2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

Adaptive algorithm for estimating the position of a passive object in a picking shelf

Ibrahim Ibrahim, Kai Rieger, Tobias Draeger, Rafael Psiuk
Self-Powered Radio Systems
Fraunhofer IIS
Nuremberg, Germany
ibrahim.ibrahim@iis.fraunhofer.de

***Abstract*—This paper presents an algorithm to estimate the position of an object in an industrial picking shelf based on low frequency magnetic fields. The proposed algorithm was designed as an improvement to the one used by the IndLoc system which has been developed at Fraunhofer Institute for Integrated Circuits. It reduces the calculation time by extracting a subtable, a portion, from a simulated look-up table during each iteration. The algorithm is tested on the IndLoc system which consists of one AC current loop, 16 receiving coils and a passive localization object that comprises three orthogonal coils. Additionally, time measurements are carried out using different subtable sizes and compared with using only one high-resolution look-up table.**

***Index Terms*—LF magnetic field, localization, algorithm, industry 4.0**

I. Introduction

In modern logistics companies the ordered goods are picked and collected by employees from different shelves, then shipped to their respective customers. To facilitate the picking process and to support the employees with preventing wrong picks, an assisting system is needed. The system shall locate the employee's hand and give him feedback if the performed pick was wrong or right. For that purpose, the IndLoc system has been developed at Fraunhofer Institute for Integrated Circuits (IIS) to support the Industry 4.0 applications. The IndLoc system is based on Low Frequency (LF) inductive localization. Such systems can be used in non line of sight environments because their magnetic fields are not affected by many obstacles including the human body, unlike camera-based localization systems, which require a clear line of sight to perform properly. Additionally, camera-based localization systems cannot be used in most companies as it is not allowed to record footage of the employees due to privacy issues. LF-based localization does not suffer from reflections and multi-path propagation problems, which are present in the radio-based localization systems [1]. All these advantages enabled its usage in several other applications such as tracking animals underground [2], goal line technology in football [3], [4], underground localization in mines [5], the positioning of electric vehicles for wireless charging stations [6] and the localization of an indoor mobile robot [7].

One critical condition for preventing wrong picks in logistics is that the localization must be accurate and the feedback is given to the employee with minimum delay. The IndLoc system locates the employee's hand by measuring the magnetic field of a passive object, which is mounted on the employee's wrist, via 16 receiving coils distributed around the shelf's area. The measured values are then compared with a look-up table that includes calculated field values for predefined object's positions. The object's position estimate is the table entry with the minimum Euclidean distance.

This approach performs well on small tables with a low grid resolution, depending on the CPU speed. However, for larger tables with a higher grid resolution, more computations are carried out and more time is needed to find the entry with the minimum distance. Hence, the employee receives a delayed feedback.

In this paper we present an adaptive localization algorithm where the number of calculations to find the table entry with the minimum Euclidean distance is reduced, which achieves much faster and more accurate localization. The proposed algorithm can be adapted and applied to any other system where large 3D look-up tables are being constantly searched to find the entry with a unique value.

The paper is structured as follows: Section II describes the system setup as well as the problem under investigation. Section III illustrates the proposed adaptive algorithm. In Section IV, the performance of the algorithm is evaluated and results from time measurements are provided. Finally, Section V concludes the paper.

II. System description

The IndLoc system is a further development of the GoalRef system described in [3], which consists of a reader, a $95 \times 102\,\text{cm}$ rectangular exciter loop and eight pairs of orthogonal receiving coils. Each pair consists of two $2 \times 32\,\text{cm}$ rectangular coils with 20 windings each. As shown in Fig. 1, the exciter loop and the receiving coils are integrated inside fiberglass tubes that form a picking shelf. The shelf is divided into 9 individual compartments. The IndLoc system is currently at the final stages of researching and its unit price should be available soon.

Fig. 2 shows the passive localization object which can be attached to the employee's wrist. It consists of three orthogonal circular coils, each with a radius of 2 cm and 20 windings.

978-1-7281-1788-1/19/$31.00 © 2019 IEEE

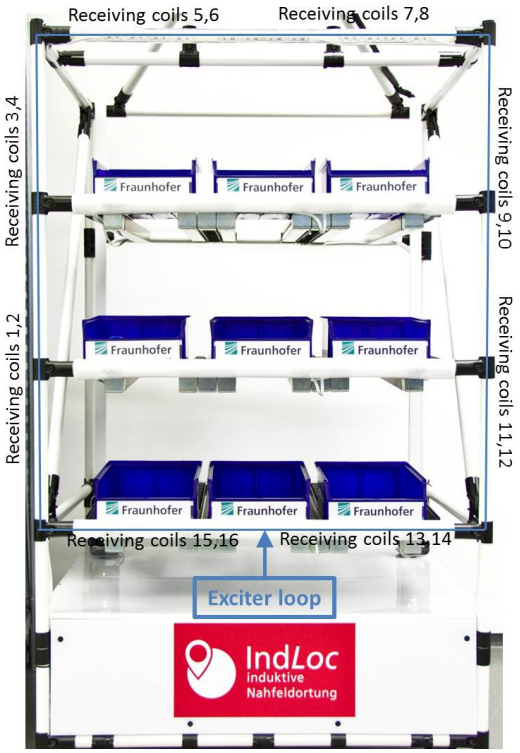

Fig. 1. The IndLoc system comprising an exciter loop and eight receiving coil pairs integrated inside fiberglass tubes forming a picking shelf.

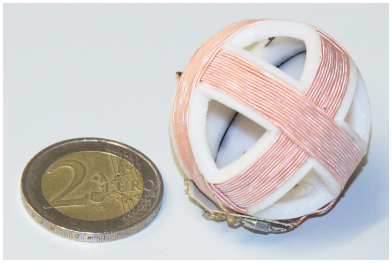

Fig. 2. Passive localization object compared with the size of a two Euro coin. The object consists of three orthogonal coils each with a radius of 2 cm and 20 windings.

A. Methodology

The exciter loop carries an AC current I_p, with a frequency of 119 kHz, which generates a primary magnetic field $\vec{H}_p$. This field induces a voltage U_c in the tuned localization object's coils according to Faraday's Law, where $\vec{A}$ is the area vector of the coil and μ is the magnetic permeability [8]:

$$U_c = -\frac{d}{dt}\int \mu \vec{H}_p \, d\vec{A} \tag{1}$$

A current flows in each of the object's coils due to the induced voltage U_c, which produces a secondary magnetic field $\vec{H}_s$ that induces voltages $\widetilde{U}_1, ..., \widetilde{U}_{16}$ in the sixteen receiving coils. As shown in Fig. 3, the measured voltages are sampled and provided to the reader hardware that performs filtering, amplification and signal processing. The reader forwards the processed samples via User Datagram Protocol (UDP) datagrams to a personal computer (PC). On the PC, a 3D localization algorithm is running, which is implemented in the Python programming language. For each sample, the object's position is estimated and forwarded to a Graphical User Interface (GUI), which helps in monitoring the picking process and gives feedback to the employee on the performed pick.

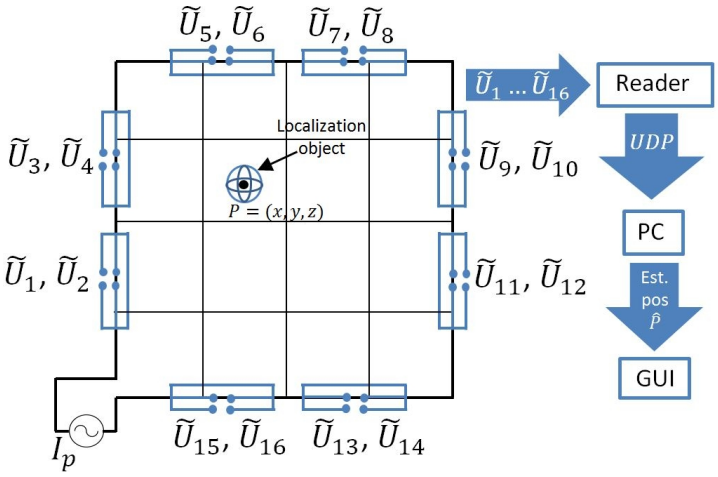

Fig. 3. The measured voltages $\widetilde{U}_1, ..., \widetilde{U}_{16}$, due to a localization object positioned at point P, are forwarded to the reader hardware. After filtering and signal processing, the voltage samples are used to find an estimate $\hat{P}$ for the object's position. The GUI gives feedback to the employee on the performed pick.

B. Generating a look-up table

The look-up table is generated as illustrated in Algorithm 1 and the induced voltages are calculated using the analytical model described in [9].

Algorithm 1 Generating a look-up table

```
X = {x_0, ..., x_m}
Y = {y_0, ..., y_n}
Z = {z_0, ..., z_l}
table = [ ]
for x in X do
    for y in Y do
        for z in Z do
            calculate induced voltages U_1, ..., U_16 from an
            object positioned at (x, y, z)
            append x, y, z, U_1, ..., U_16 to table
        end for
    end for
end for
```

Table I illustrates the structure of the generated table. The order of the three loops of Algorithm 1 can be seen in the first three columns of the table. The table's number of rows r is the product of the number of elements of the position sets

$$r = (m+1) \times (n+1) \times (l+1) \tag{2}$$

where m, n, l are the indices of the last element in the X, Y, Z position sets, respectively and they all start from zero. The table's number of columns c is 19 and its size can be expressed as (r, c).

TABLE I
THE STRUCTURE OF THE GENERATED LOOK-UP TABLE USING ALGORITHM 1.

	X	Y	Z	U_1	U_2	...	U_{16}
$i=0$	x_0	y_0	z_0	$U_1(x_0,y_0,z_0)$	$U_2(x_0,y_0,z_0)$	...	$U_{16}(x_0,y_0,z_0)$
$i=1$	x_0	y_0	z_1	$U_1(x_0,y_0,z_1)$	$U_2(x_0,y_0,z_1)$	...	$U_{16}(x_0,y_0,z_1)$
	...	...	...	...	...	...	...
	x_0	y_1	z_0	$U_1(x_0,y_1,z_0)$	$U_2(x_0,y_1,z_0)$	...	$U_{16}(x_0,y_1,z_0)$
	x_0	y_1	z_1	$U_1(x_0,y_1,z_1)$	$U_2(x_0,y_1,z_1)$	...	$U_{16}(x_0,y_1,z_1)$
	...	...	...	...	...	...	...
	x_1	y_0	z_0	$U_1(x_1,y_0,z_0)$	$U_2(x_1,y_0,z_0)$	...	$U_{16}(x_1,y_0,z_0)$
	x_1	y_0	z_1	$U_1(x_1,y_0,z_1)$	$U_2(x_1,y_0,z_1)$	...	$U_{16}(x_1,y_0,z_1)$
	...	...	...	...	...	...	...
$i=r-1$	x_m	y_n	z_l	$U_1(x_m,y_n,z_l)$	$U_2(x_m,y_n,z_l)$	...	$U_{16}(x_m,y_n,z_l)$

C. Localization

The localization algorithm compares the measured voltages from the receiving coils with each entry of the look-up table as follows:

$$d_i(\widetilde{U}_1, ..., \widetilde{U}_{16})^2 = \sum_{j=1}^{16} (\widetilde{U}_j - U_{j,i})^2 \quad (3)$$

where d_i is the sum of Euclidean distances between the measured voltages by the receiving coils and the calculated voltages $U_{j,i}$ for coil number j in the table entry with index i. The object's position estimate $\hat{P} = (\hat{x}, \hat{y}, \hat{z})$ corresponds to the table entry with the minimum Euclidean distance. It is necessary that each simulated table entry gives a unique Euclidean distance value when compared with the measured voltages. The index i^* of such an entry can be expressed as:

$$i^* = \underset{i}{\operatorname{argmin}}\, d_i(\widetilde{U}_1, ..., \widetilde{U}_{16})^2 \quad (4)$$

D. Problem description

During the aforementioned localization process, Eq. (3) is executed r times. For higher resolution tables with large X, Y, Z position sets, the number of calculations increases and hence the localization algorithm takes more time to find the entry with the minimum distance.

III. THE PROPOSED ALGORITHM

Instead of using only one look-up table during the localization process, two tables with different X, Y, Z grid resolutions are used. The first table $table1$ has a lower resolution than the second table $table2$ and $table1 \subset table2$. $table1$ is used only to get a first rough estimate of the object's position which is used afterwards to get a more accurate estimation using a subtable from $table2$.

A. The algorithm's work-flow

As illustrated in Fig. 4, $table1$ and $table2$ are loaded once into memory at the start of the algorithm, and the object's position estimate $\hat{P}$ is set to $None$. During each iteration, the algorithm checks if there is already an estimate of the object's position $\hat{P}$ and it sets $table$ to $table1$ if there is none. Each received sample from the reader is compared with all $table$'s entries to estimate the object's position. The estimated position $\hat{P}$ is fed back as an input to the if statement and $\hat{P}$ is no longer equal to $None$. This leads to an extraction of a $subtable$, which includes only few neighboring points of the previously estimated $\hat{P}$, from the high resolution $table2$. In the next step, the look-up table $table$ is set to $subtable$ and a new position is estimated via comparing the current sample with all $subtable$'s entries. The algorithm output is $\hat{P} = (\hat{x}, \hat{y}, \hat{z})$, which is forwarded to the GUI for monitoring the picking process.

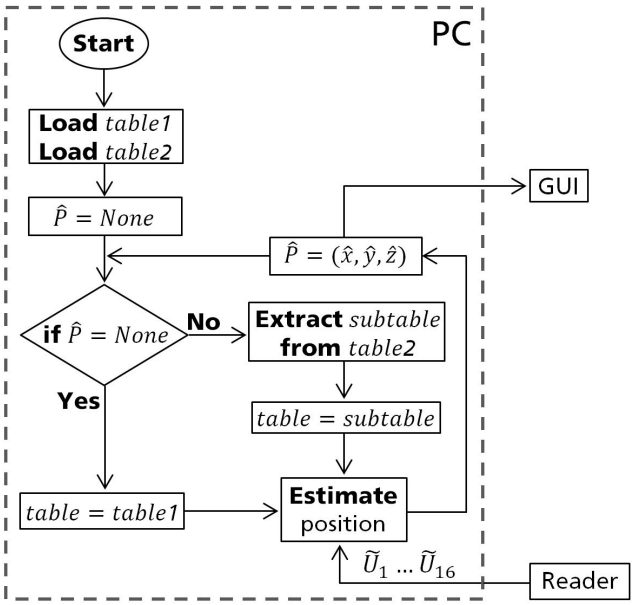

Fig. 4. Flowchart of the proposed adaptive algorithm.

B. Extracting a subtable

In order to extract a $subtable$ from the high resolution $table2$, two values are necessary. The first is the entry index of the previously estimated position $\hat{P}$ in $table2$. For any $table2$, which was generated by Algorithm 1, the index $i_{x,y,z}$ of a position entry (x, y, z) can be calculated as

$$i_{x,y,z} = \big(i_x \times (n+1) \times (l+1)\big) + \big(i_y \times (l+1)\big) + i_z \quad (5)$$

where i_x, i_y and i_z are the indices of the elements x, y, z in the X, Y, Z-position sets, respectively. For example, the following position sets $X = \{x_0, x_1\}$, $Y = \{y_0, y_1, y_2\}$ and $Z = \{z_0\}$, have the values $m = 1, n = 2, l = 0$ from Algorithm 1. Using Eq. 5, the index i_{x_1,y_2,z_0} of the position entry (x_1, y_2, z_0) can be calculated by substituting the indices i_x, i_y, i_z with $1, 2, 0$, respectively. This results in $i_{x_1,y_2,z_0} = 5$.

The second necessary value is the $Cube\ Length\ (CL)$, which is an odd integer number greater than 1 and defined as the cube root of the number of neighboring points to be included in *subtable*. For each *subtable*, Eq. 5 will be executed CL^3 times[1] to find the index of each neighboring point of the previous position estimate $\hat{P}$. Fig. 5 illustrates an extracted *subtable* from a high resolution *table2*, with $CL^3 = 3^3$ and an estimated position $\hat{P}$ at the center of *table2*. The illustrated *subtable* consists of $27\,(x, y, z)$ positions.

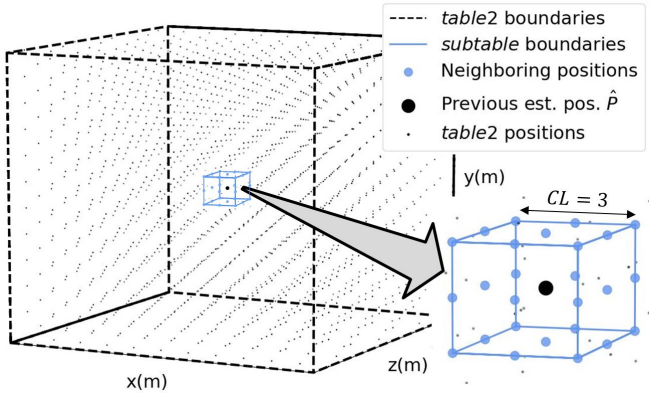

Fig. 5. Extracted *subtable* including with a total of $27\,(x, y, z)$ positions $(= CL^3)$, from a higher resolution *table2*

IV. Algorithm Performance

A. Throughput

To evaluate the throughput of the proposed localization algorithm, the object was positioned exactly in the center of the IndLoc picking shelf and 10000 samples were recorded from the reader hardware. Two look-up tables were generated using Algorithm 1. The first table is *table1*, which has X, Y, Z grid resolution of 3 cm and a total of $CL^3 = 15^3$ positions. The second table is *table2* that has a grid resolution of 1 cm and a total of $CL^3 = 57^3$ positions. To eliminate any time delay caused by the reader hardware or software, the recorded samples were loaded and provided to the localization algorithm in a separate post processing stage. The algorithm was implemented in the Python programming language. The following time measurements were carried out on a Windows PC with an Intel Core i7 processor operating at 3.40 GHz. *table1* and *table2* took up a memory space of 660 KB and 36171 KB, respectively. According to the flowchart in Fig. 4, the total calculation time t_t, to estimate the object's position from one reader sample, can be expressed as

$$t_t = t_x + t_e \tag{6}$$

where t_x is the time to extract a *subtable* from a high resolution table, and t_e is the time to calculate the minimum Euclidean distance between the sample and all the extracted *subtable*'s entries.

The proposed algorithm was used with different *subtable* sizes, starting from $CL^3 = 3^3$ to $CL^3 = 57^3$. For each CL^3 value, *table1* was used only to estimate the object's first position from the first sample of the 10000. For each subsequent sample, Eq. (5) was used to calculate the index of each neighboring point within the specified CL^3 value. Afterwards, a *subtable* was extracted from *table2* using the calculated indices and used to estimate the object's position with an accuracy of ± 1 cm. Additionally, t_x and t_e were measured and the average of their total t_t was calculated for each sample. Fig. 6 shows, that as the extracted *subtable* size increases, both t_x and t_e increase. For any CL^3 value, t_x was found to be greater than t_e. Therefore, we define a threshold value CL^3_{th}, which is the maximum recommended CL^3 to be used with the proposed algorithm without introducing an overhead on using only the high resolution *table2* for position estimation. As shown in Fig. 6, *table2* with $CL^3 = 57^3$ has a $t_{th} \approx 35.02$ ms, where t_{th} is the time taken to find the entry with the minimum Euclidean distance using (3) and (4). By intersecting the horizontal t_{th} with t_t, we deduce $CL^3_{th} \approx 31^3$.

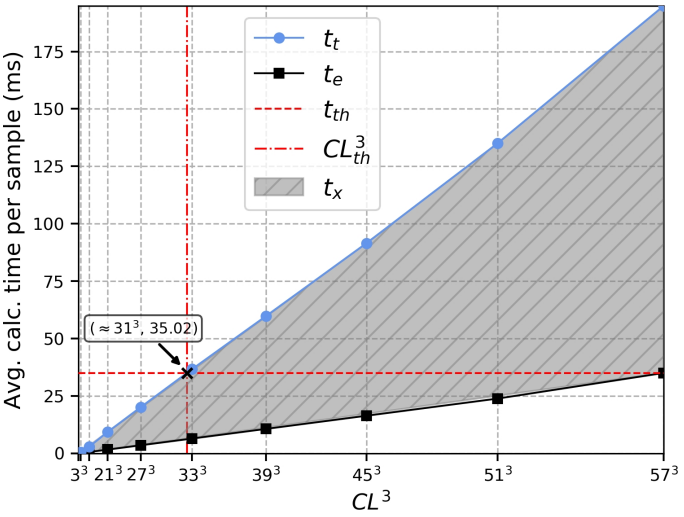

Fig. 6. The average calculation time per sample for different *subtable* sizes.

The throughput of the algorithm can be expressed as

$$Throughput = \frac{1}{t_t} \tag{7}$$

which is the number of the processed samples per second and t_t is calculated using (6) for a given CL^3 value. Table II includes the calculated $Throughput$ for different CL^3 values. It indicates that using a *subtable* with a $CL^3 = 3^3$ results in a $Throughput$ approximately 564 times greater than any of the following cases: Case 1, using all the entries of *table2* to find the minimum distance for each sample without extracting any *subtable*. Case 2, using the proposed algorithm with $CL^3_{th} = 31^3$ and extracting a *subtable* for each sample.

[1]Depending on the previous position estimate $\hat{P}$ (if it is in the center of *table2* or near it's edges/corners) some of the neighboring points might be outside the boundaries of *table2*. This results in a smaller *subtable* with fewer neighboring points and a center point that is different from $\hat{P}$.

TABLE II
THE PROPOSED ALGORITHM'S THROUGHPUT.

CL^3	$Throughput\ (S/s)$
3^3	16129.03
9^3	1666.66
15^3	347.22
21^3	106.72
27^3	49.67
$CL^3_{th} \approx 31^3$	28.55

B. Localization accuracy

To evaluate the accuracy of the proposed algorithm, two movement scenarios were performed on an empty IndLoc shelf. During each scenario 10000 samples were recorded from the reader hardware. In the first scenario, the object's movement started from the bottom left corner of the shelf towards the upper right corner. In the second scenario, the object was moved from the top left corner of the shelf towards the bottom right corner in a zigzag path. In post processing the object's positions were estimated using two approaches. Firstly, the adaptive algorithm was used with a $subtable$ size of $CL^3 = 3^3$. Secondly, all the entries of the high resolution $table2$ were used to estimate the object's positions. The localization results from both approaches are compared in Fig. 7 and Fig. 8 from the two movement scenarios respectively.

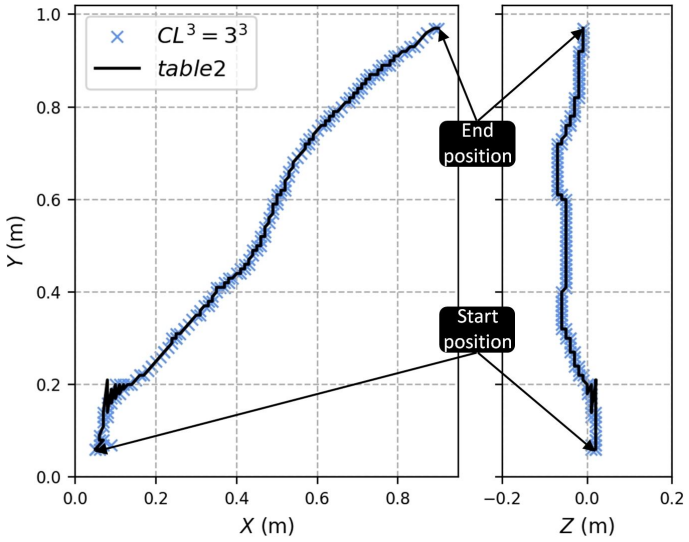

Fig. 7. The localization results from the first movement scenario using the proposed algorithm with $CL^3 = 3^3$ and compared with using $table2$.

As can be seen in both Fig. 7 and Fig. 8, the proposed algorithm with $CL^3 = 3^3$ has produced the same results as using all the entries of $table2$ except near the starting position, where a low resolution table ($table1$) was used to estimate the object's initial position from the first recorded sample.

V. CONCLUSION

We have presented an adaptive algorithm for estimating the position of a passive object within a magnetic field environment. The algorithm's performance has been evaluated

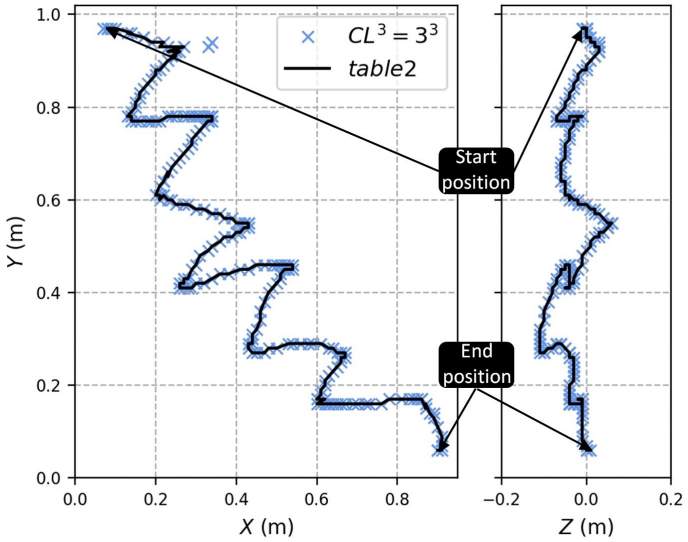

Fig. 8. The localization results from the second movement scenario using the proposed algorithm with $CL^3 = 3^3$ and compared with using $table2$.

using the IndLoc picking shelf. For different $CubeLength$ (CL) values, time measurements were conducted and an upper limit on the extracted table size was deduced. Results show that the proposed algorithm can achieve throughput up to 564 times greater than using only one fixed-size look-up table with a high resolution X, Y, Z grid. Future work will comprise analysis of the influence of metal objects such as packaging of certain products on the algorithm's localization accuracy.

REFERENCES

[1] A. Niitsoo, T. Edelhäußer, E. Eberlein, N. Hadaschik, C. Mutschler. "A Deep Learning Approach to Position Estimation from Channel Impulse Responses." *Sensors*, vol. 19, no. 1064, Mar. 2019.

[2] A. Markham, N. Trigoni, S. A. Ellwood, D. W. Macdonald. "Revealing the hidden lives of underground animals using magneto-inductive tracking." *SenSys'10, Zurich, Switzerland*, Nov. 2010.

[3] R. Psiuk, T. Seidl, W. Strauss, J. Bernhard. "Analysis of Goal Line Technology from the perspective of an electromagnetic field based approach," in *Procedia Engineering 72*, 2014, pp. 279-284.

[4] B. E. Fischer, I. J. LaHaie, D. D. Arumugam, J. D. Griffi, D. D. Stancil, D. S. Ricketts. "Three-Dimensional Position and Orientation Measurements Using Magneto-Quasistatic Fields and Complex Image Theory." *IEEE Antennas and Propagation Magazine*, vol. 56, No. 1, pp.160-173, Feb. 2014.

[5] Q. Huang, X. Zhang, J. Ma. "Underground Magnetic Localization Method and Optimization Based on Simulated Annealing Algorithm." *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing*, Aug. 2015.

[6] D. Martinovic, M. Grimm, H.-C. Reuss. "Electric Vehicle Positioning for Inductive Charging Purposes Using Magnetic Field Distortion Elimination in High-Permeability Environments." *IEEE Transactions on Magnetics*, vol. 50, no. 11, Nov. 2014.

[7] H. Chao, F. Zhongqing, R. Yupeng, C. Yueyue, L. Haixiang, W. Kai, X. Xiaodong, B. Jianmeng. "An Efficient Magnetic Localization System For Indoor Planar Mobile Robot." *34th Chinese Control Conference*, pp.4899-4904, Jul. 2015.

[8] G. Lehner. *Electromagnetic Field Theory for Engineers and Physicists*. Springer, 2010.

[9] R. Psiuk, A. Artizada, D. Cichon, H. Brauer, H. Toepfer, A. Heuberger. "Modeling of an Inductively Coupled System." *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 37, iss. 4, pp.1500-1514, 2018.