

University of Applied Sciences Telematics/Network Engineering

MASTER THESIS

Investigations on Influences on Magnetic Localization Accuracy

A master thesis submitted in partial fulfillment of the
requirements for the degree of
Diplomingenieur

Author: Ibrahim, Shawkey Ibrahim
Matriculation Number: 1510287009

Academic Tutor: FH-Prof. Dipl.-Ing. Dr. Helmut Wöllik

Tutor: Dipl.-Ing. Rafael Psiuk

Deadline: 22nd of September, 2017

Name: Ibrahim, Shawkey Ibrahim
Matriculation Number: 1510287009
Date of Birth: 25th of July, 1992
Address: Karl-Jatho-Weg 14, 90411 Nürnberg, Germany

A f f i d a v i t

On the basis of this statement I declare to have issued, produced and compiled this master thesis solely by myself. All means and sources used are quoted accordingly.

Declaration Date: 22nd of September, 2017
Location: Nürnberg, Germany
Signature:

Acknowledgement

Life is a continuous process of learning and acquiring knowledge. It all started with learning how to walk when I was a little child. To the best mentors in my life my mom and dad; I would like to say thank you for your patience and love during all these years.

Additionally, I would like to thank all the academic staff and members of Fachhochschule Kärnten for their efforts and time that they have given me during the past two years of my master studies.

"The more you know the wiser you become." quoting my academic supervisor Prof. Wöllik during our first lecture at FH-Kärnten. For him, I would like to say thank you for believing in my abilities and for teaching me how to be a better engineer.

Furthermore, I would like to thank my supervisor Ing. Rafael Psiuk for his time, help and guidance throughout the whole seven months of my master thesis work at Fraunhofer IIS in Nürnberg, Germany. He always motivated me to achieve my full potential by engaging me with challenging tasks.

To all the aforementioned, I would not be who I am today without your priceless support.

Abstract

There are various factors that influence the accuracy of magnetic field based localization systems. Some of those are geometrical imperfections, noise, man-made disturbing signals, deformation of the object to be localized or permeable and conducting obstacles near the system. Furthermore, a badly designed model can be a source of imprecision for model based localization. In this master thesis, some of the aforementioned factors will be investigated and their effect on the system's estimation accuracy of an object's position will be quantified. This will be achieved by the means of Monte-Carlo based simulations, Cramer-Rao lower bound calculations and analytic approaches.

Keywords— Magnetic field, localization, Cramer-Rao lower bound (CRLB), Monte-Carlo (MC) simulations, modeling.

List of Abbreviations

1-D One Dimensional

2-D Two Dimensional

AC Alternating Current

CRLB Cramer Rao Lower Bound

MC Monte Carlo

μ_0 Permeability of Free Space

WGN White Gaussian Noise

Contents

1. Introduction	11
2. Theoretical Background	13
2.1. Biot-Savart Law	13
2.2. Faraday’s Law	14
3. Localization System	15
3.1. Localization System Components	15
3.2. Localization Principle	16
4. Simulation Tool	18
5. Disturbing Influences	19
5.1. Internal Influences	19
5.2. External Influences	19
6. 1-D Localization	21
6.1. Goal Detection	21
6.2. Influences on Goal Detection Accuracy	23
6.2.1. Number of Sensors	23
6.2.2. Sensor Position	26
6.2.3. Orientation of Sensor	28
6.3. CRLB on Goal Accuracy	34
6.3.1. Linear Approximation for the Induced Voltage Curves	34
6.3.2. CRLB Calculations	40
6.3.3. Noise Level Effect	42
6.3.4. Sensors Constellation Effect	43
7. 2-D Localization	46
7.1. 2-D Localization Method	46
7.2. Influences on 2-D Localization	49
7.2.1. Number of the Used Sensors	49
7.2.2. Noise Level Effect on the 2-D Localization Accuracy	51
7.2.3. Orientation of the Used Sensors	54
7.2.4. Rotation of Object Coils	56
8. Summary	59
Appendices	60
A. Magnetic Sensors Types	61

B. Algorithm 1	61
C. Algorithm 2	62
D. Algorithm 3	62
E. Plotting Figure 6.6a	62
F. Plotting Figure 6.11	63
G. Plotting Figure 6.30a	65

List of Figures

1.1. Magnetic field localization system components	11
2.1. Biot-Savart law illustration [4]	13
2.2. Faraday's law illustration	14
3.1. Localization system block diagram	15
3.2. Coils' naming illustration	16
3.3. Localization environment's elements	16
4.1. Simulation tool components illustration	18
6.1. 1-D localization setup	21
6.2. (a) Ball at different z -positions (b) The sensor's measured voltage	22
6.3. Induced voltage curves	23
6.4. PDF for a Gaussian distribution with $\mu = 0$ V and $\sigma^2 = 1$ V ²	24
6.5. Goal detection using 1 sensor with $\sigma_{noise} = 10^{-6}$ V	25
6.6. Goal detection using 3 sensors with $\sigma_{noise} = 10^{-6}$ V	25
6.7. The simulated setup for a 2 cm shifted magnetic sensor	26
6.8. Goal detection using 1 sensor shifted by 2 cm from the exciter wire	27
6.9. Tilting of the magnetic sensors with an angle θ around the x -axis	28
6.10. The simulated setup for 3 tilted sensors.	28
6.11. Error distribution over the goal plane for 1° tilted sensors without noise	29
6.12. Error distribution over the goal plane for 3° tilted sensors without noise	30
6.13. Noise effect on the simulated induced voltages for different (x,y) -points	31
6.14. Error distribution over the goal plane for 1° tilted sensors with WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V noise	32
6.15. Error distribution over the goal plane for 3° tilted sensors with WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V noise	32
6.16. Linear behavior illustration	34
6.17. Illustration of the restricting factors for linear approximation	35
6.18. The configured setup using four magnetic sensors	36
6.19. The simulated (x,y) -grid	37
6.20. <i>error</i> distribution for sensor 1 and (x,y) -point number 2	37
6.21. <i>error</i> versus <i>int</i> for sensor 1 and (x,y) -point number 2	38
6.22. <i>error</i> distribution for sensor 2 and (x,y) -point number 27	38
6.23. <i>error</i> versus <i>int</i> for sensor 2 and (x,y) -point number 27	38
6.24. Magnetic sensors distribution and the chosen (x,y) -points	40
6.25. Noise-level's effect on the CRLB	42
6.26. CRLB distribution using only 1 sensor	43
6.27. CRLB distribution using 3 sensors	44
6.28. CRLB distribution using 2 sensors	44

6.29. CRLB distribution using 4 sensors	45
6.30. CRLB distribution using 10 sensors	45
7.1. 2-D localization setup	46
7.2. The simulated induced voltage distribution for the setup in Figure 7.1	47
7.3. Uncertainty in 2-D localizing of an object	48
7.4. Output of Algorithm 3 using 2 sensors for 1000 MC simulations	50
7.5. Output of Algorithm 3 using 4 sensors for 1000 MC simulations	51
7.6. Output of Algorithm 3 using 10 sensors for 1000 MC simulations	51
7.7. Illustration of $error_{xy}$ in (7.2)	52
7.8. Noise level effect on the $stdError_{xy}$ and $avgError_{xy}$ for an object positioned at $(0.93m, 0.6m)$	53
7.9. Noise level effect on the $stdError_{xy}$ and $avgError_{xy}$ for an object positioned at $(0.05m, 0.05m)$	53
7.10. Tilting of sensors anti-clockwise around the x -axis with an angle θ	54
7.11. The resulting $error_{xy}$ for an object positioned at $(x_0 = 0.05m, y_0 = 0.05m)$	54
7.12. The resulting $error_{xy}$ for an object positioned at $(x_0 = 0.93m, y_0 = 0.6m)$	55
7.13. The resulting $error_{xy}$ for an object positioned at $(x_0 = 1.85m, y_0 = 0.6m)$	55
7.14. The configured setup and the investigated (x, y) -points	56
7.15. Object coils rotation illustration	57

List of Tables

6.1.	The simulated (x,y) -grid and the z -sweep for different θ° values	29
6.2.	Summary of the observed error range for $\theta = 1^\circ, 3^\circ$ tilted sensors	33
6.3.	The simulated z -sweep	36
6.4.	Extracted measured slopes for $\Delta z = 0.001$ m and $int = 0.02$ m	39
6.5.	The investigated (x,y) -points	42
6.6.	The investigated (x,y) -grid	43
7.1.	The simulated 2-D Localization reference table form	47
7.2.	Error due to the rotation of the object with angle ϕ	58

1. Introduction

In recent years the number of applications that demand an object to be localized has increased dramatically. Ranging from smart homes objects to industrial and medical applications. Different localization methods exist such as the global positioning system and the cellular hand-sets. However, these methods have some disadvantages such as low localization accuracy, calculations complexity and long processing time [1]. On the contrary, low frequency magnetic field localization methods have higher accuracy, higher penetration ability to different materials and lower calculations complexity [1].

Nowadays, there are various applications for magnetic field based localization. Such as finding a robot position in a warehouse or a factory in which magnetic field sensors can be mounted on the ceiling or buried underground [1]. The robot position can be estimated by analyzing the magnetic coupling model between the sensors and the embeded coils inside the robot.

In medicine, it is used to track the position of a capsule endoscope within the gastrointestinal tract of the human body [2]. The capsule is equibed with a magnet and the postion of the capsule can be tracked by a magnetic sensor network that measures the flux from this magnet. In sports, magnetic localization is used to provide a clear indication to the referee whether the ball has fully crossed the goal line during football matches [3]. Loop coils are embedded inside the ball and coupled with the magnetic field produced by an AC current source. The goal can be detected by running real time algorithms on the measured data from different magnetic sensors mounted around the goal frame. Each one of the aforementioned applications requires a certain localization accuracy level but all of them do suffer from several disturbing influences.

Figure 1.1 illustrates the structure and components of a magnetic localization system.

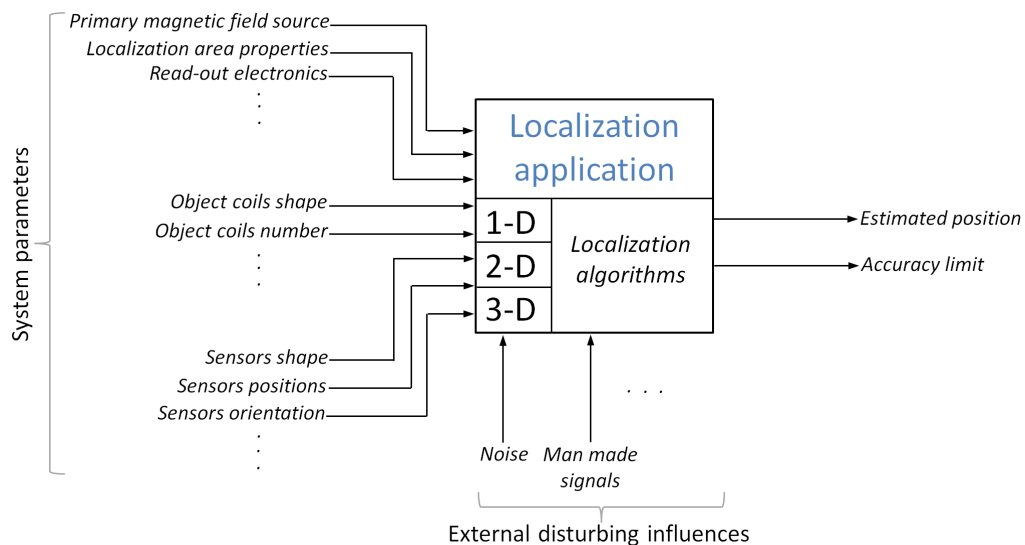


Figure 1.1.: Magnetic field localization system components

Magnetic field based localization systems consist of several components. Although each one of these components is necessary to perform a successful localization, they could limit the system localization accuracy with different amounts. Additionally, the localization process is exposed to different environments which means that external factors must be considered and analyzed to limit their effects on the system accuracy.

As can be seen in Figure 1.1, different applications share the same system parameters and they can be disturbed by several external influences such as man made signals or noise. For each application a different localization algorithm is used to analyze the sensed data and to estimate a position for the object.

In this work I am investigating different disturbing influences in order to qualify and quantify their effects on the localization accuracy. This work will be based on magnetic localization of a passive object. The outcomes of this thesis will help to build a better understanding of how a certain influence affects the accuracy level of a given application. Furthermore, such outcomes can help in defining a theoretical accuracy limit given a specific sensors constellation and a set of other parameters values.

The thesis is structured as following, in the next chapter the necessary theoretical background for magnetic localization will be provided. Afterwards, I will explain how the magnetic localization works in principle. Then, I will give an overview about the simulation tool which will be used to configure and simulate different localization setups. The first application to be investigated will be about the 1-D localization which is used for goal detection in football. Different parameters will be investigated and an accuracy limit will be defined using Cramer Rao lower bound calculations. Then, I will proceed to 2-D localization in which the noise effect on the localization accuracy will be investigated using the means of Monte Carlo simulations.

2. Theoretical Background

In this chapter, the necessary theoretical background, which forms the foundation of magnetic field based localization, is presented. Each of the following sections includes a brief descriptive overview which is necessary to understand how the localization is done. Additionally, this will be used to perform the upcoming simulations during the next chapters.

2.1. Biot-Savart Law

According to [4], the magnetic field density of a steady line current is given by the Biot-Savart law:

$$\vec{B}(r) = \frac{\mu_0}{4\pi} \int \frac{I \times \vec{r}}{r^2} d\vec{l} = \frac{\mu_0}{4\pi} I \int \frac{d\vec{l} \times \vec{r}}{r^2} \quad (2.1)$$

Where B is the magnetic flux density measured in tesla $[T]$, the integration is along the direction of the current flow, $d\vec{l}$ is an element of length along the wire, \vec{r} is the vector from the element of length to the point r (Figure 2.1), I is the current in Amperes and $\mu_0 = 4\pi \times 10^{-7} \frac{V.s}{A.m}$ is the permeability of free space.

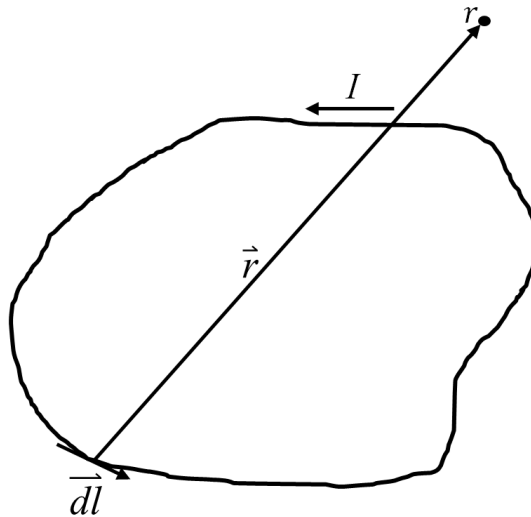


Figure 2.1.: Biot-Savart law illustration [4]

and the vector potential \vec{A} produced by this steady current at point r

$$A(\vec{r}) = \frac{\mu_0}{4\pi} \oint_a \frac{1}{r} d\vec{l} \quad (2.2)$$

2.2. Faraday's Law

"If the magnetic flux through a coiled conductor changes, a voltage proportional to the rate of change of the flux is generated between its leads" ([7], p. 632), and according to [4] this is described by Faraday's law in its integral form:

$$u_i = \oint \vec{E} \cdot d\vec{l} = -\frac{d\Phi(t)}{dt} \quad (2.3)$$

where u_i is the induced voltage in the conductor loop, \vec{E} is the electric field and Φ is the magnetic flux. For a magnetic field $\vec{B}(t)$ passing perpendicular a circular area (Figure 2.2) if

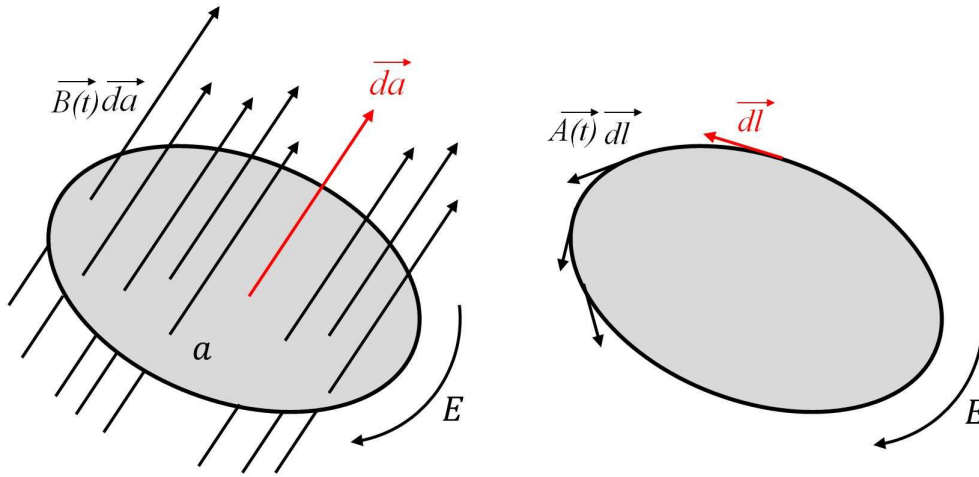


Figure 2.2.: Faraday's law illustration

B is increasing, E runs *clockwise*, and the magnetic flux through that area can be calculated by

$$\Phi = \int_a \vec{B} \cdot d\vec{a} \quad (2.4)$$

and from (2.3), the induced voltage in the loop will be

$$u_i = -\frac{\partial}{\partial t} \int_a \vec{B} \cdot d\vec{a} \quad (2.5)$$

The magnetic field density is related to the vector potential by

$$\int_a \vec{B} \cdot d\vec{a} = \oint_{\partial a} \vec{A} \cdot d\vec{l} \quad (2.6)$$

then from (2.4), Φ can be written as

$$\Phi = \oint \vec{A} \cdot d\vec{l} \quad (2.7)$$

Finally, the induced voltage is

$$u_i = -\frac{\partial}{\partial t} \oint \vec{A} \cdot d\vec{l} \quad (2.8)$$

3. Localization System

In this chapter, an overview is given about the used magnetic field based localization system. Also, a brief description about its components and their functions will be provided.

3.1. Localization System Components

As shown in Figure 3.1 , the localization system consists of four main blocks :

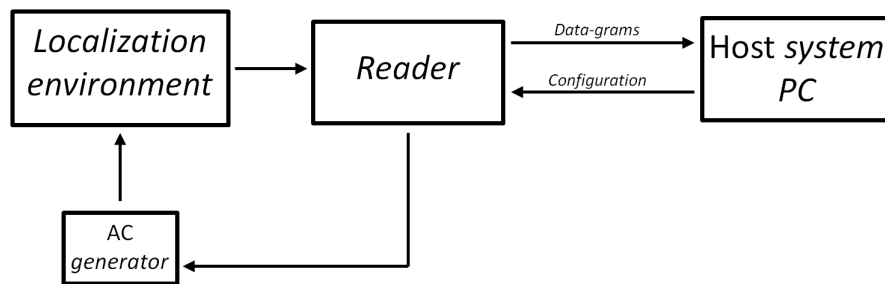


Figure 3.1.: Localization system block diagram

1. Localization environment: it consists of the following elements

- Primary magnetic field source (AC current in an exciter wire that surrounds the localization area)
- Localization object (with one or more coils embedded, each coil made of n-windings, each coil is named after the orientation of it's normal vector as illustrated in Figure 3.2)
- Magnetic field sensors (rectangular loops made of N-windings)
- Localization area's medium (air with permeability of μ_0)

Figure 3.3 illustrates one possibility of how these elements can be setup together.

2. **AC current generator:** it generates an AC current into the exciter wire that produces the primary magnetic field necessary for localization. The magnitude of the current can be adjusted by the host system through the reader.
3. **Reader:** it converts the analog measured voltages from the magnetic sensors into digital data-grams and send them to the host system through an ethernet link. The reader's sampling rate and other parameters can be adjusted from the host system via configuration commands.

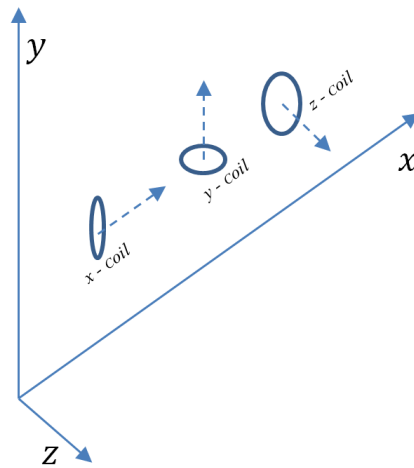


Figure 3.2.: Coils' naming illustration

4. **Host system PC:** it contains the necessary software modules written in Python which perform further processing steps on the received data from the reader. That makes them easier to be used for different applications and for visualization purposes. The host system can send configuration commands to the reader and the AC current generator.

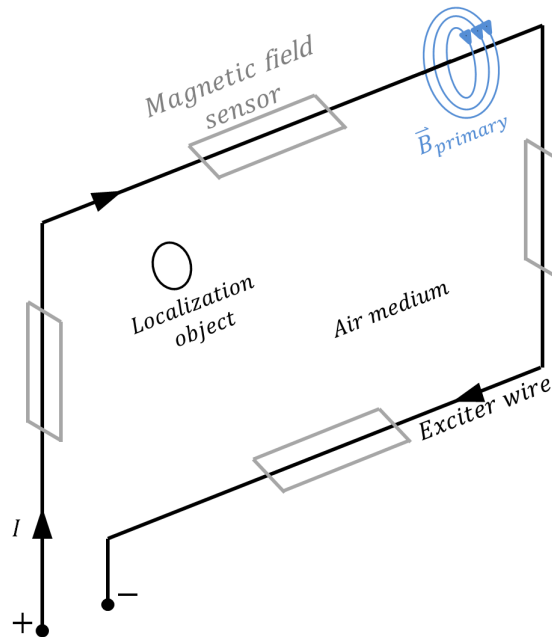


Figure 3.3.: Localization environment's elements

3.2. Localization Principle

Although The localization system can be configured to suit different applications, all of them share the same localization principle. The localization principle can be described as following:

- The flow of the current into the exciter wire is associated with a primary time-variant magnetic field
- Primary magnetic flux passes through the localization object's coil(s)' area
- According to 2.3 and 2.5, voltage is induced and current flows in the object's coil(s)
- The object's current flow is associated with a secondary time-variant magnetic field
- The secondary magnetic flux passes through the magnetic sensor(s)'s area
- According to 2.3 and 2.5, voltage is induced in each sensor
- A localization algorithm is then applied to compare the measured induced voltage(s) in each sensor with reference values for localization purposes
- The object's location is estimated

5. Disturbing Influences

The localization system under investigation suffers from several disturbing influences. Such influences will not allow a 100 % accurate localization. These influences can be either internal or external. The internal ones come from the system's imperfections. The external ones arise from the surrounding environment.

In this chapter, some of these disturbing influences are listed and a brief description is provided.

5.1. Internal Influences

Some of the internal influences are

- Magnitude of the AC current in the exciter wire: affects directly the magnitude of the primary magnetic field density and the induced voltage in the object coils.
- Shape and position of the exciter wire: affect the localization area dimensions in which the localization is being carried out.
- Shape and orientation of the object's coil(s): affect the amount and direction of the secondary magnetic flux.
- Number of windings of the Object's coil(s): affects the amount of the secondary magnetic flux.
- Shape and orientation of the magnetic field sensors: affect the magnitude of the induced voltage in the sensors.
- Constellation of the magnetic field sensors: affect the localization accuracy.
- Thermal noise from the voltage read-out electronics: affects the localization accuracy.

5.2. External Influences

- Geometrical deformations of the localization object: affect the amount, direction of the secondary magnetic flux and the object's coil(s) frequency response.
- Displacement of the sensors or the exciter wire due to external forces: affect the localization area dimensions in which the localization is being carried out.
- Additive white noise from near-by operating electronic devices: affects the localization accuracy.

- Presence of other objects with different magnetic permeability factors near the localization's area: affects the observed magnetic field density by the object.
- Magnetic fields from near-by AC current cables: affect the primary and the secondary magnetic fields that are used for localization.

6. 1-D Localization

One important application of magnetic field localization is to estimate a 1-D position of an object. Such an application can be simplified to detect whether an object has passed a specific position or not. For example, it can be used during sport events to detect goals accurately and eliminate false goal decisions. GoalRef is a magnetic field based localization system which was developed at Fraunhofer IIS for professional football leagues. It performs 1-D localization to detect goal events.

In this chapter, the goal detection process will be presented, and the selected disturbing influences of chapter 5 will be investigated.

6.1. Goal Detection

Let us consider the setup in Figure 6.1 which consists of the following:

- One magnetic field sensor in the xz -plane.
- An exciter wire passing through the middle of the magnetic sensor's loop and carrying an AC current.
- Ball object with only one embedded z -coil.

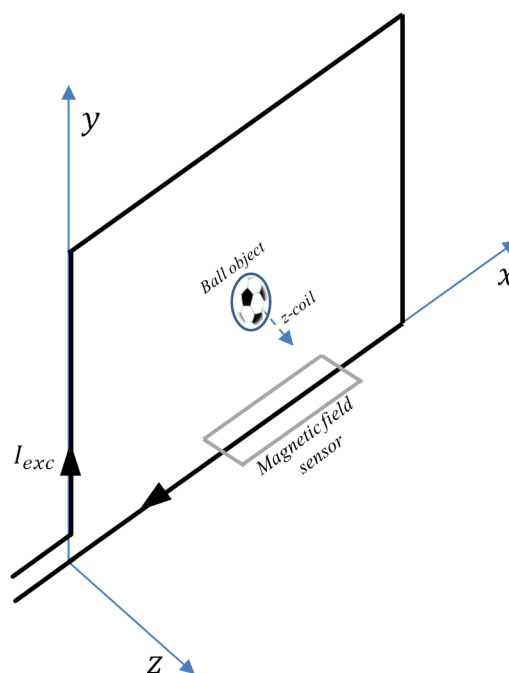


Figure 6.1.: 1-D localization setup

When the ball object starts approaching the localization setup from the negative z -direction towards the positive z -direction, the following occurs:

- Primary magnetic flux passes through the ball coil area.
- Voltage is induced in the ball coil and current flow in the coil is established.
- The current develops a secondary magnetic field.
- The secondary magnetic flux passes through the sensor area.
- Voltage is induced in the sensor.

Before ball position 1 in Figure 6.2a, the induced voltage in the sensors' coil starts increasing as the ball gets closer to xy -plane till it reaches a maximum value exactly at position number 1 (Figure 6.2b). At position number 2 in Figure 6.2a, no magnetic flux passes through the sensor area and $U_0 = 0V$.

Afterwards, the ball moves towards position 3 and the induced voltage starts reversing its polarity till it reaches a negative minimum value at position number 3 in Figure 6.2b.

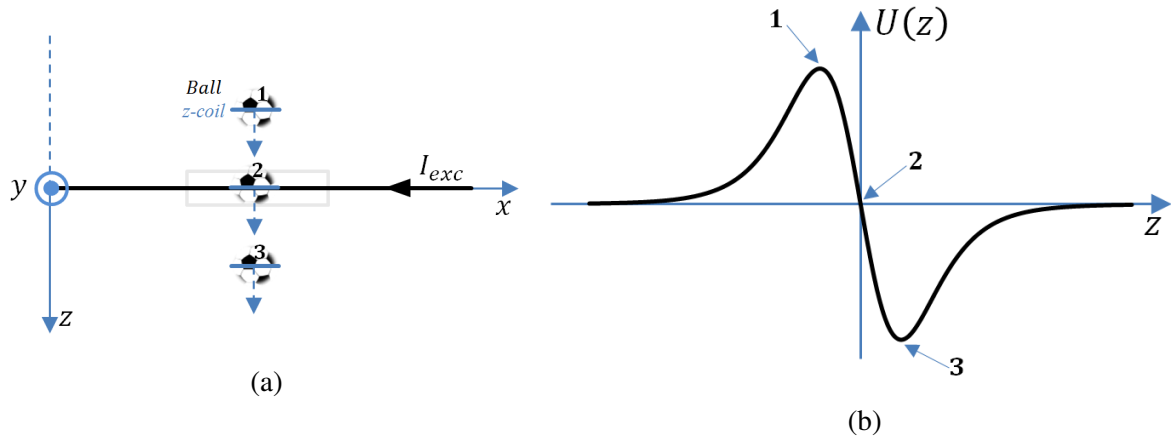


Figure 6.2.: (a) Ball at different z -positions (b) The sensor's measured voltage

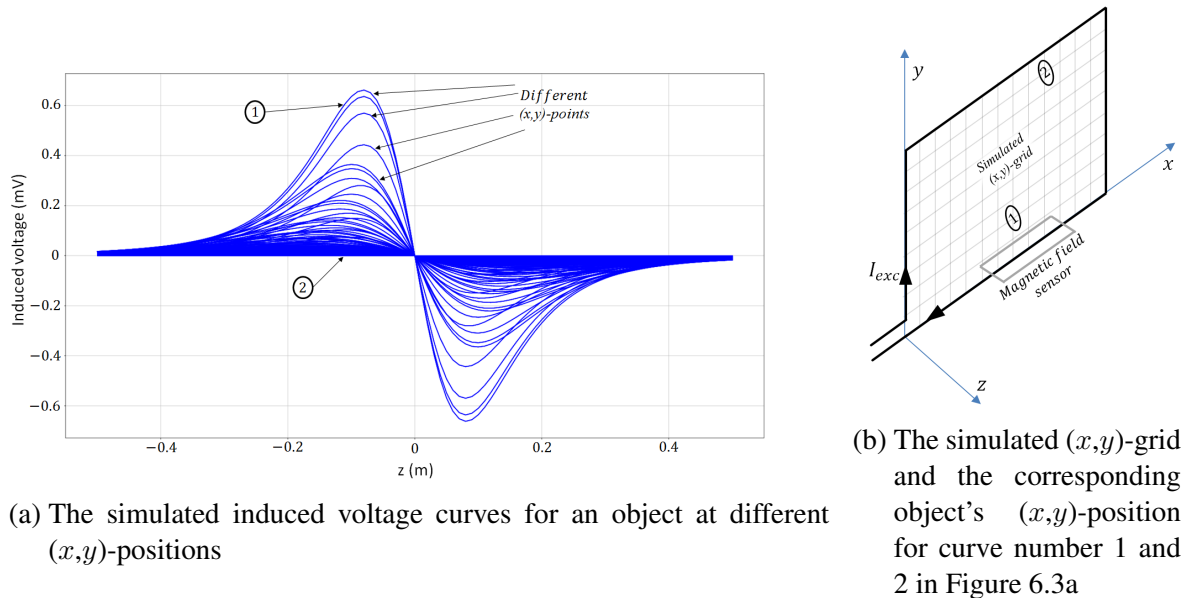
From Figure 6.1 and 6.2, the xy -plane is considered as the goal plane and the $z = 0$ m as the goal line. As a result a goal event can be detected and declared by tracking the induced voltage in the magnetic sensor until its polarity changes when crossing the zero V value. Algorithm 1 describes how a goal event is detected. It takes the sensor's induced voltage values as an input. Then a for loop runs over the length of the input values with a condition at line 4. This condition detects the voltage zero crossing by comparing the sign of two successive voltage values. If the sign of two successive values are different, a goal event is declared in line 5, and if not, the loop continue running.

Algorithm 1 Goal detection

- 1: **Input:** Induced voltage in the sensor U_{ind}
 - 2: **for** i **in range** $(0, \text{len}(U_{ind})-1)$ **do**
 - 3: buffer = $U_{ind}[i]$
 - 4: **if** $\text{sign}(\text{buffer}) \neq \text{sign}(U_{ind}[i + 1])$ **then**
 - 5: Declare a goal event
 - 6: **break;**
-

Using the simulation tool described in Chapter 4, the induced voltage curves of Figure 6.3a were simulated. Each curve represents the induced voltage in the sensor for a ball object placed at a certain (x,y) -point and moving along the z -axis.

For example, Figure 6.3b shows the object's positions in the xy -plane that induced the voltage curves number 1 and 2 in Figure 6.3a.



(a) The simulated induced voltage curves for an object at different (x,y) -positions

(b) The simulated (x,y) -grid and the corresponding object's (x,y) -position for curve number 1 and 2 in Figure 6.3a

Figure 6.3.: Induced voltage curves

6.2. Influences on Goal Detection Accuracy

During this section, the following influences are investigated:

- Constellation of the magnetic field sensors.
- Position of the exciter wire.
- Orientation of the sensors.
- Thermal noise effect.

6.2.1. Number of Sensors

I will investigate the influence of the number of sensors on the goal detection accuracy while considering the noise effects. The simulation tool is used to configure the used sensors' dimensions. Then, an object is simulated which moves along the z -axis for a grid of (x,y) positions. For each (x,y,z) position, an induced voltage value is calculated. Afterwards, the noise effect is simulated as following

$$U_{sim} \Big|_{x,y,z} = U_{ind} \Big|_{x,y,z} + w \quad (6.1)$$

Where U_{sim} is the simulated voltage value including noise for a certain object position, U_{ind} is the calculated induced voltage value using the simulation tool for a certain (x,y,z) -point, w is an independent random noise sample. During this thesis, White Gaussian Noise (WGN) is used to simulate the noise effect with the following Probability Density Function (PDF)

$$f(w; \mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(w - \mu)^2}{2\sigma^2} \quad (6.2)$$

Where μ is the mean of the random variable w and σ^2 is its variance. Figure 6.4 illustrates the PDF of (6.2) for $\mu = 0$ V and $\sigma^2 = 1$ V². The standard deviation σ value represents the uncertainty in the random variable w .

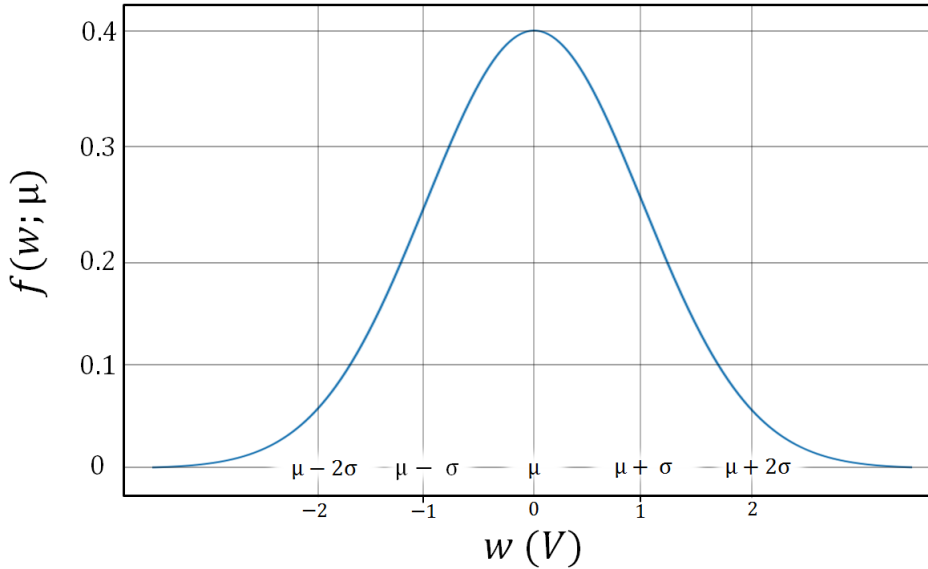


Figure 6.4.: PDF for a Gaussian distribution with $\mu = 0$ V and $\sigma^2 = 1$ V²

Algorithm 1 will be used to detect the zero crossings for goal events. It takes the U_{sim} voltage values as an input to consider the noise effect.

For the first simulation, one magnetic field sensor is used. The simulation setup parameters are shown in Figure 6.5b. The exciter wire passes exactly through the sensors' loop to get a theoretical voltage zero crossing at $z = 0$ m exactly. Using this fact, the error in the goal decision can be defined as following

$$error = z_{detected} - z_{theoretical} \quad (6.3)$$

where $error$ is measured in meters, $z_{detected}$ is the z -position of the ball at which a zero crossing where detected using Algorithm 1, $z_{theoretical}$ is the aforementioned theoretical position at $z = 0$ m where the zero crossing occurs.

The used magnetic sensor has a 50×4 cm² rectangular loop. WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V will be used for the noise samples w calculations. Figure 6.5a illustrates the resulting distribution of the induced voltage zero crossing positions over the xy -goal plane.

A 0.03 mm error of this setup means that the zero crossing and hence the goal decision occurred 0.03 mm after the object has passed the theoretical goal line at $z = 0$ m. The maximum error value is -0.99 mm which means that the goal decision occurred 0.99 mm before

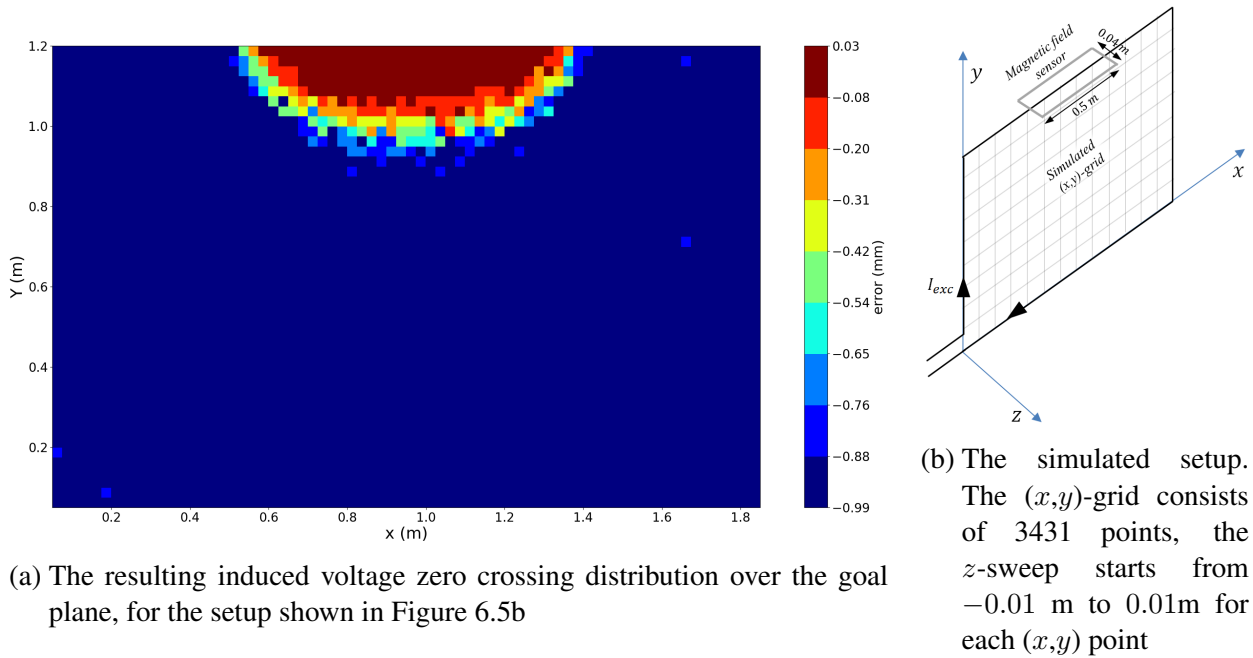


Figure 6.5.: Goal detection using 1 sensor with $\sigma_{noise} = 10^{-6}$ V

the object has passed the goal line.

For the next simulation, three identical magnetic field sensors will be used. Each with a 50×4 cm² rectangular loop. This means that the calculations in (6.1) will be performed three times, one time for each sensor. Figure 6.6b shows how the sensors are distributed around the goal plane. WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V is considered. The sum of the three sensors voltages is used to calculate the resulting zero crossings positions which are shown in Figure 6.6a. The resulting zero crossing distribution due to the use of three sensors

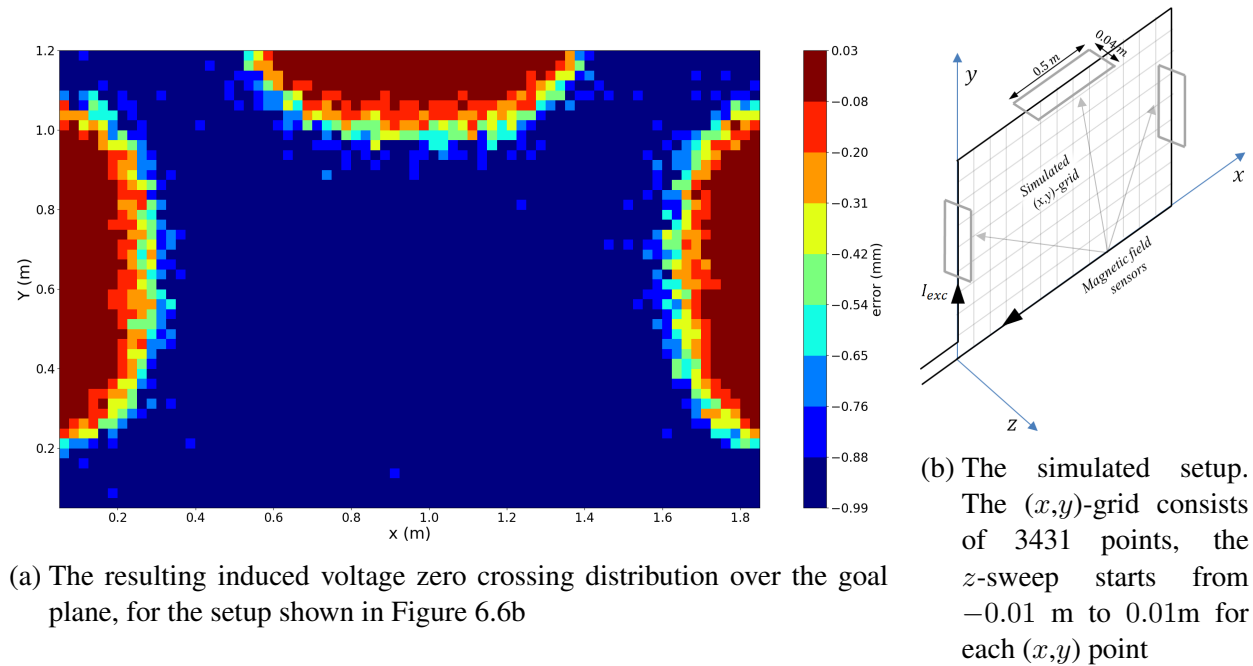


Figure 6.6.: Goal detection using 3 sensors with $\sigma_{noise} = 10^{-6}$ V

has a maximum error value of -0.99 mm, which is the same as the case of using only one

magnetic field sensor.

As a conclusion, the (x,y) -points in the middle of the goal are heavily affected by the added noise. That resulted in an early zero crossing because the induced voltage values were comparable to the noise level of 10^{-6} V. And increasing the number of magnetic sensors did not reduce the maximum resulting error value for a fixed noise level. But it increased the areas which have a lower error values near the sensors positions.

6.2.2. Sensor Position

In this section, I am investigating the influence of the position of the exciter wire on the goal detection accuracy. The exciter wire will not pass exactly through the middle of the magnetic sensor. The simulation tool will be used to create a ball object with three orthogonal embedded coils. One magnetic sensor will be positioned at the middle bottom of the goal plane and shifted with 2 cm towards the positive z -direction.

Figure 6.7 shows the simulated setup with the same (x,y) -grid as in Section 6.2.1 and a sweep of the ball's z -positions starting from -0.05 m to 0.05 m for each (x,y) point.

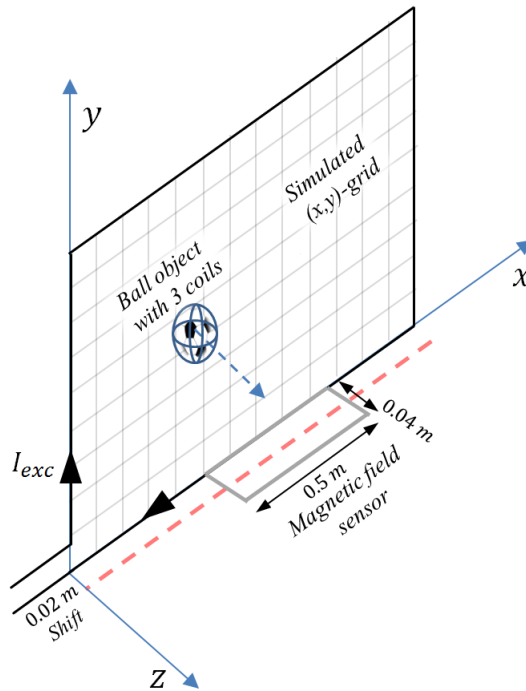
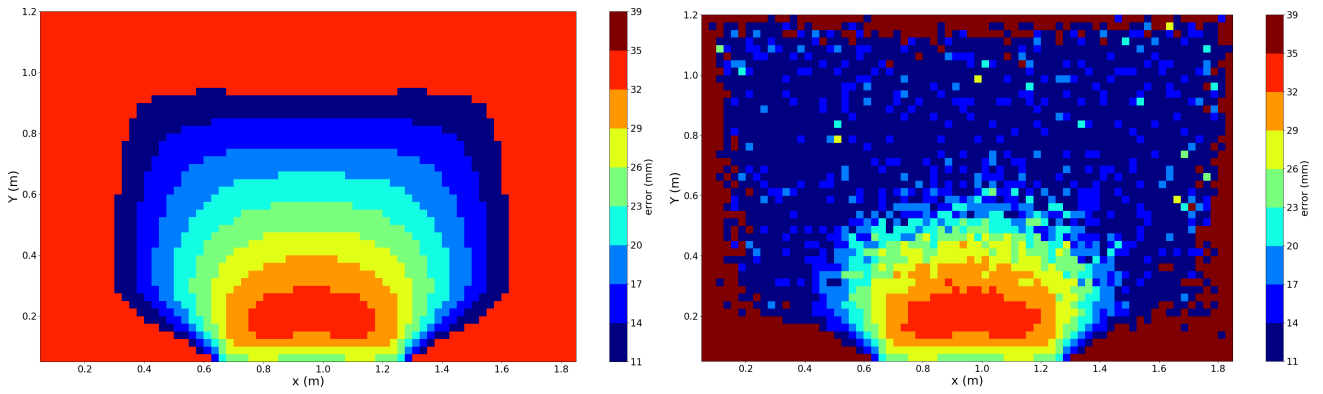


Figure 6.7.: The simulated setup for a 2 cm shifted magnetic sensor

I have used Algorithm 1 to extract the zero crossings for the induced voltages of two different cases. The first case is without considering the noise effect. The resulting zero crossing distribution is shown in Figure 6.8a. In the second case WGN samples of $\mu = 0$ V and $\sigma = 10^{-6}$ V were added to the induced voltage values and the corresponding zero crossing is shown in Figure 6.8b.

By comparing both figures, the maximum error value has reached 39 mm in Figure 6.8b instead of 33 mm in Figure 6.8a. But the presence of the noise has resulted in less error values for the middle and the upper (x,y) -points. The added noise caused several (x,y) -points to have an early voltage zero crossings.



- (a) The resulting induced voltage zero crossing distribution over the goal plane, for the setup shown in Figure 6.7 without considering any noise effect
- (b) The resulting induced voltage zero crossing distribution over the goal plane, for the setup shown in Figure 6.7 with WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V

Figure 6.8.: Goal detection using 1 sensor shifted by 2 cm from the exciter wire

The error's average for both cases has been calculated by summing all the error values together then dividing the sum by the number of (x,y) -points. 26.4 mm is the error's average without considering the noise effect. While 20.5 mm was the error's average with considering the noise effect.

It can be concluded that, although the maximum error value due to the presence of noise was higher, the error's average was lower than the case without noise. The presence of the noise has compensated for the shift in the magnetic sensor's position.

6.2.3. Orientation of Sensor

During this section, the influence of the magnetic sensors' orientation on the goal detection accuracy will be investigated. I will use the simulation tool to configure three identical and tilted magnetic sensors and a ball object with three orthogonal coils. Figure 6.9 illustrates how the sensors are being tilted. The tilting is done by an angle θ around the $-x$ -axis in the anti-clock wise direction.

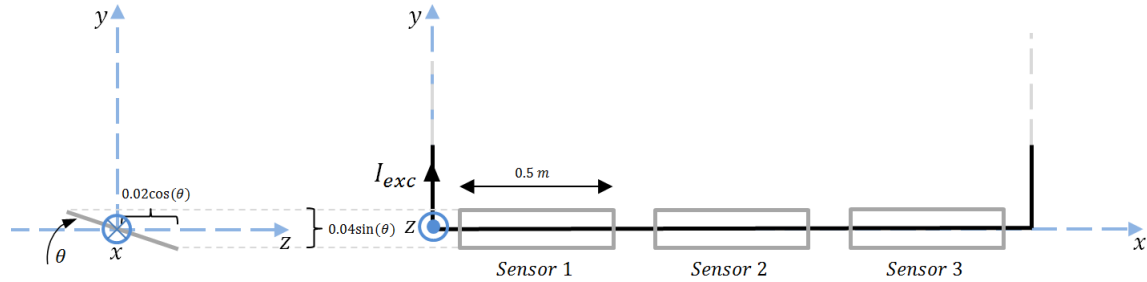


Figure 6.9.: Tilting of the magnetic sensors with an angle θ around the x -axis

I will use $\theta = 1^\circ$ and $\theta = 3^\circ$ during this investigation. I have chosen small θ values because it is hard to notice such small titling angles during a real life situation. The simulation tool will be configured two times, one time for each θ value.

The induced voltage values per each sensor will be simulated. Then Algorithm 1 will be used two times to extract the voltage zero crossing for each simulated (x,y,z) -position. Figure 6.10 shows the simulated setup. The simulated (x,y) -grid and the z -sweep are defined in Table

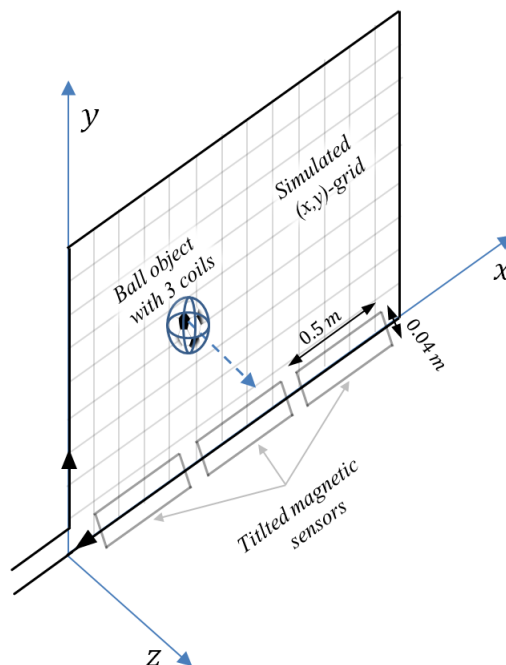


Figure 6.10.: The simulated setup for 3 tilted sensors.

6.1 for both θ values. I am running this investigation one time without considering any noise effect. Then another time with considering a WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V.

θ°	x (m)	y (m)	z (m)
1	[0.05, 1.85], step \approx 0.03	[0.16, 1.22], step \approx 0.02	[-0.01, 0.01], step = 0.001
3			[-0.03, 0.03], step = 0.001

Table 6.1.: The simulated (x,y) -grid and the z -sweep for different θ° values

Tilting by $\theta = 1^\circ$ without noise

The resulting error distribution over the goal plane using 1° tilted sensors is shown in Figure 6.11.

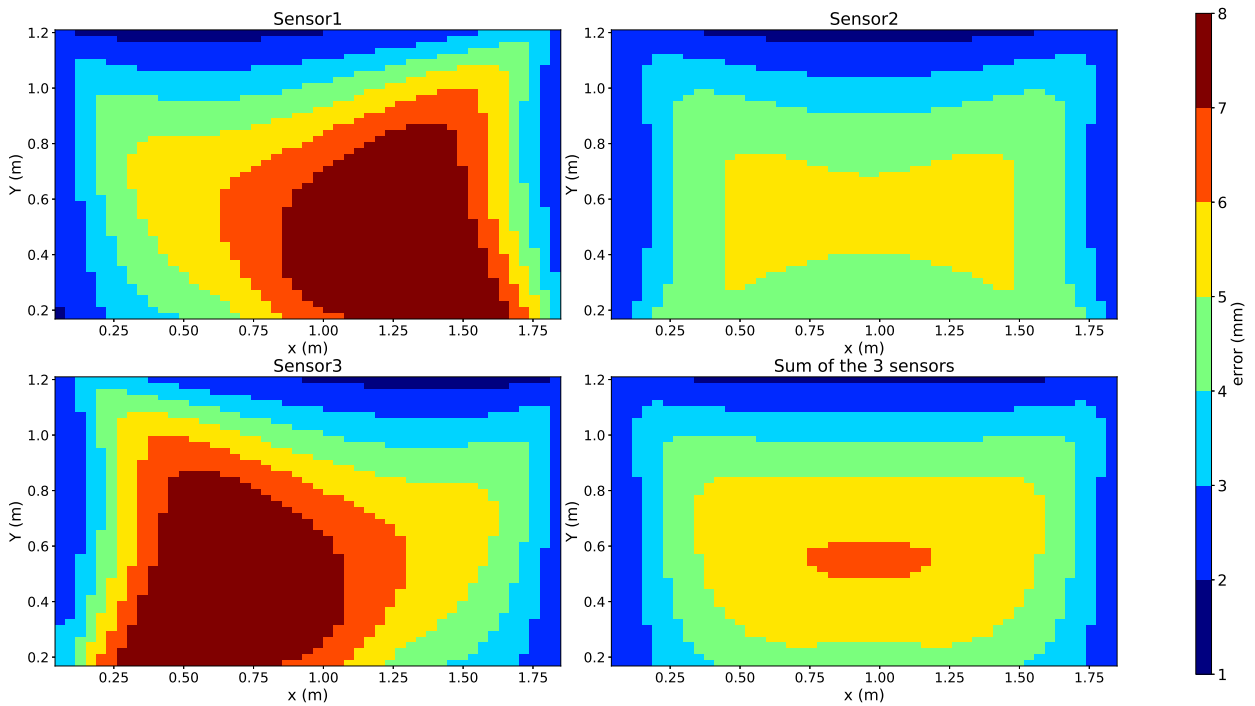


Figure 6.11.: Error distribution over the goal plane for 1° tilted sensors without noise

For sensor 1 and sensor 3, the error ranges from 1 to 8 mm. This means the goal decision was made 1 to 8 mm after the object's has passed the $z = 0$ m theoretical goal line. By comparing the sensors' heat maps, sensor number 2 has the lowest error value of 6 mm. To generate the sum of the 3 sensors heat map, the induced voltage values for the three sensors were added together. Then Algorithm 1 was used to extract the zero crossings of the sum. The maximum error value for the sum is indicated by the red area in the middle of the goal plane with 7 mm.

Tilting by $\theta = 3^\circ$ without noise

The resulting error distribution over the goal plane for 3° tilted sensors is shown in Figure 6.12. The error range has increased in comparison with the 1° tilting. For sensor 1 and 3,

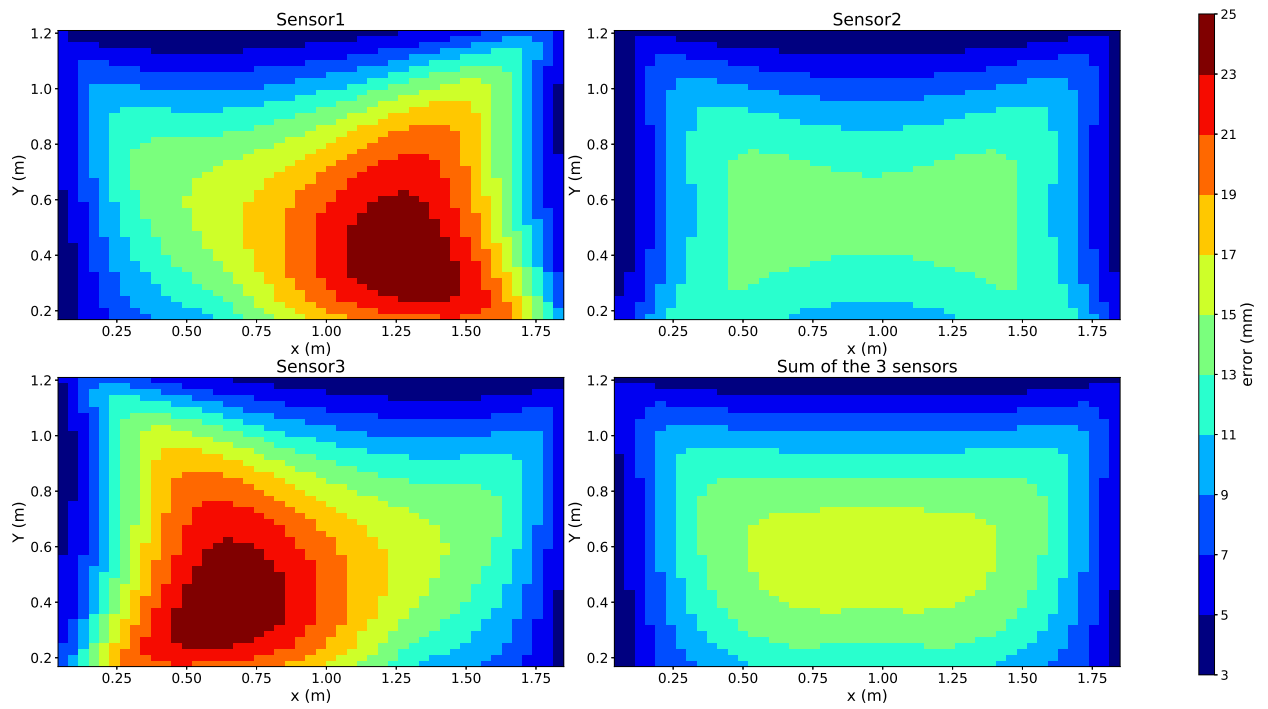


Figure 6.12.: Error distribution over the goal plane for 3° tilted sensors without noise

the error ranges from 3 to 25 mm. Among the three sensors, sensor number 2 has the lowest error value of 15 mm. And the maximum error value for the sum of the three sensors has reached 17 mm.

Tilting by $\theta = 1^\circ$ with noise

Figure 6.13 illustrates the effect of the WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V on the simulated induced voltages. Early voltage zero crossings occurred because the noise level was comparable with the voltage level of some (x,y) -points. For that, a threshold voltage value must be set in order to differentiate between the correct zero crossings and the false ones. I have chosen a threshold value of $10 \times \sigma_{noise}$ V.

A voltage zero crossing will be considered as a goal event indicator only if the peak of its corresponding voltage curve is 10 times higher than the noise level. The simulation tool

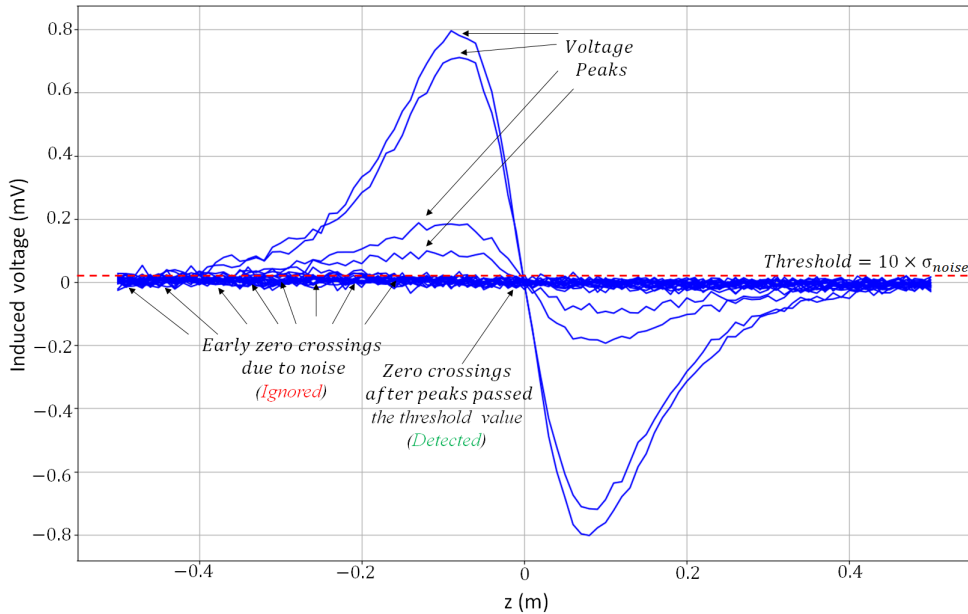


Figure 6.13.: Noise effect on the simulated induced voltages for different (x,y) -points

was used to get the induced voltage curves. Then noise elements were added to each curve. Algorithm 1 was used to extract the zero crossings while considering the threshold value. Figure 6.14 shows the resulting error distribution over the goal plane using 1° tilted sensors and considering the noise effect. The dashed areas in these heat maps represent the ignored (x,y) -points. Such points can not be considered as goal event indicators due to their relatively low induced voltage levels as compared to the noise level.

Tilting by $\theta = 3^\circ$ with noise

Figure 6.15 shows the resulting error distribution over the goal plane using 3° tilted sensors and considering the noise effect. The error range has increased in comparison with the 1° tilting. The heat map of the sum of the three sensors includes some points that reached -14 mm error value. This happened because the sum includes the three sensors' noise elements together.

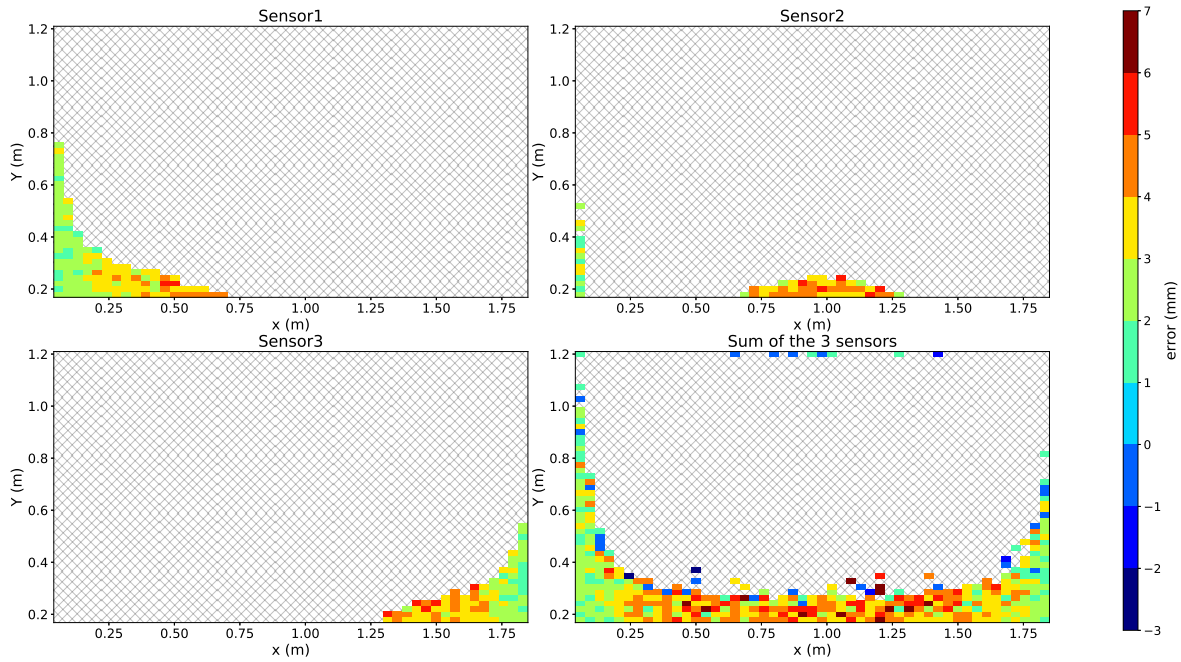


Figure 6.14.: Error distribution over the goal plane for 1° tilted sensors with WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V noise

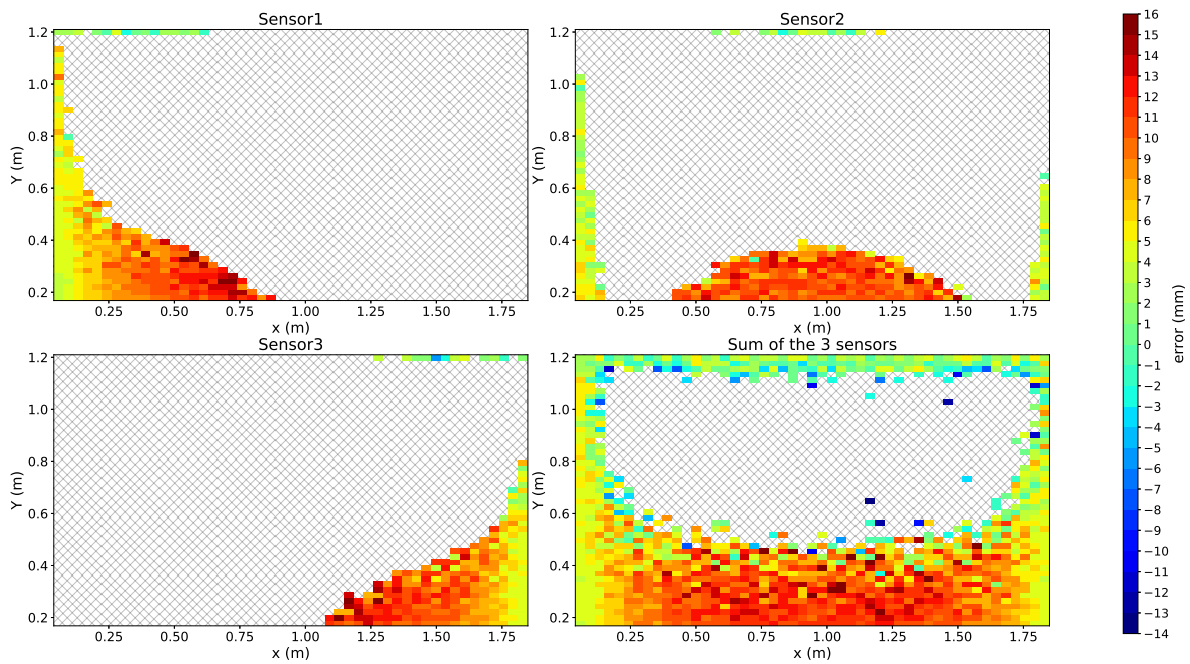


Figure 6.15.: Error distribution over the goal plane for 3° tilted sensors with WGN of $\mu = 0$ V and $\sigma = 10^{-6}$ V noise

From the outcomes of this investigation, it can be concluded that tilting of the magnetic sensors has resulted in an error distribution for detecting goal events. The error values depend on the tilting angle θ and the noise level. To wrap up, Table 6.2 includes a summary of all the error ranges for different setups.

	Err. range in (mm), without noise				Err. range in (mm) , with WGN of $(\mu = 0 , \sigma = 10^{-6})\text{V}$			
	Sen. 1	Sen. 2	Sen. 3	Sum of 3 sen.	Sen. 1	Sen. 2	Sen. 3	Sum of 3 sens.
1°	[1, 8]	[1, 6]	[1, 8]	[1, 7]	[1, 6]			[-3, 7]
3°	[3, 25]	[3, 15]	[3, 25]	[3, 17]	[-4, 16]			[-14, 16]

Table 6.2.: Summary of the observed error range for $\theta = 1^\circ, 3^\circ$ tilted sensors

6.3. CRLB on Goal Accuracy

In this section, I will perform Cramer Rao Lower Bound calculations to get the theoretical accuracy limit that can be achieved using localization system under investigation. To do that, I will start with simplifying the induced non-linear voltage curves into linear ones under some specific conditions. Afterwards, it will be much easier to deal with linear equations to investigate the different factors that affect the accuracy limit.

6.3.1. Linear Approximation for the Induced Voltage Curves

Figure 6.16 shows the induced voltage curves for different (x,y) -points. The curves are non linear but the parts inside the zoomed-in circle have a linear behavior. In this chapter I will exploit this feature and perform linear approximations for the induced voltage curves. The results of these approximations will be used during the next chapter.

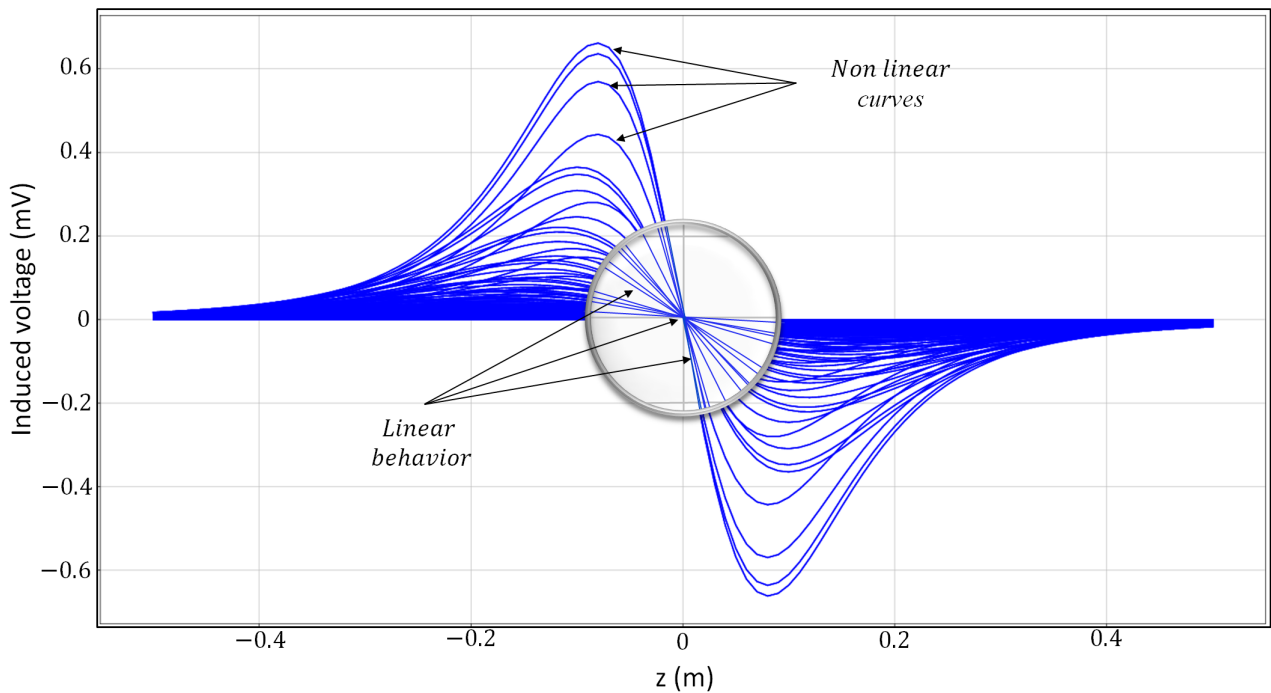


Figure 6.16.: Linear behavior illustration

The goal of this chapter is to find a function

$$U_{approx}(z) \Big|_{x,y,n} = s_{x,y,n}z \quad (6.4)$$

where $U_{approx}(z)$ is the approximated induced voltage in sensor number n for an object placed at a certain (x,y) -point, $s_{x,y,n}$ is the slope of the linearly approximated induced voltage in sensor number n for an object placed at a certain (x,y) -point, and z is the object's position along the z -axis.

The following factors affect the accuracy of a linear approximation

- *int*: the interval where the linear approximation is being carried out.

- Δz : the horizontal difference between two points that is used to calculate the slope.

Figure 6.17 illustrates both factors using different int and Δz values. The blue solid curve is the induced voltage to be approximated. Linear approx. 1 was created using point P_0 and P_1 . This approximation matches the solid curve during a Δz_1 period then it start deviating from it. This deviation represents an error value.

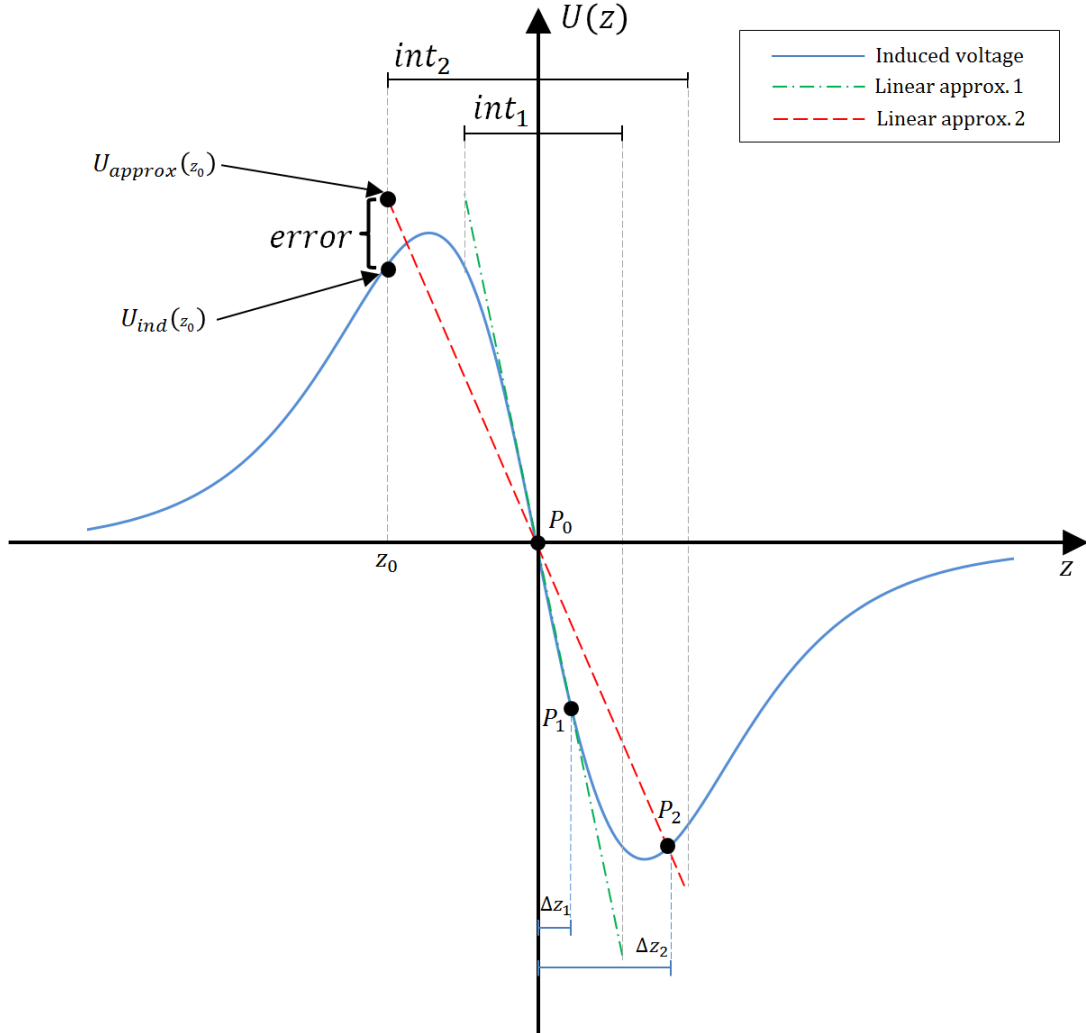


Figure 6.17.: Illustration of the restricting factors for linear approximation

For example, for an object positioned at $z = z_0$ m, the error can be expressed as

$$error(z_0) \Big|_{x,y,n} = U_{ind}(z_0) \Big|_{x,y,n} - U_{approx}(z_0) \Big|_{x,y,n} \quad (6.5)$$

where $error$ is measured in Volts. Then the total error can be calculated as

$$error_{tot} \Big|_{x,y,n} = \sum_{z=-int/2}^{z=int/2} \sqrt{(U_{ind}(z) - U_{approx}(z))^2} \quad (6.6)$$

where $error_{tot}$ is the total accumulated error value along the int between the induced voltage curve and the linearly approximated.

In Figure 6.17, for a fixed $int1$ value, Linear approx. 2 accumulates more error than Linear approx. 1. Because Δz_2 is bigger than Δz_1 which allowed for more deviation from the induced voltage solid curve. This resulted in a smaller slope value for linear approx. 2 than linear approx. 1. The smaller the Δz the higher the slope and vice versa.

Increasing the interval to $int2$ results in more accumulated error for both, linear approx. 1 and linear approx. 2.

I have used the simulation tool to configure the setup shown in Figure 6.18. This setup consists of four magnetic sensors and a ball object with three orthogonal coils. The simulated (x,y) -grid is shown in Figure 6.19 where the red points are the nearest to each sensor. And the object will be moved along the z -axis for each (x,y) -point and the induced voltage for each magnetic sensor will be simulated.

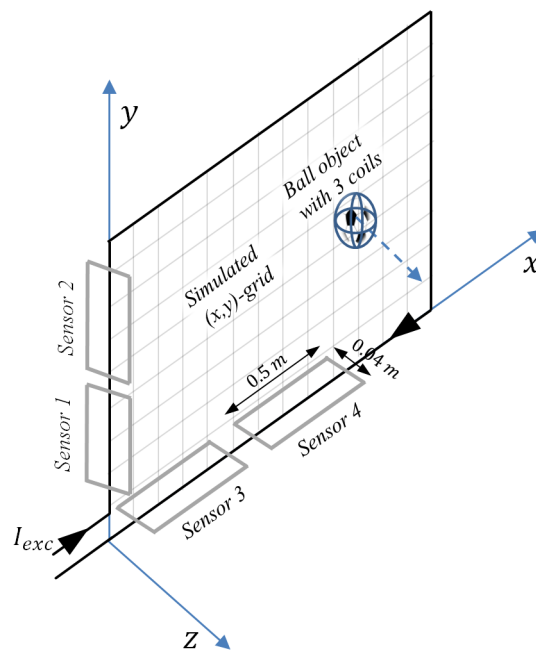


Figure 6.18.: The configured setup using four magnetic sensors

The simulated z -sweep is included in Table 6.3.

z -sweep (m)
$[-0.05, 0.05], \Delta z = 0.001$

Table 6.3.: The simulated z -sweep

For the sensor 1 and (x,y) -point number 2 in Figure 6.19, the $error(\Delta z)$ (calculated using (6.5)) distribution for different int values is shown in Figure 6.20. For small int values, the $error$ increases slowly with increasing the Δz . The highest $error$ which is indicated by the red area at the left upper corner of the heat map was a result of a small Δz and a high int of 0.05 m.

Figure 6.21 shows the $error$ versus the int value. The highest Δz value of 0.05 m has resulted in a maximum $error$ of around 1000 mV at an int of 0.05 m. And the lowest Δz

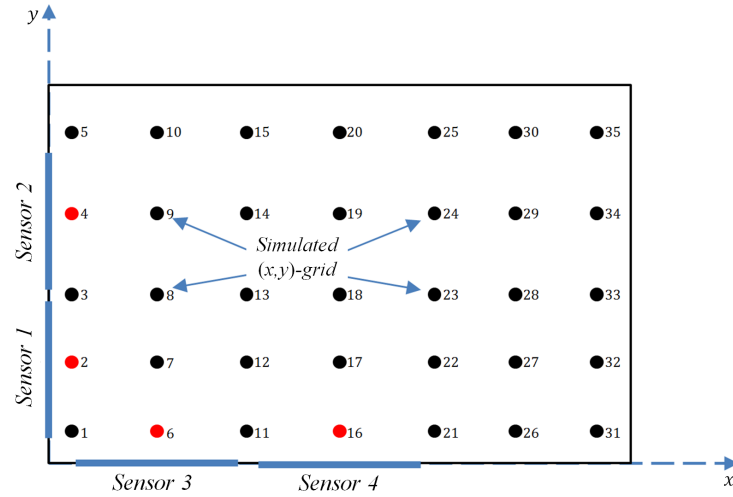


Figure 6.19.: The simulated (x,y) -grid

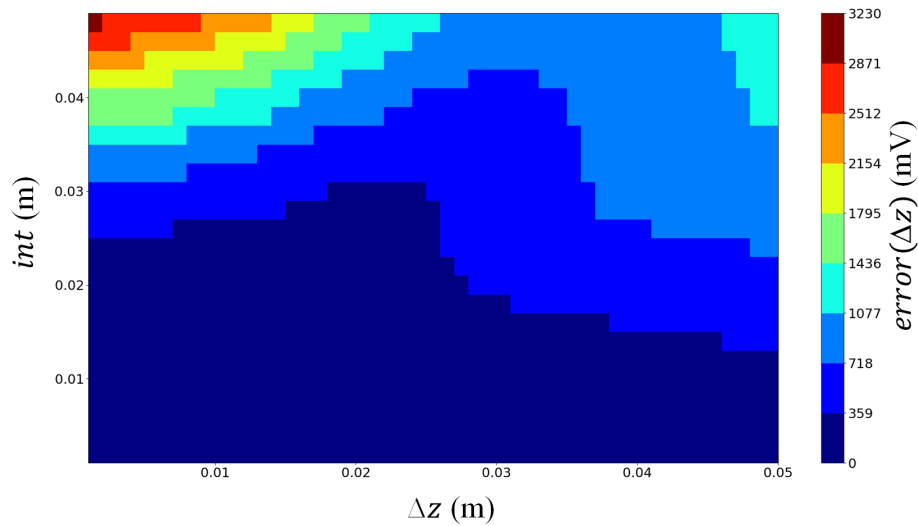


Figure 6.20.: *error* distribution for sensor 1 and (x,y) -point number 2

value of 0.001 m has resulted in a maximum *error* of around 3000 mV at an *int* of 0.05 m.

For the sensor 2 and (x,y) -point number 27 in Figure 6.19, the $error(\Delta z)$ (calculated using (6.5)) distribution for different *int* values is shown in Figure 6.22. The highest *error* which is indicated by the red area at the right upper corner of the heat map was a result of a Δz of 0.05 m and a high *int* of 0.05 m. The *error* range is much smaller than the results in Figure 6.20 because the (x,y) -point number 27 is not near the magnetic sensor.

Figure 6.23 shows the *error* versus the *int* value. The highest Δz value of 0.05 m has resulted in a maximum *error* of around 0.2 μV at an *int* of 0.05 m. And the lowest Δz value of 0.001 m has resulted in a maximum *error* of around 0.1 μV at an *int* of 0.05 m.

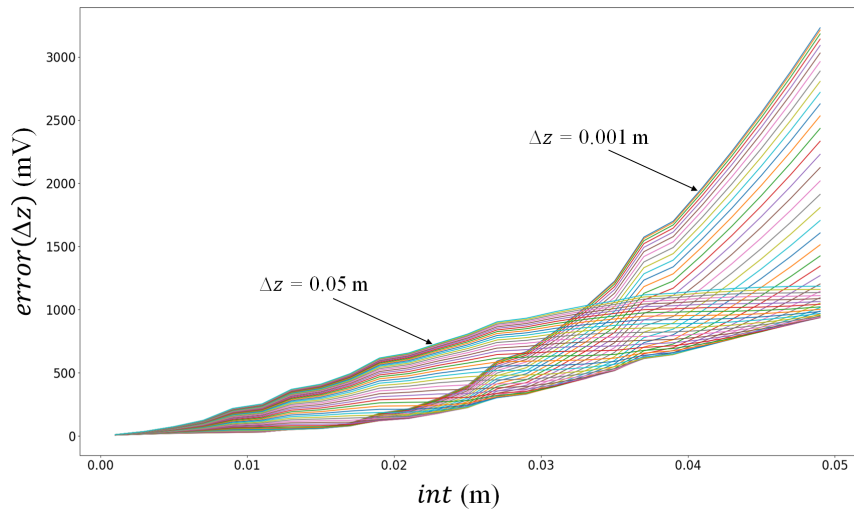


Figure 6.21.: *error* versus *int* for sensor 1 and (x,y) -point number 2

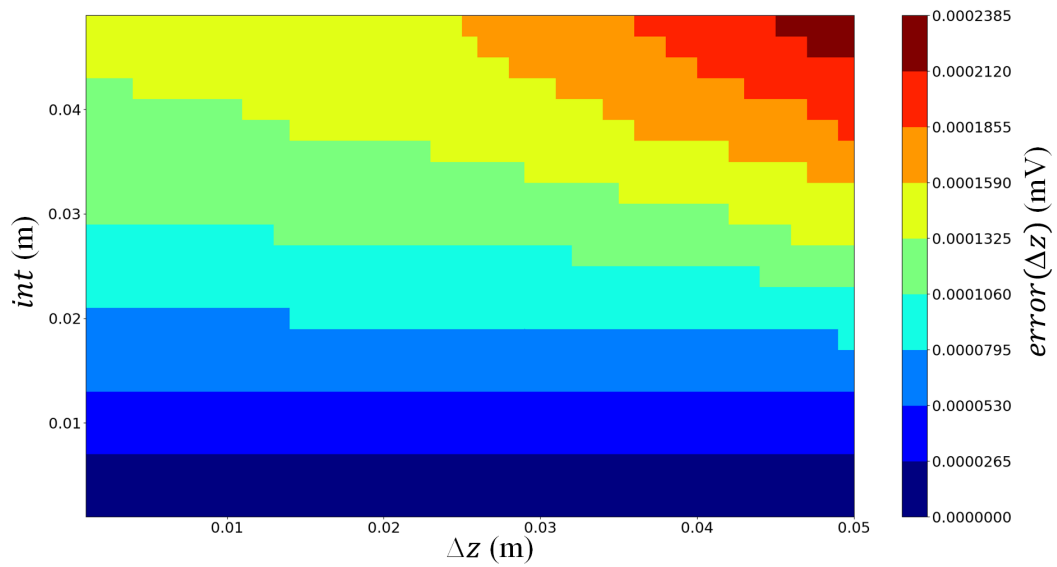


Figure 6.22.: *error* distribution for sensor 2 and (x,y) -point number 27

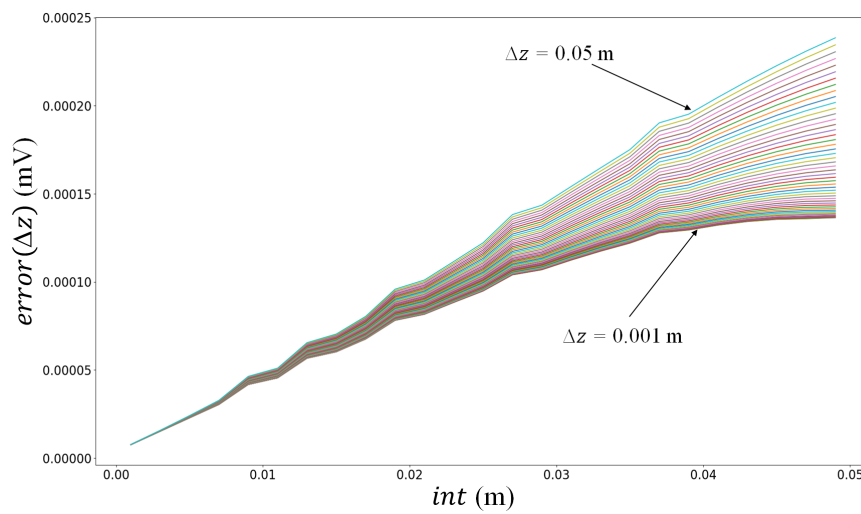


Figure 6.23.: *error* versus *int* for sensor 2 and (x,y) -point number 27

For goal detection, the theoretical goal line is fixed at $z = 0$ m. Therefore I have chosen an $int = 0.02$ m and $\Delta z = 0.001$ m for performing the linear approximations.

I have written a Python code to calculate the $s_{x,y,n}$ value for each linearly approximated induced voltage curve. Table 6.4 includes the calculated slopes for each of the 35 (x,y) -points in Figure 6.19.

(x,y) -point	Sensor 1	Sensor 2	Sensor 3	Sensor 4
1	-0.35	-2.97e-05	-0.35	-2.97e-05
2	-3.89	-2.32e-4	-3.49e-3	-8.16e-05
3	-0.21	-0.21	-5.05e-4	-6.46e-05
4	-2.29e-4	-3.85	-1.38e-4	-3.86e-05
5	-2.04e-05	-0.24	-5.74e-05	-2.44e-05
6	-3.48e-3	-8.15e-05	-3.89	-2.31e-4
7	-1.52e-3	-1.05e-4	-1.52e-3	-1.05e-4
8	-6.36e-4	-6.36e-4	-1.34e-4	-3.69e-05
9	-9.78e-05	-1.41e-3	-3.37e-05	-1.60e-05
10	-3.13e-05	-1.34e-3	-2.24e-05	-1.39e-05
11	-5.02e-4	-6.41e-05	-0.21	-0.21
12	-1.30e-4	-3.57e-05	-6.16e-4	-6.16e-4
13	-7.39e-05	-7.39e-05	-7.39e-05	-7.39e-05
14	-3.24e-05	-1.18e-4	-2.23e-05	-2.23e-05
15	-2.34e-05	-1.83e-4	-1.83e-05	-1.83e-05
16	-1.35e-4	-3.78e-05	-2.25e-4	-3.77
17	-2.91e-05	-1.38e-05	-8.43e-05	-1.21e-3
18	-1.90e-05	-1.90e-05	-2.76e-05	-1.00e-4
19	-1.24e-05	-2.62e-05	-1.24e-05	-2.62e-05
20	-1.37e-05	-4.90e-05	-1.27e-05	-2.05e-05
21	-5.01e-05	-2.13e-05	-1.78e-05	-0.21
22	-1.03e-05	-6.38e-06	-1.44e-05	-6.16e-4
23	-7.37e-06	-7.37e-06	-9.44e-06	-7.39e-05
24	-5.79e-06	-9.35e-06	-6.24e-06	-2.23e-05
25	-7.78e-06	-1.83e-05	-7.78e-06	-1.83e-05
26	-2.28e-05	-1.24e-05	-3.66e-06	-2.31e-4
27	-5.29e-06	-3.80e-06	-4.30e-06	-1.05e-4
28	-4.17e-06	-4.17e-06	-4.16e-06	-3.69e-05
29	-3.54e-06	-4.92e-06	-3.53e-06	-1.60e-05
30	-4.80e-06	-8.80e-06	-4.71e-06	-1.39e-05
31	-1.88e-05	-1.20e-05	-1.80e-06	-2.97e-05
32	-1.23e-05	-9.74e-06	-6.73e-06	-8.16e-05
33	-1.14e-05	-1.14e-05	-8.84e-06	-6.46e-05
34	-9.64e-06	-1.22e-05	-8.38e-06	-3.86e-05
35	-8.28e-06	-1.29e-05	-7.55e-06	-2.44e-05

Table 6.4.: Extracted measured slopes for $\Delta z = 0.001$ m and $int = 0.02$ m

The slopes in green boxes represent the highest slopes in magnitude and they were a result of the nearest (x,y) -point to each sensor. The ones in the red boxes represent the lowest slopes in magnitude and they were a result of the farthest (x,y) -point to each sensor.

6.3.2. CRLB Calculations

In this section, I will use the linearly approximated voltages to perform CRLB (Cramer Rao Lower Bound) calculations. CRLB is a minimum bound on the variance of the estimated z -position of the ball object and hence the of the goal detection accuracy. This bound depends only on the behavior of the noise and independent of any localization algorithm.

The following calculations are only valid for an *int* of 0.02 m which was used during Section 6.3.1 to linearly approximate the induced voltage curves.

Recall (6.4)

$$U_{approx}(z) \Big|_{x,y} = s_{x,y,n} z$$

where z takes any value from -0.01 to 0.01 m. Then, the measured voltage value by sensor number n can be calculated as following

$$M_n \Big|_{x,y} = U_n(z) \Big|_{x,y} + w_n \quad (6.7)$$

where w_n is a random and independent WGN of $\mu = 0$ V and a standard deviation σ V observed by sensor number n . I have considered N number of observing sensors, where $n = 1, 2, \dots, N$. Figure 6.24 shows the distribution of $N = 10$ sensors around the localization area.

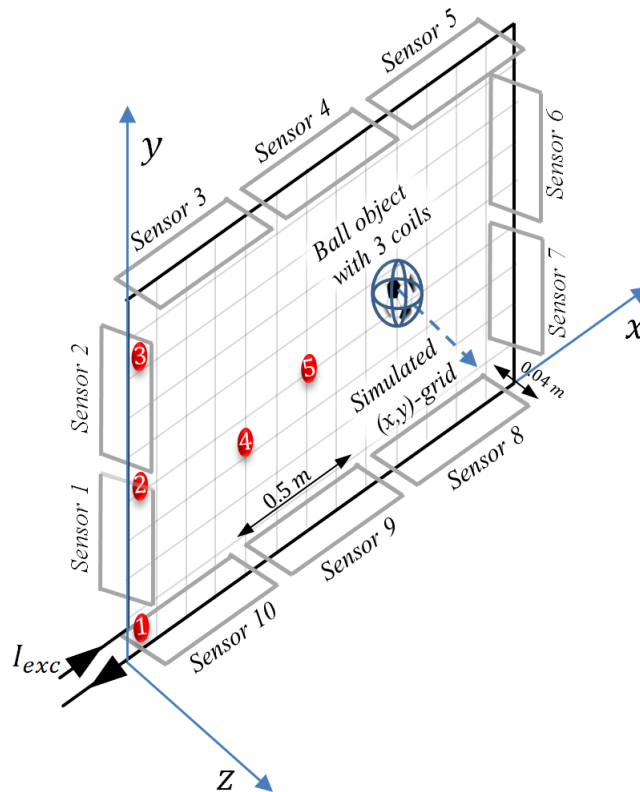


Figure 6.24.: Magnetic sensors distribution and the chosen (x,y) -points

According to [9], the probability density function is expressed as

$$f(M; z) = \prod_{n=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[\frac{-(M_n(z) - U_n(z))^2}{2\sigma^2} \right] \quad (6.8)$$

which tells us how probable it is to measure a certain M voltage value provided that the ball object is located at a certain location z .

Please note that for simplicity reasons the terms M and $U(z)$ mean the measured and the approximated voltages respectively as a function of z for an object located at a certain (x,y) -point, and the term s_n means the approximated voltage's slope observed by sensor n for a ball object located at a certain (x,y) -point.

The next step is to take the \ln of (6.8) which will become

$$\ln f(M; z) = \ln \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^N - \frac{1}{2\sigma^2} \sum_{n=1}^N (M_n(z) - U_n(z))^2 \quad (6.9)$$

Now, I will fix M and take the first partial w.r.t z

$$\frac{\partial}{\partial z} \ln f(M; z) = \frac{1}{\sigma^2} \sum_{n=1}^N s_n (M_n(z) - U_n(z)) \quad (6.10)$$

then I take the partial derivative one more time

$$\frac{\partial^2}{\partial z^2} \ln f(M; z) = -\frac{1}{\sigma^2} \sum_{n=1}^N s_n^2 \quad (6.11)$$

and since equation 6.11 does not depend on M , the CRLB on the variance of estimating z in the *int* of 0.02 m can be calculated as following

$$\begin{aligned} CRLB &= \frac{1}{-\frac{\partial^2}{\partial z^2} \ln f(M; z)} \\ &= \frac{\sigma^2}{\sum_{n=1}^N s_n^2} \end{aligned} \quad (6.12)$$

As can be seen from equation 6.12, the CRLB depends on the noise's standard deviation and the sum of the squared slopes of the observing sensors.

For a fixed noise level, the bound on the variance of the object's z -position declines with increasing the number of observing magnetic sensors. For a fixed number of sensors, it increases with increasing the noise level.

6.3.3. Noise Level Effect

For investigating the noise-level's effect on the resulting CRLB, I will fix the number of sensors to $N = 10$ and calculate the corresponding CRLB of (6.12). This will be done for the five red (x,y) -points in Figure 6.24 and defined in Table 6.5.

Point number	(x,y) (m)
1	(0.05,0.05)
2	(0.05,0.6)
3	(0.05,1.2)
4	(0.3,0.5)
5	(0.93,0.6)

Table 6.5.: The investigated (x,y) -points

The slope values $s_{x,y,n}$ were calculated for the aforementioned five points and ten sensors in an *int* of 0.02 m.

Figure 6.25 illustrates the calculated CRLB for the chosen (x,y) -points. Point 5 has the highest bound on estimating the z -position of the object due to the fact that its the farthest point to all of the sensors. And consequently the induced voltages in the sensors have the lowest slope values.

Point 1 has the lowest bound because it is positioned near to sensor number 1 and 10. This resulted in a high slope value which lowered the CRLB according to equation 6.12.

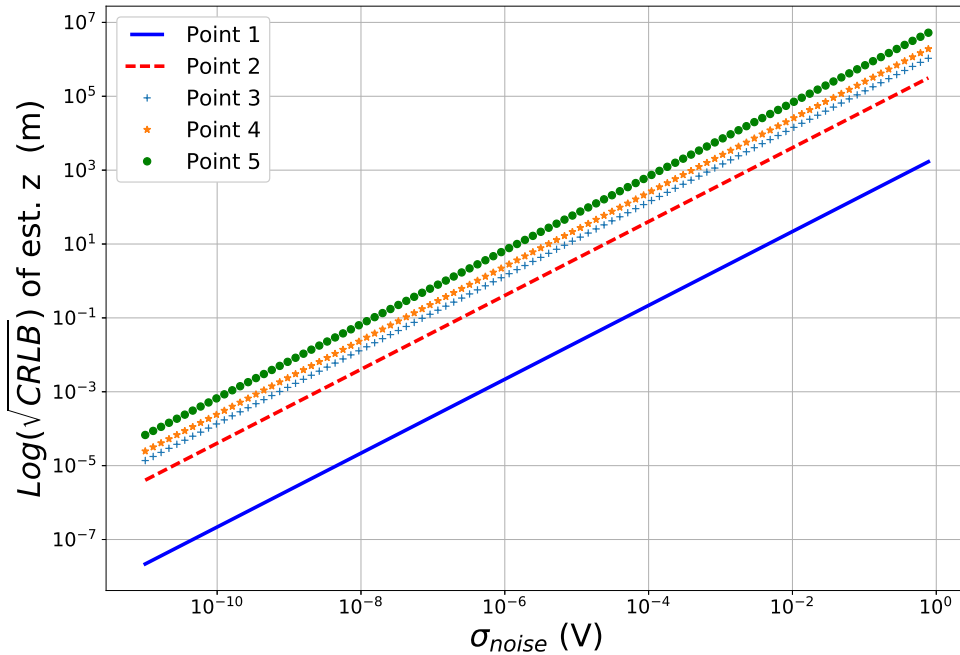


Figure 6.25.: Noise-level's effect on the CRLB

6.3.4. Sensors Constellation Effect

In this section, I will fix the noise level and investigate the CRLB for different sensors' constellations. Table 6.6 includes the investigated (x,y) -grid.

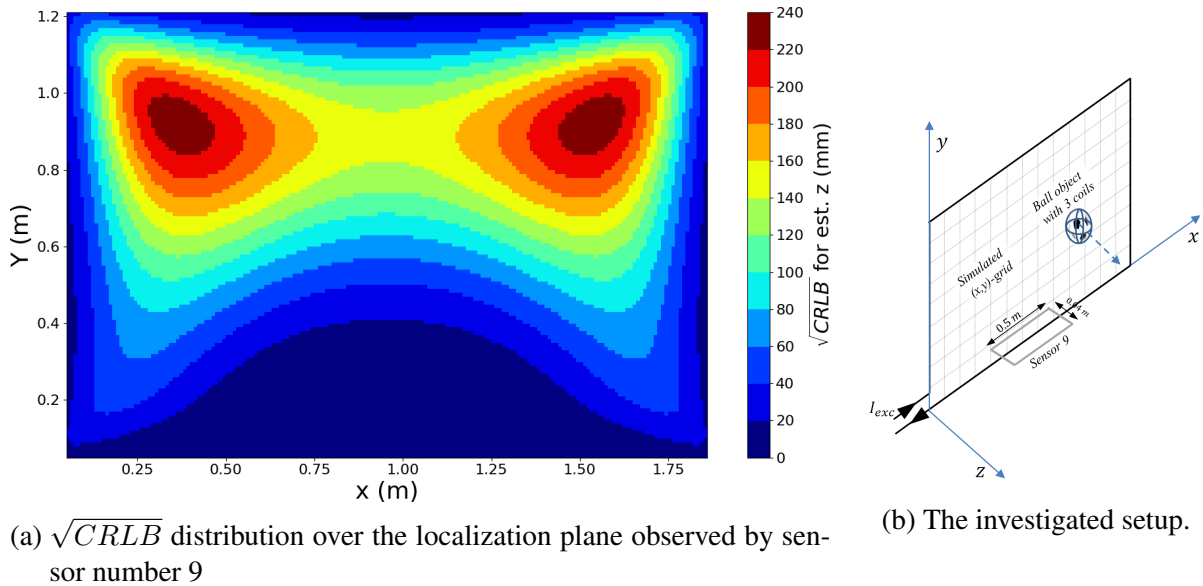
x -sweep (m)	y -sweep (m)
$[0.05, 1.85], \text{step}= 0.01$	$[0.05, 1.22], \text{step}= 0.01$

Table 6.6.: The investigated (x,y) -grid

By inspecting Figure 6.25, noise levels which are higher than 10^{-8}V have resulted in a \sqrt{CRLB} in the meter and kilometer range for points 2,3,4 and 5. Which seems to be a high uncertainty range for a real system. On the contrary, noise levels which are less than 10^{-8}V have resulted in a \sqrt{CRLB} in the millimeter and micrometer range for the same points. And that seems a bit ideal.

So, I have chosen $\sigma = 10^{-8}\text{V}$ to be the fixed noise level during this investigation. Because it has resulted in a \sqrt{CRLB} in the centimeter and millimeter range. Which could be a reasonable uncertainty range for a real goal detection system.

Using only sensor number 9 (in Figure 6.26b), Figure 6.26a shows the corresponding bound distribution over the localization plane with a maximum Cramer bound in the range of 240 mm indicated by the red areas.



(a) \sqrt{CRLB} distribution over the localization plane observed by sensor number 9

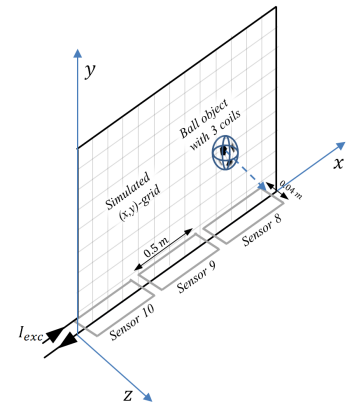
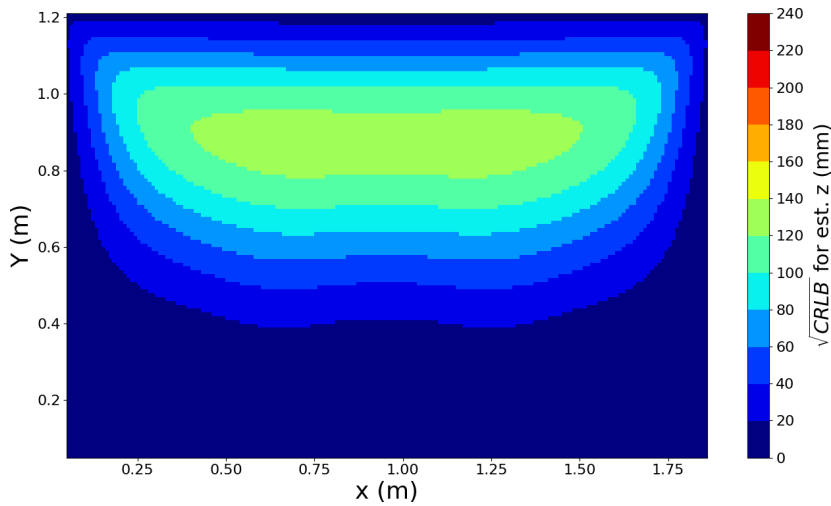
(b) The investigated setup.

Figure 6.26.: CRLB distribution using only 1 sensor

Now, I will increase the number of observing sensors to three sensors as illustrated in Figure 6.27b which results in a \sqrt{CRLB} distribution as shown in Figure 6.27a. And as a result, the \sqrt{CRLB} dropped to a maximum of 140 mm which is consistent with what was expected following (6.12).

For the next constellation, I will use sensors number 4 and 9 as shown in Figure 6.28b. This constellation resulted in a \sqrt{CRLB} distribution with a maximum value of around 100 mm as shown in Figure 6.28a.

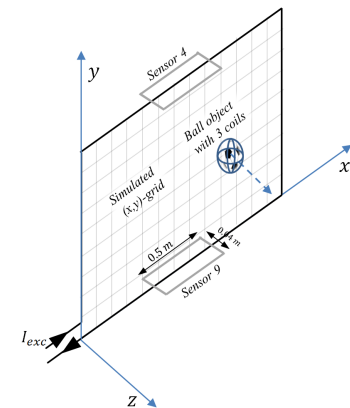
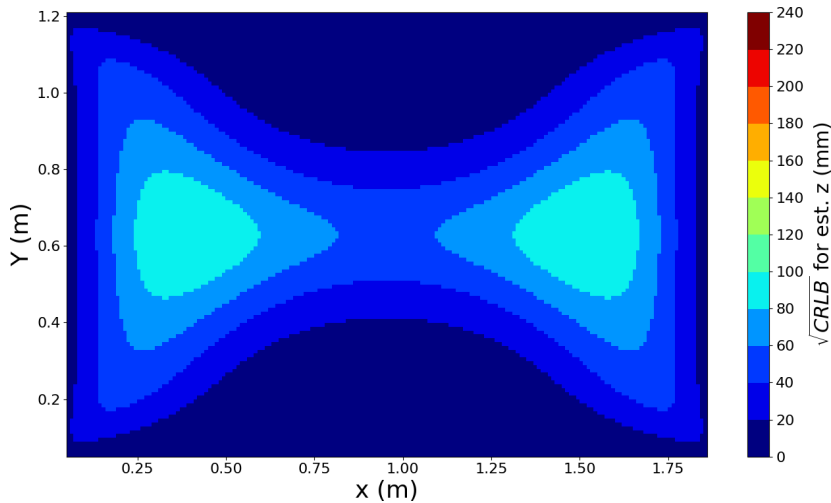
This maximum value is lower than the maximum bound value when three sensors were used in Figure 6.27a.



(a) \sqrt{CRLB} distribution over the localization plane observed by sensors number 8, 9, 10

(b) The investigated setup.

Figure 6.27.: CRLB distribution using 3 sensors



(a) \sqrt{CRLB} distribution over the localization plane observed by sensors number 4, 9

(b) The investigated setup.

Figure 6.28.: CRLB distribution using 2 sensors

Using sensors number 1, 4, 6 and 9 has resulted in the distribution shown in Figure 6.29a with a maximum Cramer bound of around 80 mm.

Finally, all the ten sensors in Figure 6.30b have been used and the resulting distribution is represented by the heat-map of Figure 6.30a with a maximum bound of 60 mm.

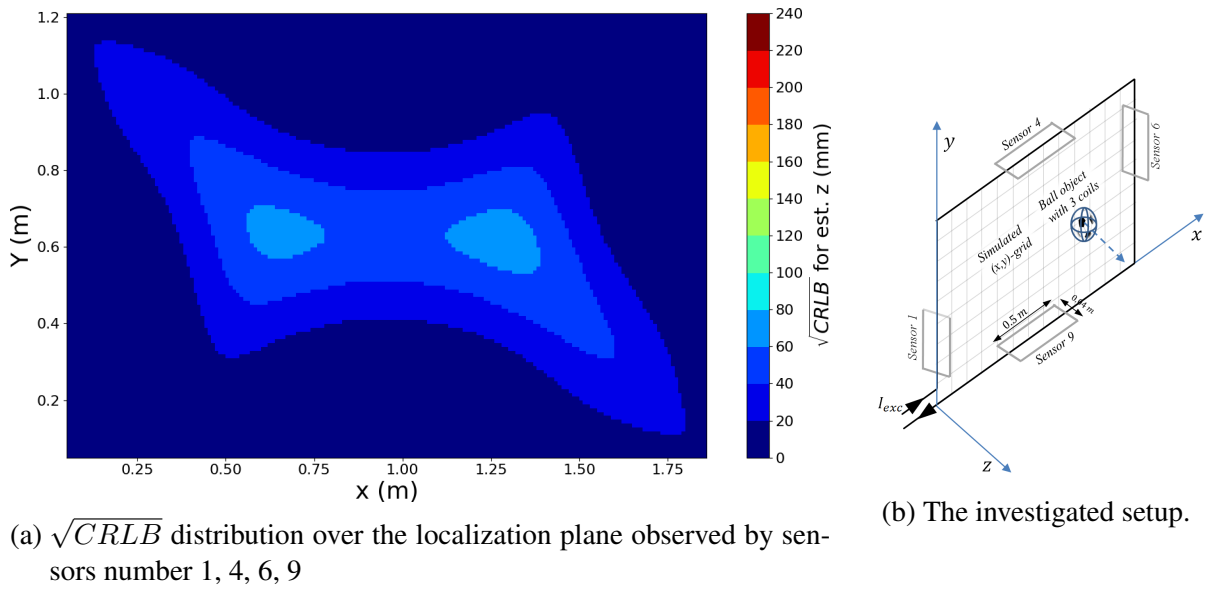


Figure 6.29.: CRLB distribution using 4 sensors

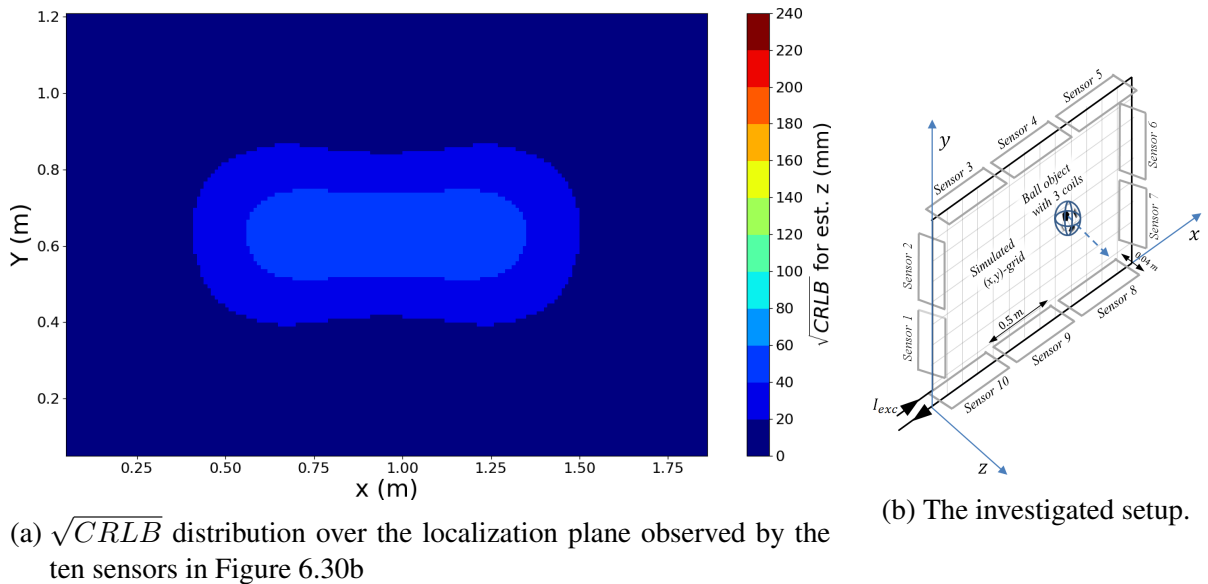


Figure 6.30.: CRLB distribution using 10 sensors

The aforementioned figures illustrate that increasing the number of the observing sensors lowers the CRLB. The bound is also affected by how the sensors are being distributed.

For example, using $N = 4$ sensors (shown in Figure 6.29b) has resulted in a maximum \sqrt{CRLB} of 80 mm while using $N = 10$ sensors (shown in Figure 6.30b) has a maximum bound of 60 mm which means only a 25% improvement.

The results of such an investigation can help in:

- Defining the accuracy limit of the system using a certain sensors' constellation
- Optimizing the number of the required sensors to achieve a specific accuracy limit

7. 2-D Localization

Magnetic field localization can be used to estimate a 2-D position of an object equipped with one or more coils. In this chapter, an overview about the 2-D localization process will be given and some disturbing influences of chapter 5 will be investigated.

7.1. 2-D Localization Method

2-D localization can be achieved by performing a comparison algorithm between the measured voltage value by sensor number n and a simulated induced voltage values for the same sensor. And depending on the minimum output from the algorithm, a 2-D position will be estimated.

For 2-D localization in the xy -plane, a reference table with all the corresponding voltage values including different (x,y) -positions must be created. I have used the simulation tool to configure the setup shown in Figure 7.1.

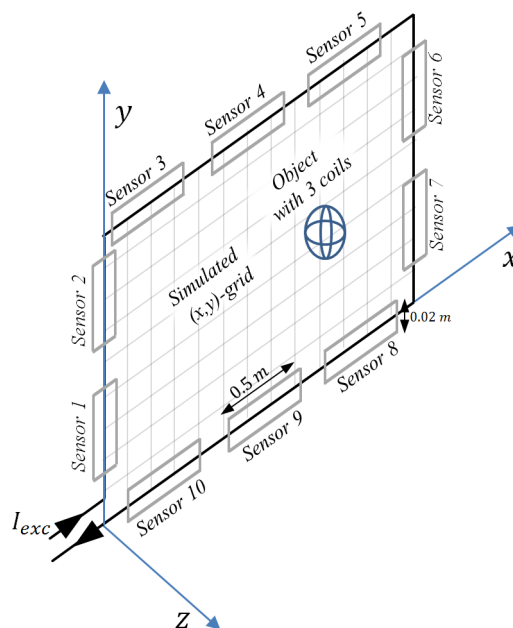


Figure 7.1.: 2-D localization setup

This setup consists of the following

- Ten magnetic sensors positioned in the xy -plane, 0.5×0.02 m each
- Exciter wire that passes exactly through the middle of each sensor and carries an AC current of 1 A

- An object with three orthogonal circular coils oriented in the x , y , z directions as illustrated in Figure 3.2. Each one has a radius of 3.81 cm

The object was moved along the (x,y) -grid and the induced voltage in each sensor was calculated. A grid of 1 cm has been selected. Hence the maximum achievable localization accuracy is 1 cm. Figure 7.2 shows a heat map of the sum of the induced voltage distribution over the localization plane.

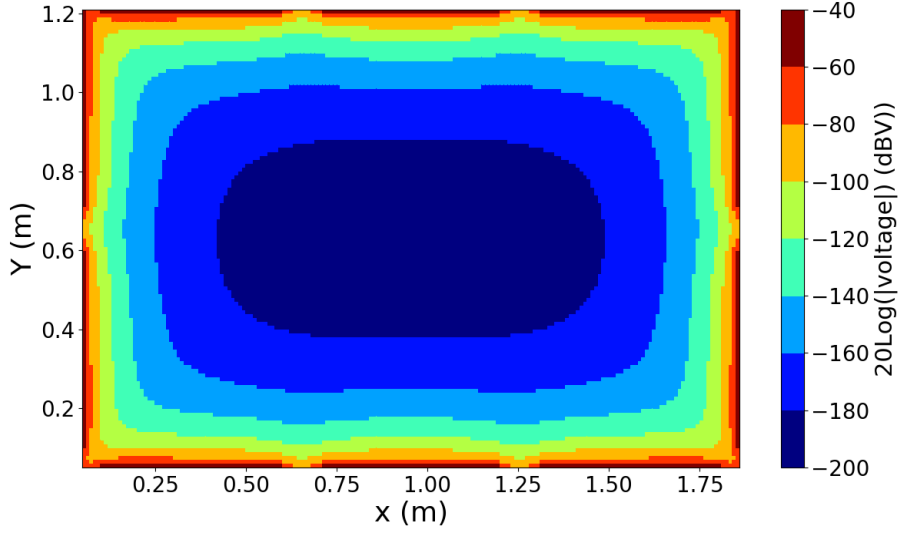


Figure 7.2.: The simulated induced voltage distribution for the setup in Figure 7.1

The simulated table has the form of Table 7.1 with ml rows and $N + 2$ columns. Where

Grid		U_n			
x -Pos. (m)	y -Pos. (m)	Sen. 1 (V)	Sen. 2 (V)	...	Sen. N (V)
x_1	y_1	U_{1,x_1y_1}	U_{2,x_1y_1}	...	U_{N,x_1y_1}
x_1	y_2	U_{1,x_1y_2}	U_{2,x_1y_2}	...	U_{N,x_1y_2}
x_1	y_3	U_{1,x_1y_3}	U_{2,x_1y_3}	...	U_{N,x_1y_3}
·	·	·	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·
x_m	y_l	U_{1,x_my_l}	U_{2,x_my_l}	...	U_{N,x_my_l}

Table 7.1.: The simulated 2-D Localization reference table form

m and l are the simulated x and y points respectively. N is the total number of the used magnetic sensors. U_n is the simulated induced voltage in a sensor n for an object positioned at a specific (x,y) -position.

For example, I have considered the localization setup in Figure 7.3b with an object placed exactly in the middle of the localization xy -plane at $(0.93,0.6)$ m. One magnetic sensor will be used for the localization.

Algorithm 2 will be used to determine the object's position. The input to the algorithm is the xy -grid and the simulated voltage U_9 for $(0.93,0.6)$ m from Table 7.1. Additionally, I have chosen the threshold value E_{thr} to be $10\mu\text{V}$. E_{thr} represents the selection criteria for

(x,y) -point to be considered as a possible object's position.

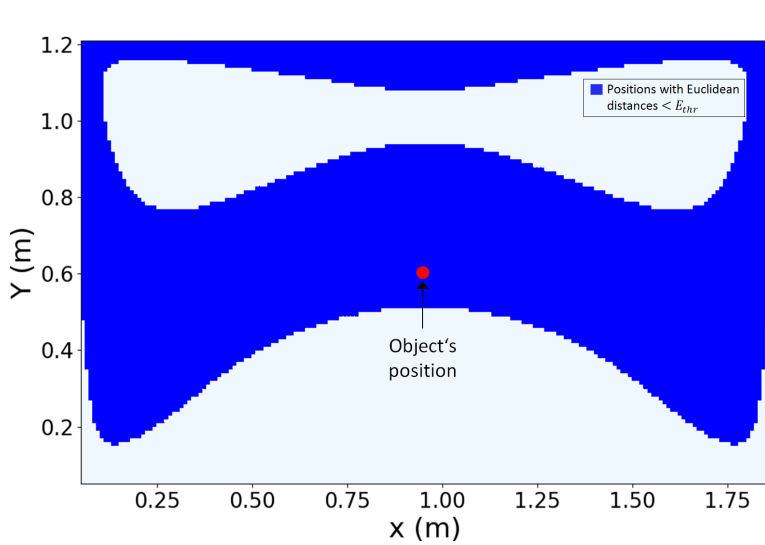
Figure 7.3a illustrates the output from Algorithm 2. The blue colored points represent the possible object's positions. Several (x,y) -positions have produced an Euclidean distances less than E_{thr} . Which means that the current position of the object does not result in a unique Euclidean distance.

Algorithm 2 2-D localization based on a threshold Euclidean distance

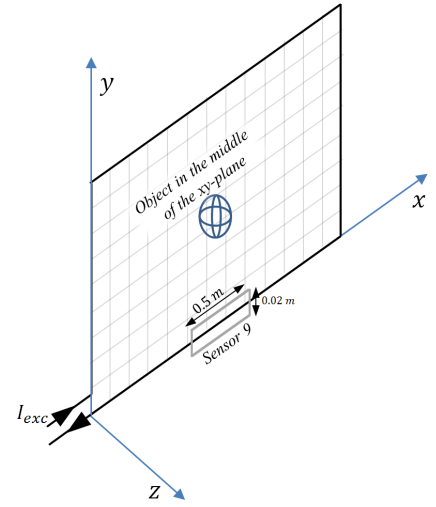
```

1: Input: Grid( $ml \times 2$ ),  $U_9(ml \times N)$ ,  $U_9$ ,  $E_{thr}$ 
2: Output: possiblePositions
3: EuclideanDistances = []
4: for  $p$  in range (0, $ml$ ) do
5:   EuclideanDistances.append  $\left( \sqrt{(U_n - U_n[p])^2} \right)$ 
6: possiblePositions = []
7: for  $i$  in range (0,len(EuclideanDistances)) do
8:   if EuclideanDistances[ $i$ ] <  $E_{thr}$  then
9:     possiblePositions.append([Grid[ $i$ ][0], Grid[ $i$ ][1] ])
10: return possiblePositions

```



(a) Algorithm 2 output presenting the possible (x,y) -positions of the object for a certain induced voltage



(b) Localization setup with an object positioned in the middle of the xy -plane and one magnetic sensor

Figure 7.3.: Uncertainty in 2-D localizing of an object

For 2-D localization, $N > 1$ sensors should be used to decrease the uncertainty in determining the object's position. During the following sections, the effect of increasing the number of sensors on the localization accuracy will be investigated.

7.2. Influences on 2-D Localization

During this section, the following influences are investigated:

- Number of the magnetic field sensors.
- Thermal noise effect.
- Orientation of the sensors.
- Orientation of the object coils.

7.2.1. Number of the Used Sensors

In this section, I am investigating how the number of the used sensors affects the 2-D localization accuracy. Additionally, the noise effect will be considered.

In real life, the presence of noise will not allow a 100% accurate localization. Therefore, the measured voltage M by sensor n for an object positioned at (x,y) -point can be expressed as

$$M_n \Big|_{x,y} = U_n \Big|_{x,y} + w_n \quad (7.1)$$

where U_n is the simulated induced voltage in sensor n for an object positioned at (x,y) -point, and w_n is a random WGN of $\mu = 0\text{V}$ and a standard deviation $\sigma = 10^{-6}\text{V}$. I have chosen $\sigma = 10^{-6}\text{V}$ because measurements were carried out on the real system and the measured noise level was in the range of $[10^{-8}, 10^{-6}]\text{V}$. Each w_n sample is random and independent for each sensor.

I have implemented Algorithm 3 in Python to determine the object's position based on the minimum Euclidean distance between the measured sensors' voltages and the simulated ones from the reference table.

Algorithm 3 2-D localization based on the minimum Euclidean distance

```
1: Input: Grid( $ml \times 2$ ),  $U_n(ml \times N)$ ,  $M_n(1 \times N)$ 
2: Output:  $x$ -pos,  $y$ -pos
3: EuclideanDistances = []
4: for  $p$  in range ( $0, ml$ ) do
5:     EuclideanDistances.append  $\left( \sqrt{\sum_{n=1}^N (M_n - U_n[p])^2} \right)$ 
6: minDist = min(EuclideanDistances)
7: xyIndex = EuclideanDistances.index(minDist)
8:  $x$ -pos = Grid[xyIndex] [0]
9:  $y$ -pos = Grid[xyIndex] [1]
10: return  $x$ -pos,  $y$ -pos
```

Algorithm 3 calculates all the Euclidean distances between the measured voltage by sensor n and all the simulated voltages for each other sensor. Then the sum of these distances, for each (x,y) -point in the grid, is appended to EuclideanDistances list. Afterwards, the minimum Euclidean distance is calculated. The corresponding index of the minimum distance is extracted. The x -pos and y -pos which have the same index will be returned.

$N = 2$ Sensors

As a start, two sensors are used (sensors 4 and 9 in Figure 7.1) to localize an object positioned in the middle of the localization plane at $(x = 0.93m, y = 0.6m)$.

Depending on the noise sample w_n , Algorithm 3 will produce a different (x,y) -position of the object. Therefore, Monte Carlo(MC) simulations method is useful in such a case. MC method simply means running the same simulation many times but each time a randomly generated noise value will be used. This will produce a statistical distribution for the possible object's positions.

Figure 7.4 shows the possible (x,y) -positions for an object positioned at the red reference point. To generate this distribution, Algorithm 3 was used 1000 times and each time a random w_n value was used. Several (x,y) positions resulted in an induced voltage value which gives a shorter Euclidean distance than the distance produced by the correct object's position.

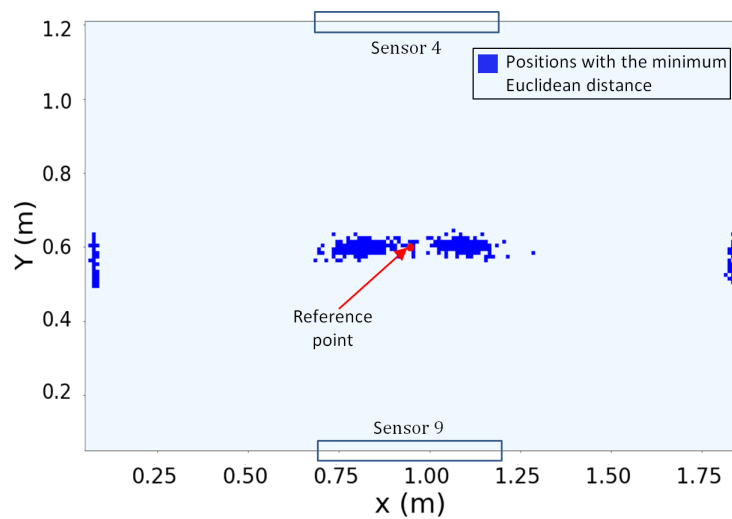


Figure 7.4.: Output of Algorithm 3 using 2 sensors for 1000 MC simulations

$N = 4$ Sensors

Now, the number of the used sensors is increased to four. Two more sensors will be added to the previous setup, namely sensor 1, and 7 in Figure 7.1. Again, Algorithm 3 was executed 1000 times and each time a random w_n value was used.

Figure 7.5 includes the possible (x,y) -positions for the same object positioned at the red reference point. But this time the blue points are concentrated near the reference point with less ambiguity. The uncertainty in determining the object's position is more confined than the case of using two sensors.

$N = 10$ Sensors

All the ten sensors of Figure 7.1 were used during this section. The same 1000 MC simulations were conducted. Each time a random w_n value was used. Figure 7.6 shows the resulting uncertainty area of the outputs from Algorithm 3.

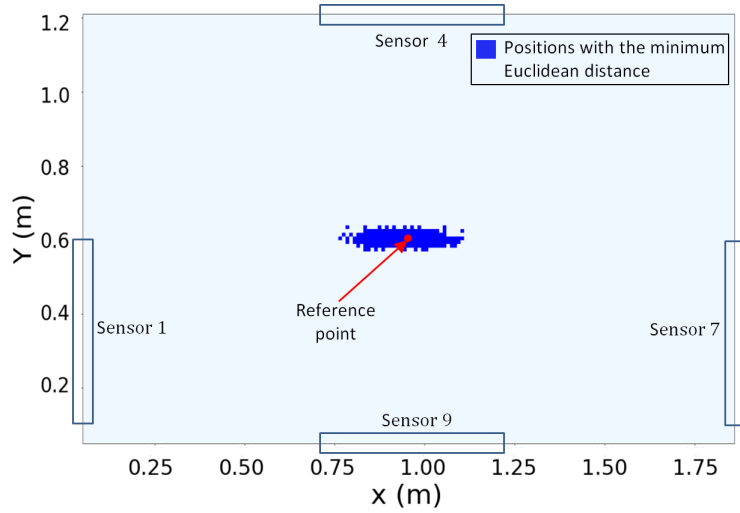


Figure 7.5.: Output of Algorithm 3 using 4 sensors for 1000 MC simulations

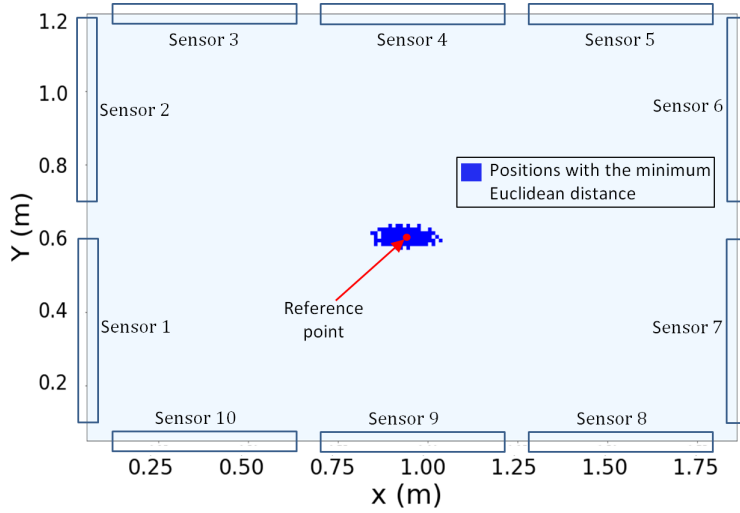


Figure 7.6.: Output of Algorithm 3 using 10 sensors for 1000 MC simulations

From the outcomes of this investigation, its obvious that the number of the used sensors affects the accuracy of 2-D localization. The more sensors the more accurate an object can be localized in 2-D space.

7.2.2. Noise Level Effect on the 2-D Localization Accuracy

In this section, the effect of the noise level on the localization accuracy will be investigated. All the ten sensors of Figure 7.1 will be used to carry on the localization process. I have selected ten noise levels in the range $[10^{-10}, 10^{-1}]V$ to be investigated.

Firstly, the object is positioned at $(x_0 = 0.93m, y_0 = 0.6m)$. For each noise level, M_n in (7.1) is calculated 1000 times and in each time w_n is randomly generated and Algorithm 3 is executed. After each time Algorithm 3 is executed, an error value $error_{xy}$ is calculated as following

$$error_{xy} = \left(\sqrt{(x_0 - x_{output})^2 + (y_0 - y_{output})^2} \right) \quad (7.2)$$

where it is measured in meters, x_0 is the object's correct x -position, x_{output} is the output x -position from Algorithm 3, y_0 is the object's correct y -position, y_{output} is the output y -position from Algorithm 3. Figure 7.7 illustrates what is meant by $error_{xy}$. As can be seen, it is the euclidean distance between the localization algorithm output and the correct object's position.

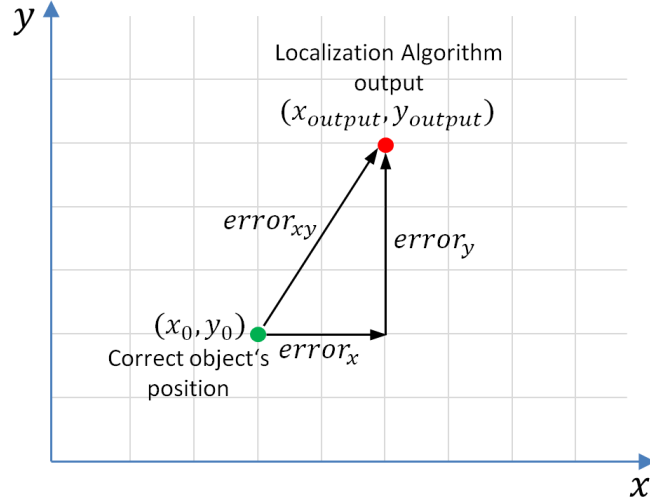


Figure 7.7.: Illustration of $error_{xy}$ in (7.2)

After 1000 $error_{xy}$ values were calculated, for each noise level, their average can be expressed as

$$avgError_{xy,\sigma_{noise}} = \frac{\sum_{n=1}^{1000} error_{xy,n}}{1000} \quad (7.3)$$

Then their corresponding standard deviation can be calculated as

$$stdError_{xy,\sigma_{noise}} = \sqrt{\frac{1}{1000} \sum_{n=1}^{1000} (error_{xy,n} - avgError)^2} \quad (7.4)$$

Figure 7.8 shows the noise effect on both $stdError_{xy}$ and $avgError_{xy}$ for an object positioned exactly in the middle of the localization plane at $(0.93m, 0.6m)$.

The same object will be positioned in the left bottom corner of the localization plane at $(x_0 = 0.93m, y_0 = 0.6m)$ close to the exciter wire. The previous calculations will be repeated for the new object's position. Figure 7.9 shows the resulting noise effect on both $stdError_{xy}$ and $avgError_{xy}$.

From Figure 7.8 and Figure 7.9, it can be concluded that localization of an object placed near the exciter wire is more robust against noise. This is because the nearer the object to the exciter wire the more voltage it induces in each sensor. For example, in Figure 7.9 the error starts to occur at $\sigma_{noise} > 10^{-3}V$. Which means that the induced voltages were in the milli-voltage range. Furthermore, in Figure 7.8 the error starts to occur at $\sigma_{noise} > 10^{-7}V$. Which means that the induced voltages were in the micro-voltage range.

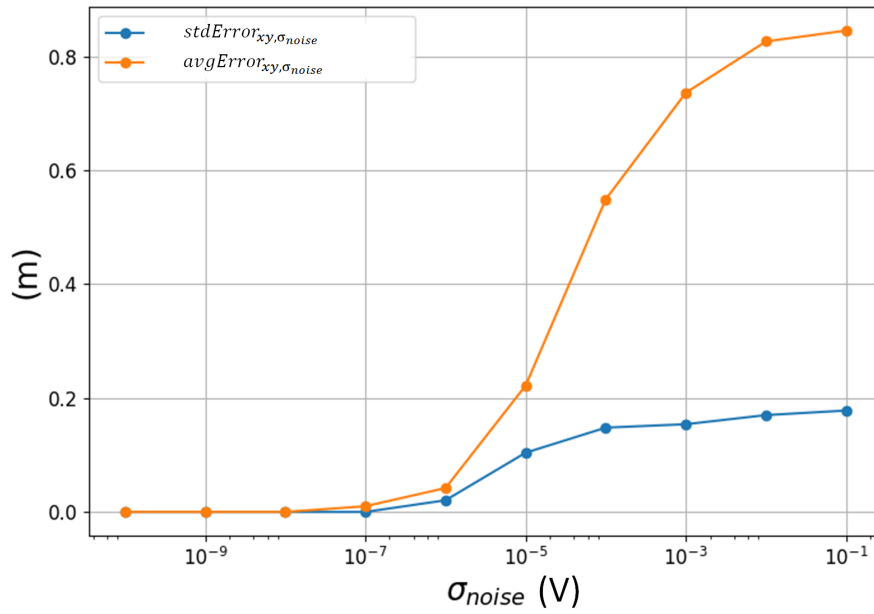


Figure 7.8.: Noise level effect on the $stdError_{xy}$ and $avgError_{xy}$ for an object positioned at $(0.93m, 0.6m)$

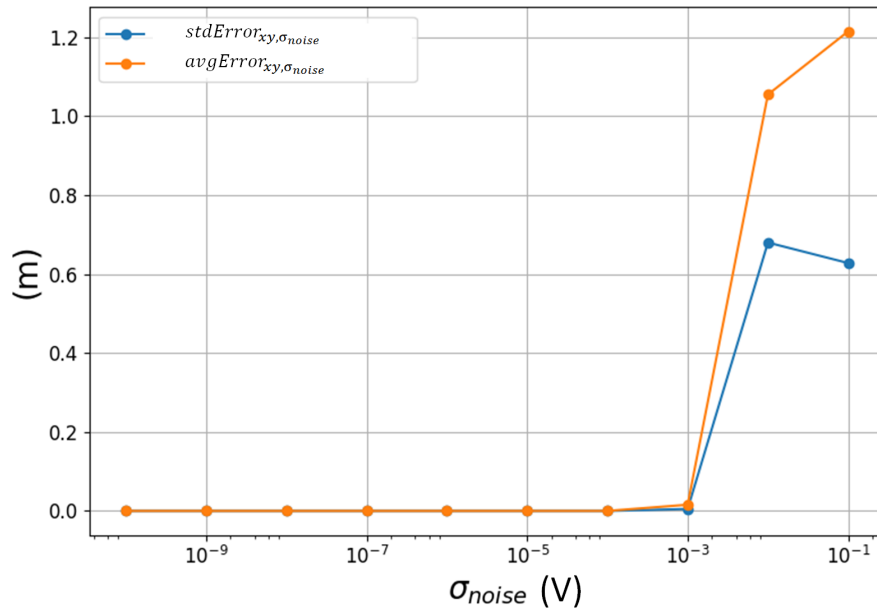


Figure 7.9.: Noise level effect on the $stdError_{xy}$ and $avgError_{xy}$ for an object positioned at $(0.05m, 0.05m)$

7.2.3. Orientation of the Used Sensors

Another factor that affects the 2-D localization accuracy is how the measuring sensors are oriented. In this section I am investigating how tilting of the 10 sensors affects the localization accuracy. I have used the simulation tool to configure the same setup in Figure 7.1 but every sensor will be tilted anti-clockwise around the $-x$ -axis with an angle θ .

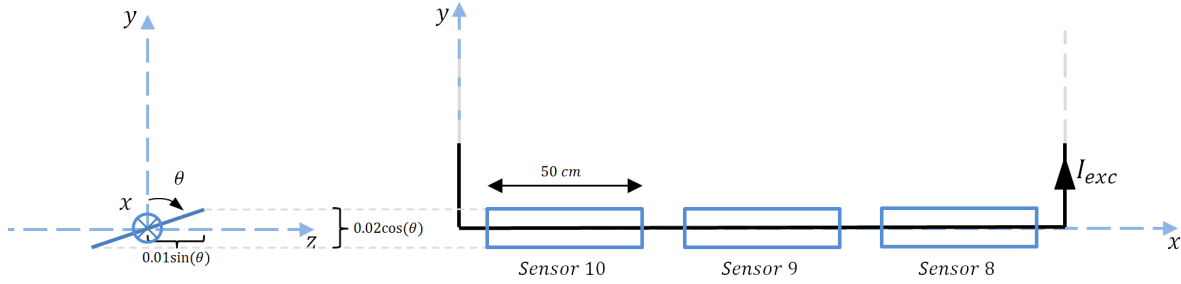
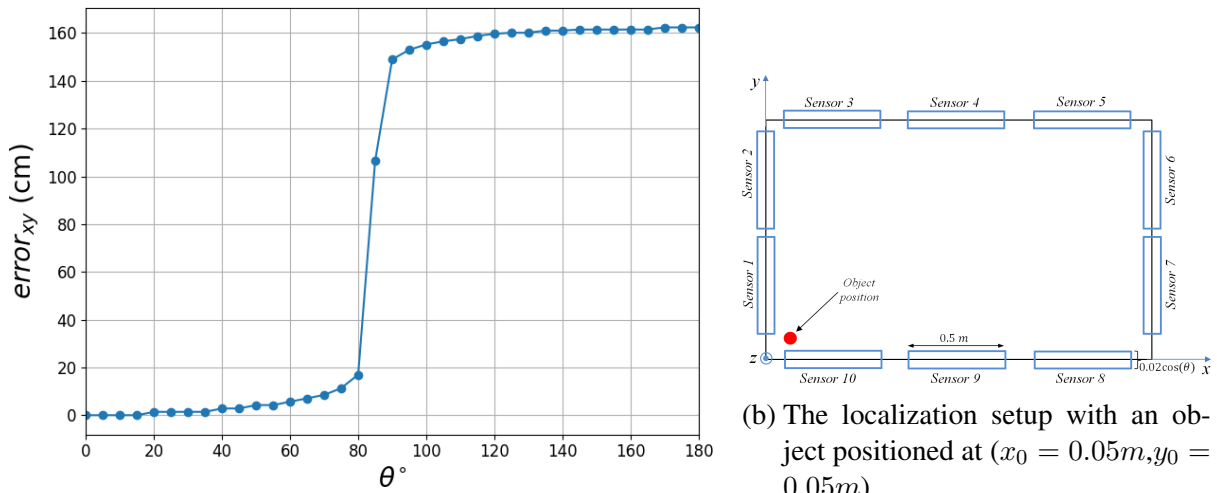


Figure 7.10.: Tilting of sensors anti-clockwise around the x -axis with an angle θ

Firstly, the object is placed at $(x_0 = 0.05m, y_0 = 0.05m)$ as shown in Figure 7.11b. Then the $error_{xy}$ from (7.2) was calculated for each tilting angle θ . In this case, x_{output} and y_{output} in (7.2) are the output (x,y) -position from Algorithm 3. Additionally, M_n in Algorithm 3 is the measured voltage value in sensor n due to a tilting angle θ without considering the noise effect. Figure 7.11a shows the resulting $error_{xy}$ due to tilting of the sensors. The maximum error level has reached 160 cm with a tilting angle $\theta = 180^\circ$

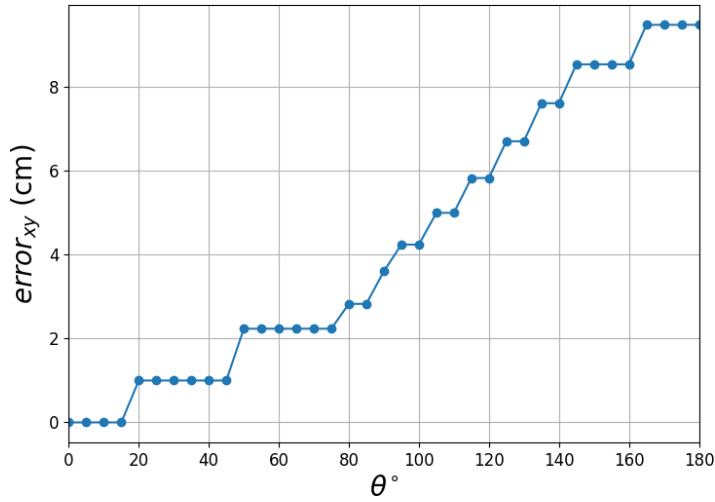


(a) The resulting $error_{xy}$ due to tilting of the sensors with angle θ

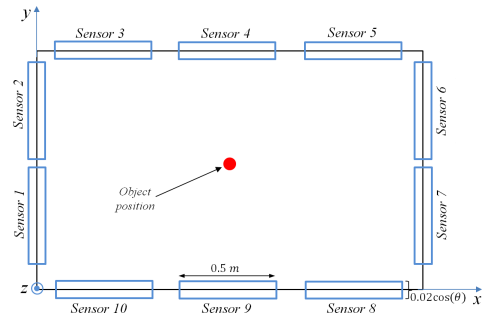
Figure 7.11.: The resulting $error_{xy}$ for an object positioned at $(x_0 = 0.05m, y_0 = 0.05m)$

Secondly, the object is placed at $(x_0 = 0.93m, y_0 = 0.6m)$ as shown in Figure 7.12b. Then the $error_{xy}$ from (7.2) was calculated for each tilting angle θ . Figure 7.12a shows the resulting $error_{xy}$ due to tilting of the sensors. The maximum error level has reached 9 cm with a tilting angle $\theta = 180^\circ$

Finally, the object is placed at $(x_0 = 1.85m, y_0 = 0.6m)$ as shown in Figure 7.13b. Then the $error_{xy}$ from (7.2) was calculated for each tilting angle θ . Figure 7.13a shows the resulting



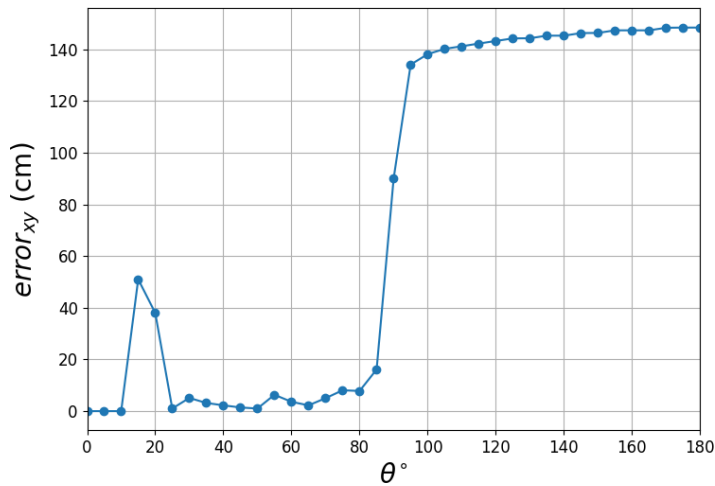
(a) The resulting $error_{xy}$ due to tilting of the sensors with angle θ



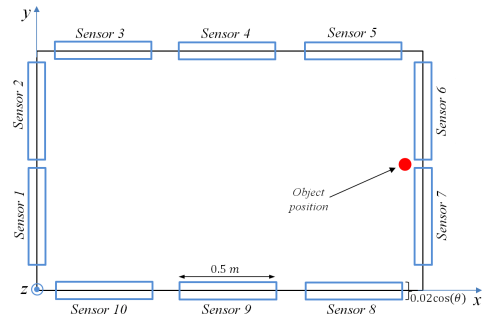
(b) The localization setup with an object positioned at $(x_0 = 0.93m, y_0 = 0.6m)$

Figure 7.12.: The resulting $error_{xy}$ for an object positioned at $(x_0 = 0.93m, y_0 = 0.6m)$

$error_{xy}$ due to tilting of the sensors. The maximum error level has reached 150 cm with a tilting angle $\theta = 180^\circ$



(a) The resulting $error_{xy}$ due to tilting of the sensors with angle θ



(b) The localization setup with an object positioned at $(x_0 = 1.85m, y_0 = 0.6m)$

Figure 7.13.: The resulting $error_{xy}$ for an object positioned at $(x_0 = 1.85m, y_0 = 0.6m)$

From the results of this investigation, the position in the middle of the localization plane has produced less error than those near to the exciter wire. The reason is the same as mentioned before which relates to the amount of induced voltages in the sensors due to the object's magnetic field. The further the object from the exciter wire the lower the induced voltage in each sensor and the lower the output values from Algorithm 3. Then the terms of (7.2) results in less $error_{xy}$ value.

7.2.4. Rotation of Object Coils

The object's coils are the main source of the secondary magnetic field which is necessary for localization. Each coil produces a certain amount of magnetic flux depending on the amount of the primary magnetic flux which passed through its area. Although the coil's area is constant, the effective coil's area which encloses primary magnetic flux changes with the coil orientation.

Therefore, in this section I am investigating how the orientation of the object's coils affects the induced voltages in the sensors and hence the localization accuracy. Noise effect will not be considered.

The simulation tool was used to configure the setup shown in Figure 7.14. Including ten sensors and an object with three orthogonal coils. Each coil is oriented in the x , y or z direction. The object is positioned at each of the three red points in Figure 7.14. At each point, the object is rotated around each of the (x,y,z) -axes with an angle ϕ . I am investigating two rotation angles, namely $\phi = 45^\circ$ and $\phi = 90^\circ$.

Figure 7.15 illustrates how the rotation around each axis is done. The rotation angle ϕ is increasing in the anti-clockwise direction when looking towards the positive direction of the rotation axis.

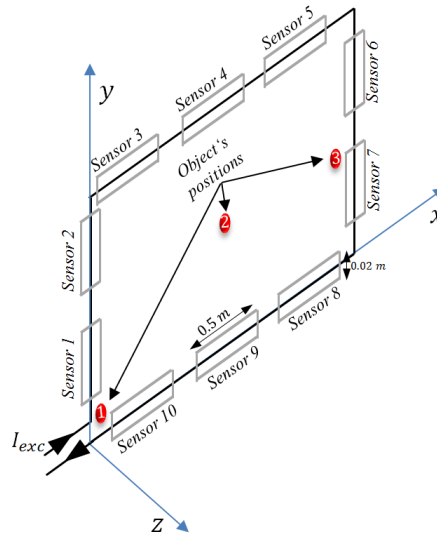


Figure 7.14.: The configured setup and the investigated (x,y) -points

For each investigated point, the $error_{xy}$ from (7.2) was calculated for each rotation angle ϕ for each axis. In this case, x_{output} and y_{output} in (7.2) are the output (x,y) -position from Algorithm 3. Additionally, M_n in Algorithm 3 is the measured voltage value in sensor n due to a rotation angle ϕ without considering the noise effect.

Table 7.2 includes the resulting $error_{xy}$ for each investigated point. The check marks represent a correct 2-D localization with no errors.

For point 1 which is located near to the exciter wire, an error value of 0.042m occurred when the object was rotated around x or y -axis with $\phi = 45^\circ$.

No errors have occurred when the object was positioned at point 2 for any rotation angle around any axis.

Point 3 suffered also from errors because of the fact that its located near the exciter wire.

From Table 7.2, it can be seen that rotating the object around the z -axis with any ϕ angle did not result in any errors. Because this rotation only affects the orientation of both x ,

and y coils which are the less contributors to the overall secondary magnetic flux. In the meanwhile, the orientation of the z -coil remains intact in the xy -plane which produces the majority of the magnetic flux that passes through the sensors' areas.

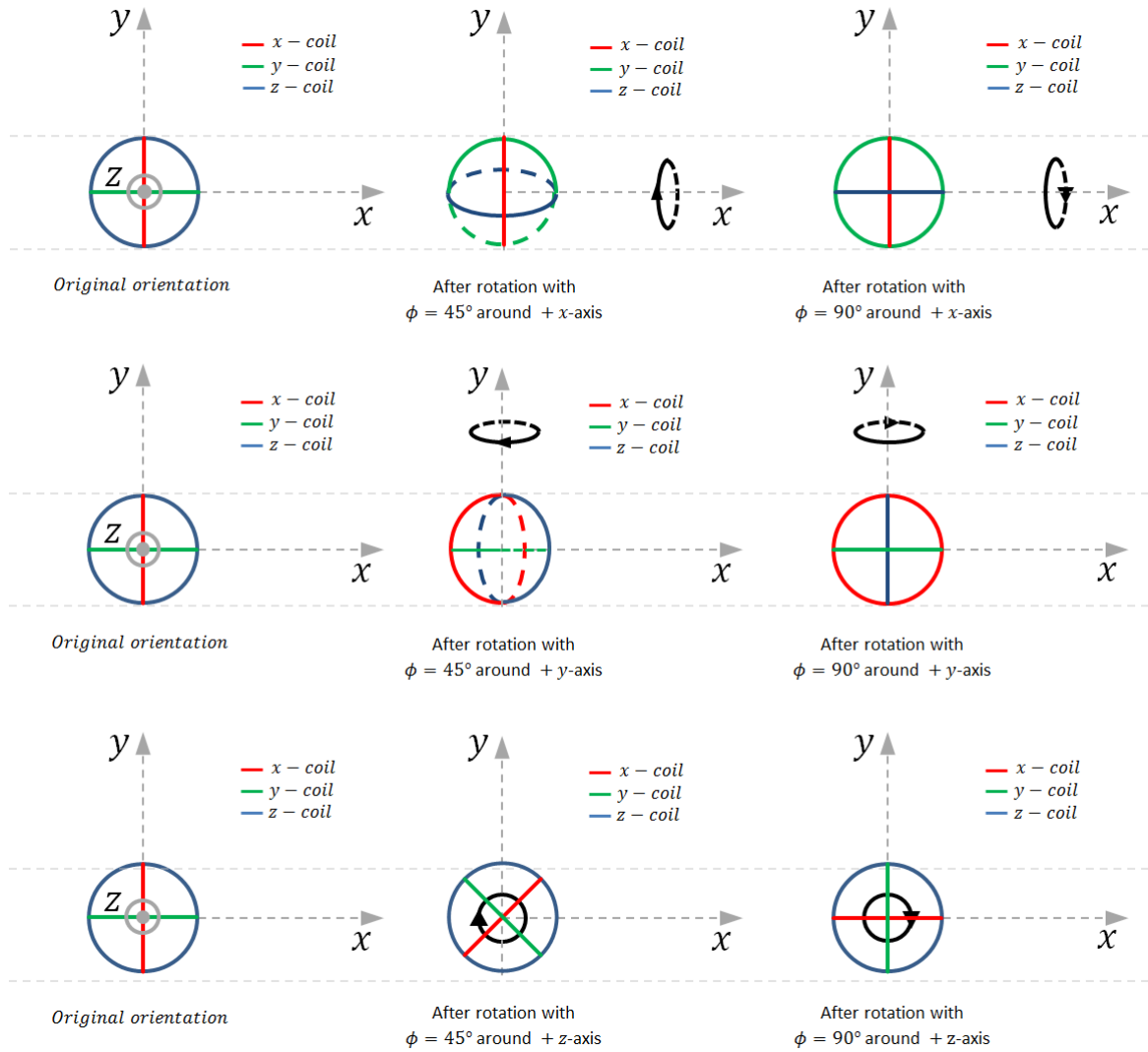


Figure 7.15.: Object coils rotation illustration

rotation axis	ϕ	Point 1 (m) ($x = 0.05, y = 0.05$) Bottom left corner	Point 2 (m) ($x = 0.93, y = 0.6$) Middle	Point 3 (m) ($x = 1.85, y = 0.6$) Middle right-side
x	45°	$err_{xy} = 0.042\text{m}$	✓	$err_{xy} = 0.022\text{m}$
x	90°	✓	✓	✓
y	45°	$err_{xy} = 0.042\text{m}$	✓	$err_{xy} = 0.028\text{m}$
y	90°	✓	✓	✓
z	45°	✓	✓	✓
z	90°	✓	✓	✓

Table 7.2.: Error due to the rotation of the object with angle ϕ

From the results of this investigation, its noticeable that the localization accuracy is more vulnerable to errors near the the exciter wire due to the rotation of the object's coils.

8. Summary

Systems that use a magnetic field for localizing objects do suffer from various disturbing influences. These influences can come from the system's imperfections or from the surrounding environment. In this thesis, a selected group of these influences was investigated for different localization applications.

For 1-D localization, I have explained the principle of the GoalRef system which indicates whether a ball has crossed the goal line. The number of sensors influence on the goal detection accuracy was investigated while considering WGN effect. Additionally, the orientation of the sensors was examined by simulating sensors with tilting angles of $\theta = 1^\circ$ and 3° .

Without considering the noise effect, maximum error values of 8 mm and 25 mm have occurred at $\theta = 1^\circ$ and 3° sensors tilt respectively. The resulting error distribution over the goal plane was consistent with what already has been observed by the GoalRef team at Fraunhofer IIS previously. The presence of noise has affected heavily the 1-D position estimation in the middle of the goal plane. This resulted in an early voltage zero crossings which can not be considered for a correct goal detection. To facilitate the CRLB derivation, the object was positioned at different (x,y) -points and moved along the z -axis. Additionally, The resulted non linear induced voltage curves were linearly approximated. I have fixed the number of sensors to be $N = 10$ and the corresponding CRLB was calculated for different noise levels. Five (x,y) -positions were selected and the resulting CRLB was in millimeter and centimeter range. Then, I have fixed the noise level to be 10^{-8}V and different sensor constellations were investigated. This has shown that not only increasing the number of sensors lowers the bound but also how they are distributed around the goal plane.

For 2-D localization, the method of estimating the (x,y) -position of an object was provided. The simulation tool was used to configure the 2-D localization setup. Unlike the 1-D localization, $N > 1$ sensors should be used to decrease the uncertainty and to estimate a useful 2-D position of the object. The number of the used sensors was investigated while considering noise. Increasing the sensors has reduced the ambiguity in estimating the object's 2-D position. Furthermore, the sensors were tilted by an angle θ between 10° and 180° and the error in estimating the object's 2-D position was calculated for each angle. The positions in the middle of the localization plane have produced less error than those near to the exciter wire. WGN noise effects on the localization accuracy were investigated. The number of sensors was fixed to $N = 10$ and I have investigated the effects of the noise with standard deviation of $\sigma_{noise} = [10^{-10}, 10^{-1}]\text{V}$ on the localization accuracy. The (x,y) -positions near the exciter wire were more robust against noise than those in the middle of the goal plane. Finally, the object coils were rotated around each of the (x,y,z) -axes with $\phi = 45^\circ$ and $\phi = 90^\circ$. As a result, the localization accuracy was more vulnerable to errors for the (x,y) -positions near the the exciter wire.

The methods and the results of this work can be used by the engineers during the design process of the systems. For example, the outcomes of the CRLB investigations can be helpful in choosing the sensors constellation for a specific localization accuracy limit.

Appendices

A. Magnetic Sensors Types

"Magnetic sensors can be classified according to whether they measure the total magnetic field or the vector components of the magnetic field."([7], p. 631) In our investigations we will be using stationary vector magnetometers.

The principle behind this type of magnetometer is Faraday's law of induction (Section 2.2). If the magnetic flux through a coiled conductor changes, a voltage proportional to the rate of change of the flux is generated between its leads. The flux through the coil will change if the coil is in a magnetic field that varies with time, if the coil is rotated in a uniform field, or if the coil is moved through a nonuniform field. ([7], p. 632)

Sensors of this type can detect fields as weak as 10^{-10} mT ([8], Table 48.1) and their useful frequency range is typically from 0.1 Hz to 1 MHz ([8], Table 48.1), which is suitable for our localization applications.

B. Algorithm 1

```
import numpy as np

def goalDetect(U_ind):
    goalEvent = 0
    for i in range(0, len(U_ind[i]) - 1):
        buffer = U_ind[i]
        if np.sign(buffer) != np.sign(U_ind[i + 1]):
            goalEvent = 1
            break
    return goalEvent
```

C. Algorithm 2

```
import numpy as np

def loc2D(grid, U_9sim, U_9, E_thr):
    euclideanDistances = []
    for p in range(0, len(grid)):
        euclideanDistances.append(np.sqrt
                                   (np.square(U_9-U_9sim[p])))
    possiblePositions = []
    for i in range(0, len(euclideanDistances)):
        if euclideanDistances[i] < E_thr:
            possiblePositions.append([grid[i][0],

    return possiblePositions
```

D. Algorithm 3

```
import numpy as np

def loc2D_min(grid, U_n, M_n):
    euclideanDistances = []
    for p in range(0, len(grid)):
        sum = 0
        for n in range(0, len(M_n)):
            sum += np.square(M_n[n] - U_n[p][n])
        euclideanDistances.append(np.sqrt(sum))
    minDist = min(euclideanDistances)
    xyIndex = euclideanDistances.index(minDist)
    xPos = grid[xyIndex][0]
    yPos = grid[xyIndex][1]
    return xPos, yPos
```

E. Plotting Figure 6.6a

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors

# Plotting a heatmap for the voltage
# zero crossings of three sensors

plt.rc('font')
plt.rc('xtick', labelszize=20)
plt.rc('ytick', labelszize=20)
fig, ax = plt.subplots()
goalWidth = 1.90
goalHeight = 1.25
ypos = np.linspace(0.05, goalHeight - 0.05, num=47, endpoint=True)
xpos = np.linspace(0.05, goalWidth - 0.05, num=73, endpoint=True)
y, x = np.meshgrid(ypos, xpos)
z = []
for i in range(0, len(self.sum_xyz_list_threeSensors), len(ypos)):
    temp = self.xyz_list4[i:i + len(ypos)]
    temp1 = []
    for element in temp:
        temp1.append(element[2])
    z.append(temp1)

z = np.array(z)
for i in range(0, len(z)):
    for ii in range(0, len(z[i])):
        z[i][ii] = z[i][ii] * 1000
ticks = np.linspace(z.min(), z.max(), 10)
for i in range(0, len(ticks)):
    ticks[i] = "%.2f " % ticks[i]
bounds = ticks
norm = colors.BoundaryNorm(boundaries=bounds, ncolors=256)
pcm = ax.pcolormesh(x, y, z, norm=norm, cmap='jet')
cbar = fig.colorbar(pcm, ax=ax, orientation='vertical')
cbar.set_label('error (mm)', size=22)
ax.set_ylabel('Y (m)', fontsize=26)
ax.set_xlabel('x (m)', fontsize=26)
plt.show()
```


F. Plotting Figure 6.11

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors

# Plotting 4 heatmaps, one for each sensor
# with the error distribution for one degree
# tilted sensors
plt.rc('font')
plt.rc('xtick', labelsz=16)
plt.rc('ytick', labelsz=16)
fig, axes = plt.subplots(2,2)

fig.subplots_adjust(wspace = 0.1, hspace=0.2, right=0.8)
axes = axes.ravel()
x1=[];y1=[];z1=[];x2=[];y2=[];
z2=[];x3=[];y3=[];z3=[];x4=[];y4=[]
; z4=[]; z_2D1=[]; z_2D2=[]; z_2D3=[]; z_2D4=[]
for i in range(1,5):
    x = eval("x%d" % i)
    y = eval("y%d" % i)
    z = eval("z%d" % i)
    xyz_list = eval("xyz_list%d" % i)
    # Preparing the lists for plotting heatmaps
    for element in xyz_list:
        x.append(element[0])
        y.append(element[1])
        z.append(element[2] * 1000)
    z_2D = eval("z_2D%d" % i)
    for ii in range(0, len(z), len(yValues_Extracted)):
        tempList = []
        tempList = z[ii:ii + len(yValues_Extracted)]
        z_2D.append(tempList)
    # Setup the 2D grid with Numpy
    x, y = np.meshgrid(xValues_Extracted, yValues_Extracted)
    # Convert intensity (list of lists)
    # to a numpy array for plotting
    z = np.array(z_2D)
    tiks = [1,2,3,4,5,6,7,8]
    bounds = tiks
    norm_mess = matplotlib.colors.BoundaryNorm
        (boundaries=bounds, ncolors=256)
    ax = axes[i-1].pcolormesh(x, y, np.transpose(z),
        norm=norm_mess, cmap='jet')
    axes[i-1].set_ylim(bottom=min(yValues_Extracted)
        , top=max(yValues_Extracted))
```

```

axes[i-1].set_xlim(left=min(xValues_Extracted)
                    ,right=max(xValues_Extracted))
axes[i-1].set_ylabel('Y (m)', fontsize=18)
axes[i-1].set_xlabel('x (m)', fontsize=18)
if i !=4:
    axes[i-1].set_title('Sensor'+str(i), fontsize=20)
else:
    axes[i-1].set_title('Sum of the 3 sensors', fontsize=20)
cax = plt.axes([0.85, 0.1, 0.01, 0.8])
cbar = plt.colorbar(ax,cax=cax,ticks=tiks)
cbar.set_label('error (mm)', size=18)
plt.tick_params(labelsize=18)
plt.xticks(fontsize = 18)
plt.yticks(fontsize=18)
plt.show()

```

G. Plotting Figure 6.30a

```

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors

# Plotting one heatmap for the
# CRLB distribution over the goal plane
# using 10 sensors and noise level of 1e-8V
slopes = []
for i in range(0, len(main), 2):
    temp=[]
    temp.append(main1[i+1]-main1[i]/0.001)
    temp.append(main2[i + 1] - main2[i] / 0.001)
    temp.append(main3[i + 1] - main3[i] / 0.001)
    temp.append(main4[i + 1] - main4[i] / 0.001)
    temp.append(main5[i + 1] - main5[i] / 0.001)
    temp.append(main6[i + 1] - main6[i] / 0.001)
    temp.append(main7[i + 1] - main7[i] / 0.001)
    temp.append(main8[i + 1] - main8[i] / 0.001)
    temp.append(main9[i + 1] - main9[i] / 0.001)
    temp.append(main10[i + 1] - main10[i] / 0.001)
    slopes.append(temp)
sigma = 1e-8
CRLB = []
for elment in slopes:
    sum = 0

```

```

    for slope in element:
        sum = sum + (slope*slope)
    CRLB.append(np.sqrt
        (sigma*sigma/float(sum))*1000)
# Preparing the heat map
xpos = np.arange(0.05, 1.86, 0.01)
ypos = np.arange(0.05, 1.21, 0.01)
x= xpos.tolist()
y = ypos.tolist()
x,y = np.meshgrid(x,y)
z= []
for i in range(0,len(CRLB),len(ypos)):
    temp = CRLB[i:i + len(ypos)]
    z.append(temp)
# Plotting the HeatMap
plt.rc('font')
plt.rc('xtick', labelsizes=16)
plt.rc('ytick', labelsizes=16)
fig, ax = plt.subplots()
tiks=[0,0.03,0.06,0.09,0.12
      ,0.15,0.18,0.21,0.24,0.27,0.3,0.33]
bounds = tiks
norm = colors.BoundaryNorm(boundaries=bounds,
    ncolors=256)
pcm = ax.pcolormesh(x, y,
    np.transpose(z), norm=norm, cmap='jet')
cbar = fig.colorbar(pcm, ax=ax,
    orientation='vertical', ticks=tiks)
plt.axis([x.min(), x.max(), y.min(), y.max()])
cbar.set_label(r'$\sqrt{\text{CRLB}}$ for est. z (mm)', size=22)
ax.set_ylabel('Y (m)', fontsize=26)
ax.set_xlabel('x (m)', fontsize=26)
plt.show()

```

Bibliography

- [1] HU Chao et al.: *An Efficient Magnetic Localization System For Indoor Planar Mobile Robot*. Proceedings of the 34th Chinese Control Conference, 2015: 4899-4904
- [2] Babak Moussakhani et al.: *On localizing a Capsule Endoscope using Magnetic Sensors*. 34th Annual International Conference of the IEEE EMBS San Diego, California USA, 2012: 4058-4062
- [3] Rafael Psiuk et al.: *Analysis of Goal Line Technology from the perspective of an electromagnetic field based approach*. Conference of the International Sports Engineering Association, 2014: 297 -284
- [4] David J.Griffiths, Reed College. *Introduction to Electrodynamics* (3rd ed.). Upper Saddle River, New jersey: Prentice Hall, 1999.
- [5] Klaus Finkenzeller. *RFID Handbook* (3rd ed.). Wiltshire, UK: Wiley Ltd, 2010.
- [6] John D.Jackson. *Classical Electrodynamics* (3rd ed.). USA: Wiley INC, 1999.
- [7] James Lenz, Alan S. Edelstein. *Magnetic Sensors and Their Applications*. IEEE SENSORS JOURNAL, VOL. 6: 2006: 631-649
- [8] Steven A. Macintyre. *Magnetic Field Measurement*. CRC Press LLC: 1999
- [9] Steven M. Kay. *Statistical Signal Processing* (Vol. 1, pp. 28-32). Upper Saddle River, New jersey: Prentice Hall, 1993.