The R815 system at Solar-Terrestrial Environment Laboratory (STEL) in Nagoya University is used a computational joint research program. The FX1 system at Information Technology Center in Nagoya University is used as a STEL computational joint research program, a JHPCN Program and Nagoya University HPC Program. The computational resource at JAXA is provided by Iku Shinohara. The FX10 system at Information Technology Center in the University of Tokyo is used as a Large-Scale HPC Challenge Program. The CX400 and FX10 system at Research Institute for Information Technology in Kyushu University is used as an Advanced Computational Scientific Program.

References

1. Umeda, T., Togano, K., Ogino, T.: Two-dimensional full-electromagnetic Vlasov code with conservative scheme and its application to magnetic reconnection. Comput. Phys. Commun. 180, 365–374 (2009)
2. Umeda, T.: A conservative and non-oscillatory scheme for Vlasov code simulations. Earth Planets Space 60, 773–779 (2008)
3. Umeda, T., Nariyuki, Y., Kariya, D.: A non-oscillatory and conservative semi-Lagrangian scheme with fourth-degree polynomial interpolation for solving the Vlasov equation. Comput. Phys. Commun. 183, 1094–1100 (2012)
4. Schmitz, H., Grauer, R.: Comparison of time splitting and backsubstitution methods for integrating Vlasov's equation with magnetic fields. Comput. Phys. Commun. 175, 86–92 (2006)
5. Umeda, T., Fukazawa, K., Nariyuki, Y., Ogino, T.: A scalable full electromagnetic Vlasov solver for cross-scale coupling in space plasma. IEEE Trans. Plasma Sci. 40, 1421–1428 (2012)
6. Schmitz, H., Grauer, R.: Kinetic Vlasov simulations of collisionless magnetic reconnection. Phys. Plasmas 13, 092309, 10 (2006)
7. Ghizzo, A., Huot, F., Bertrand, P.: A non-periodic 2D semi-Lagrangian Vlasov code for aser-plasma interaction on parallel computer. J. Comput. Phys. 186, 47–69 (2003)
8. Umeda, T., Miwa, J., Matsumoto, Y., Nakamura, T.K.M., Togano, K., Fukazawa, K., Shinohara, I.: Full electromagnetic Vlasov code simulation of the Kelvin-Helmholtz instability. Phys. Plasmas 17, 052311, 10 (2010)
9. Umeda, T., Togano, K., Ogino, T.: Structures of diffusion regions in collisionless magnetic reconnection. Phys. Plasmas 17, 052103, 6 (2010)
10. Umeda, T., Kimura, T., Togano, K., Fukazawa, K., Matsumoto, Y., Miyoshi, T., Terada, N., Nakamura, T.K.M., Ogino, T.: Vlasov simulation of the interaction between the solar wind and a dielectric body. Phys. Plasmas 18, 012908, 7 (2011)
11. Umeda, T.: Effect of ion cyclotron motion on the structure of wakes: A Vlasov simulation. Earth Planets Space 64, 231–236 (2012)
12. Idomura, Y., Ida, M., Kano, T., Aiba, N., Tokuda, S.: Conservative global gyrokinetic toroidal full-f five-dimensional Vlasov simulation. Comput. Phys. Commun. 179, 391–403 (2008)
13. Umeda, T., Ito, Y., Fukazawa, K.: Global Vlasov simulation on magnetospheres of astronomical objects. J. Phys.: Conf. Ser. 454, 012005 (2013)

Performance Evaluation of Magnetohydrodynamics Simulation for Magnetosphere on K Computer

Keiichiro Fukazawa[1,2,3], Takeshi Nanri[1,3], and Takayuki Umeda[4]

[1] Research Institute for Information Technology, Kyushu University, Japan
{fukazawa,nanri}@cc.kyushu-u.ac.jp
[2] International Center for Space Weather Science and Education, Kyushu University, Japan
[3] CREST JST, Japan
[4] Solar-Terrestrial Environment Laboratory, Nagoya University, Japan
umeda@stelab.nagoya-u.ac.jp

Abstract. To use the massively parallel scalar type supercomputer for the large scale numerical magnetospheric simulation, we made performance evaluation of a three-dimensional Magnetohydrodynamic (MHD) simulation code for space plasma simulations of planetary magnetosphere on the K computer with 262,144 cores at RIKEN. For parallelization of the MHD code, we use four different methods, i.e., regular one-dimensional, two-dimensional, three-dimensional domain decomposition methods and a cache-hit type of three-dimensional domain decomposition method. We found that the one-dimensional decomposition and cache-hit type of three-dimensional decomposition achieves the best performance and our MHD code can scale over 200,000 processes on the K computer.

Keywords: MHD simulation, parallel computing, performance evaluation.

1 Introduction

No less than 99.9 % of the matter in the visible Universe (except for the dark matter distribution which is visible through gravitational lensing) is in the plasma state, i.e., in the state of ionized gases. The stellar system is filled with plasmas ejected from the upper atmosphere of the stars. The stream of plasma carries the magnetic field of the stars, in particular, we call it the solar wind in the solar system. Various phenomena, such as geomagnetic storms, are caused by interactions between the solar wind and a planetary intrinsic magnetic field. Recently these affect to the GPS satellites and its measurements accuracy on the ground. Thus studies of space plasma are important to understand our Universe.

Since the density of space plasma is very low, collisions between individual plasma particles are neglected. Then, kinetic dynamics of such collisionless plasma is described by the first-principle Vlasov-Maxwell system equations [1]. The Vlasov-Maxwell system equations are commonly used for studies of electron-scale processes in space plasma. For larger-scale processes, however, electron-scale processes can be sometimes neglected, and various approximations are applied to the Vlasov-Maxwell

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 570–576, 2013.

system equations. For investigating global structures of plasma, such as planetary magnetospheres, the Magnetohydrodynamic (MHD) equations are used, in which full kinetics of plasma are neglected by taking the moments of the Vlasov equations. The Vlasov-Maxwell system equations and MHD equations are highly nonlinear and are very complex to solve by hand calculations. Thus computer simulations play essential roles in studies of space plasma and have been performed from the first vector computer in Japan.

The numerical MHD code for space plasma has been optimized for vector-type supercomputers for a long time because most of supercomputers with vector processors have high performance till 1990's. These codes often have achieved a very high computational efficiency (the ratio of the effective performance to the theoretical performance). However, almost 100 % of the "top 500" supercomputer systems in the world recently adopt the scalar type processors and more than 90 % of systems consist of the 64bit-x86 processor architecture. The other scalar type computers are POWER and SPARC architectures [2].

The purpose of the present study is to make performance tuning of MHD code for space plasma simulations on scalar-type massively parallel supercomputer systems [3] [4]. In general, the computing efficiency of user applications on a scalar-type computer tends to be low (~5 %) [5], although the computing efficiency of LINPACK sometimes exceeds 80%. In this paper, a performance measurement study of MHD codes is carried out on the K computer at RIKEN which is the fastest supercomputer system in Japan.

2 Magnetohydrodynamic Simulation Model

As mentioned above the MHD equation is derived from the Vlasov equation and Maxwell equation [1].

$$\begin{cases} \dfrac{\partial \rho}{\partial t} = -\nabla \cdot (\vec{v}\rho) + D\nabla^2 \rho \\ \dfrac{\partial \vec{v}}{\partial t} = -(\vec{v} \cdot \nabla)\vec{v} - \dfrac{1}{\rho}\nabla P + \dfrac{1}{\rho}\vec{J} \times \vec{B} + g + \dfrac{\boldsymbol{\Phi}}{\rho} \\ \dfrac{\partial P}{\partial t} = -(\vec{v} \cdot \nabla)P - \gamma P \nabla \vec{v} + D_p \nabla^2 P \\ \dfrac{\partial \vec{B}}{\partial t} = \nabla \times (\vec{v} \times \vec{B}) + \eta \nabla^2 \vec{B} \end{cases} \tag{1}$$

$$* \ \vec{J} = \nabla \times (\vec{B} - \vec{B}_d)$$

Here ρ is the plasma density, $\vec{v}$ is the plasma velocity, P is the pressure, $\vec{B}$ is the magnetic field, $\vec{j}$ is the current density. $D = D_p$ is the diffusion coefficient, g is acceleration of gravity, $\boldsymbol{\Phi} \equiv \mu \nabla^2 \vec{v}$ is the viscosity, $\gamma = 5/3$ is the three dimensional ratio of specific heat, η is the resistance and $\vec{B}_d$ is the dipole magnetic field of planet.

We use a three-dimensional MHD code [6], which has been used for studies on global structures and dynamics of planetary magnetospheres [7] [8]. The MHD code uses the "Modified Leapfrog" method, in which partial difference equations are solved by the two-step Lax-Wendroff method for one time step and then by the Leapfrog scheme for $(l - 1)$ time steps and the procedure is repeated. Thus the method has second-order accuracy in both space and time. The Modified Leapfrog method is a kind of combination technique which balances numerical stability of the two step Lax-Wendroff method and dissipationlessness of the Leapfrog method. That is, smaller l is desirable for numerical stability, while larger l is desirable for suppression of numerical dissipation. Based on the linear analysis and preliminary simulations, the value of $l = 8$ has been chosen.

3 Parallelization Model

As a parallelization technique, the domain decomposition method for dividing three-dimensional space is adopted. Usually in parallel computing on a distributed-memory computer, the domain decomposition is an easy way to decompose three-dimensional Eularian variables. In the case of a three dimensional model, the dimension of domain decomposition can be chosen as one dimension, two dimensions, or three dimensions as shown in Fig. 1. In general, three-dimensional domain decomposition is most efficient for a scalar parallel computer, while two-dimensional domain decomposition is most efficient for a vector parallel computer. This is because vectorization of the most inner loop can be vectorized efficiently. Effective use of cache memory is important for obtaining better performance on a scalar processor due to the small memory bandwidth compared to the vector processor. Since the size and structure of cache memory depend on CPU architecture and model, it is not easy to obtain the highest performance on scalar processor computers.

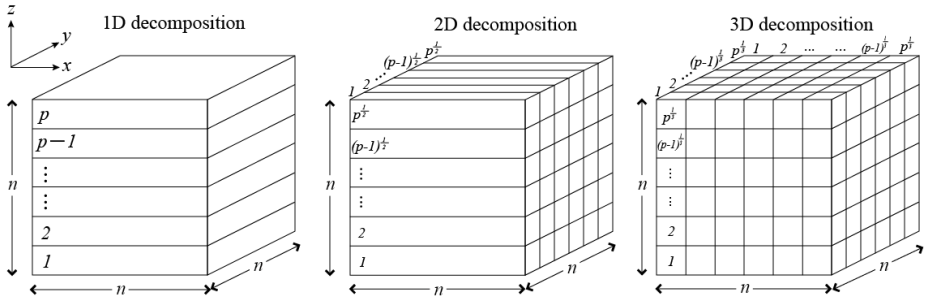


Fig. 1. Schematics of three kinds of the domain decomposition methods

In the MHD model, the number of treated physical variables is eight: plasma density N, velocity (V_x, V_y, V_z), pressure P, and magnetic field (B_x, B_y, B_z). In the present MHD code, an array of these physical variables are defined as $f(n_x, n_y, n_z, n_m)$ (here referred as "Type A") with $n_m = 1$ and $n_m = 5$ being N and P, $n_m = 2$ being V_x, $n_m = 3$

being V_y, $n_m = 4$ being V_z, etc. n_x, n_y and n_z are the array size in x, y, and z-direction, respectively. Since these physical variables on the same grid point are used repeatedly in computation, an array defined as $f(n_m, n_x, n_y, n_z)$ (here referred as "Type B") is considered to be more effective for the cache hit. In this study, the Fortran is used as a programming language not C or C++. Thus the memory allocation order of multi-dimensional array is different among them (in Fortran the array is stored from the lower dimension). If using C and C++, there is a possibility that the following our results are changed. That is, in those cases (C and C++) Type B may not be effective for cache hit. In the present study, we evaluate performance of both Type A and B on scalar processor.

To minimize the communication time, we use a buffer array which stores all the boundary data for inter-core and inter-node communications. In these communication processes, the `MPI_sendrecv` function is adopted because `MPI_sendrecv` is a bidirectional communication function which includes both sending data and receiving data in one process. The other communication functions, for example pairs of `MPI_send/MPI_isend` and/or `MPI_irecv/MPI_recv`, need two processes for each sending and receiving processes. It is also noted that we have suffered the data overflow using non-blocking communication (ie: `MPI_isend, MPI_irecv`) thus we do not use the non-blocking communication which is often said to shorten the communication time [3][4].

4 Results

The K computer consists of 88,128 nodes of clusters. Each node has one Fujitsu SPARC64 VIIIfx processor (8 cores, 2.0 GHz) and 16 GB memory with 64 GB/s memory bandwidth. As for the inter-node connections, each node connects through the 6 dimensional mesh/torus inter connect (5 GB/sec per 1 link), which is called "Tofu Interconnect". The system has 10.51 PFlops peak performance which is the top performance from 6/2011 to 11/2011 [2]. More detailed information of these systems is in the web site [9]. On K computer, the Fujitsu Fortran Compiler is installed. As compiler options, we adopt

```
-Kfast, visimpact, nomfunc, noalias=s, fsimple,
prefetch_indirect, prefetch_strong, noparallel
```

We use the array of 64 MB/core for the computational domain and additionally 192 MB/core for workspaces for computing the MHD equations with the Modified Leapfrog method. That is, the weak scaling and flat MPI are used in this study.

Fig. 2 shows the results of performance measurements of MHD code with the K computer. In this study we could use up to 262,144 cores. Using the K computer, we obtained 798.6 TFlops (19 % of efficiency) at maximum in the one- and there-dimensional decomposition Type B case. This means the long vector length and the

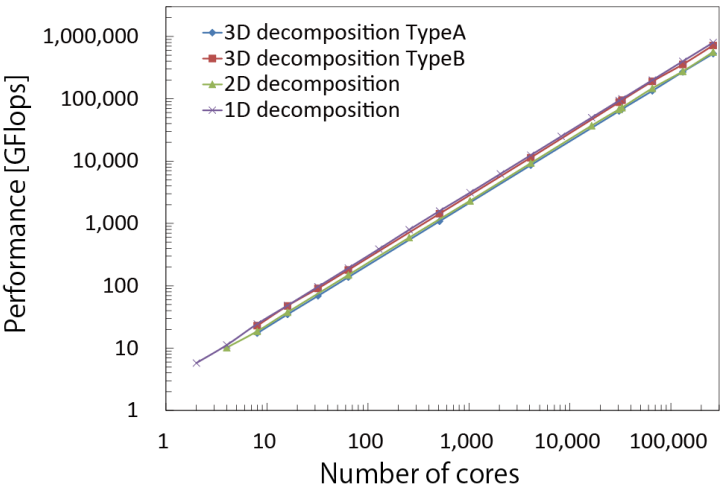


Fig. 2. Effective performance of MHD simulation code with K computer

cache hit type are suitable for the K computer. There is about 10 % difference of computational efficiency between the three-dimensional decomposition of Type A and Type B case. This result has the same trend of the POWER and previous SPARC64 architectures [3] [4] whish are RISC processors. SPARC64 VIIIfx processor has a kind of vectorization however the cache hit tuning is also effective. Additionally this is the adverse result of x86 type computer [3] [4].

The parallel scalability of MHD code with the K computer is shown in Fig. 3. One-dimensional decomposition is the best scalability. The degradation of parallel performance has not appeared over 65,536 cores in all decomposition cases. Over 65,536 cores only one-dimensional decomposition achieved the flat scalability. To consider the value of parallel scalability, in spite of using good inter node connection, the scalabilities of each decomposition are less than 90 %. To examine the reason, we found the problem in an array copy. In the parallel computing, as mentioned in section 2, we gather the boundary data into a buffer array in each process, and then transfer the array to the other processes. It takes more time to gather the boundary data on the K computer. This decreases the scalability at ~7 %. These data transfers in three-dimensional decomposition are required 3 times than that of one-dimensional decomposition. Thus the scalability of one-dimensional decomposition is the best. To run on the peta and exa Flops computers, this problem should be considered.

From the both results, it becomes clear that the optimization is very important to perform the large scale computing due to the about 200 TFlops difference between the best and worst results on 262,144 cores.

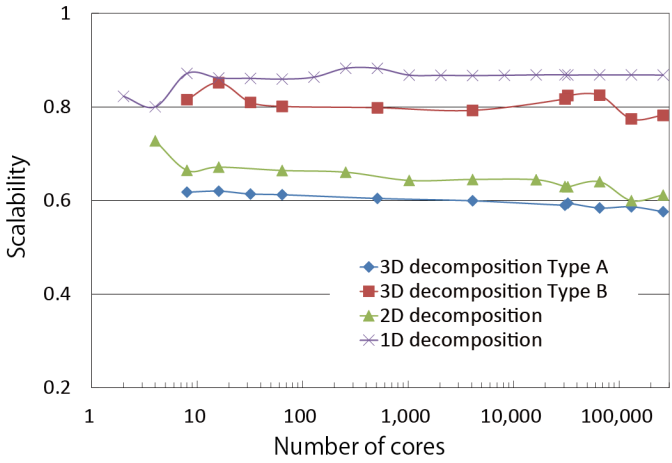


Fig. 3. Scalability of MHD simulation code with K computer

5 Summary

In this study we have made performance measurements of MHD code for the K computer. This supercomputer system consists of SPARC64 processors. For the MHD code, we evaluated four types of decomposition methods: regular one-dimensional, two-dimensional, three-dimensional domain decompositions and a cache hit considered type of three-dimensional decomposition. We found that the one- and three-dimensional decomposition Type B methods are the most suitable for K computer. The best scalability id obtained for one-dimensional decomposition. This comes from the quantity of data transfer generated by parallel computing. As the results, we obtained the best performance of 798.6 TFlops (19 %) with 262,144 cores on K computer. In this study we only use the flat MPI between the not only inter-node but also inner node communication to see the effect of decomposition. Note that using OpenMP and MPI (so called hybrid MPI), we achieved 964 TFlops (23 %) with three-dimensional decomposition on 262,144 cores.

Acknowledgements. This work was supported by CREST, the Japan Science and Technology Agency (JST) and Grant-in-Aid for Young Scientists (B) No. 25800275 (K.F.). The computational resource was provided by the K computer at the RIKEN Advanced Institute for Computational Science and by pursuing HPCI Systems Research Projects (Proposal Number hp120118).

References

1. Chen, F.F.: Introduction to Plasma Physics. Plenum Press, NY (1974)
2. Top500 Supercomputing Sites, `http://www.top500.org/`

3. Fukazawa, K., Umeda, T., Miyoshi, T., Terada, N., Matsumoto, Y., Ogino, T.: Performance measurement of magneto-hydro-dynamic code for space plasma on the various scalar type supercomputer systems. IEEE Trans. Plasma Sci. 38(9), 2254 (2010)
4. Fukazawa, K., Umeda, T.: Performance measurement of magnetohydrodynamic code for space plasma on the typical scalar type supercomputer systems with the large number of cores. International Journal of High Performance Computing Applications (2012), doi:10.1177/1094342011434813
5. Oliker, L., Canning, A., Carter, J., Shalf, J., Ethier, S.: Scientific computations on modern parallel vector systems. In: Proc. SC 2004: High Performance Computing, Networking, and Storage Conference (2004)
6. Ogino, T., Walker, R.J., Ashour-Abdalla, M.: A global magnetohydrodynamic simulation of the magnetopause when the interplanetary magnetic field is northward. IEEE Trans. Plasma Sci. 20, 817–828 (1992)
7. Fukazawa, K., Ogino, T., Walker, R.J.: A simulation study of dynamics in the distant Jovian magnetotail. J. Geophys. Res. 115, A09219 (2010a), doi:10.1029/2009JA015228
8. Fukazawa, K., Ogino, T., Walker, R.J.: A Magnetohydrodynamic Simulation Study of Kronian Field-Aligned Currents and Aurora. J. Geophys. Res. (2011) (revised)
9. Supercomputer K RIKEN Advanced Institute for Computational Science site, http://www.aics.riken.jp/en/

Approaches to Adaptability Pre-assessment of Facilities in New Line Connected Transfer Station

Wang Yuexin and Han Baoming

Traffic and Transportation, Beijing Jiaotong University, Beijing 100044

Abstract. Construction of urban rail transit requires large investment and long cycle, so unified planning and phased construction are generally adopted. Operation of a new line would bring the variation of passenger flow in the transfer station. Urban rail transfer station is a station with two lines or more, which is the important node in the network planning. Whether the current facilities in the station could meet new passenger demand, adaptability pre-assessment needs to be undertaken. This study firstly introduces the definition and content of adaptability assessment in urban rail transit station, and advances two assessment approaches: stationary index analysis and dynamic simulation experiment. Then through the case study, elaborate the specific process and method application. At last, put forward path optimization schemes, ensuring high operational efficiency and service level.

Keywords: urban rail transit, transfer station, adaptability pre-assessment, facility, pedestrian simulation.

1 Background

With the rapid development of urban rail transit, the network is gradually being formed in many cities. When a new line is put into operation having a linking with existing lines, crowded chaos might happen for the ability deficiency or irrational distribution of facilities in transfer stations. This phenomenon is due to the fact that urban rail transfer station has two main parts: station internal environment and passengers, in whom the interaction decides the operation organization efficiency and service level. The opening of a new line brings the change of passenger flow, not only the quantity but also the walking paths, which greatly influences the existing interaction. Thus, before the operation of the new line, it is a need to pre-assess whether the station status could adapt to the change, and based on the assessment results, take necessary measures.

Based on the characteristics of pedestrian behavior and passenger flow, Sui Xiaobo [1] analyzes the coordination of facilities and then builds the related assessment system. Li Minghua [2] summarizes characteristics of pedestrian behavior and uses Netlogo as the tool of modeling to analyze adaptability of pedestrian facilities. Wu Xianyu and Yuan Zhenzhou [3] establish the adaptability evaluation system of urban rail transit hub and come up with the simulation design thought of facilities' layout. Xu Qianqian [4] uses VISSIM to model the station and study the coordination of facilities in the transfer station.

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 577–586, 2013.

2 Adaptability Pre-assessment

'Adaptability' is a term in ecology that refers to the ability in organism to adapt to the ecological environment. As to urban rail transfer stations, adaptability refers to the ability of station internal facilities adapt to the passenger distribution, which reflects rationality of facilities layout and efficiency of passenger organization. Therefore, we can know whether the station status could adapt to the change of passenger flow through adaptability pre-assessment.

Adaptability pre-assessment mainly includes stationary index analysis and dynamic simulation experiment. When a new line connected to the transfer station, the significant change is passenger quantity. What's more, according to train arrival and departure, passenger flow is distributed as pulse type, which would impact on the facilities regularly. Given to this circumstances, it is a need to calculate the capacity of internal facilities with static index analysis. However, passenger quantity is not the only change, passenger walking path is also different. New paths would appear and old paths would vary, of which the influence to the operation organization can't be calculated theoretically. Thus, pedestrian simulation must be undertaken to find out what problems might happen in the station.

2.1 Static Assessment

Static adaptability assessment mainly focuses on assessing facilities capacity. Facilities capacity is reflected by two indicators: facilities saturation [6] and facilities capability matching rationality.

Facilities saturation is the ratio of hourly passenger volume at peak hours to the design passing capacity. If the ratio is larger, the facilities saturation is higher, which means the satisfaction of facilities to passenger demand at peak hours is lower. For the pulse type of passenger flow and by computing facilities saturation, the bottleneck restricting the whole facilities passing capacity in the station can be found.

Facilities capability matching rationality refers to whether the passing capacity of two adjacent facilities in the same passenger streamline is coordinated. Facilities capability matching rationality is reflected by the difference value of adjacent facilities saturation. If the value is smaller, the facilities capability matches more rational.

Combining with two indicators to make a comprehensive analysis, static adaptability assessment results are obtained.

2.2 Dynamic Assessment

Dynamic adaptability assessment is to use the computer simulation method, simulating the station internal passenger flow space-time distribution and passenger travel behavior**.** According to the system putout, analyze and assess the facilities adaptability. Simulation process mainly includes setting up simulation, implementing simulation scheme and analyzing experimental results.

By using basic data like station logical devices, train schedules, passenger flow lines, passenger flow generation regular and ticket information, get simulation control parameters and create the space model. Before implementing simulation experiment, it is a necessity to prove the validity of the model with historical data. After the test, statistically analyze the highest number of gathered people and regional services level and give the dynamic adaptability assessment results.

This passage mainly introduces the usage of SRail rail transit hub passenger distributing simulation system [7], independently developed by Beijing Jiaotong University, to implement dynamic adaptability assessment process.

2.3 Adaptability Improvement

Through adaptability assessment, we can know unreasonable facilities layout in the transfer station and take corresponding improvement measures. Adaptability improvement is mainly done from three aspects: passenger flow organization, engineering reform, equipment update.

Passenger flow organization is to optimize passenger flow lines and improve the utilization efficiency of facilities by guiding passenger behaviors. Engineering reform is to adjust the station civil engineering and improve station layout. Equipment update is to improve the ability of facilities and facilities status through updating or replacing the old facilities, changing the quantity of facilities. Operation efficiency and service level can be effectively improved by reasonably using these measures.

3 Case Study

Set an urban rail transit transfer station of line 1 and line 2 in a city as an example. This station is located in a busy intersection surrounded by many points that attracting a great quantity of passenger flow. Fig. 1 shows the main body of the station. The internal passenger flow lines include in-station flow lines, out-station flow lines and transfer flow lines of line 1 and line 2. Now line 1 is under operation and line 2 is to be open, so it is necessary to make the adaptability pre-assessment of this station.

3.1 Station Static Assessment

The predicted hourly passenger volume at peak hour in the station is 30867. The transfer passenger volume is 12167 with 49.3% for line 1 to line 2 and 50.7% for line 2 to line 1. The in-and-out station passenger volume is 18700. The passing passenger volume of facilities can be computed shown as Table 1.

Refer to the metro design code [8], we can get station internal facilities design passing capacity and calculate facilities saturation shown as Table 1.

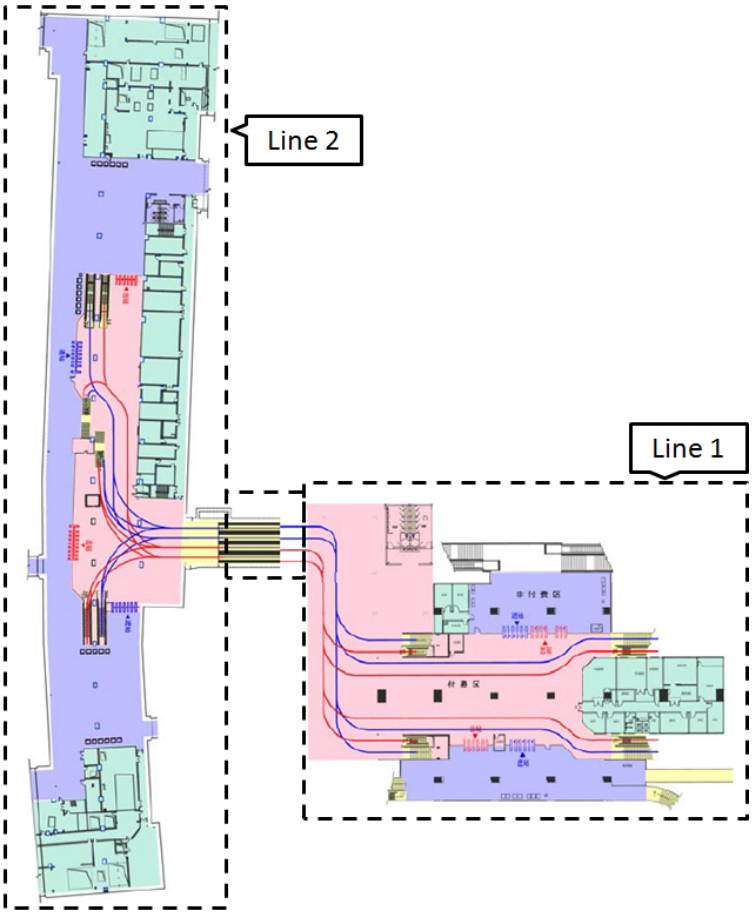


Fig. 1. Station structure and passenger flow lines

Table 1. Facilities design passing capacity and saturation

name	Width (m)	feature	passing capacity	Passenger volume	Saturation (%)
Stair between hall and platform	1.9	Bidirectional mixed flow	6080	1963	32.3
escalator between hall and platform	0.9	0.47m/s*	7614	1895	24.9
Transfer channel (1)	15.0	Bidirectional mixed flow	60000	12167	20.3
Transfer channel (2)	10.0	Bidirectional mixed flow	40000	12167	30.4
Transfer escalator	0.9	0.47m/s	7614	3042	40.0
Automatic ticket vending machine	/	/	300	4675	15.6
Artificial ticket vending machine	/	/	1200		
Fare gate	/	Three-bar form, IC card	1500	18700	62.3

(*transmission speed)

Table 1 shows the saturation of fare gate is too high and the difference with adjacent facilities is too big; in the transfer flow line, the facilities capability matching rationality between transfer channel (2) and transfer escalator (a difference of 19.7) is low.

3.2 Station Dynamic Assessment

In the process of dynamic simulation assessment of the station, use SRail rail transit hub passenger distributing simulation system as the tool.

Simulation Model Building.

(1) Station logical device

Station logical device is the file that describes characteristics, condition, parameters of functional area and facilities in the station. In this case, handle the transfer channel to a passage way, with transfer passenger flow of line 1 to line 2 regarded as in-station flow, and transfer passenger flow of line 2 to line 1 regarded as out-station flow.

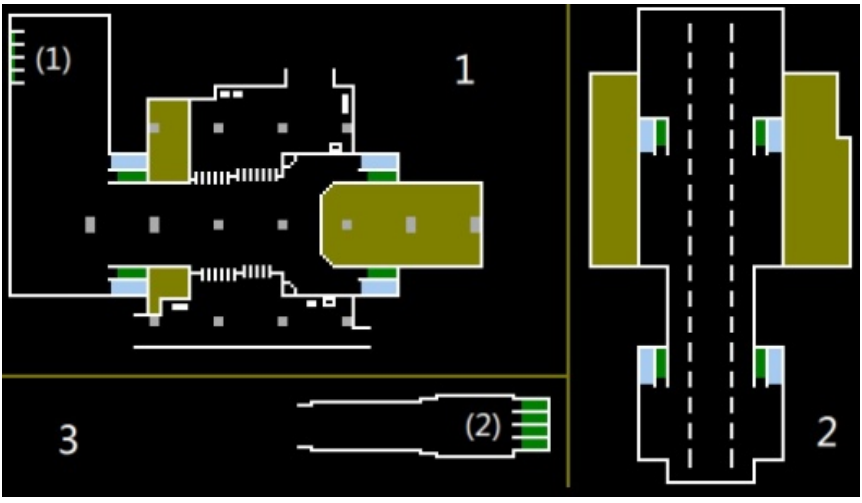


Fig. 2. Station facilities plan

Fig.2 is the station plan of line 1 and transfer channel. Area 1 is the hall of line 1 and transfer channel (1). There are 9 in-station fare gates, 11 out-station fare gates and both automatic ticket vending machines and artificial ticket vending machines. Area 2 is the platform of line 1. Area 3 is the transfer channel (2). Transfer channel (1) and transfer channel (2) is linked by four escalators in the right-side. The station has three passage ways without security inspection and equipped with two sets of in-and-out station automatic fare gate. The location of ticket vending machines is marked as rectangle in the hall and stairs along with escalators are showed as shadowed areas with an open side.

(2) Passenger generation regular

Passenger generation regular file provides the generation mode and method, temporal and spatial distribution, structure of passengers. These data (showed in 3.1) could be gotten through passenger flow forecast and spot investigation.

(3) Train timetable

Train timetable file is the main basis of event-driven passenger generation and influence passengers accumulating and dispersing in the station. It also sets the basic parameters of the train. In this case, the trains in line 1 adopt 4 sections marshalling and every section has 4 doors, with capacity of 210.Departure interval is 5min and station dwell time is 1min in peak hours.

(4) Passenger streamlines

Passenger streamline file fixes the sequence of the work flow and work place in the station. The system determines the streamlines by selecting control points. Control point is the key location in the processing of a passenger's walking path. In this case, passenger streamline is reflected in Fig. 3.

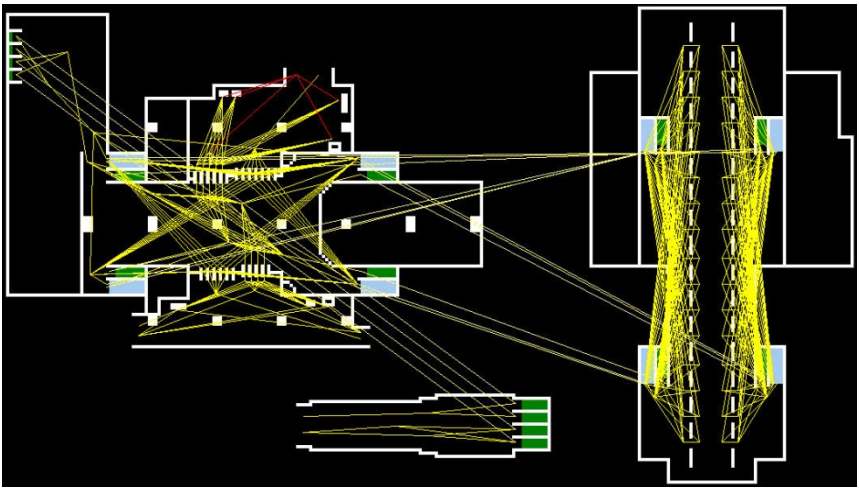


Fig. 3. Passenger streamline

Simulation Model Validation. For verifying the correctness of the model, use current passenger flow's data to implement a simulation experiment. Select the number of area gathered people as validation index and compare the outcome of the simulation experiment with spot investigation data. If the average fitting degree is under 5%, the simulation model can be used in the following experiment.

In this case, calculate two areas' gathered number-hall and platform, and then analyze the fitting degree. After twice model modification, the average fitting degree could meet the accuracy requirement.

Simulation Experiment. Based on the basic data and model in peak hour, implement the simulation experiment. Fig. 4 is the area density map in the simulation process.

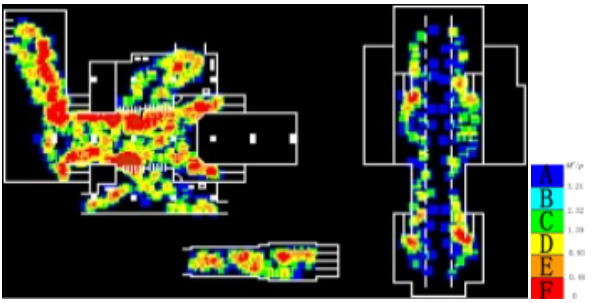


Fig. 4. Area density map

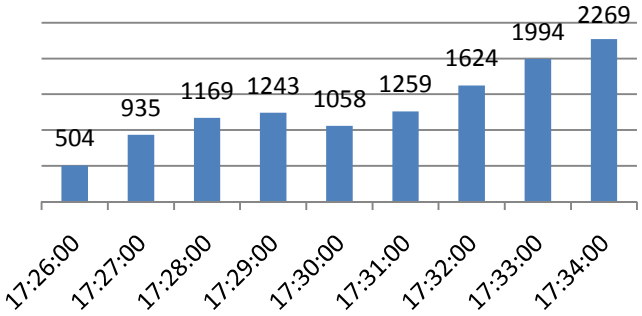


Fig. 5. The highest number of gathered people

By using the simulation results, calculate the highest number of gathered people and regional services level. Fig. 5 shows within 10min of the simulation experiment, the highest number of gathered people has been over 2000 and presents a trend of increasing; regional services levels in Table 2 are below grade C except for platform's, which indicates a low station service level.

Table 2. Typical regional service levels

Typical regional	service levels
Hall (paid area)	E
Platform	B
Transfer channel	C
Stair and escalator	F

Experimental results show that:

(1) Distribution effect in the station is poor. The passenger density of stairs, escalators and fare gates is high, which results in congestion. Passengers can't get on the train, transfer or leave the station quickly but strand in the hall.
(2) Passengers can't walk freely in the station. The facilities set has problems, leading to passenger flow interweaved heavily.

3.3 Optimization Scheme

Through the static and dynamic adaptability assessment, we can find that the current station internal facilities layout couldn't meet the passenger demand after the new line open. Thus, it is a need to adjust passenger travel lines and optimize the facilities in view of the problems found during the assessment. The concrete measures are as follows:

(1) Increase the number of facilities.
Increase the number of fare gate or replace the three-bar form with door type; add a set of stair beside the transfer escalators. This can enhance the passing capacity and facilities saturation will be lower.
(2) Adjust the ticket vending machines' location.
Centralize the vending machines to the middle area in the south of the hall. This area is far from out-station flow path, so the conflict of the paths can be reduced in non-pay zone. Moreover, the vending machines are near passage ways, which is convenient for passengers buying tickets.
(3) Separate in-station passenger flow and transfer passenger flow.
Exchange the relative position of two sets of fare gate in the north of the hall. After the relocation, in-station fare gates are near stairs and escalators in the south. This will prompt people to use these stairs and escalators to reach platform, which can separate the in-station flow and transfer flow.
(4) Change the escalator running direction.
Change the running direction of escalators in the west from up to down. This measure brings three effects: firstly enhance evacuation efficiency of the platform; secondly prevent transfer passengers reach the platform in a short time as an impact; thirdly attract passengers in the middle of the platform to choose western stairs and escalators getting the hall, which can reduce the disturbance of out-station flow to in-station flow.

Modify the original model with station facilities parameters and layout, as well as passenger walking path, and then implement the simulation experiment again. Fig. 6 is the area density map in the simulation process. Compared with Fig. 4, passenger accumulation is significantly reduced and also the density.

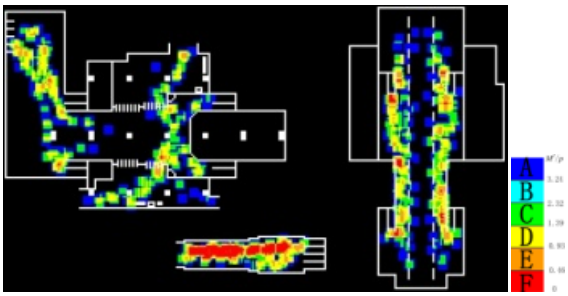


Fig. 6. Area density map

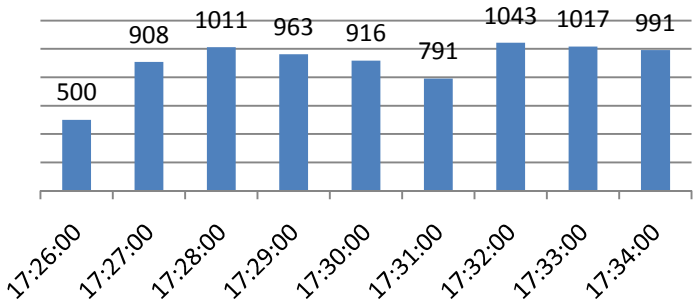


Fig. 7. The highest number of gathered people

Fig. 7 shows after the optimization, the highest number of gathered people has kept around 1000 with a regular float. It indicates that distribution effect is much better than before and passenger can get on the train, transfer or leave the station more smoothly. Besides, regional services levels all reach grade A.

Through the comparative analysis of assessment outcome, it proves that the measures improve station facility adaptability well and the optimization scheme achieves a good result.

4 Conclusions

This passage defines the urban rail transit transfer station adaptability assessment. Using static index calculation and dynamic simulation experiment and combining case study, explain the practical application way of adaptability assessment. This verifies the feasibility and effectiveness of the theory and provides means to optimize the facilities layout and organization in the station.

Acknowledgement. This research was supported by the Key Issues Study of Beijing Urban Rail Transit Identification, Prediction, Safety Warning of Operational Status, (Project No.: T13E300010), the People's Republic of China.

References

1. Sui, X.: Study on Infrastructure Coordination of Urban Rail Transfer Stations. Beijing Jiaotong University, Beijing (2008)
2. Li, M.: Analysis on Adaptability of Pedestrian Walking Facilities in Urban Railway Transit Hub. Beijing Jiaotong University, Beijing (2008)
3. Wu, X., Yuan, Z.: Framework Design of Simulation Evaluation System about Urban Rail Transit Hubs Facilities' Adaptability. J. Urban Mass. Transit. 14(2), 21–28 (2011)
4. Xu, Q.: Study on Evaluation and Optimization of Coordination of Transfer Station Facilities for Urban Rail Transit. Beijing Jiaotong University, Beijing (2010)
5. Qi, Z., Baoming, H., Fang, L.: Interactions Between Passengers and Station Environment of Urban Railway Transit. J. Urban Mass. Transit. 14(10), 44–48 (2011)
6. Wang, B., An, S., Li, X.: Evaluation System for the Transfer In-frastructure of Urban Rail Transit. J. Urban Rapid Rail Transit. 20(4), 40–43 (2007)
7. Li, D.: Modeling and Simulation of Microscopic Pedestrian Flow in MTR Hubs. Beijing Jiaotong University, Beijing (2007)
8. GB 50157-2003 The Metro Design Code. S. China planning press, Beijing (2003)
9. Wang, J.: Study on the Service Level and Capacity Calculation Method of Facilities and Equipments of the Urban Rail Station. Beijing Jiaotong University, Beijing (2011)

Feed-Forward Neural Network Approximation Applied to Activated Sludge System

Muhammad Sani Gaya[1], Norhaliza Abdul Wahab[1,*], Yahya Md Sam[1], and Sharatul Izah Samsuddin[2]

[1] Faculty of Electrical Engineering, Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
muhdgayasani@gmail.com, {aliza,yahaya}@fke.utm.my
[2] Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia
76100 Durian Tunggal, Melaka, Malaysia
sharatul@utem.edu.my

Abstract. The dynamic behavior of an activated sludge system is highly complex and uncertain. To efficiently control and operate the system, a reliable model capable of accurately describing the several time-varying parallel processes of the system is needed. Most of the existing models are too complex to use for design or control purposes. This paper presents a feed-forward neural network model for the system. The model validation was achieved through the use of appropriate international accepted data of the benchmark simulation model no. 1 (BSM1). Simulation studies revealed that the neural network model exhibited an outstanding performance in predicting the effluent quality, root mean square error (RMSE) of 0.0464, mean absolute deviation (MAD) of 0.0347, correlation coefficient (R) of 0.979 for chemical oxygen demand (COD) and RMSE of 0.1103, MAD of 0.0794, R of 0.841 for the total nitrogen (TN) could be acheived. The model is quite effective and suitable tool for the activated sludge system.

Keywords: Model, neural network, learning algorithm, prediction.

1 Introduction

Activated sludge system has been recognized as the most versatile technology used for municipal wastewater treatment system. The system comprises of an aeration unit where the consortium microorganisms capable of degrading the organic pollutants are mixed with raw wastewater in the presence of oxygen and a settler in which the treated water is separated from the solids. However, optimal operation of the system depends on a suitable model capable of predicting the dynamics of the system. The available models proposed by the international association on water quality (IAWQ) task group on mathematical modeling for design and operation of the activated sludge system such as activated sludge model no. 1 (ASM1), ASM2, ASM2D and ASM3 have been used in the wastewater treatment industries. The models have proven to be essential tools for better understanding of mechanisms of

G. Tan et al. (Eds.): AsiaSim 2013, CCIS 402, pp. 587–598, 2013.

activated sludge system and forecasting, particularly the ASM1 that received international acceptability among researchers and practitioners. The ASM1 model was designed purposely for carbon and nitrogen in domestic wastewater treatment [1], [2]. The model is structurally complex and difficult to use for analysis and control [3] [4]. Complexities and uncertainties associated with several processes in the system contributed immensely to difficulty in modeling the system. Neural network presents a better alternative in dealing with uncertainty and complexity of the system.

Several neural network models were proposed for activated sludge system [5], [6], [7], [8], [9] and have demonstrated greater effectiveness in predicting the dynamic behavior of the systems. Nevertheless, most of these models were applied to industrial wastewater treatment plants; they do not focus on domestic wastewater treatment for nutrients' removal. Therefore, the objectives of this paper are to propose a straightforward and efficient neural network model based on ASM1/BSM1 configuration and investigate the performance of the model in predicting the measured (experimental) effluent quality of BSM1. Neural networks are versatile for nonlinear function approximation and can yield a very promising prediction even if the data is complex and rowdy. Attractive features such as adaptability, fault tolerance and fast convergence made neural network a preferred choice in modeling nonlinear system such as the activated sludge system.

Error measures such as a root mean square (RMSE), mean absolute deviation (MAD) and correlation coefficient (R) were used to evaluate the prediction performance of the model. A minimal value of the error measures and high value of R indicates accurate prediction. The paper is organized as follows: Section 2 briefly explains the plant. Section 3 discusses neural network and neural network model implementation, results are presented in section 4. Section 5 concludes the paper.

2 Benchmark Simulation Model No. 1 Plant

The plant consists of a biological reactor tank where the microorganisms responsible for treating the wastewater are allowed to grow and survive and a settler in which the solids (sludge) are separated from the treated water as shown in Fig. 1. A portion of the sludge is recycled to the tanks so as to maintain the microorganisms, and the excess sludge is removed as a waste. The reactor tank is subdivided into five units; the unit 1 and 2 are anoxic units having a volume of $1000m^3$ each; units 3 to 5 are aerobic unit having a volume of $1333m^3$ each, placed in series with the 10 layers settler having a total volume of $6000m^3$. The biological activities taking place in the reactor tanks are described by the ASM1 model. The ASM1 has 13 state variables and eight separate processes. Each of the state variables is described by nonlinear differential equation as:

$$\frac{d\psi}{dt} = \frac{Q}{V}(\psi_{in} - \psi) + r(\psi) \tag{1}$$

where Q is the influent flow rate, V is the reactor volume, ψ is the state variable concentration, ψ_{in} is influent state variable concentration

$$\psi = \begin{bmatrix} S_I & S_S & X_I & X_S & X_{BH} & X_{BA} & X_P & S_O & S_{NO} & S_{NH} & S_{ND} & X_{ND} & S_{ALK} \end{bmatrix}^T$$

$$\psi_{in} = \begin{bmatrix} S_{I,in} & S_{S,in} & X_{I,in} & X_{S,in} & X_{BH,in} & X_{BA,in} & X_{P,in} & S_{O,in} & S_{NO,in} & S_{NH,in} & S_{ND,in} & X_{ND,in} & S_{ALK,in} \end{bmatrix}^T$$

and $r(\xi)$ is the reaction rate of a variable given by:

$$r(S_I) = 0 \tag{2}$$

$$r(S_S) = -\frac{1}{Y_H}\mu_H\left(\frac{S_S}{K_S + S_S}\right)\left(\frac{S_O}{K_{OH} + S_O}\right)X_{BH} - \frac{1}{Y}\mu_H\left(\frac{S_S}{K_S + S_S}\right)\left(\frac{K_{OH}}{K_{OH} + S_O}\right)\left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right)\eta_g X_{BH} + k_h \frac{X_S / X_{BH}}{K_X + \left(X_S / X_{BH}\right)}\left[\left(\frac{S_O}{K_{OH} + S_O}\right) + \eta_h\left(\frac{K_{OH}}{K_{OH} + S_O}\right)\left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right)\right]X_{BH} \tag{3}$$

$$r(X_I) = 0 \tag{4}$$

$$r(X_S) = (1 - f_p)b_H X_{BH} + (1 - f_p)b_A X_{BA} - k_h \frac{X_S / X_{BH}}{K_X + \left(X_S / X_{BH}\right)}\left[\left(\frac{S_O}{K_{OH} + S_O}\right) + \eta_h\left(\frac{K_{OH}}{K_{OH} + S_O}\right)\left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right)\right]X_{BH} \tag{5}$$

$$r(X_{BH}) = \mu_H\left(\frac{S_S}{K_S + S_S}\right)\left(\frac{S_O}{K_{OH} + S_O}\right)X_{BH} + \mu_H\left(\frac{S_S}{K_S + S_S}\right)\left(\frac{K_{OH}}{K_{OH} + S_O}\right)\left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right)\eta_g X_{BH} - b_H X_{BH} \tag{6}$$

$$r(X_{BA}) = \mu_A\left(\frac{S_{NH}}{K_{NH} + S_{NH}}\right)\left(\frac{S_O}{K_{OA} + S_O}\right)X_{BA} - b_A X_{BA} \tag{7}$$

$$r(X_P) = f_p b_H X_{BH} + f_p b_A X_{BA} \tag{8}$$

$$r(S_O) = -\frac{1-Y_H}{Y_H}\mu_H\left(\frac{S_S}{K_S+S_S}\right)\left(\frac{S_O}{K_{OH}+S_O}\right)X_{BH} - \frac{4.57-Y_A}{Y_A}\mu_A\left(\frac{S_{NH}}{K_{NH}+S_{NH}}\right)\left(\frac{S_O}{K_{OA}+S_O}\right)X_{BA} \quad (9)$$

$$r(S_{NO}) = \frac{1-Y_H}{2.86Y_H}\mu_H\left(\frac{S_S}{K_S+S_S}\right)\left(\frac{K_{OH}}{K_{OH}+S_O}\right)\left(\frac{S_{NO}}{K_{NO}+S_{NO}}\right)\eta_g X_{BH} + \frac{1}{Y_A}\mu_A\left(\frac{S_{NH}}{K_{NH}+S_{NH}}\right)\left(\frac{S_O}{K_{OA}+S_O}\right)X_{BA} \quad (10)$$

$$r(S_{NH}) = -i_{XB}\mu_H\left(\frac{S_S}{K_S+S_S}\right)\left(\frac{S_O}{K_{OH}+S_O}\right)X_{BH} + i_{XB}\mu_H\left(\frac{S_S}{K_S+S_S}\right)\left(\frac{K_{OH}}{K_{OH}+S_O}\right)\left(\frac{S_{NO}}{K_{NO}+S_{NO}}\right)\eta_g X_{BH} - \left(i_{XB}+\frac{1}{Y_A}\right)\mu_A\left(\frac{S_{NH}}{K_{NH}+S_{NH}}\right)\left(\frac{S_O}{K_{OA}+S_O}\right)X_{BA} + k_a S_{ND}X_{BH} \quad (11)$$

$$r(S_{ND}) = -k_a S_{ND}X_{BH} + k_h\frac{X_S/X_{BH}}{K_X+\left(X_S/X_{BH}\right)}\left[\left(\frac{S_O}{K_{OH}+S_O}\right)+\eta_h\left(\frac{K_{OH}}{K_{OH}+S_O}\right)\left(\frac{S_{NO}}{K_{NO}+S_{NO}}\right)\right]X_{BH}\left(X_{ND}/X_S\right) \quad (12)$$

$$r(X_{ND}) = \left(i_{XB}-f_p i_{XP}\right)b_H X_{BH} + \left(i_{XB}-f_P i_{XP}\right)b_A X_{BA} - k_h\frac{X_S/X_{BH}}{K_X+\left(X_S/X_{BH}\right)}\left[\left(\frac{S_O}{K_{OH}+S_O}\right)+\eta_h\left(\frac{K_{OH}}{K_{OH}+S_O}\right)\left(\frac{S_{NO}}{K_{NO}+S_{NO}}\right)\right]X_{BH}\left(X_{ND}/X_S\right) \quad (13)$$

$$r(S_{ALK}) = \frac{-i_{XB}}{14}\mu_H\left(\frac{S_S}{K_S+S_S}\right)\left(\frac{S_O}{K_{OH}+S_O}\right)X_{BH} + \left(\frac{1-Y_H}{14*2.86*Y_H}-\frac{i_{XB}}{14}\right)\mu_H\left(\frac{S_S}{K_S+S_S}\right)\left(\frac{K_{OH}}{K_{OH}+S_O}\right)\left(\frac{S_{NO}}{K_{NO}+S_{NO}}\right)\eta_g X_{BH} - \left(\frac{i_{XB}}{14}+\frac{1}{7*Y_A}\right)\mu_A\left(\frac{S_{NH}}{K_{NH}+S_{NH}}\right)\left(\frac{S_O}{K_{OA}+S_O}\right)X_{BA} + \frac{1}{14}k_a S_{ND}X_{BH} \quad (14)$$

These reaction rates of the state variables are due to combinations of fundamental processes [10]. Three (3) data files of the plant were provided by IWA tasks group on benchmarking of control strategies, which are meant to represent three different

weather condition (dry, storm and rain weather), each of the data file contains 14 days sampled at an interval of 15 minutes. Therefore, each data file contained 1344 (4x24x14) samples of data. This paper makes use of chemical oxygen demand (COD) and total nitrogen (TN) to develop the models, since they are the parameters used in reality to measure the strength of wastewater and determine the treatment efficiency of the plant. These parameters are expressed in ASM1/BSM1 as:

$$COD = S_S + X_S + S_I + X_I + X_P + X_{BH} + X_{BA} \tag{15}$$

$$TN = S_{NO} + S_{NH} + S_{ND} + X_{ND} + i_{XB}\left(X_{BH} + X_{BA}\right) + i_{XP}\left(X_P + X_I\right) \tag{16}$$

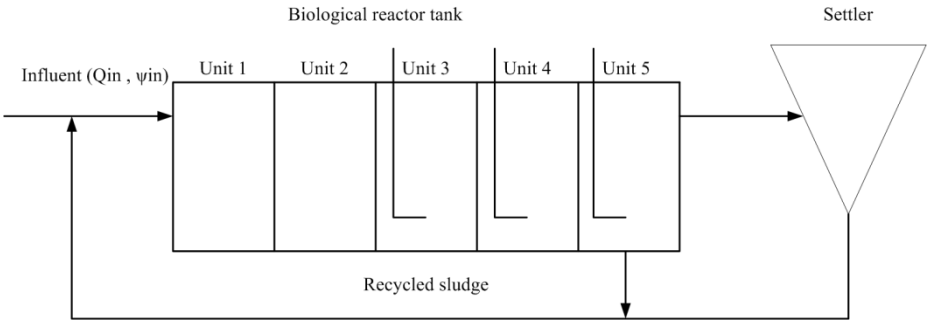


Fig. 1. BSM1 plant

3 Neural Network

Learning abilities and adaptability made neural networks quite attractive. Neural networks have been used to solve several complex real-world problems. They are composed of nodes (artificial neurons) and links. The nodes receive an input signal, process the signal and yield an output. The artificial neurons are the computing elements of neural networks and cooperate to perform the desired task. The neuron is expressed in terms of [11] McCulloch Pitt's model:

$$\gamma = \sum_{i=1}^{k} w_i x_i + \beta \tag{17}$$

where k the number of inputs is, w_i is the weight associated with $i-th$ input x_i and β is the bias. The output y is given by:

$$y = g(\gamma) \tag{18}$$

where g represent the activation function which is applied to the weighted sum of the inputs of the neuron to yield an output.

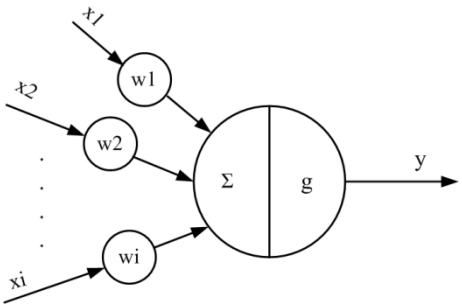


Fig. 2. The neuron model

Most of the neural networks use sigmoid function as the activation function (transfer function). The weights are updated during training the network by an algorithm. In neural networks, the links indicate the direction of information flow, which can be in one direction or bidirectional. Neural networks are classified based on their structure as feed-forward or recurrent neural network. Feed-forward networks are easier to build and have unidirectional flow of information as shown in Fig. 3. This type of neural network is the most commonly used for many applications. Recurrent neural networks have feedback originating from an output layer to the input layer or from hidden layer to the input layer [12]. The existence of feedback increases the network performance and learning capabilities. Recurrent networks are valuable tools for nonlinear identification and control; they are usually implemented in parallel or series-parallel mode.

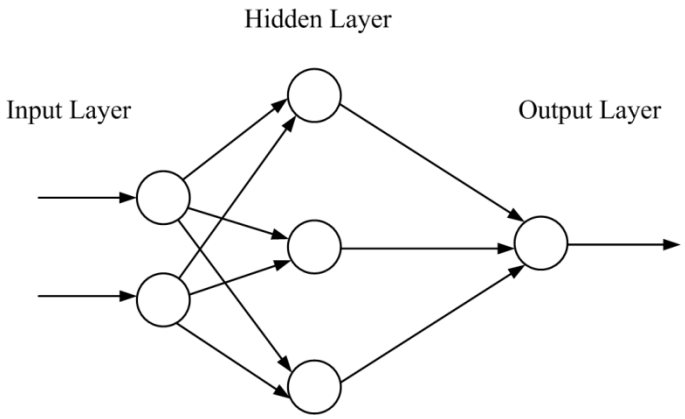


Fig. 3. The feed-forward neural network structure

Learning composed of updating the weights of the networks through learning algorithm when data sets are made available. A learning algorithm refers to a method where the learning rules are utilized in order to optimize the weights [13]. The learning paradigm can be supervised or unsupervised learning. In supervised learning paradigm, inputs and the targets are provided to the network, and the weights are

updated so that the responses (outputs) of the network are as close as the targets. Unlike supervised learning, only inputs are provided in unsupervised learning. The network updates its weights so that the inputs yield corresponding outputs. Both these two types learning update the weights according to a learning algorithm. There are many learning algorithms to train a network, and the objective of the algorithms is to minimize the error between the target and output of the network.

3.1 Back-Propagation Algorithm

1. Once the input propagates through the network, error between the target and network output is computed, the error is propagated backward, and the weights are updated in order to minimize the error. Back-propagation algorithms utilize supervised learning. The network is provided with the input –output data to be approximated. The training of the network to realize the desired target (output) starts with arbitrary weights, and the weights are updated until error is reduced. The error is defined [14] as:

$$E(t) = \frac{1}{2}\sum_{j=1}^{m}\left(y_j(t) - T_j(t)\right)^2 \tag{19}$$

where y and T represent the network output and target at time t respectively.

The overall error measure is given by

$$E = \sum_{t=1}^{T} E(t) \tag{20}$$

Back-propagation algorithm is a gradient descent approach of reducing error. The weights are updated using the gradient descent

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \tag{21}$$

where η is the learning rate.

3.2 Determining a Neural Network Structure

Performance and learning capabilities of a network rely on an appropriate network structure. Therefore, determining network structure is an essential step in developing a neural network model. Network structure comprises of determining the number of hidden layers and number of hidden neurons. Number of studies shows that one or two hidden layers are enough for fast convergence [12]. However, with regard to the number of hidden neurons, there is no systematic procedure for selecting the number of hidden neurons. In most cases, trial-and-error approach is adapted. Notwithstanding, there exists upper and lower limits for the numbers as proposed by [15].

$$N^{\alpha} \leq 2N^{\theta} + 1 \tag{22}$$

where N^{α} depicts the number of hidden neurons and N^{θ} is the number of inputs.

In order to avoid over-fitting the training data, [16] proposed a relationship between the number of samples of training data and number of hidden neuron.

$$N^{\alpha} \leq \frac{N^{\xi}}{\left(N^{\theta} + 1\right)} \tag{23}$$

where N^{ξ} is the number of samples of training data.

3.3 Feed-Forward Network Implementation

In this paper, six hundred and seventy two (672) data samples of BSM1 were used. To improve the performance of the network, the data samples were transformed into a trainable form by normalizing the data between zero and one using minimum-maximum method.

$$X_i = \frac{x_i - v_{\min}}{v_{\max} - v_{\min}} \tag{24}$$

where X_i is the normalized value of the sample, x_i represent the sample value, $v_{\min}$ is the minimum value in the data samples and $v_{\max}$ depicts the maximum value.

The normalized data were divided into 470 samples (70%) for training and 202 samples (30%) for testing. The selection of the training and testing data were done randomly. Based on the proposal [16], a three-layer network was selected having input layer containing the input variables, connected to the hidden layer (with 15 neuros) trailed by the output layer. A tan-sigmoid and purelin transfer functions were chosen for the hidden layer and output layer respectively. The network was trained with the back-propagation algorithms through two thousand (2000) training epochs.

4 Result and Discussion

Fig. 4 shows the influent COD and TN concentrations for the seven-day period (672 samples) which are representative of the system inlet variations. The prediction performance of the neural network model was evaluated through simulation using the root mean square error (RMSE), mean absolute deviation (MAD) and correlation coefficient (R). These measures are expressed as:

$$RMSE = \sqrt{\frac{\Sigma\left(x_i - y_i\right)^2}{N}} \tag{25}$$

$$MAD = \frac{\Sigma \mid x_i - y_i \mid}{N} \tag{26}$$

$$R = \frac{\Sigma_{i=1}^{N} \left(x_i - \bar{x} \right) \left(y_i - \bar{y} \right)}{\sqrt{\left(\Sigma_{i=1}^{N} \left(x_i - \bar{x} \right)^2 \right) \left(\Sigma_{i=1}^{N} \left(y_i - \bar{y} \right)^2 \right)}} \tag{27}$$

where x_i is the measured value, $\overline{x_i}$ is mean of the measured value, y_i is the predicted value, $\overline{y_i}$ is mean of the predicted value and N is the number of samples.

Training data set are used to ascertain potentially estimative correlation and mostly chosen to be the largest percentage of the whole data set, 70% was selected in this study. Testing data are utilized to measure the capability of the model in responding to an unseen or unusual data set. Usually, testing data takes at most 30% of the complete data set. The results from the training and testing are presented in Table 1. The FFNN model predictions during testing for the measured COD and TN were illustrated in Fig. 5 and 6 respectively.

Table 1. Model prediction performance

Variable	Training			Testing		
	RMSE	MAD	R	RMSE	MAD	R
COD	0.0439	0.0317	0.981	0.0464	0.0347	0.979
TN	0.1023	0.0738	0.856	0.1103	0.0794	0.841

The FFNN model predictions for the measured COD and TN were illustrated in Fig. 5 and 6 respectively. During the training, the correlation coefficient (R) between the measured COD and the network model predicted COD is 0.981 indicating a very strong correlation. From the Fig. 5 when testing, the predicted values of COD by the neural network model were able to follow well the measured COD. The error measures were still low, and the correlation is very strong as value of R is 0.979.

Despite some drifts during testing of the network model for TN variable which could be connected to the prediction errors, but, still there is the good agreement between the measured TN and predicted TN by the network model as depicted in Fig. 6 and the correlation is quite strong which shows that the reliable prediction could be achieved .

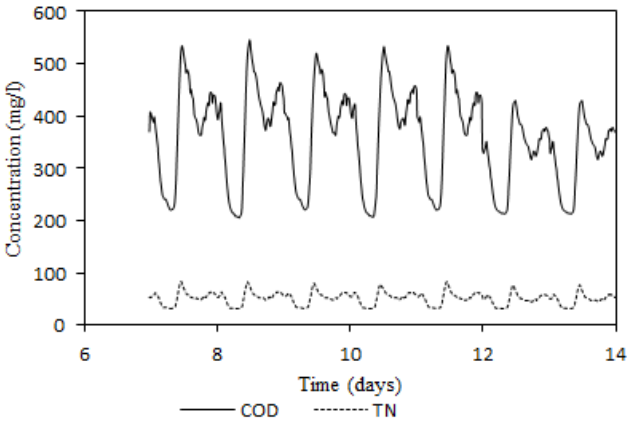


Fig. 4. The influent concentrations

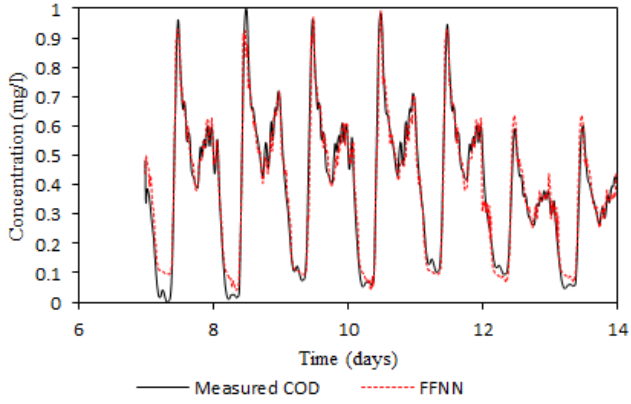


Fig. 5. Model testing pattern for COD variable

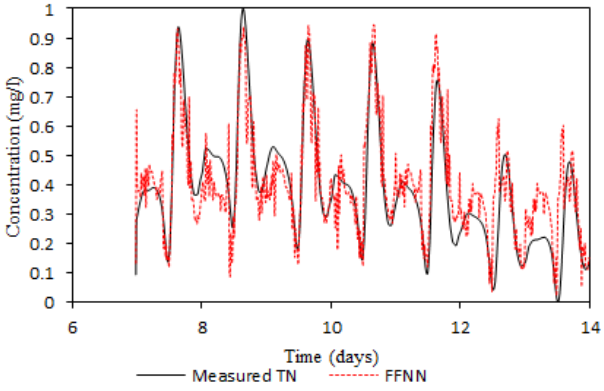


Fig. 6. Model testing pattern for TN variable

5 Conclusion

The paper has presented a feed-forward neural network model. With the back-propagation learning algorithm, the model was able to predict well the measured COD and TN with high degree of accuracy. The results demonstrated the efficiency and robustness of the network model, thus proving that the feed-forward network model is suitable for the wastewater treatment plant. The model can be used to design a control algorithm for the system which may results in increasing the efficiency, reducing operational/maintenance cost and producing the desired high quality effluent that nature can handle.

Acknowledgement. The authors wish to thank Ministry of Higher Education (MOHE), Universiti Teknologi Malaysia and Research University Grant (GUP) vote Q.J130000.2533.02H70 for their financial support. This support is gratefully acknowledged.

References

[1] Henze, M., Gujer, W., Mino, T., Loosedrecht, V.: Activated Sludge Models ASM1, ASM2, ASM2D and ASM3. IWA Publishing, London (2000)
[2] Gujer, W.: Activated Sludge Modelling: past, present and future. Water Science and Technology 53(3), 111–119 (2006)
[3] Jeppson, U.: A Simplified Control-Oriented Model of the Activated Sludge Process. Mathematical Modelling of Systems 1(1), 3–16 (1995)
[4] Tay, J., Zhang, X.: A Fast Predicting Neuro Fuzzy Model for High-Rate Anaerobic Wastewater Treatment Systems. Water Research 34(11), 2849–2860 (2000)
[5] Gontarski, C., Rodrigues, P., Mori, M., Prenem, L.: Simualtion of an Industrial wastewater Treatment Plant using Artificial Neural Networks. Computers and Chemical Engineering 24, 1719–1723 (2000)
[6] Oliveira-Esquerre, K., Mori, M., Bruns, R.: Simulation of an Industrial Wastewater Treatment Plant using Artificial Neural Networks and Principal Component Analysis. Brazilian Journal of Chemical Engineering 19(4), 365–370 (2002)
[7] Nasr, M., Moustafa, M., Seif, H., El-kobrosy, G.: Application of the Artificial Neural Network for the Prediction of El-agamy Wastewater Treatment Plant Performance-Egypt. Alexandria Engineering Journal 51, 37–43 (2012)
[8] Mjalli, F., Al-Asheh, S., Al-Fadala, H.: Use of Artificial Neural Network Black Box Modelling for the Prediction of Wastewater Treatment Plants Performance. Journal of Environmental Management 83(3), 329–338 (2007)
[9] Kriger, C., Tzenova, R.: Neural Networks for Prediction of Wastewater Treatment Plant Disturbance. In: 8th IEEE-Africon Conference, Windhoek, Namibia (2007)
[10] Alex, J., Benedetti, L., Copp, J., Garnaey, K., Jeppson, U., Nopens, I., Pons, M., Rosen, C., Steyer, J., Vanrolleghem, P.: Benchmark Simulation Model no. 1. IWA publishing, London (2008)
[11] McCulloch, W., Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics 5, 115–133 (1943)

[12] Chiang, Y., Chang, L., Chang, F.: Comparison of Static feed-forward and Dyanmic Feedback Neural Networks for Rainfall-Runoff Modelling. Journal of Hydrology 290, 297–311 (2004)
[13] Jain, A., Mao, J., Mohluddin, K.: Artificial Neural Networks: A Tutorial. IEEE-Computer Magazine 29(3), 31–44 (1996)
[14] Rumelhart, D., Hinton, G., Williams, R.: Learning Internal Representations by Error Propagation. Parallel Distributed Processing 1, 318–362 (1986)
[15] Hetch, N.: Kolmogorov's Mapping Neural Network Existence Theorem. In: IEEE- First Annual International Conference on Neural Networks, New York (1987)
[16] Rogers, L.L., Dowla, F.U.: Optimization of Groundwater Remediation using Artificial Neural Networks with Parallel Solute Transport Modelling. Water Resource Research 30(2), 457–481 (1994)

Author Index