Bachelor Project

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Indoor Robot Localization Using Magnetometer Data

Tomáš Fiala

Supervisor: Ing. Vladimír Smutný, Ph.D.
February 2022

Acknowledgements

I would like to express my deepest thanks to Ing. Vladimír Smutný, PhD. for his persistent help and much helpful suggestion.

Special thanks to Matěj Boxan, whose work I built on and who introduced me to the issue.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university thesis.

Prague, February 7, 2022

....

Signature

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 7. února 2022

....

Podpis autora práce

Abstract

The magnetic field in reinforced concrete buildings is often inhomogeneous and can be used for localization. Magnetometer measurements along with a previously known magnetic map can improve the odometry accuracy. This bachelor thesis deals with the localization of the ground mobile robot in the indoor environment using a magnetometer.

In the first part, we demonstrate how it is possible to record the inhomogeneity of the magnetic field inside buildings. Measuring the dataset and converting it into a magnetic map is necessary for possible localization.

In the second part, we design and implement localization in a magnetic field as an optimization problem. This localization uses the already mentioned magnetic maps to adjust the odometry.

Keywords: localization, mobile robot, magnetometer, magnetic field, ROS, odometry

Supervisor: Ing. Vladimír Smutný, Ph.D.
Prague, CIIRC,
Jugoslávských partyzánů 1580

Abstrakt

Magnetické pole v železobetonových budovách je často nehomogenní a lze ho využít k lokalizaci. Měření magnetometrem spolu s předem známou magnetickou mapou může zlepšit přesnost odometrie. Tato bakalářská práce se zabývá lokalizací pozemního mobilního robotu ve vnitřním prostředí za pomoci magnetometru.

V první části ukážeme, jak je možné zaznamenat nehomogenitu magnetického pole uvnitř budov. Naměření datasetu a jeho konverze na magnetickou mapu jsou klíčové pro možnou lokalizaci.

Ve druhé části navrhneme a implementujeme lokalizaci v magnetickém poli jakožto optimalizační úlohu. Tato lokalizace využije již zmíněné magnetické mapy k úpravě odometrie.

Klíčová slova: lokalizace, mobilní robot, magnetické pole, magnetometr, ROS, odometrie

Překlad názvu: Lokalizace robotu za pomoci magnetometru

Contents

Figures

Tables

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Fiala Tomáš**	Personal ID number:	**492185**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Cybernetics and Robotics**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Indoor Robot Localization Using Magnetometer Data

Bachelor’s thesis title in Czech:

Lokalizace robotu za pomoci magnetometru

Guidelines:

1. Get familiar with Robot Operating Systém and robot localization using NDT maps.
2. Measure intensity and 3D orientation of magnetic field in different buildings.
3. Propose an algorithm using the magnetometer data and magnetic field map to increase accuracy of robot localization.
4. Implement the algorithm, make experiments, evaluate results.

Bibliography / sources:

[1] Gozick, Brandon & pathapati subbu, Kalyan & Dantu, Ram & Maeshiro, Tomyo. (2011). Magnetic Maps for Indoor Navigation. IEEE T. Instrumentation and Measurement. 60. 3883-3891. 10.1109/TIM.2011.2147690.
[2] Wahlström, Niklas. (2013). Localization using Magnetometers and Light Sensors, Master Thesis.
[3] Montella, Corey. (May 2011). The Kalman Filter and Related Algorithms: A Literature Review.
[4] Siebler, Benjamin & Sand, Stephan & Hanebeck, Uwe. (2020). Localization With Magnetic Field Distortions and Simultaneous Magnetometer Calibration. IEEE Sensors Journal. PP. 10.1109/JSEN.2020.3024073.

Name and workplace of bachelor’s thesis supervisor:

Ing. Vladimír Smutný, Ph.D. Robotic Perception CIIRC

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **15.12.2021** Deadline for bachelor thesis submission: **15.08.2022**

Assignment valid until: **30.09.2023**

Ing. Vladimír Smutný, Ph.D. Supervisor’s signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt Student’s signature

Chapter 1

Introduction

Robot localization is the process of determining where a mobile robot is located with respect to its environment. A robot is typically equipped with sensors that observe the environment or measure the robot's inner parameters. The localization problem is to determine the robot's position based on the data from these sensors.

Calculating the position based on data from incremental angular sensors on robot wheels is called odometry. The robot's initial position in the environment is necessary for this localization method. In simulations, odometry data can be perfect. In real life, a minor defect on an angular sensor or model describing the robot, such as different tire pressures than expected, can cause an error in the position estimate. The errors accumulate over time.

Inertial measurement unit (IMU) provides angular velocity data. By integrating the data, we determine the robot's orientation. Those data can be merged with odometry data and provide a better estimate of the robot's position. This estimate is also affected by error accumulating over time.

Possible solution seems to be the use of magnetometer to measure the intensity and direction of the magnetic induction. In undisturbed environments on Earth including oceans or forests, Earth's magnetic field is locally homogeneous. If we could measure the Earth's magnetic field, we might improve the accuracy of the position estimate.

The Earth's magnetic field is locally homogeneous, however, measured magnetic field is not likely to be homogeneous. The source of the magnetic field is a permanent magnet or electric current. In many cases, load-bearing structures in modern buildings are made of ferromagnetic materials, which can be considered as permanent magnets. A magnetometer measures the sum of all magnetic fields. Therefore all other disruptive fields are part of the measurement. In an environment with disruptive magnetic fields, the orientation adjustment for example utilized in ocean ships navigation, cannot be used without certain modifications.

1.1 Problem definition

This work aims to improve the localization of the ground mobile robot in the indoor environment with the use of a magnetometer. If the magnetic field is inhomogeneous, it is necessary to have the knowledge of the magnetic induction in the area where we locate the robot. There are many ways to measure and represent this inhomogeneity.

One of the ways is to measure the magnetic induction at predefined points. This measurement can be done in a regular square grid. The resulting raster map contains magnetic induction values. For a continuous map, it is necessary to interpolate the raster.

Another way is navigating the robot equipped with a magnetometer in the measured area. We measure magnetic induction with high frequency along the robot's trajectory. The measurement result is a dataset with the magnetometer's values and position. Ideally, the individual measurements are evenly distributed over the measured area. The regular square grid with magnetic induction values can be generated from the dataset. In this case, the vector assigned to the individual grid points is not an exact value of the magnetic induction but only a weighted mean of the surrounding measurements. The resulting raster must again be interpolated. A raster with assigned magnetic induction values we call a magnetic map.

As already mentioned, robot localization can utilize various sensors and methods. The measured data from the magnetometer together with the magnetic map can improve the accuracy of the resulting localization. We assume that this improvement could enhance the localization accuracy in situations where the state-of-the-art method Normal Distribution Transform (NDT) fails. Such places are especially long corridors with parallel walls.

The robot localization using the data from the magnetometer and the magnetic map can be based on various principles. One of the options is to record data from a magnetometer and find the most probable position of the robot in the environment. This approach leads to solving the task of locating agent in a maze. Another option is the odometry adjustment, which ensures the most accurate match of the measured magnetic induction with the magnetic map throughout the robot's movement.

1.2 State of the art

The magnetic field inside buildings arises from natural and artificial sources, such as steel and reinforced concrete structures, electric power systems, and electric appliances. The magnetic field anomalies inside a building provide a unique magnetic fingerprint that can be utilized in global self-localization. Proposed Haverinen's and Kemppainen's navigation [1] is one-dimensional and shows the potential of magnetic field navigation in long narrow corridors. It is also worth mentioning that they registered a passage through the door and used it as a guiding point because of its iron frame.

The stability of the magnetic induction over time periods and the repeatability of the magnetometer measurements are studied in [2]. The authors also came up with a system of landmarks and guide points. Those came out of the nature of modern buildings. The guide points are explained as the effect of pillars made out of ferromagnetic material.

A holistic view of this issue is presented in [3]. Siebler and others demonstrate the localization in the magnetic field according to the magnetic map and simultaneous magnetometer calibration. It focuses on locating on a previously known route. It gives the example of determining the position of a model train on tracks. The proposed method is very complex using the Kalman filter. It is suitable for heavy platforms in industrial environments due to the possibility of calibrating the magnetometer only during the testing and use. Standard magnetometer calibration requires sensor motion around all axis, which can be difficult when the magnetometer is a part of a heavy platform.

Combining magnetometry data and light level data in an outdoor environment can be utilized for localization [4]. This alternative localization method is inspired by animal navigation. The research article draws attention to the fact that the magnetometer is not affected by dust, noise, humidity, etc.

The similarity of magnetic field navigation with submarine navigation is described in [6]. The method the authors propose needs a magnetic induction intensity map. The environment in which the described method is tested perfectly matches our case. The resulting localization accuracy is around 60 cm. The authors came up with a leader-follower scenario with even better results.

Reinforced concrete often makes it impossible to use GPS indoors. Orrient's so-called Indoor Positioning Technology [25] attempts to replace GPS in places where GPS fails with localization based on magnetic field inhomogeneity. They use a smartphone commonly equipped with a magnetometer, simple IMU and pedometer. Such localization has more potential in reinforced concrete buildings than GPS due to the possibility of 3D positioning. It is therefore possible to estimate even the floor in which the localized object is located.

Work dealing with the localization of people or ground robots [26] shows what localization results can be achieved using a combination of odometry data, magnetometer and magnetic map. They measured the magnetic map systematically by a robot using a ground-truth localization system. The resulting map has a resolution of 1 cm. They repeated the magnetic map measurements after 4 months and concluded that the magnetic field was stable. The average localization error in their experiments was 9 cm.

Chapter 2

Maps of magnetic field

2.1 Magnetometry data

Magnetometers measure magnetic induction along a sensor x, y, and z-axis. Accurate magnetic field measurements are essential for sensor fusion and determining the intensity and orientation of magnetic induction. Magnetometers typically need to be calibrated to compensate for environmental noise and manufacturing defects.

An ideal magnetometer measures magnetic induction along the orthogonal x, y, and z-axis. If magnetometer measurements are taken as the sensor is rotated through all possible rotations, the measurements should lie on a sphere. The calibration task is to find such calibration parameters that will result in the transformation of a general ellipsoid into a sphere.

The rotation of the robot equipped with a magnetometer to all possible orientations is complicated. If the robot moves only horizontally and one axis of the magnetometer is orthogonal to the floor, 2D calibration is sufficient.

Our sensor has undergone 2D calibration. We subtract the value of magnetic induction measured in the suburban park area in Prague from the values from the magnetometers z-axis. The value of the magnetic induction in the z axis is thus the difference between the measured magnetic induction and the magnetic induction caused by the Earth's magnetic field. The magnetic induction is calculated from the sensor output data $\mathbf{B}_,$ as follows:

$$\mathbf{B} = (\mathbf{B}_m - \mathbf{H}) \cdot \mathbf{S} \tag{2.1}$$

$$\mathbf{H} = \begin{pmatrix} 32676.85 \\ 32903.00 \\ 32306.34 \end{pmatrix} \tag{2.2}$$

$$\mathbf{S} = \begin{pmatrix} 1.00277 & 0 & 0 \\ 0 & 0.99724 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

However, sensor calibration does not tell us anything about the units we measure the magnetic field. The units are, therefore, relative. In this work, we would be satisfied with relative units, but for clarity, we will also state the ratio of the size of the measured magnetic induction with the Earth's magnetic induction on the surface. The dataset measured in the suburban park area in Prague will serve as a reference. We will denote this ratio unit by E. Furthermore, we can consider the value of the magnetic field on the surface as $0.2\ G = 0.2 \cdot 10^{-5}\ T$, measured at the observatory in Budkov near Prachatice [14]. We are aware that the value of the magnetic field in a city outdoor area and the observatory may differ. But it helps to put the measured data into context.

$$1\ E \approx 0.2 \cdot 10^{-5}\ T \tag{2.4}$$

2.2 Measurement procedure

When measuring a dataset of magnetometry data, it is necessary to determine the position of the magnetometer. We use a transformation matrix to describe the position of the magnetometer. The position is estimated by *NDT SLAM Respecting Visibility* implemented by Boxan [12]. The input of the *NDT SLAM* is the initial *NDT* map and lidar data. The *NDT* map determines the coordinate system. The initial *NDT* map can be generated from a *DXF* file, which is often part of building documentation. This conversion from *DXF* to *NDT* was implemented by Pánek [13]. It is worth mentioning the possibility of choosing *CAD* layers, which affect the final appearance of the map. In particular, glass walls are generally not detected by lidar, but when the surface is treated, for example, by sandblasting or matt adhesive foil, the glass is perceived as a solid wall. For each wall in *DXF* file, there are two lines representing its edges. If both of these lines extend into one *NDT* cell, the result is one gaussian in the middle of the wall. The lidar detects the edge of the wall, which the *NDT* method tries to match on the Gaussian representing the center of the wall. This results in a deterioration in the accuracy of the *NDT* method. If we know that we will not detect one side of the wall during the measuring procedure, then it is possible to remove the excess edge and represent the wall using only one line. The gaussian is then exactly where the lidar detects the wall. By suitable modification of the *DXF* file before conversion to the *NDT* map, it is therefore possible to achieve better results of the *NDT* method. The resulting *NDT* map also serves as a background for plotting the measured data.

We measure magnetic induction with the highest possible frequency. The robot equipped with the magnetometer can move during the measuring procedure. Navigating the robot systematically in the environment is necessary to provide an adequate dataset. The navigation is performed by the Robot Operating System (*ROS*) package `Move Base` together with the already mentioned *SLAM*. The inputs to the `Move Base` are robot poses (x, y, θ). Generating the poses appropriately, we cover the area with the measured magnetic induction samples.

2.3 Magnetic field map definition

Magnetic field in a point $(x, y, z)^T$ can be described as a vector of magnetic induction B.

$$\mathbf{B} = (b_x, b_y, b_z)^T \tag{2.5}$$

A single measurement of magnetic induction using magnetometer we denote as:

$$\widetilde{\mathbf{B}} = (b_x, b_y, b_z, x, y, z)^T, \tag{2.6}$$

where b_x, b_y, b_z are components of magnetic induction measured in point $(x, y, z)^T$ with respect to the origin of the map.

When measuring $\widetilde{\mathbf{B}}$ along a trajectory of the robot we obtain a dataset $\mathbf{M}$ containing n measurements.

$$\mathbf{M} = (\widetilde{\mathbf{B}}_1, \dots, \widetilde{\mathbf{B}}_n) \tag{2.7}$$

The robot moves only horizontally while measuring a dataset. The magnetometer is located in unchanging height. Therefore, all the measured data are in a plane. Coordinate z of all the measurements in the dataset is a constant.

The dataset needs to be converted into regular square grid with assigned vectors of magnetic induction. Those vectors represent the magnetic induction in the grid points as a weighted average of measurements obtained in its neighborhood. This grid with assigned vectors we call a magnetic map. The grid points are specified by the the coordinates x, y. The vector $\mathbf{B}$ assigned to the grid point with coordinates x, y is calculated as:

$$\mathbf{B}_{x,y} = \Big(\sum_{k=1}^{n} w_k\Big)^{-1} \sum_{k=1}^{n} w_k \cdot (b_{xk}, b_{yk}, b_{zk})^T, \tag{2.8}$$

where n is the number of measurements $\widetilde{\mathbf{B}}$ in the dataset $\mathbf{M}$. The vector $(b_{xk}, b_{yk}, b_{zk})^T$ is the magnetic induction from the measurement $\widetilde{\mathbf{B}}_k$. The weighting factor w_k is zero if the measurement coordinates $(x_k, y_k)^T$ are outside the neighborhood U of the grid point.

$$U = \{(x_r, y_r)^T \in \mathbb{R}^2, ||(x, y)^T - (x_r, y_r)^T|| < d\} \tag{2.9}$$

$$w_k = ||(x, y)^T - (x_k, y_k)^T||^{-1}, \;\; (x_k, y_k) \in U \tag{2.10}$$

$$w_k = 0, \;\; (x_k, y_k) \notin U \tag{2.11}$$

The parameter d in equation 2.6 is dependent on the properties of given dataset and the density of grid points.

2.4 Interpolation of the magnetic map

The magnetic map is a regular square grid in which components have a 3D vector assigned or are empty. If the component is empty then there are no measurements in the neighborhood U of given grid point. We can also look at the map as a plane in which we know the estimate of the magnetic induction at some points. If we want to get a value outside these points, we must interpolate the map. The interpolation by elements assigns a 3D vector to each point. By interpolating the magnetic map, we obtain the function f_m, which provides us with information about the magnetic induction at point $(x, y)^T$.

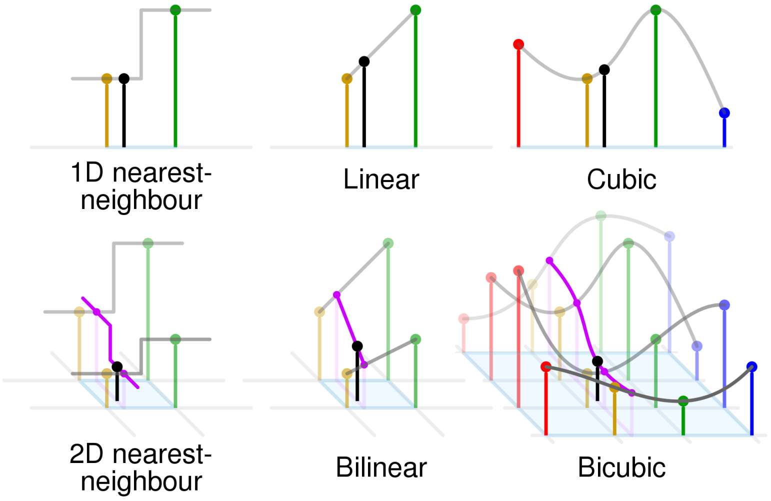


Figure 2.1: Comparison of 1D and 2D interpolations [11]

The choice of interpolation method depends on continuity, differentiability, or computational time requirements. The 2D-nearest-neighbor method finds the cell to which the point corresponds and uses its center's value as the output. This function is neither continuous nor differentiable.

Another interpolation method is a bilinear interpolation [8]. The resulting function of this method is continuous throughout the domain. It is not differentiable on the vertical and horizontal links connecting the grid points.

Some applications need both continuity and differentiability of the function. These requirements can be achieved with bicubic convolution interpolation [7]. The standard bicubic interpolation provides both continuity and differentiability but only on the eight-neigbor of the given cell. We consider cells whose boundaries are the vertical and horizontal links of the grid points.

We chose the bilinear interpolation for our application. Low computational time [10] is preferred for the optimization task. Derivation can be additionally defined in places where it does not exist. The existence of the derivative of the function f_m would allow the use of more advanced optimization methods. However, we will use such an optimization method in this work, which does not need the derivation defined.

2.5 Bilinear interpolation

Bilinear interpolation leads to a solution that gives us a continuous function. Vectors of magnetic induction estimated in the four nearest surrounding grid points to a point $(x, y)^T$ are used as an input into the interpolation.

$$\mathbf{m}_{i,j},\ i, j \in \{1, 2\} \tag{2.12}$$

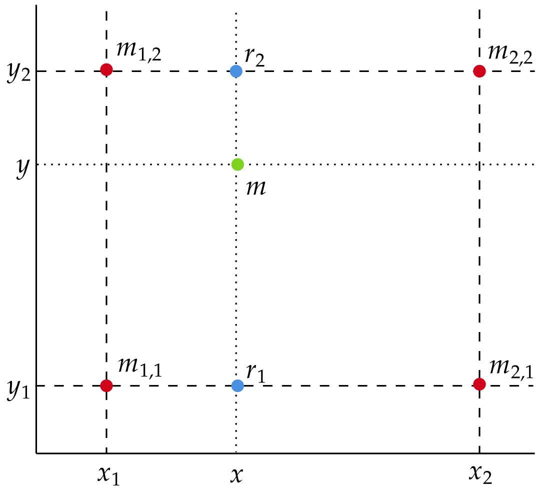


Figure 2.2: Bilinear interpolation diagram. The four red dots are the data points and the green dot is the point at which we want to interpolate.

Vector of magnetic induction is then calculated as:

$$\mathbf{m}(x, y) = \mathbf{r}_1 \cdot \frac{y_2 - y}{y_2 - y_1} + \mathbf{r}_2 \cdot \frac{y - y_1}{y_2 - y_1}, \tag{2.13}$$

$$\mathbf{r}_1 = \mathbf{m}_{1,1} \cdot \frac{x_2 - x}{x_2 - x_1} + \mathbf{m}_{2,1} \cdot \frac{x - x_1}{x_2 - x_1}, \tag{2.14}$$

$$\mathbf{r}_2 = \mathbf{m}_{1,2} \cdot \frac{x_2 - x}{x_2 - x_1} + \mathbf{m}_{2,2} \cdot \frac{x - x_1}{x_2 - x_1}. \tag{2.15}$$

Chapter 3

Robot localization using magnetometer data

Locating the robot in corridors with parallel walls is challenging in several aspects. The *NDT* fails when there is a lack of features in lidar data. In such cases, odometry is usually used as the primary source of data for localization. The error in odometry is not significant, but it can accumulate over time. This chapter will focus on ways to improve localization by using a magnetometer and magnetic map.

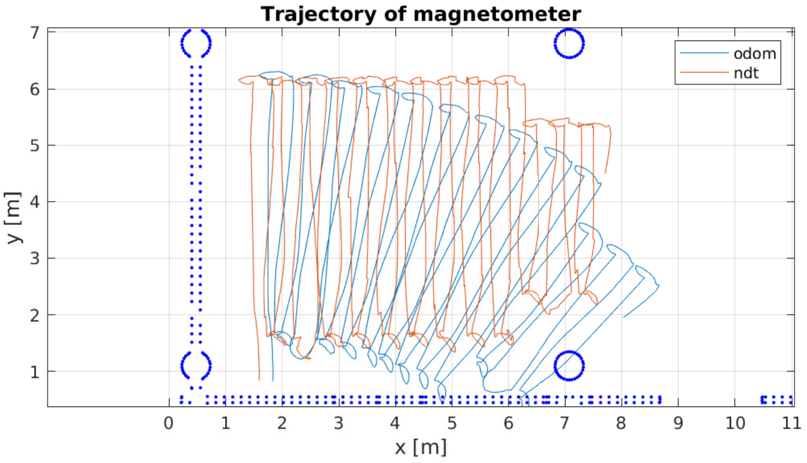


Figure 3.1: The top view of the B670 room. Dark blue parts indicate obstacles such as walls and pillars. The two curves show the trajectory of the magnetometer mounted on the robot when measuring magnetic field. The blue curve shows the trajectory estimated by odometry data only. The red curve shows the robot's trajectory estimated by NDT.

3.1 MAG method

The robot localization uses several sensors. Determining the position of a robot based on incremental angular sensors on the robot's wheels is called odometry. Determining the robot's initial position in the environment is necessary for this localization method. The *IMU* provides angular velocity data. By integrating these data, we determine the orientation of the robot.

A Kalman filter (*KF*) is an optimal estimation algorithm. The *KF* produces estimates of hidden variables based on uncertain and inaccurate measurements. The *KF* is based on linear dynamical systems discretized in the time domain. Common applications of *KF* include localization [17], computer vision systems, signal processing, etc. The Extended Kalman filter (*EKF*) is an extension of *KF* for non-linear systems where non-linearity is approximated using the first or second derivative.

We use Extended Kalman Filter (*EKF*) to combine odometry data and *IMU* data. The result of the *EKF* is the most probable position of the robot according to the *IMU* and odometry. The position estimate from the odometry and *IMU* we call filtered odometry.

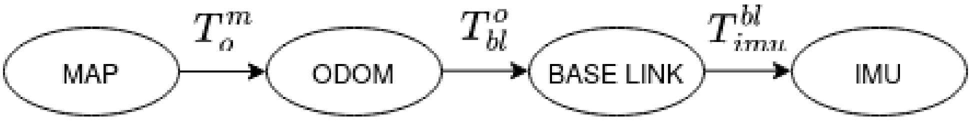


Figure 3.2: Diagram of important coordinate systems. Transformations between coordinate systems are stored in ROS TF.

The localization task is to find the transformation matrix T_{bl}^m as a function of time. The transformation $T_{bl}^m(t)$ can be described using two transformation matrices $T_o^m(t)$ and $T_{bl}^o(t)$. The transformation $T_{bl}^o(t)$ is the result of the filtered odometry. When considering the odometry localization method, the transformation matrix $T_o^m(t)$ is constant and is given by the initial position of the robot in the environment. Changing the transformation $T_o^m(t)$ can be considered as an adjustment of the odometry. This adjustment is provided by *NDT* or *MAG* method.

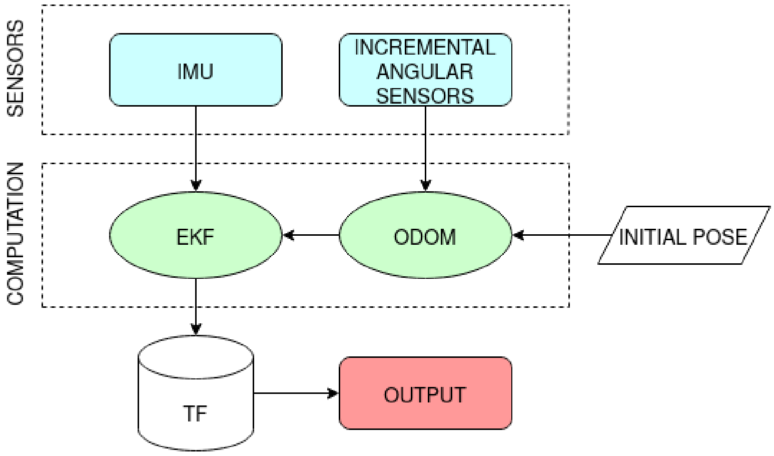


Figure 3.3: Scheme of localization using filtered odometry only. The arrows represent the data flow.

Filtered odometry can be burdened by an error that integrates over time. Therefore, it is necessary to use a method that is not affected by the integration error. This error can be prevented in most cases by using lidar data and the *NDT* method (see figure 3.3).

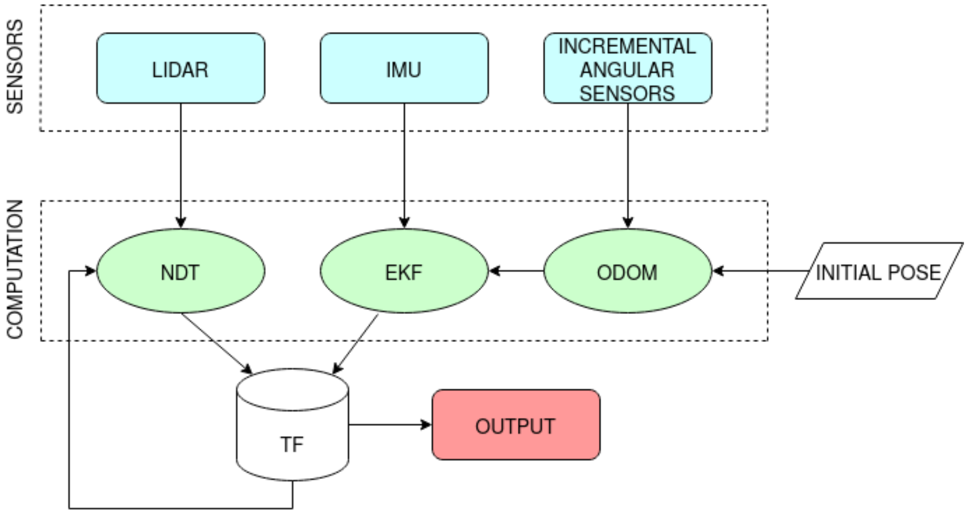


Figure 3.4: Scheme of localization using filtered odometry along with the NDT (NDT + ODOM). The arrows represent the data flow.

Using the knowledge of the magnetic field inhomogeneity together with the data from the magnetometer could positively affect the filtered odometry. The method using the latest position estimate, magnetometry data, and the magnetic map we call *MAG* method (see figure 3.4).

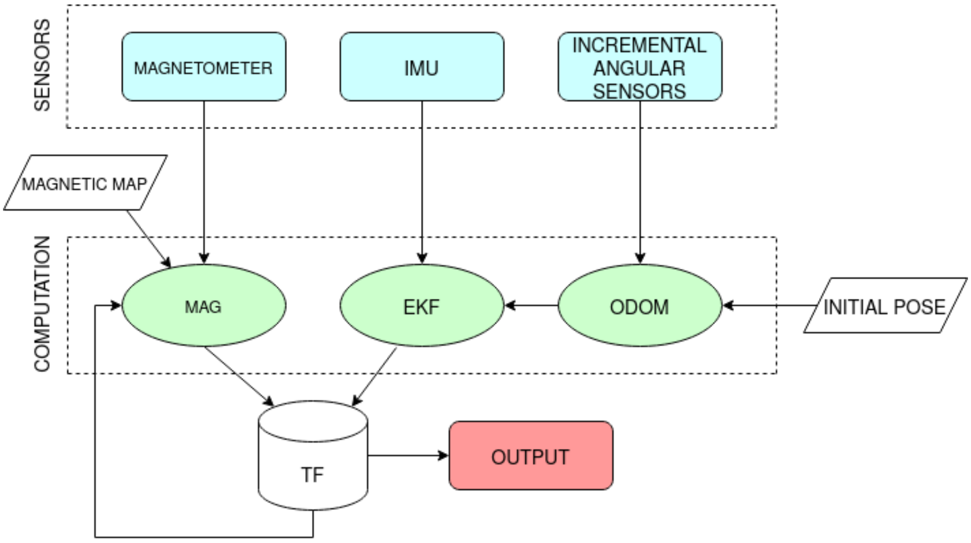


Figure 3.5: Scheme of localization using filtered odometry along with *MAG (MAG + ODOM)*. The arrows represent the data flow.

The *MAG* method adjusts the position estimate so that the measured magnetic induction matches the magnetic map better. We describe the change in a position estimate as an optimization task. By finding the minimum of the criterion function, we determine the new position estimate. The last robot's position estimate, the last measured magnetometry data, and the magnetic map are the input to the criterion function.

3.1.1 MAG algorithm

One iteration of the MAG method consists of 4 steps:

1. 1. Save the last magnetometer measurement and current transformations T^{m}_{imu} and T^{imu}_{o}.
2. Optimize the criterion function.
3. Calculate the new transformation T^{m}_{imu} based on the result from the optimization task.
4. Calculate and publish the transformation T^{m}_{o}.

3.2 Criterion function used in the MAG method

The mobile ground robot has three degrees of freedom. Therefore, we need three coordinates to describe the robot's position. The robot's position at time t is described in relation to the position at time $t-1$ obtained from the last iteration of *MAG*. The first coordinate is the distance traveled d. We can define the distance traveled d between times $t-1$ and t.

$$d_t = \left\| \begin{pmatrix} x \\ y \end{pmatrix}_t - \begin{pmatrix} x \\ y \end{pmatrix}_{t-1} \right\| \tag{3.1}$$

The robot's position at time t is on a circle with center $(x, y)^T$ (robot's position in time t-1) and radius d. The second coordinate is the oriented angle θ measured from the y axis, which determines the point on the circle where the robot is located. The third coordinate is the oriented angle α measured from the y-axis determining the robot's orientation.

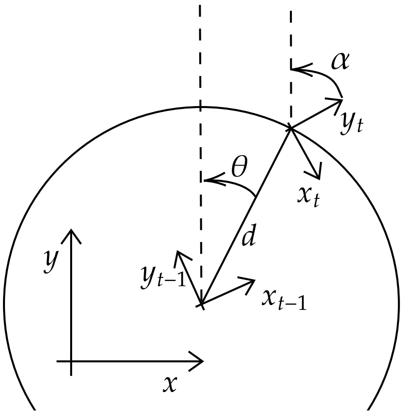


Figure 3.6: Coordinate diagram describing the position of the robot at time t according to the robot's position in time t-1.

The criterion function g, whose inputs are changes in coordinates around the last position estimate, has two parts. Part A expresses the difference between the measured magnetometry data and the magnetic map. Part B results in the preference of the original position estimate. Both parts of the criterion function are weighted by the parameter W.

$$g(\Delta\theta, \Delta\alpha, \Delta d) = W_A A + W_B B \tag{3.2}$$

Part A of the criterion function g is the magnitude of the vector of the difference between the measured values and the values from the magnetic map. This part of the criterion function prefers the change in the robot's position that will better match the measured values with the magnetic map. The part A is optimal when the vectors of magnetic induction from the map and measured match.

$$A = \left\| \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} - \begin{pmatrix} \widetilde{b}_x \\ \widetilde{b}_y \\ \widetilde{b}_z \end{pmatrix} \right\| \tag{3.3}$$

In section 2.4, we have defined the function f_m, which assigns an estimate of the magnetic induction to point $(x, y)^T$.

$$\begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = f_m\left(\begin{pmatrix} x \\ y \end{pmatrix}_t + (d + \Delta d) \cdot \begin{pmatrix} \sin\theta \\ \cos\theta \end{pmatrix}\right) \tag{3.4}$$

We rotate the magnetic induction vector measured by the magnetometer $(b_{mx}, b_{my}, b_{mz})^T$ around the z-axis when the α and θ coordinates change.

$$\begin{pmatrix} \widetilde{b}_x \\ \widetilde{b}_y \\ \widetilde{b}_z \end{pmatrix} = R_z(\alpha + \theta) \begin{pmatrix} b_{mx} \\ b_{my} \\ b_{mz} \end{pmatrix} \tag{3.5}$$

R_z is the rotation matrix around the z-axis.

$$R_z(\delta) = \begin{pmatrix} \cos\delta & -\sin\delta & 0 \\ \sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.6}$$

Part B of the criterion function g is a negative 3D Gaussian function. Covariances are assigned to individual coordinates. The part B is optimal with zero input.

$$B = -\exp\left(\frac{-\Delta\alpha^2}{2\sigma_\alpha^2} + \frac{-\Delta\theta^2}{2\sigma_\theta^2} + \frac{-\Delta d^2}{2\sigma_d^2}\right) \tag{3.7}$$

3.3 Testing the MAG method

The localization task is to find the transformation matrix $T_{bl}^{m}(t)$ in time. To test the *MAG* method, we propose three scenarios for the overall localization of the robot. The first scenario (see figure 3.3) is standard localization using *NDT*. The second scenario (see figure 3.4) consists in replacing the *NDT* method with the *MAG* method. The third scenario (see figure 3.2) is to locate the robot using filtered odometry only.

The result of the filtered odometry is the transformation matrix $T_{bl}^{o}(t)$. It is necessary to determine the transformation matrix $T_{o}^{m}(t)$ to estimate the robot's position on the map. If we use only filtered odometry, $T_{o}^{m}(t)$ is constant and is given by the robot's position on the map at the beginning of the experiment. In the case of a method involving *NDT* or *MAG*, the transformation matrix $T_{o}^{m}(t)$ changes over time using the feedback information from lidar or magnetometer.

By comparing the transformation matrices $T_{o}^{m}(t)$ in the mentioned scenarios, we verify the functionality and check the accuracy of the proposed localization using the magnetic map and magnetometer.

Chapter 4
Implementation

This section will introduce the experimental platform and describe relevant programs in detail. The code was included in the *ROS* package Jackal Magnetometer Experiments. The co-author of the package is Matěj Boxan, whose work precedes mine.

4.1 Experimental platform

The robot Jackal [19] from Clearpath Robotics forms the basic framework of our experiments. It is small, entry-level field robotics research platform, currently equiped with Sick TiM561 LIDAR [20] and Inertial Measurement Unit [21] (*IMU*). The *IMU* can operate as a magnetometer, which is crutial for this work. Another present accessories include a router, information LED interface, safety foam elements, ceiling pointing digital camera, and a plate with reflectors for the Vicon system.

The onboard computing power is distributed into two computers. The inner PC is reserved for robot control and data colection. The second computer (NUC PC) is placed on top of the mobile platform and provides the majority of the computing power. When performing experiments, it is advisable to use a third computer (portable laptop) to run such programs, which are not affected by a longer *ROS* response time, such as visualization.

Distribution of the important nodes and programs		
NUC PC	Laptop	Jackal inner PC
NDT SLAM Move Base Magnetometer driver Rosbag recorder	Mag-pose-handler Rviz	Jackal Node

Table 4.1: Distribution of the important *ROS* nodes and programs into the individual computers during the final experiment.

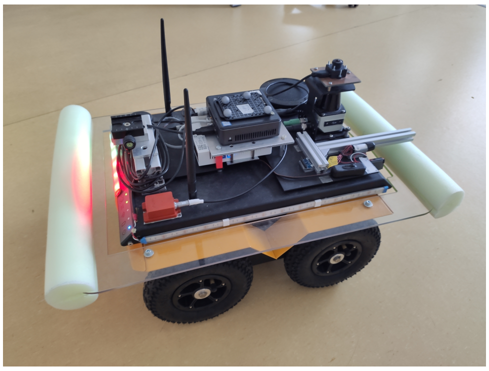


Figure 4.1: Robot Jackal from Clearpath Robotics equipped with sensors.

IMU Xsens MTi-30-2A5G4 offers inertial and orientation data. It also supplies 3-axis magnetometer data. The standard full range is ± 8 G with total RMS noise 0.5 mG and non-linearity 0.2 %. Resolution of the magnetometer is 0.25 mG. Sampling/output frequency is 10/2 kHz.

4.2 Robot Operating System

Robot Operating System (*ROS*) is a robotics middleware suit [18]. A *ROS* project usually consists of various nodes [16]. Topics serve for sending and receiving messages between nodes. A great deal of the work presented in this paper has been done with rosbags. Rosbags contain the record of message communication. For visualisation purposes we use *Rviz*.

We use Boxan's *NDT SLAM Respecting Visibility* [12]. The *NDT* map generated from a *DXF* file serves as a background map [13]. The size of each *NDT* cell is 25 cm. The *ROS* package *Move Base* provides navigation through the map. That allows us to navigate the robot to a particular pose (x, y, θ) on the map.

4.3 Mag-pose-handler program overview

The program `mag-pose-handler` is a *ROS* node that controls the robot and at the same time provides magnetometry data collection together with the positions of the magnetometer. The input of the program are pre-defined poses of the robot in map. The program sends individual poses to the *Move Base* and in the process creates a dataset, which can then be transformed into a magnetic map.

The node `mag-pose-handler` subscribes the topic `/imu_data_str` on which the magnetometer publishes its data. When the callback is triggered it reads the transformation T_{imu}^{map} and appends python dictionary with both magnetometry data and position of magnetometer in map.

The node reads a *csv* file in which are pre-generated poses for *Move base*. The path of the *csv* file can be set as *ROS* parameter `source_path`.

The program sends a pose message to the client and waits for the positive confirmation. Another pose is sent if the robot reaches the pose on the map. The remaining poses are published as markers to the topic `/poses_marker`. They can be visualized in *Rviz*. The visualization helps to assure that the poses were correctly read. The program saves the dictionary as *mat* file when all the poses are reached.

4.4 Ndt2poses-gui program overview

Creating poses for `mag-pose-handler` is provided by the `ndt2poses-gui` program. It is a graphical user interface program written in *Python*. The input of the program is an *NDT* map of the environment in which we want to control the robot. The user selects the area on the map that he wants the robot to systematically go through, and select the appropriate program settings.

The user must first select a *NDT* map in the search line and then enter data describing the selected map. The `Width`, `Height` and `Square` parameters describe the map. The `Width` and `Height` parameters indicate the size of the map. The `Square` parameter specifies the size of one *NDT* cell in meters.

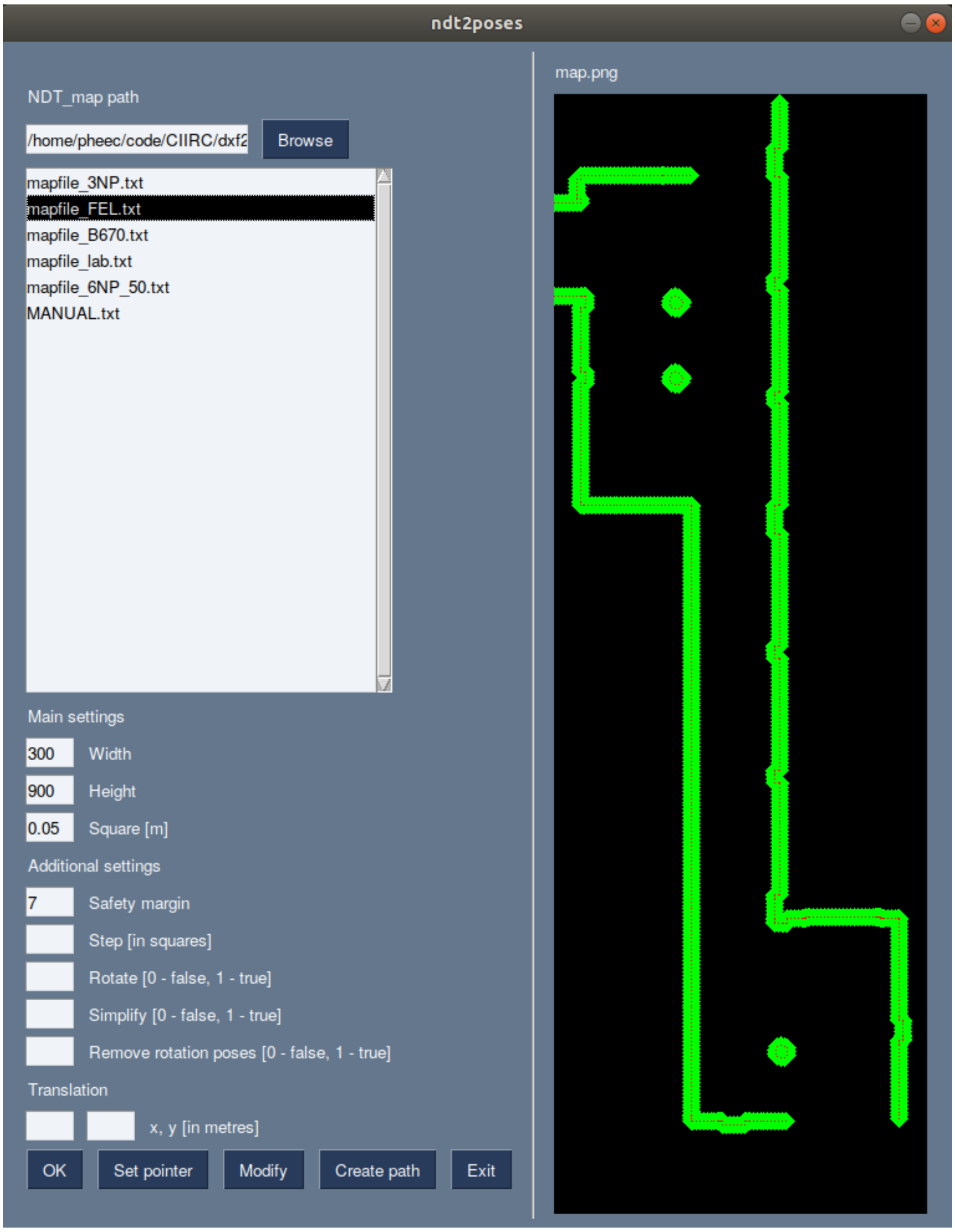
We convert the *NDT* map into a bit map representing the occupancy grid. If there is a nonzero gaussian in the cell, we denote it as occupied. One pixel corresponds to one *NDT* cell. The `Safety Margin` (SM) parameter determines how many times binary dilation is performed on the bitmap. With higher SM, free space on the map decreases. User can also modify the bitmap using the `Modify` button. Adding no-go zones will again reduce the free space on the map.

The `Step` parameter specifies the distance between the poses as the number of *NDT* cells between them. The `Rotate` parameter changes the preferred direction of the flood algorithm, which results in a change in the horizontal and vertical direction of the path.

To eliminate the odometry error that occurred during the simultaneous rotation and forward movement of the Jackal robot, we added so-called rotation poses. The movement of the robot is divided into movement in lines and rotation on the spot. These rotation poses can be removed by setting the `Remove Rotation Poses` parameter to 1 (the default value is 0).

In many cases, there is a situation where the movement in a straight line is divided into several poses. Excessive poses can be removed by setting the `Simplify` parameter to 1 (default value is 0).

In our case, the goal is the systematic movement of the magnetometer in the environment. If the sensor is not in the center of the robot (base link), it is necessary to enter this offset as the `Translation` parameter.



NDT_map path

/home/pheec/code/CIIRC/dxf2

mapfile_3NP.txt
mapfile_FEL.txt
mapfile_B670.txt
mapfile_lab.txt
mapfile_6NP_50.txt
MANUAL.txt

Main settings

300 Width
900 Height
0.05 Square [m]

Additional settings

7 Safety margin
Step [in squares]
Rotate [0 - false, 1 - true]
Simplify [0 - false, 1 - true]
Remove rotation poses [0 - false, 1 - true]

Translation

x, y [in metres]

Figure 4.2: The `ndt2poses` user interface with NDT map modified with Safety Margin

There may be several closed regions in the map where poses could potentially be generated. It is therefore necessary to select a specific one by using the `Set pointer` button

The `Create Path` button will generate a *csv* file with poses according to the settings listed above. The final shape of poses for each setting can be found in the appendix of this thesis.

The created path is displayed in the *GUI*. For a more accurate view, it is advisable to use the `plot_path()` function defined in the `generate_points.m` file in Matlab.

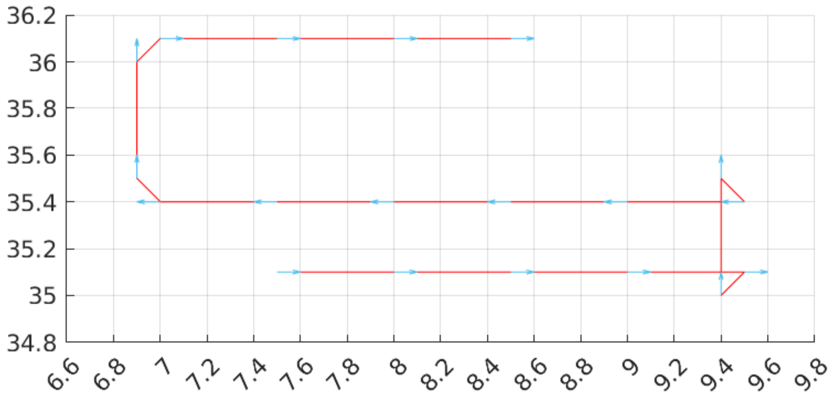


Figure 4.3: Visualization of several poses generated in `ndt2poses-gui`.

4.5 Magnetic maps

Conversion of datasets into magnetic maps is implemented in the Matlab file `mag-maps.m`. The Matlab file also contains magnetic map connection with the *NDT* map. Multiple datasets can be merged into one before the conversion.

The file `mag-map-class.py` contains a python class dealing with magnetic maps. It includes loading magnetic map files, printing magnetic maps, and implemention of bilinear interpolation.

All magnetic maps presented in this thesis are done with 20 cm resolution.

4.6 Mag-handler program overview

The node `mag-handler` represents the *MAG* block in the localization scheme *MAG + ODOM* (see figure 3.5). The filtered odometry is required when using the *MAG* method. When initializing, it first reads the config file containing the path to the magnetic map file, initial robot position, *MAG* parameters and magnetometer calibration parameters (the config file format is part of the appendix). Subsequently, the node loads the magnetic map using the python class for magnetic maps.

The node mag-handler subscribes to the topic `/imu_data_str` and reads transformations T_m^{imu} and T_{imu}^o. Magnetometer data needs to be modified with calibration parameters provided by the sensors calibration.

The program optimizes a criterion function described in section 3.2 using the *Nelder-Mead* method from scipy library. The *Nelder-Mead* optimization method is commonly used for optimization problems in which derivates are not known.

The result of the optimization task is the new estimate of the magnetometer position. This result reflects in the transformation T_m^o, which is calculated as a matrix multiple of the new T_m^{imu} and T_{imu}^o. The node publishes the transformation T_m^o.

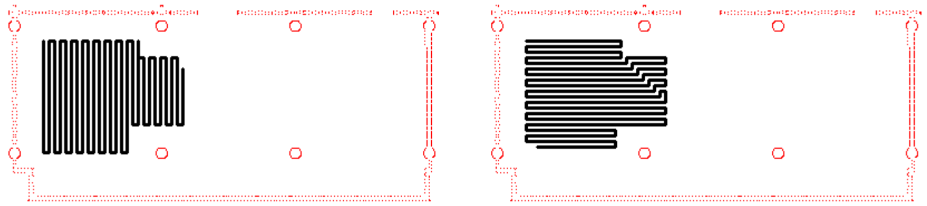
Chapter 5

Experimental results

5.1 Magnetic maps

5.1.1 Auditorium B670 in the CIIRC building

The auditorium B670 is located in the new part of the Czech Institute of Informatics, Robotics, and Cybernetics (CIIRC) building. The load-bearing structure in this part of the building is made out of reinforced concrete [24]. The main load-bearing elements are round concrete pillars. The pillars, floor, and ceiling are reinforced with rebars. Furniture in the form of benches and chairs was removed before the measurement.



(a) : The path of magnetometer when performing the first measurement procedure.

(b) : The path of magnetometer when performing the second measurement procedure.

Figure 5.1: The floor plan of the B670 room with solid obstacles (red) such as walls and pillars. The black curve is the predefined trajectory of the magnetometer measuring the magnetic field.

The resulting dataset consists of two measurement procedures. The robot trajectories during the measurement procedure were chosen to form a rectangular network. We mapped the back part of the auditorium measuring approximately 7x5 meters.

The measurement was performed twice with a time interval of four weeks to confirm the stability of the magnetic field. The similarities between two maps constructed four weeks apart suggest that the magnetic field is temporally stable over long periods of time. We therefore conclude, that the magnetic field is generated by ferromagnetic structures, not by electrical currents, which are mostly alternating.

Although the axis z of the magnetic induction is not calibrated with respect to the other two axes, it provides information about the nature of the magnetic anomalies. We could not find a relationship between the shape of the magnetic map and the distribution of ferromagnetic material in the building or another interfering source.

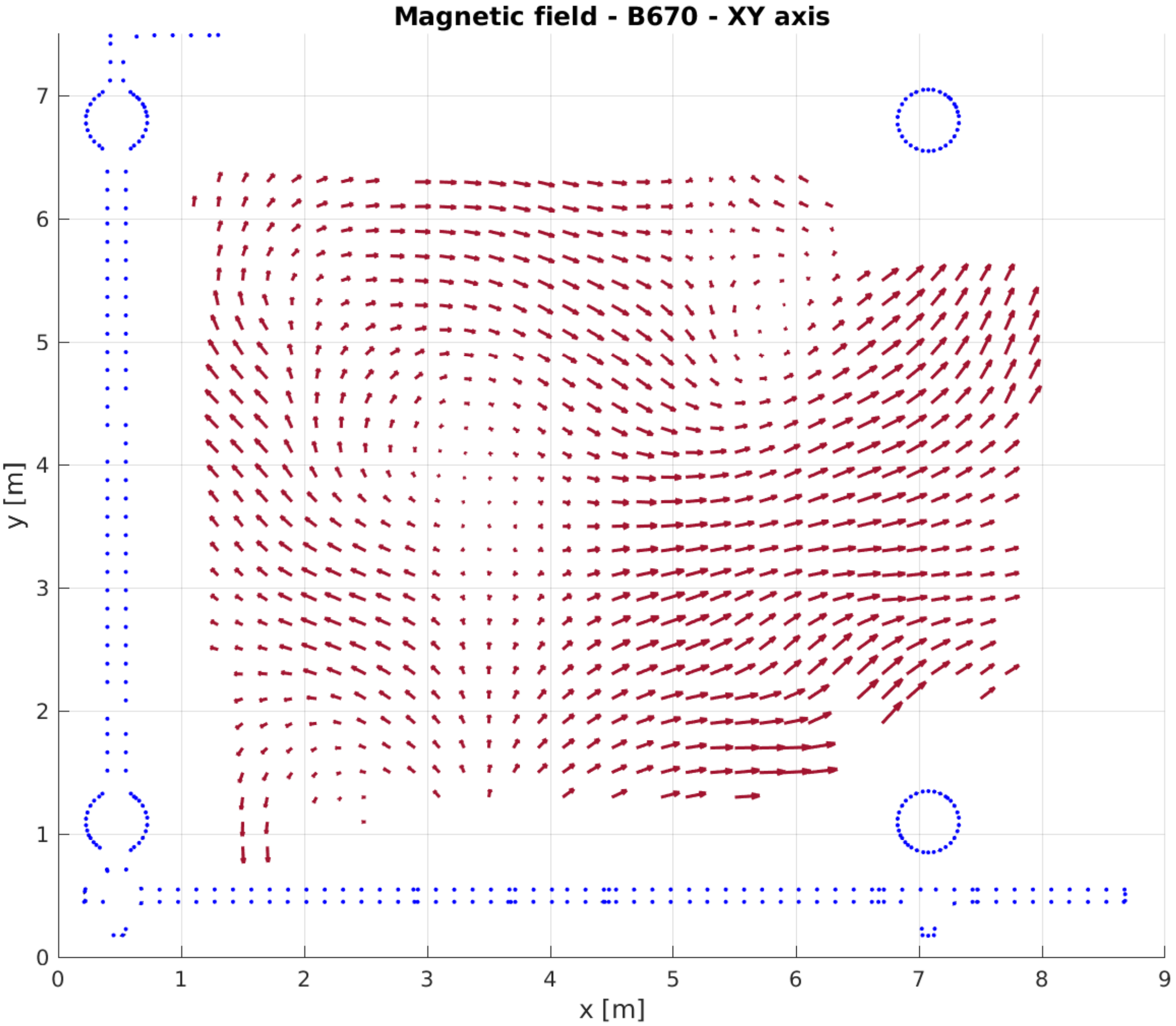


Figure 5.2: The floor plan of the B670 room. Dark blue parts indicate obstacles such as walls and pillars. The arrows represent the x and y component of the magnetic map as 10 E/m.

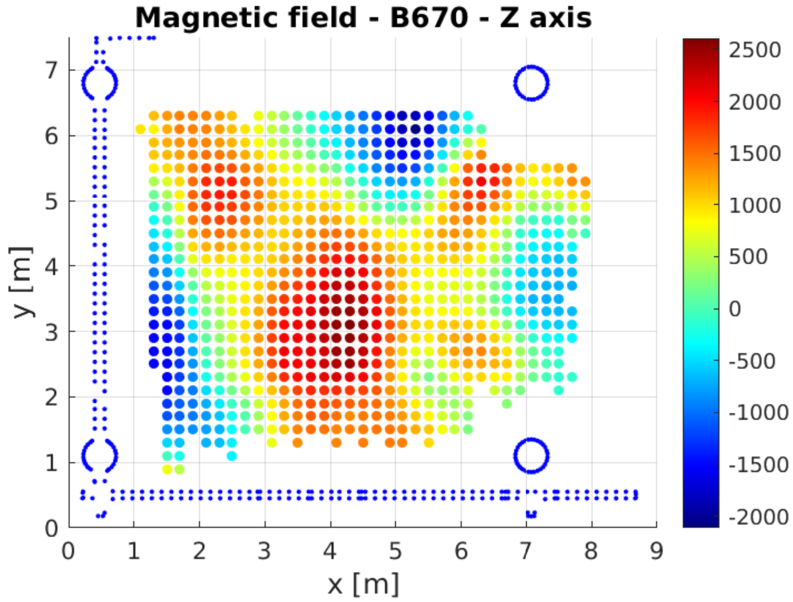


Figure 5.3: The floor plan of the B670 room. Dark blue parts indicate obstacles such as walls and pillars. The colors represent the z component of the magnetic map. The values of magnetic induction are in mE units.

5.1.2 Corridor of the 6th floor of the CIIRC building

Together with room B670, we also measured the magnetic field on a section of the corridor on the same floor of the CIIRC building. The measurement was repeated several times due to the unreliability of the NDT SLAM between the parallel walls. The inhomogeneity seems to permeate the entire new part of the CIIRC building.

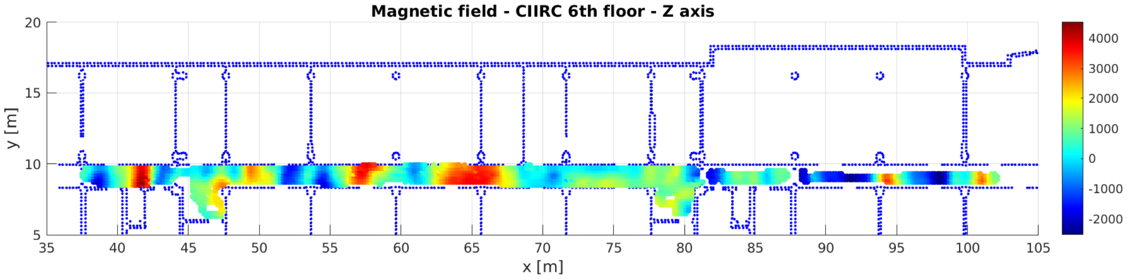


Figure 5.4: The floor plan of the 6th floor of the CIIRC building. Dark blue parts indicate obstacles such as walls and pillars. The colors represent the the z component of the magnetic field map. The values of magnetic induction are in mE units.

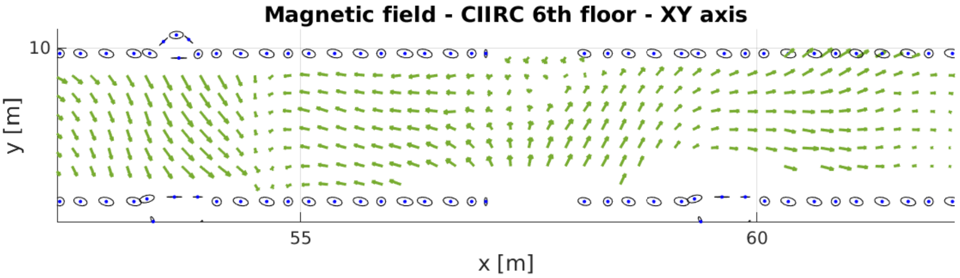


Figure 5.5: The section of floor plan in which we tried measuring the magnetic field with a hand compass. Blue parts indicate obstacles such as walls and pillars. The arrows represent the x and y component of the magnetic map as 10 E/m.

A magnetic field of 1E or higher should be measurable not only with a magnetometer but also with a hand compass. To test this hypothesis, we selected a section of the corridor where the magnetic field in the horizontal plane turns 180 degrees over a short distance (see figure 5.5). The compass pointed in the direction of magnetic induction from the magnetic map and was dynamically changing when moving through the corridor.

5.1.3 The 3rd floor of the CIIRC building

The measurements on the third floor of the CIIRC building differed from the previous in using the *Vicon system* [22]. The *Vicon system* allows the robot's localization with a millimeter accuracy.

It was necessary to mount a plate with reflectors to the robot, the position of which is tracked by the *Vicon*. We placed the plate in the base link of the robot (see figure 4.1).

The measurement procedure took 90 minutes, during which poses for `Move Base` were randomly generated. The result of this measurement procedure was a dataset of magnetometry data and a rosbag containing all the messages and ground-truth position of the robot provided by the *Vicon system*. The functional space of the *Vicon system* and its surroundings offer a large number of lidar features. Thanks to the measured rosbag, we were able to compare the ground-truth robot's position with the result of the *NDT + ODOM* method. If we do not consider the moments when it was necessary to rescue the robot due to blocking by an obstacle manually, the *NDT* method was able to eliminate the integration error of odometry.

The third floor is the old part of the building [23], which has undergone only a minor renovation [24]. The main supporting elements are again reinforced concrete rectangular pillars. One of these is in the middle of the mapped area. Since this building was built in 1975, we expected that the disturbing elements would not be so significant. This hypothesis has been confirmed. There is not a single vortex in the map. The magnetic field in this part of the building is more homogeneous.

A unique magnetic imprint is created around a reinforced concrete pillar, near which we observe the most significant intensities and directions of magnetic induction. The values change in a relatively small interval in the z-axis, but the most extensive changes are around the already mentioned pillar.

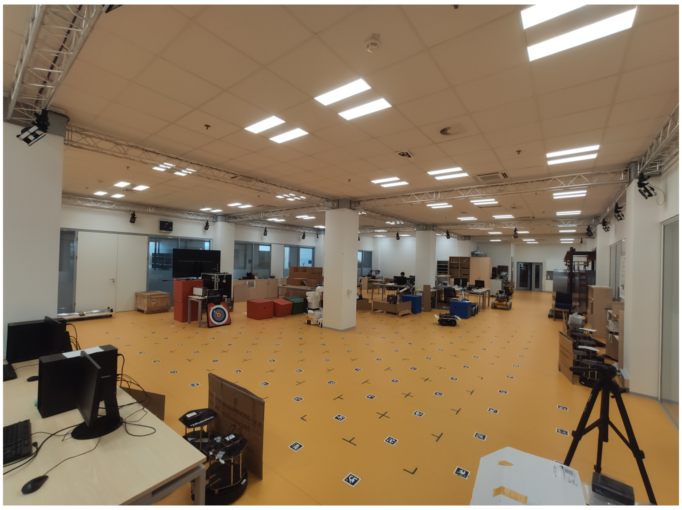


Figure 5.6: The *Vicon system* on the third floor of the CIIRC building.

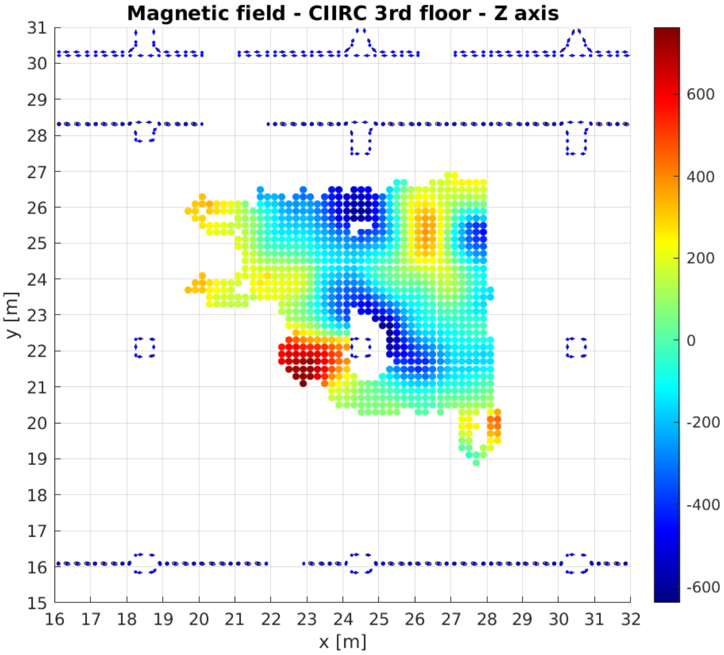


Figure 5.7: The floor plan of the 3$^{\text{rd}}$ floor of the CIIRC building. Dark blue parts indicate obstacles such as walls and pillars. The colors represent the z component of the magnetic map. The values of magnetic induction are in mE units.

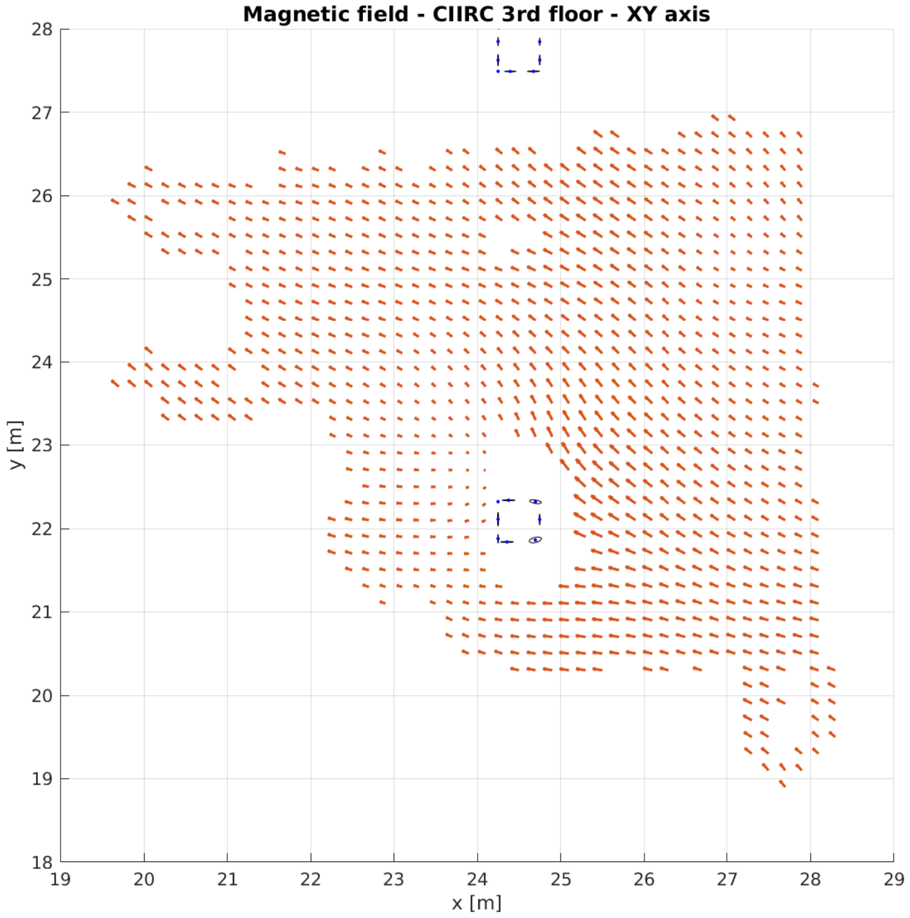


Figure 5.8: The floor plan of the 3$^{\mathrm{rd}}$ floor of the CIIRC building. Dark blue parts indicate obstacles such as walls and pillars. The arrows represent the x and y components of the magnetic map as 10 E/m.

5.1.4 The Building of the Faculty of Electrical Engineering

Since its construction, the FEE building from 1960 has not undergone significant reconstruction. The main load-bearing structure consists of a reinforced concrete monolithic skeleton lined with bricks. The spiral mushroom system forms the ceiling [23]. We chose a part of the corridor in section C4 on the 5th floor for the measurement. The walls form almost a square room with one column in the middle.

We expected the magnetic interference to be similar to that on the third floor of the CIIRC building. The x and y components of the magnetic map are almost homogeneous. There is not a single vortex. The magnitude of the magnetic induction fluctuates but not significantly.

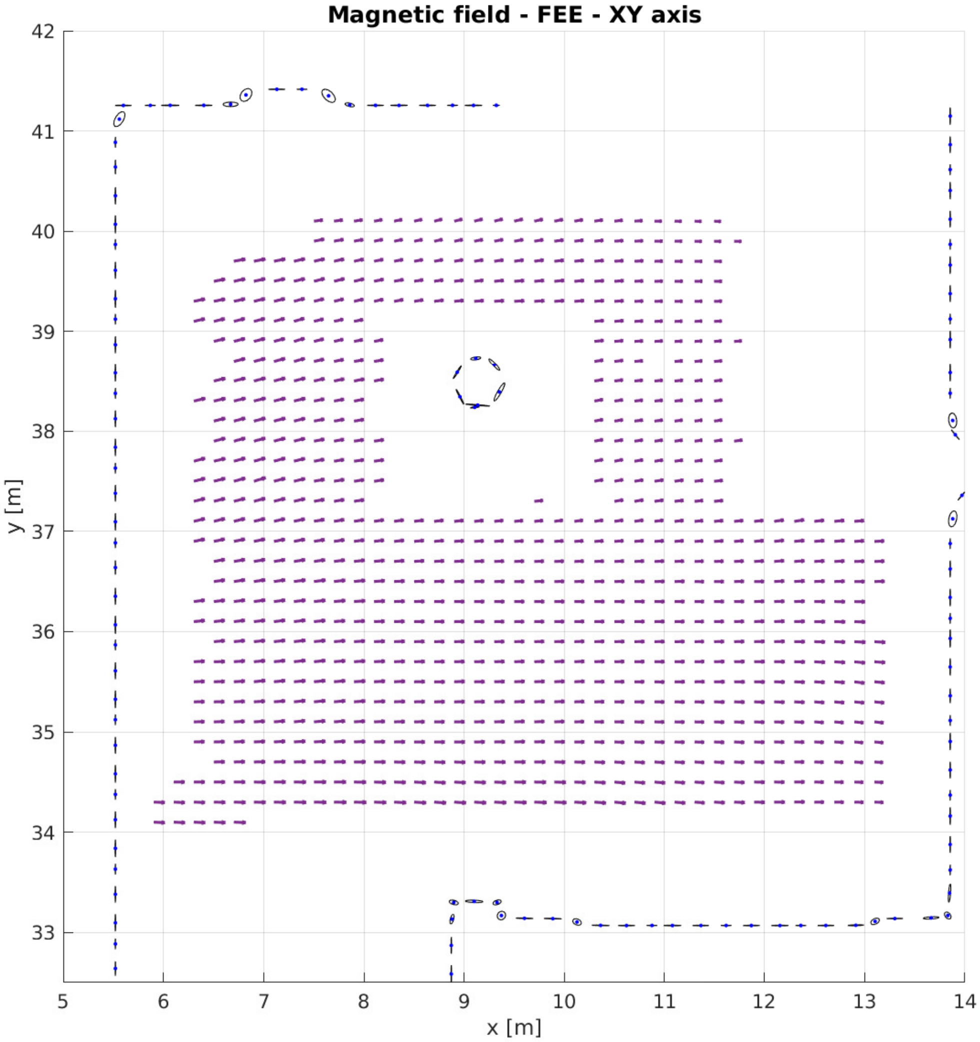


Figure 5.9: The floor plan of the measured area in the FEE building. The arrows represent the x and y components of the magnetic map as 10 E/m.

An error probably occurred during the measurement, which added an offset to an angle of magnetic induction in xy plane of the magnetic map. This inaccuracy should not affect the interpretation of the data.

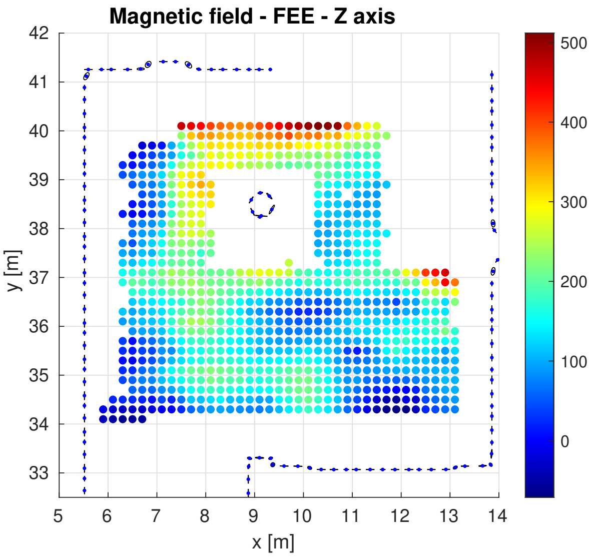


Figure 5.10: The floor plan of the measured area in the FEE building. The colors represent the z component of the magnetic map. The values of magnetic induction are in mE units.

5.2 Robot localization

5.2.1 Magnetometer noise

The magnetometer's noise and the robot's effect on the magnetic field is crucial for measuring and using magnetic maps. The robot's electric motors generate a magnetic field. We placed the robot with a magnetometer on a base that allowed the wheels to turn without the robot moving. We performed two experiments within a short time interval.

During the first experiment, the robot was inactive. We did 500 magnetic induction measurements. In the second experiment, we set the robot's speed to 0.5 km/s. It resulted in even wheel rotation. Again, we made 500 measurements.

The standard deviation at the first measurement was 21.06 mE. The standard deviation of the second measurement was 41.92 mE. The effect of the robot's motor torque has almost doubled the standard deviation. Due to the torque of the motors, it resulted in an offset of the magnetic induction values.

When comparing the standard deviation with the change in magnetic induction in the magnetic map (B670), the difference is approximately 5 mm. We therefore conclude that the sensor noise is negligible. We also consider the offset that occurred during the steady torque of the robot's motors to be negligible. However, finding the dependence of the magnetic induction offset on wheel rotation could improve the results.

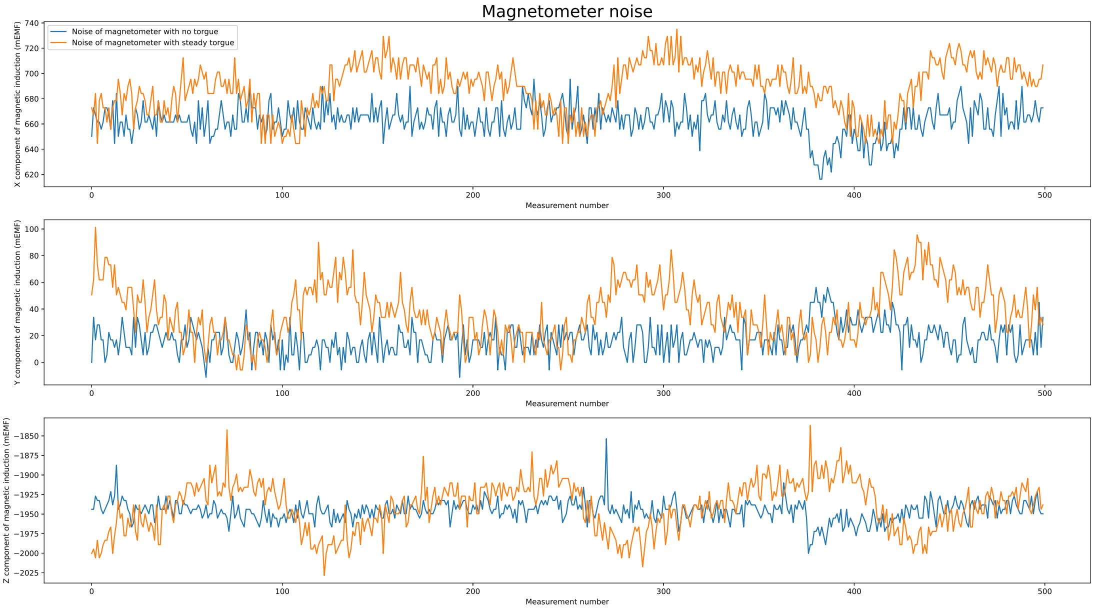


Figure 5.11: Measured magnetic induction during the robot's inactivity (blue curve) and with the robot's wheels rotating (orange curve).

Magnetometer noise		
	No torque	Steady torque
Standard deviation	21.06 mE	41.92 mE
X offset	-	26.15 mE
Y offset	-	21.03 mE
Z offset	-	5.41 mE

Table 5.1: Results of the experiments determining the noise of the magnetometer and the robot's effect on the magnetic field.

5.2.2 Localization at uniform motion with one degree of freedom

The robot's localization becomes more challenging with the increasing degrees of freedom. In our case of an indoor ground robot, we have three degrees of freedom. First, we will try to locate the robot with only one degree of freedom. We will consider the uniform motion of the robot. This experiment might show the accuracy of the localization using only a magnetometer.

By keeping the robot in the middle of the corridor with parallel walls, we achieve movement in a straight line. The robot thus loses one degree of freedom. If we limit the robot's movement to only one direction, we fix the rotation angle and thus get to the localization with one degree of freedom.

The robot measures magnetometry data along a trajectory at the highest possible frequency. We know that magnetic induction samples are measured equidistantly because of the assumption of uniform motion.

We get data from the magnetic map along the line that the robot was moving along. After fitting the measured magnetic induction on the magnetic map, it turns out that the robot stopped at 65.85 meters. We used the least squares method for fitting the curves. We minimized the square distance of the points on the curves. The optimized parameter is the length of the robot's trajectory. After measuring the position with a tape measure, we determined that the robot's actual position was 65.45 meters.

The high spatial variability of the magnetic field allowed us in this simple case to locate the robot with an accuracy of less than 0.5 m. The aim of future work should be to achieve similar accuracy, when extending the problem to 3 degrees of freedom.

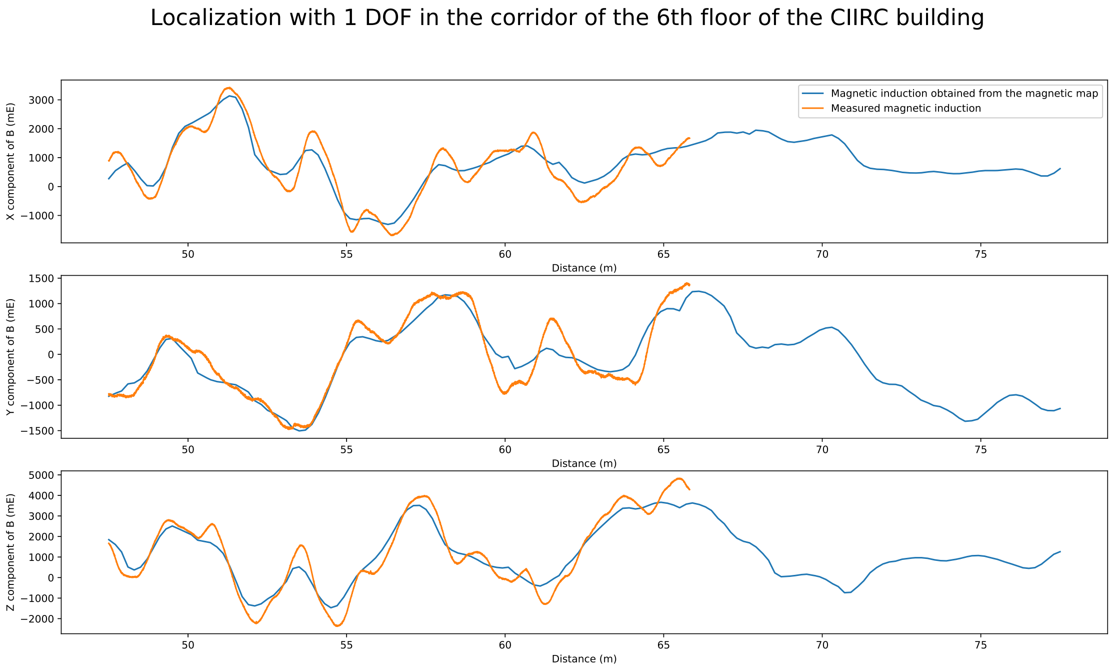


Figure 5.12: Results of robot's localization at uniform motion with one degree of freedom. The orange curve represents the measured data on the estimated position of their measurement.

The process of creating magnetic map has the characteristics of a low-pass filter (see figure 5.12). The data from the magnetic map are more smooth than the measured data. The individual peaks of the magnetic induction measured along the trajectory are higher than the data from the magnetic map.

5.2.3 MAG method

To test the MAG method, it was necessary to record a rosbag while the robot was moving in the area with the mapped magnetic field. We chose the B670 auditorium room for the experiment as it has an inhomogeneous magnetic field and has suitably distributed solid obstacles for the *NDT* method. Four columns complement the straight walls at the back of the room.

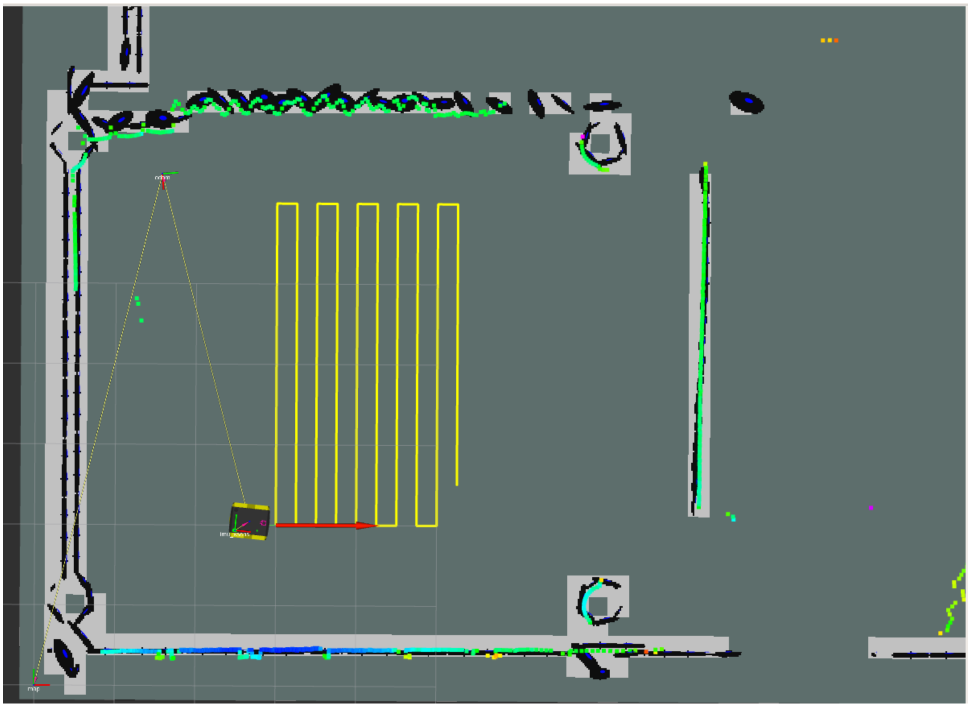


Figure 5.13: Screenshot from the visualization tool *Rviz* when recording the rosbag. The black spots represent gaussians from the *NDT* method. The yellow curve is a predefined trajectory of the robot.

In the `ndt2poses` program, we created poses for `Move Base` so that the robot moves approximately in the middle of the mapped area (see picture 5.13). The *NDT + ODOM* method was used as the default method when locating the robot during the experiment. The result of the experiment was not only the rosbag, but also the transformation matrix $T_o^m(t)$ generated by the *NDT* method. The experiment lasted about 9 minutes. The robot traveled distance of approximately 60 meters.

$$T_o^m = \begin{pmatrix} cos\theta & -sin\theta & 0 & x \\ sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.1}$$

The $T_o^m(t)$ transformation was then removed from the rosbag. The experiment was repeated in a simulation environment with the *MAG* method. Therefore, we get two transformation matrices (the first was generated by the

MAG method and the second one was generated by the *MAG* method) that vary over time for comparison. In the case of filtered odometry, we consider $T_o^m(t)$ constant, determined by the position of the robot at the beginning of the experiment

MAG parameters	
W_1	1
W_2	10000
σ_θ	0.006
σ_α	0.008
σ_d	0.004

Table 5.2: Parameters of the *MAG* method during the experiment.

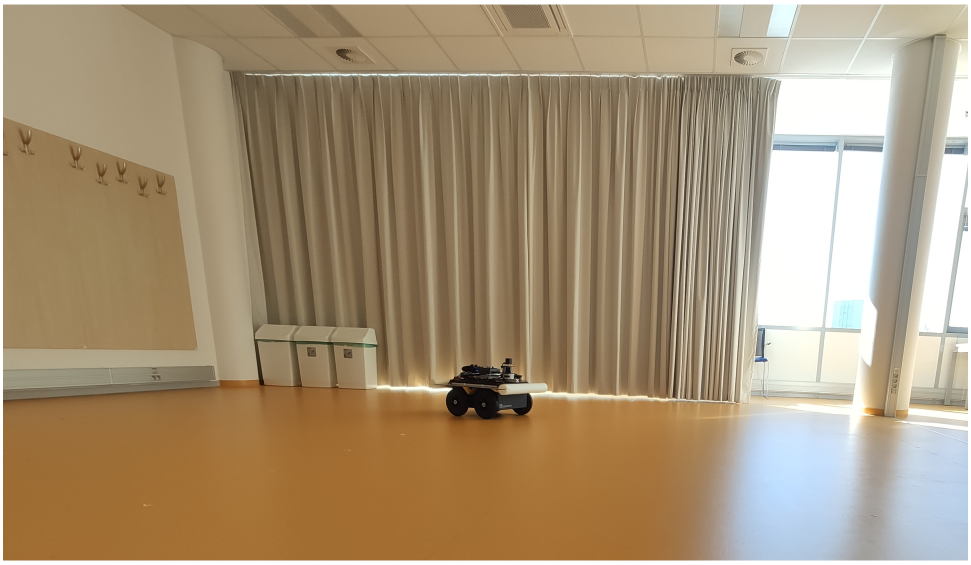


Figure 5.14: Robot Jackal in the room B670. In the back, there is a curtain, which caused a deterioration in the accuracy of the *NDT* method in the final experiment because of its movement caused by a wind.

We know from an experiment using the *Vicon system* that the *NDT* method can eliminate the integration error of odometry if there are enough lidar features in the environment. The final experiment was performed in room B670, suitable for the *NDT* method. Thus, it is possible to consider the transformation T_o^m generated by the *NDT* method as an accurate odometry correction. In figure 5.15, we see the transformation matrices adjusting the odometry in time. The transformation generated by the *NDT* method is not smooth. This is due to several factors. The first factor is the computational time of one iteration of the *NDT*, which results in a low frequency of the odometry correction. The second factor is the slight movement of the curtain in the room (see figure 5.14), which at certain time intervals formed the primary source of lidar data for the *NDT* method. The third factor is the way the robot Jackal turns, during which the wheels naturally slip and the odometry makes a significant mistake in a short period of time. This error must be corrected during one iteration of the NDT method, and thus the transformation matrix T_o^m suddenly changes.

The transformation matrix T_o^m is described by three parameters x, y, and θ. Let us compare the parameters for individual localization methods. In figure 5.15, we see that at the beginning of the experiment there was almost no adjustment of the odometry. This was due to the small difference between the measured magnetic induction and the magnetic induction from the map. A more significant adjustment of the odometry occurred at a time when this difference was more significant.

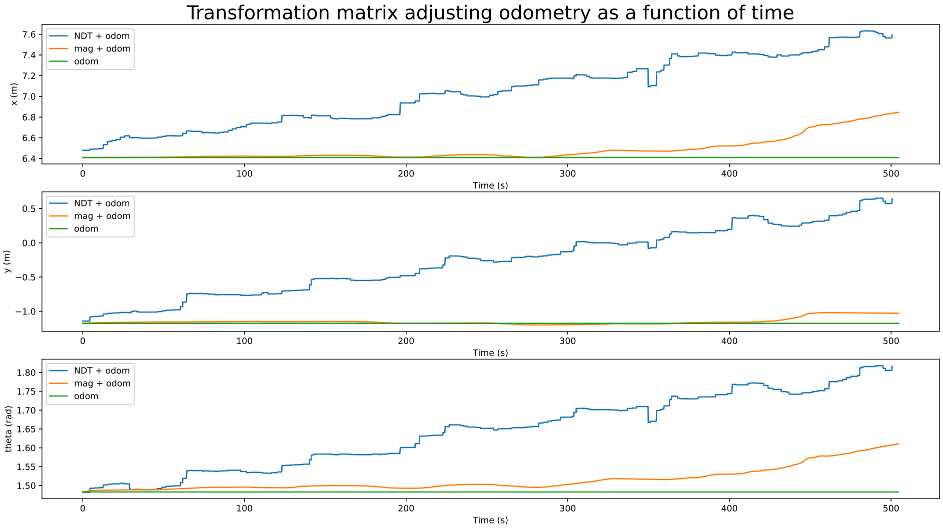


Figure 5.15: Comparison of the transformation matrices adjusting odometry. The transformation matrices are described using the parameters x, y, θ. The blue curve represents the filtered odometry together with the *NDT*. The orange curve represents the filtered odometry together with the *MAG* method. The green curve represents the filtered odometry only.

In Figure 5.16 we see the x and y coordinates of the robot's position in time, according to the individual localization methods. The *MAG + ODOM* method had less tendency to take on an integration error than filtered odometry.

Figure 5.17 shows the difference between the robot's position in the final experiment and the other two localization methods. In the case of filtered odometry, the mean value of this difference is 34 cm. Assuming the *MAG* method, the mean value is 29 cm. The *MAG* method achieved 15% better results compared to filtered odometry. However, this conclusion is based on only one experiment. The method must therefore be subjected to further experiments.

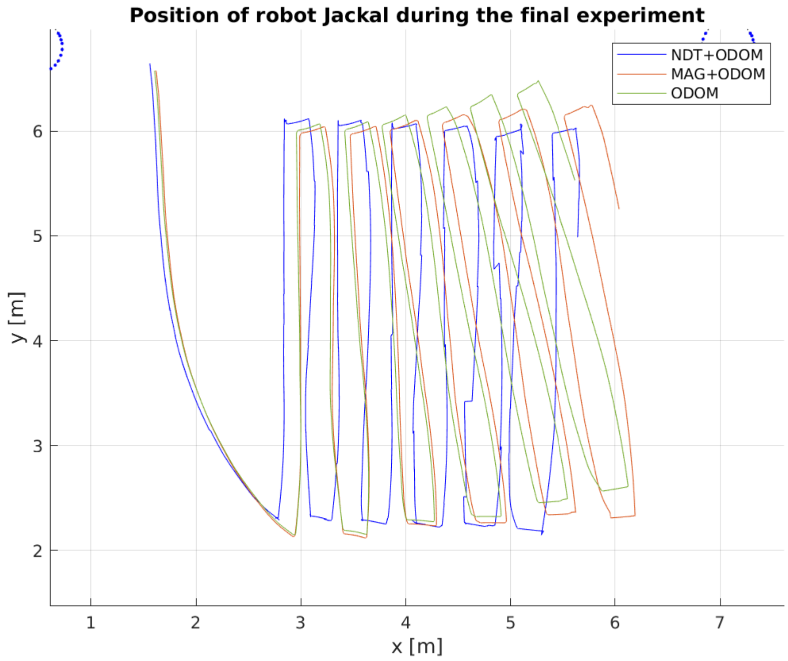


Figure 5.16: Position of robot Jackal during the final experiment according to each localization method.

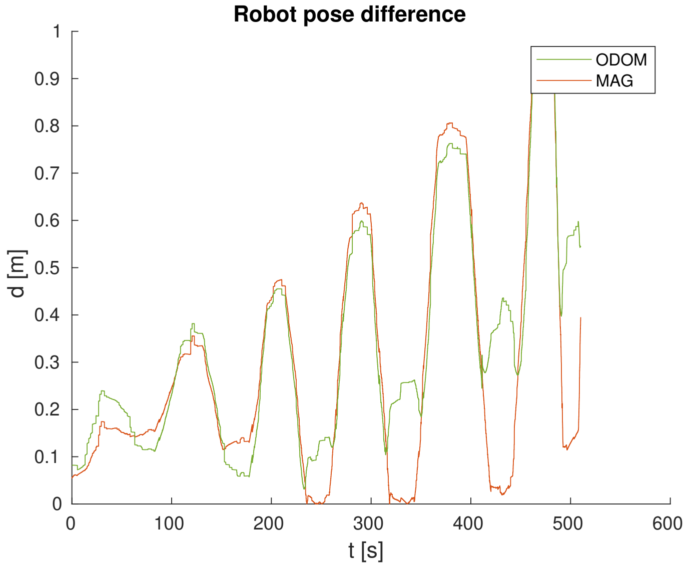


Figure 5.17: The difference of the robot's position in the final experiment according to the *NDT* method and the *MAG* method (red line) or filtered odometry (green line)

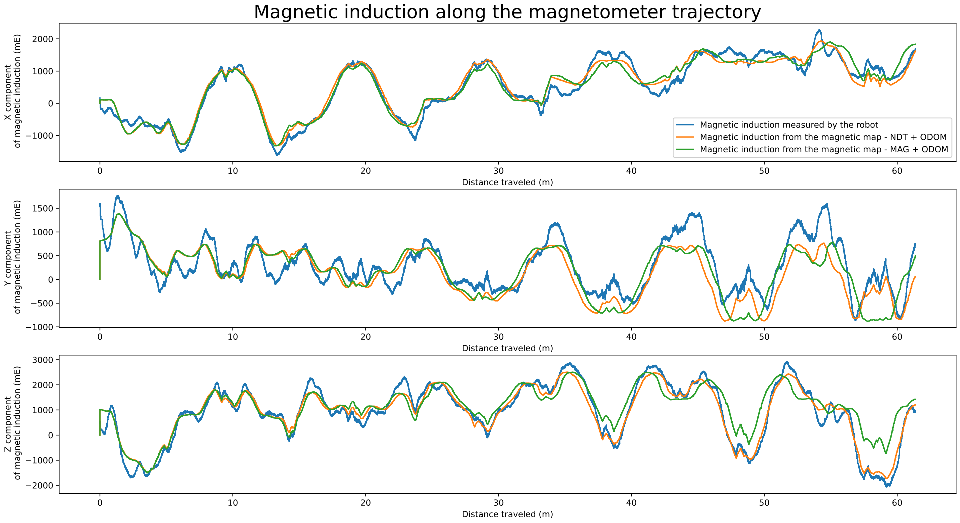


Figure 5.18: Graph of the magnetic induction along the robot's trajectory. The blue curve represents the magnetic induction measured by the robot. The orange curve represents the magnetic induction obtained from the magnetic map according to the robot's position estimated by the *NDT + ODOM* method. The green curve represents the magnetic induction obtained from the magnetic map according to the robot's position estimated by the *NDT + ODOM*.

The measured magnetic induction along the robot's trajectory in the final experiment is shown in figure 5.17. For comparison, we also plotted the values of magnetic induction from the magnetic map corresponding to the estimated position according to *NDT + ODOM* and *MAG + ODOM* methods.

Chapter 6

Future work

This bachelor thesis was created based on a project to improve the robot indoor localization. The knowledge of the magnetic field inside a building can improve location accuracy.

We work with static magnetic maps, which must be measured and generated before using them for localization. The measurement of new data cannot be reflected on the map. The principle of generating a map from a dataset requires a recalculation of the entire map when incorporating new data, which is possible but not in real-time. Animation showing such a scenario is in this video [Appendix C]. Expanding the dataset from which we generate magnetic maps is not a suitable solution because we only have limited memory in the real case. It is, therefore, necessary to find such a model of the representation of the magnetic map, which will be possible to calculate in real-time and will only work with limited memory.

One of the possible solutions is to assign not only the magnetic induction value to the grid-point but also its weight. If we make a new measurement of magnetic induction $\tilde{B}$ in the neighborhood of the grid-point, we assign it a weight $\tilde{w}$ depending on the distance from the grid-point. The new value of magnetic induction B_n in grid point is then calculated as:

$$B_n = \frac{B_{n-1} w_{n-1} + \tilde{B}\tilde{w}}{w_{n-1} + \tilde{w}}, \tag{6.1}$$

where B_{n-1} is the preceding value of magnetic induction in the grid point and w_{n-1} is the corresponding weight. The weight w_n of magnetic induction B_n is updated as follows:

$$w_n = w_{n-1} + \tilde{w}, \quad w_{n-1} + \tilde{w} \leq m \tag{6.2}$$

$$w_n = m, \quad w_{n-1} + \tilde{w} > m \tag{6.3}$$

The most significant disadvantage of the *MAG* method is that it does not take the history of measured magnetometry data into consideration. The information about the measurement history should be incorporated into the criterion function. Such a solution could completely eliminate the integral error of the odometry.

Furthermore, it is necessary to design a complex localization using all sources of the position estimate. A possible solution is the Kalman filter or any of its equivalents.

When working with an extensive *NDT* map, i.e., an entire building floor with a cell size of 25 cm or less, the frequency of transferring the map to the visualization program is very low, making the operator's work difficult. If we consider sharing the map between several robots, a quick update of the shared map is necessary. A solution could be to send only such gaussians that have changed, which means not necessarily sending the entire *NDT* map in each iteration, but only individual cells.

Another problem of the *NDT SLAM* implementation used in this thesis is mapping temporary objects in spaces that appear infinite to the lidar. These objects that disappear after mapping cannot be unmapped because no data is returned to the lidar from behind the object; hence the probability of lidar data (gaussian) cannot decrease. That results in improper *SLAM* performance on corridors longer than the lidar range, i.e., longer than 10 m in our case. A person walking down the corridor in front of the robot will leave behind gaussians that cannot be removed. A solution could be to replace points at infinity with points at a distance of the lidar range. However, the algorithm would only use these to reduce the probability of lidar data occurring between them and the lidar.

Chapter 7

Conclusions

In order to propose a localization algorithm using magnetometry data, it is necessary to have the knowledge of the magnetic field in the robot's surroundings. In particular, information on the variability and inhomogeneity of magnetic induction is important.

The data describing the magnetic induction can be stored in magnetic maps. We have shown how to systematically measure the magnetic field in buildings and convert the measured datesets into magnetic maps. Magnetic maps converted from datasets with different measurement dates were compared. The comparison confirmed the stability of the magnetic field in the CIIRC building in the four-week span. The time-invariant magnetic field can be used for robotic localization.

We measured the magnetic field in three different indoor environments. The magnetic field in the new part of the CIIRC building is the least homogeneous. This part of the building contains numerous iron rebars, which are probably the primary source of the magnetic field in the building. We could not find a correlation between the magnetic field and the load-bearing elements of the building. In the old part of the CIIRC building, the magnetic field is much more homogeneous. However, there is a slight disturbance of homogeneity around the reinforced concrete column. In the FEE building, the magnetic field is homogeneous.

We designed a localization algorithm using a magnetometer and magnetic maps. Our proposed localization method adjusts the position estimate from the filtered odometry to achieve a closer match of the measured magnetic induction with the magnetic map.

The experimental results show that the *MAG + ODOM* method achieves slightly better accuracy than localization provided by filtered odometry only. The inaccuracy of magnetic maps may be the cause of the limitations of such localization method. The accuracy of the method is also slightly affected by the magnetometer noise and the magnetic field generated by the robot's motors.

Bibliography

[1] Haverinen, J., & Kemppainen, A. (2009). Global indoor self-localization based on the ambient magnetic field. Robotics and Autonomous Systems, 57(10), 1028-1035.

[2] Gozick, B., Subbu, K. P., Dantu, R., & Maeshiro, T. (2011). Magnetic maps for indoor navigation. IEEE Transactions on Instrumentation and Measurement, 60(12), 3883-3891.

[3] Siebler, B., Sand, S., & Hanebeck, U. D. (2020). Localization with magnetic field distortions and simultaneous magnetometer calibration. IEEE Sensors Journal, 21(3), 3388-3397.

[4] Wahlström, N. (2013). Localization using magnetometers and light sensors (Doctoral dissertation, Linköping University Electronic Press).

[5] Biber, P., & Straßer, W. (2003, October). The normal distributions transform: A new approach to laser scan matching. In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453) (Vol. 3, pp. 2743-2748). IEEE.

[6] Storms, W., Shockley, J., & Raquet, J. (2010, October). Magnetic field navigation in an indoor environment. In 2010 Ubiquitous Positioning Indoor Navigation and Location Based Service (pp. 1-10). IEEE.

[7] Keys, R. (1981). Cubic convolution interpolation for digital image processing. IEEE transactions on acoustics, speech, and signal processing, 29(6), 1153-1160.

[8] Mastyło, M. (2013). Bilinear interpolation theorems and applications. Journal of Functional Analysis, 265(2), 185-207.

[9] Sendov, B., & Andreev, A. (1994). Approximation and interpolation theory. Handbook of numerical analysis, 3, 223-462.

[10] Gribbon, K. T., & Bailey, D. G. (2004, January). A novel approach to real-time bilinear interpolation. In Proceedings. DELTA 2004. Second IEEE International Workshop on Electronic Design, Test and Applications (pp. 126-131). IEEE.

[11] By Cmglee - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=53064904

[12] Boxan, M. (2020) NDT SLAM Respecting Visibility.Bachelor thesis, Department of Cybernetics, Czech Technical University in Prague.

[13] Pánek, V. (2018) Map import for mobile robot from CAD drawing. Bachelor thesis, Department of Cybernetics, Czech Technical University in Prague.

[14] Current measurements of the geomagnetic field: Measurement of the intensity of the Earth's magnetic field from the Budkov Observatory. Geofyzikální ústav Akademie věd ČR, v.v.i. [online]. Prague: AVČR, Accessed: 2022-02-07, Accessible from: https://www.ig.cas.cz/vyzkum-a-vyuka/oddeleni/geomagnetika/aktualni-mereni-geomagnetickeho-pole/

[15] Hurák, Z., & Řezáč, M. (2010, December). Delay compensation in a dual-rate cascade visual servomechanism. In 49th IEEE Conference on Decision and Control (CDC) (pp. 1639-1643). IEEE.

[16] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).

[17] Negenborn, R. (2003). Robot Localization And Kalman Filters. Utrecht Univ., Utrecht, Netherlands, Master's thesis INF/SCR-0309.

[18] Stanford Artificial Intelligence Laboratory et al. (2018). Robotic Operating System. Retrieved from https://www.ros.org

[19] Jackal UGV - Small Weatherproof Robot-Clearpath. https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/. Accessed: 2022-05-10.

[20] TiM561-2050101 | Detection and ranging solutions | SICK. https://www.sick.com/us/en/detection-and-ranging-solutions/2d-lidar-sensors/tim5xx/tim561-2050101/p/p369446 Accessed: 2022-05-10.

[21] IMU XSENS MTI-30-2A5G4 | Inertial measurement unit | https://www.xsens.com/products/mti-10-series Accessed: 2022-05-10.

[22] Vicon system | Motion capture | https://www.vicon.com/applications/engineering/ Accessed: 2022-05-10

[23] Vorlík, P. (2014). Areál ČVUT v Dejvicích v šedesátých letech.

[24] Franta, P. (2020). Český institut informatiky, robotiky a kybernetiky – CIIRC. Stavebnictví, 01-02, 14-20

[25] Oriient | Geomagnetic indoor positioning technology | https://www.oriient.me/geomagnetic-indoor-positioning-technology/ Accessed: 2022-06-10

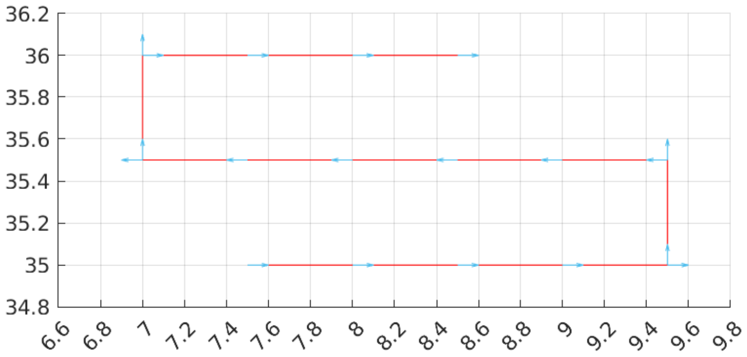
[26] Frassl, M., Angermann, M., Lichtenstern, M., Robertson, P., Julian, B. J., & Doniec, M. (2013, November). Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 913-920). IEEE.

Appendix A

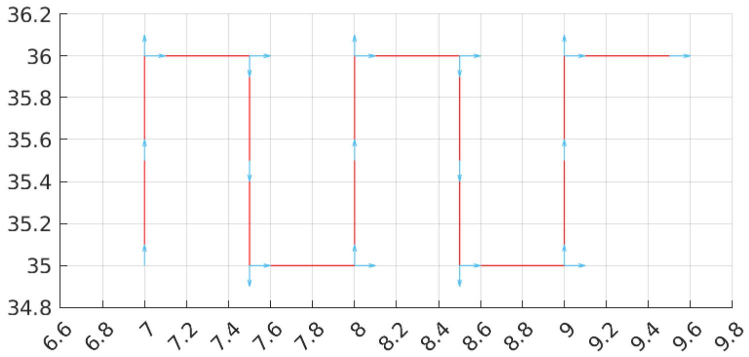
Ndt2poses-gui settings

When generating poses in `dxf2poses_gui`, we have many possible combinations of optional parameters. Some of possible results follow.

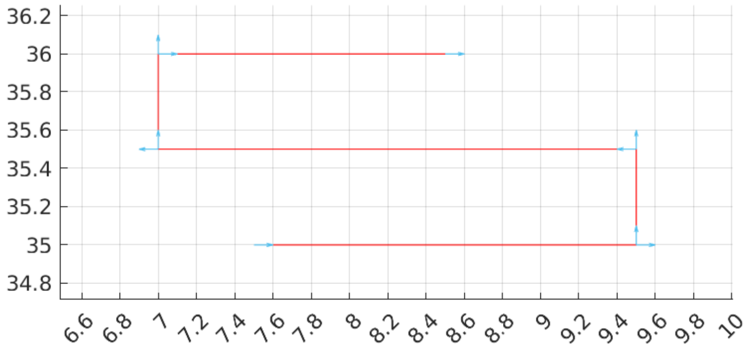
1. `Rotate = 0, Simplify = 0, Remove Rotation Poses = 0, Translation = (0, 0)`



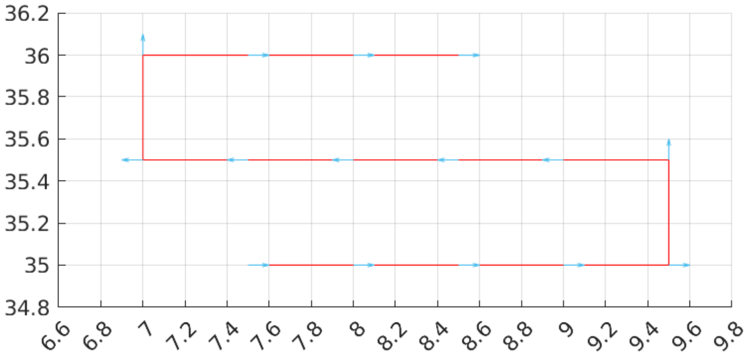
2. `Rotate = 1, Simplify = 0, Remove rotation poses = 0, Translation = (0, 0)`



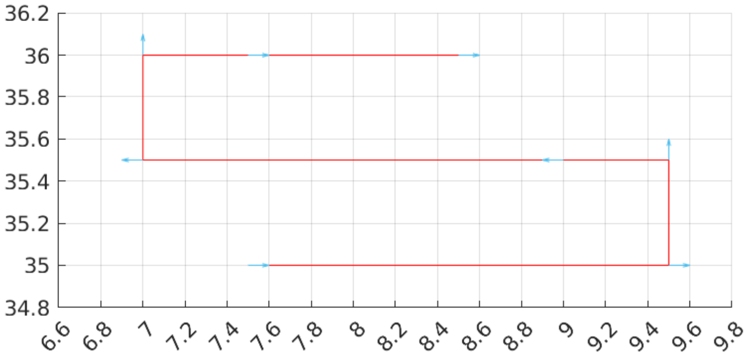
3. `Rotate = 0, Simplify = 1, Remove rotation poses = 0, Translation = (0, 0)`



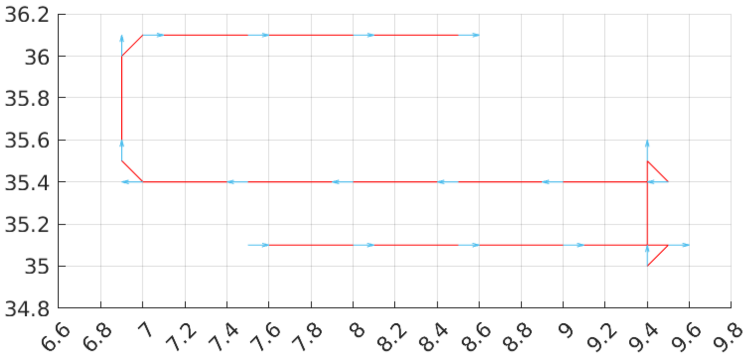
4. `Rotate = 0, Simplify = 0, Remove rotation poses = 1, Translation = (0, 0)`



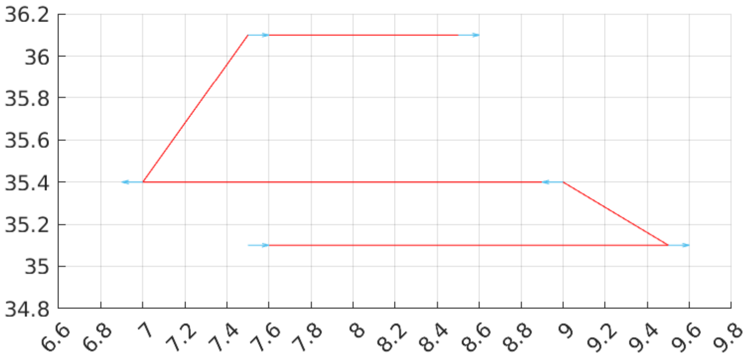
5. `Rotate = 0, Simplify = 1, Remove rotation poses = 1, Translation = (0, 0)`



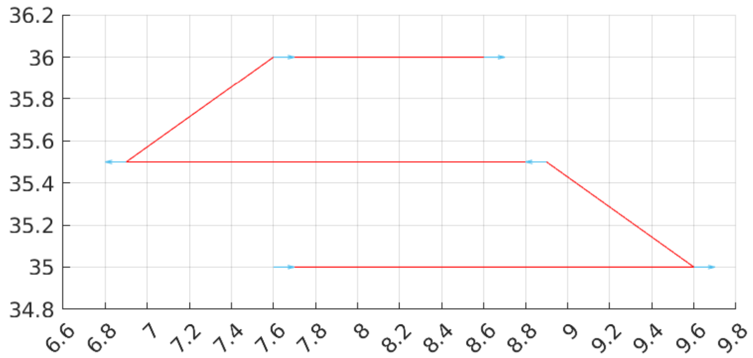
6. `Rotate = 0, Simplify = 0, Remove rotation poses = 0, Translation = (0, 0.1)`



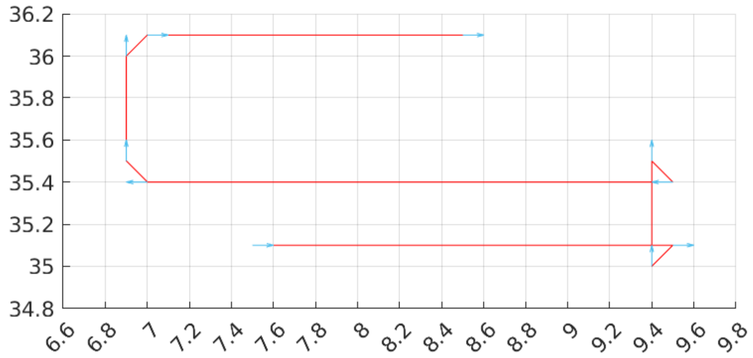
7. `Rotate = 0, Simplify = 1, Remove rotation poses = 1, Translation = (0, 0.1)`



8. `Rotate = 0, Simplify = 1, Remove rotation poses = 1, Translation = (0.1, 0)`



9. `Rotate = 0, Simplify = 1, Remove rotation poses = 0, Translation = (0, 0.1)`



Appendix B

Mag-handler config file

A text file in the following format is required for running the *MAG* method. It contains the path to the magnetic map file, the initial position of the robot, *MAG* method parameters, and the magnetometer calibration parameters.

```
map = ./mag_maps/B670/B670.mat

x0 = 1.179
y0 = 6.409
theta0 = -1.482

w1_weight = 1.0
w2_weight = 10000.0
sigma_theta = 0.004
sigma_alpha = 0.004
sigma_d = 0.004

HI_x = 32676.85450557868
HI_y = 32903.00937109157
HI_z = 32306.34412470024

SI_x = 1.002770083102493
SI_y = 0.997245179063361
SI_z = 1
```

Appendix C

Magnetic map measurement procedure - video

`https://youtu.be/4hLkKZsrHDs`

The video captures one magnetic map measurement procedure in room B670 in the new part of the CIIRC building. The acquired dataset is not part of the magnetic map presented in chapter 5.1.1.

The first part of the video is the visualization of the *NDT* map. The blue ellipses correspond to Gaussians representing the static part of the map, i.e., the pre-generated *NDT* map from the *DXF* file. Green ellipses correspond to gaussians representing the variable part of the map. The predefined path of the robot is marked in yellow.

An animation called "Magnetometer raw data" shows every twentieth magnetometer measurement. Magnetic induction has x, y, and z components. The x and y components are represented by the arrow and the z component by the color. Specific numerical values are not captured in the video.

The animation called "Magnetic map" shows the shape of the magnetic map generated from the dataset, which corresponds in time to the "Magnetometer raw data" animation. The data is displayed the same way as in the previous animation, i.e., using arrows and colors.