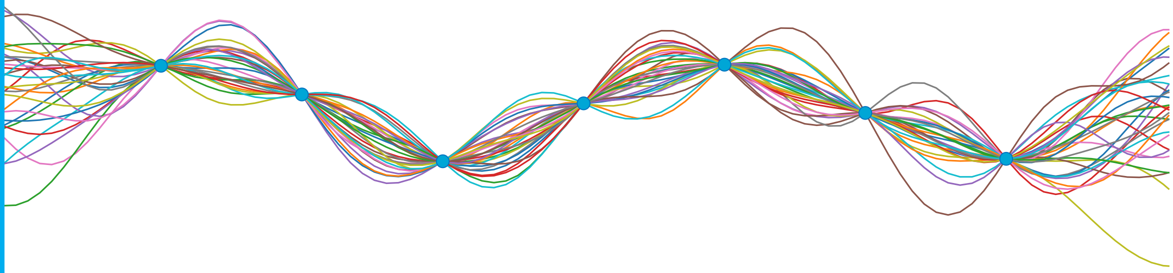
Magnetic field SLAM

using an inertial human motion suit and reduced rank Gaussian process regression

Thijs Veen

Master of Science Thesis



Delft University of Technology

Delft Center for Systems and Control

Magnetic field SLAM

using an inertial human motion suit and reduced rank Gaussian process regression

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Thijs Veen

December 5, 2022

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Magnetic field SLAM

by

Thijs Veen

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: December 5, 2022

Supervisor(s): ______________________
dr. Manon Kok

____
Mostafa Osman

Reader(s): ______________________
dr.-Ing Jens Kober

Abstract

Indoor localisation is a growing field of interest in recent studies. While GPS (global positioning system) is a standard for outdoor localisation, no such solution exists for indoor applications. The literature provides several methods to obtain the location of indoor systems, often using optical sensors. A small number of recent studies use the indoor magnetic field for localisation. Ferromagnetic materials in the structure of buildings cause magnetic anomalies that are distinct enough to use for localisation. To use the magnetic field for localisation, a map has to be created.

In this thesis, a sensor setup different from other studies is used. This sensor setup consists of a inertial HMTS (human motion tracking suit), containing seventeen IMUs (inertial measurement units) with magnetometers. This suit uses advanced techniques to obtain a better pose estimate than a single IMU can achieve. The combination of an inertial motion tracking suit with a magnetic localisation approach has not been studied before. Inspired by the state of the art approaches, a SLAM (simultatenous localistation and mapping) algorithm is proposed that is able to use information from the inertial HMTS. The algorithm consists of a reduced rank GP (Gaussian process) to create a map of the magnetic field. A RBPF (Rao-Blackwellized particle filter) is used to localise the HMTS. The algorithm allows for the use of multiple magnetometers to create the magnetic field map instead of a single one, which is a novelty.

The proposed method is tested with real-life data. Live odometry obtained from the HMTS can be post-processed for improved inertial odometry. Both the live and post-processed odometry data from the HMTS is used in the proposed algorithm and the results are compared. Additionally, the differences between a magnetic field map constructed with a single magnetometer and a map constructed with multiple magnetometers is investigated. The trajectory estimated by the RBPF is compared to a groundtruth, obtained by an optical tracking system. The RBPF shows higher performance for trajectories longer than 250 seconds compared to the inertial odometry. The use of multiple magnetometers does not improve the performance of the algorithm.

Table of Contents

List of Figures

List of Tables

Preface

The subject of this MSc-thesis was proposed by dr. Manon Kok, department of Systems and Control, Delft University of technology. The thesis will investigate the combination of a inertial tracking suit and magnetic field localisation and mapping, to obtain a good pose estimate.

The goals of this master thesis study are to survey and get acquainted with different magnetic field mapping techniques, localisation and mapping problems and inertial human motion tracking that have been proposed by the scientific community, define a new method for magnetic SLAM using an inertial HMTS and to test the new method with experiments

I would like to thank my supervisors dr. Manon Kok and Mostafa Osman for their ideas and feedback and guidance.

I also want to thank Xsens for the use of their inertial and Fabian Girrbach who helped me with what data the suit contained and helped me to extract the data needed for this thesis.

I addition, I would like to thank Maarten ten Voorde for his help during the experiments.

Delft, University of Technology
December 5, 2022

Thijs Veen

Chapter 1

Introduction

In this thesis, the combination of inertial HMTSs (human motion tracking suits) and magnetic field SLAM (simultatenous localistation and mapping) is researched. The inertial HMTS used in this thesis is the commercially available MVN Link [1, 2], provided by Xsens (Enschede, The Netherlands). The first section discusses the relevant background on both topics. In the second section the research questions are formulated. The chapter ends with an overview of the structure of this thesis.

1-1 Background

Localisation is a field of study that has seen growing interest over the last decades. A more specific topic within localisation is human pose estimation, with a wide variety of applications like healthcare [3], sports [4] and animation [5]. To obtain the positions and orientation of the parts of an human body, devices like an HMTS can be used.

A motion tracking suit is a wearable for the entire body, capable of estimating the pose of the wearer. One approach is to use an OTS (optical tracking system). Markers are placed upon different parts of the suit and several cameras track these markers, capturing the motion of the human body. This approach, while accurate, has several downsides. Since the sensors are optical, they require a line of sight to the marker. Additionally, the motion can only be tracked in a specified space inside the field of view of the cameras.

These downsides have given rise to inertial tracking suits. These suits contain several IMUs (inertial measurement units), giving an location and orientation estimate of the body part they are attached to. The IMU is small enough to be placed on microelectronics and can be found in all modern smartphones. Modern IMUs are equipped with an accelerometer, a gyroscope and a magnetometer. Since no part of the sensor setup is not on the suit, the suit can be used outside of prepared locations. Tracking the pose of humans with inertial sensors is commonly called *inertial human motion tracking*. While inertial tracking does not suffer from spatial restrictions like the OTS, it is subject to drift. This drift is caused by integrating the accelerometer and gyroscope measurements twice to obtain the position and orientation [6]. To compensate for this drift, the inertial HMTSs use multiple IMUs. These IMUs, placed on different body parts, allow for the use of kinematic constraints and the zero-velocity update. A overview of inertial motion tracking methods is presented in [7], and a summary of commonly used methods is given in [8].

Several studies [i.e. 9–11] have been done to overcome the downsides of the inertial sensors for motion tracking suits. Attempts to use accelerometers to measure the movement of human bodies go back as far as 1953 [12], although the accelerometers were found lacking. Morris claimed that, with the proper accelerometers, only six independent measurements are required to determine the movement of the human body [13]. Furthermore, the claim is made that gyroscopes are not required, because rotational movement is large, and can be sufficiently determined by the acceleration.

More recent methods, however, do use more than just the accelerometers for inertial motion tracking suits. In [9], Roetenberg uses inertial and optical sensor with a KF (Kalman filter). Additionally, a combination of inertial sensors and magnetometers, which measure the local magnetic field, to obtain estimates about human motion is investigated. The obtained results were promising and opened the way for self-contained motion tracking suits, meaning that no equipment is placed outside of the suit. Magnetometers are often part of modern IMUs, increasing their use in motion tracking suits [14, 15].

The parts of the human body can be represented by a kinematic constraint model. Since the IMUs are placed on the human body, this model can be used to constrain the relative positions and orientations of these IMUs. The constraint model can be used in an optimisation framework, so that the different body parts stay connected and do not suffer as much from drift [11].

An additional approach to improve the inertial human motion tracker is the zero velocity update [16–20]. This uses the knowledge that the foot of the stance leg has zero velocity. The literature provides several methods to detect the moment there is a stance leg in the gait [e.g. 21, 22]. This further reduces the drift and improves the position estimate.

Inertial HMTS require a sensor-to-segment calibration before use. The segments represent different body parts. The calibration aligns the segments with the IMUs. The Xsens MVN Link uses several length measures of the wearer to determine the position of the IMUs with respect to the body segment it is attached to. Additionally, it requires the user to stand in a known pose, such as the N- or T-pose, followed by a short walk and a turn-around before recording any data [1].

The Xsens MVN suit uses a motion capture engine that requires ‘warming up’ [2, Section 3.5.5]. The recommended practice is to move slowly for the first 30 seconds after calibration. The motion capture engine aims to overcome major sources of errors, like those caused by anomalies in the magnetic field, disturbing the magnetometers [2, Section 23.8].

To measure the performance of an inertial motion suit, it can be compared to a optical tracking suit, or with global navigation satellite system (GNSS). The commercially available Xsens MVN [1] is a inertial motion suit using seventeen IMUs. Several studies have validated the accuracy of the suit.

Supej uses the Xsens MVN to capture the motions of a alpine skier [23]. The study shows a drift of 6.3 cm at the neck after a 15 m walk back and forth, with a single turn. Although the drift is rather low, the researches expect a trajectory with more turns will introduce a larger error. Such a trajectory is likely to occur in an indoor environment. The performance of the suit in a skiing experiment was only validated by visually comparing the video with the 3D animation. No high amount of drift was observed. A more extensive accuracy analysis was done by Zhang et al. [24]. The study presented very small differences with a optical tracking system for flexion and extension. The rotation around the other two rotational axis performed significantly worse.

The drift that is still present in the inertial HMTS causes growing errors over time. To further improve the pose estimates of the inertial HMTS, the Earth's ambient magnetic field can be used.

The Earth's ambient magnetic field is uniform, with its direction the magnetic north pole. This allows magnetometers that are part of the IMUs to provide heading information, without drift. Due to ferromagnetic materials present in buildings, the Earth's ambient magnetic field is disturbed. This creates anomalies in the otherwise uniform field, preventing the IMUs from accurately obtaining its heading. Some inertial tracking approaches try to filter the disturbances present in indoor environments [18, 20, 25], while others try to use those disturbances for localisation [26].

Recent studies have used these anomalies in the magnetic field to tackle the SLAM problem [see e.g. 27–29]. In this problem, the location is estimated using a map. In magnetic field SLAM, the ambient magnetic field is mapped. Cross referencing a magnetometer measurement with this map yields the locations where the sensor can be.

However, this still relies on an accurate map that has been constructed beforehand. In SLAM there is no map available and the map has to be constructed while navigating the environment. This is a chicken and egg problem, since a map is required for localisation, and the locations are needed to construct a map.

To overcome this problem some estimation of movement, *the odometry*, is required. Aside from the HMTS, the odometry can be obtained for example from the wheels of a robot [see e.g. 30, 31] or an optical sensor [see e.g. 28]. Additionally, some distinct features that can be mapped are needed to tackle SLAM. For indoor environments, this can be magnetic field [27]. Since the indoor magnetic field is disturbed by the ferromagnetic materials used in buildings, it becomes distinct enough to be used for localisation [32].

1-2 Research goal

The goal of this research is to establish a method to improve the position estimates provided by the inertial HMTS. This method uses a SLAM approach that uses the inertial odometry provided by the HMTS. An improvement to these estimates would lead to better performance in the fields where these suits are applied.

The inertial HMTS combines naturally with magnetic field SLAM, since the suit provides both the odometry and the magnetic field measurements that can be used to construct the map. There are two major novelties in the research on this combination. The first novelty is the use of an inertial HMTS for magnetic field SLAM. This leads to first research question:

What are the improvements to the position estimate of an inertial human motion tracking suit when combined with magnetic field simultatenous localistation and mapping?

Since an inertial HMTS uses multiple IMUs, multiple magnetometers can be used to construct the magnetic field map. This is the second novelty and leads to the following research question:

How does using multiple magnetometers to construct a magnetic field map in a magnetic field simultatenous localistation and mapping approach differ compared to a map constructed with a single magnetometers?

1-3 Organisation

The goal of this this thesis is to propose and test a method to obtain the position of the wearer of the inertial tracking suit, whilst creating a map of the local magnetic field and using this map for localisation.

In Section 2-1 different approaches to modelling the magnetic field are presented. Section 2-2 discusses the methods used in the literature for localisation and mapping.

In Chapter 3, a method is derived to combine the HMTS with magnetic field SLAM. It starts with the notation used throughout the thesis. The available data from the inertial HMTS is presented in Section 3-2. Section 3-3 presents the odometry model used in the SLAM algorithm. The measurement model is described in Section 3-5. Section 3-6 presents the adjustments to the algorithm when large spaces need to be mapped. Finally, the full algorithm is summarised in Section 3-7.

Multiple experiments are performed to evaluate the performance of the proposed algorithm. The results are presented in Chapter 4. The chapter starts with a verification of the method in Section 4-1. In Section 4-2, the algorithm is run with only a single magnetometer to construct the magnetic field map, while multiple magnetometers are used in Section 4-3. Chapter 5 presents the conclusion from the results, and answers the research question. The glossary at the end of the thesis contains all used abbreviations and symbols. The appendix contains preliminary knowledge about coordinate frames and quaternions. It also includes additional information on Gaussian processes.

Chapter 2

Related work

In this chapter, the existing literature is reviewed. The first section discusses the different models used to model the magnetic field. The second section presents the localisation methods found in the literature that use the magnetic field.

2-1 Magnetic field representation

Earth's outer core is the most influential source of the magnetic field on its surface. This field is dominant enough for traditional navigation, since it causes a compass needle to point to magnetic north. However, close to ferromagnetic materials, the field changes noticeably. These disturbances are seen as a cause for errors in some literature [e.g. 33], but Suksakulchai et al. were one of the first to show that these distortions can be used as distinctive and recognisable features in a localisation framework [32].

2-1-1 Spatial and temporal stability

Although the Earth's magnetic field fluctuates over a time span of one day, for localisation purposes it is often assumed to be temporally stable. If the field would change significantly over time, the map of the magnetic field could become useless after some time. An example of the temporal stability is given by Frassl et al. Daily fluctuations are less than 0.1% of the average magnitude [34]. Further study in to the stability of the field is done in [35]. A massive moving metal object, like an elevator cart, also changes the ambient magnetic field. Several studies show that this effect is only significant very close, up to approximately one meter [35–37]. This validates the use of the magnetic field for localisation, even when large moving ferromagnetic objects are present.
In [38], Vallivaara measures the spatial variation of an indoor magnetic field for different representations of the field. More spatial variation results in more distinct magnetic field, improving the localisation accuracy. The results show a high deviation in the plane parallel to the Earth's surface, the xy-plane, and less in the direction of gravity, or z-direction.

2-1-2 Magnitude and full vector representation

The magnetic field can be represented by a three dimensional vector . In this representation, both the magnitude and the direction of the field are captured. It requires the orientation of the magnetometer to be known. Studies with an IMU often use the full vector, since the orientation is known [see e.g 28, 39–41].
When the orientation is not available, the full vector cannot be used. For mobile robots that move across the floor, a yaw independent representation can be used [35].
If no orientation information is available, the magnitude of the field can still be used [see e.g 42–44], since the observed magnitude is independent of the orientation of the magnetometer. Studies comparing the performance of these representation conclude that the full vector is most desirable if the orientation is known, while the magnitude representation suffers from the lack of information on the direction of the field [34, 45–47].

2-1-3 Magnetic field models

Unlike cameras or range-lasers, magnetometers and Wi-Fi receivers can only make observations at their current location. This either means that every location needs to be visited to map it, or that the map of the unvisited areas has to be estimated using the observations made elsewhere. Another method would be to map landmarks, and recognise those after revisiting.

Landmark Based Maps

The earliest work on magnetic field navigation use maps based on landmarks. Suksakulchai et al. extracted signatures based on electronic compass readings in a one dimensional environment [32]. The system was able to recognise different parts of a hallway. Taking inspiration from speech recognition, where sequences of measurements at different rates are matched, Subbu, Gozick, and Dantu expand on the hallway localisation with DTW (dynamic time warping) [48]. This system enabled pedestrian localisation in a hallway using a smartphone, and was capable of dealing with different walking speeds. Another example of landmark based maps is proposed in [49]. The map consists of large structures that disturb the magnetic field, like steel pillars. The authors also use guideposts, strategically placed by architects for additional landmarks. This would mean that one of the advantages of using the magnetic field as map, that it does not require additional infrastructure, is no longer valid. More recent work use techniques from machine learning to extract features of the magnetic field [50, 51].

Interpolation Maps

An intuitive way to fill the magnetic field map is to interpolate between measurements, creating an estimation for unvisited locations [see e.g. 52]. These maps are often used with a RBPF (Rao-Blackwellized particle filter) for localisation. Examples of this combination are [40, 43, 45], where linear interpolation was used. More sophisticated interpolation techniques can improve the localisation performance, as shown in [53], where bicubic interpolation was compared to bilinear interpolation.

In [54], Vallivaara et al. use a local interpolant using the polygonal neighbourhood. The proposed interpolation method share desirable properties with the Sibson interpolant. The authors show improved performance compared to the Sibson interpolant when the data is sparse, and worse performance when data is dense. While this might seem undesirable, it is crucial when considering the initial stages of SLAM.

Gaussian Processes

Another option to model the magnetic field is GPR (Gaussian process regression). GPR was introduced by O'Hagan in 1978 [55]. GPs (Gaussian processes) provide a probabilistic approach to learning in kernel machines [56]. In machine learning, GPR is used to learn the input-output mapping from measurements. In the case of modelling the magnetic field, the input could be the state with position, and the output would be an estimate of the magnetic field at that location.

Mean function and kernel GPs are build using a mean function and a kernel, which are parametrised by hyperparameters. Within these functions, *a priori* knowledge about the magnetic field can be used to improve the magnetic field map. For instance, the magnetic field is continuous. A kernel that guarantees smoothness in the output, is the SE (squared exponential) kernel. Most studies where the magnetic field is mapped by a GP choose the SE as kernel [see e.g 27, 30, 57–59].
In [60], Wahlström et al. use the Maxwell equations by adapting covariance function, using the SE kernel as a basis. While this extended kernel performed only slightly better in estimation perfomance compared to the approaches using the SE kernel, it allows for modelling the magnetic sources. The kernel proposed by Wahlström et al., was used for navigation and SLAM in [61] and [28] respectively. While the curl-free property of the magnetic field was used, the divergence-free property, as proposed in [60], was not used in [61] and [28].

Hyperparameters An advantage of GPR is that it does not rely on tuning parameters, since the hyperparameters in the mean function and kernel can be learned from the data. A common way to obtain the hyperparameters is to maximise the log likelihood of the observations [57, 60, 62]. In [27, 30], the hyperparameters are learned, although it is not specified how. In the case of sequential updating, learning the hyperparameters from data is not practical. Maximising the log likelihood requires batches of data, which are unavailable at the start. Therefore in the studies where the magnetic field was used with SLAM, the hyperparameters are considered tuning parameters [28, 59]. While recursive updating of the hyperparameters is possible, Solin et al. suggest to optimise the hyperparameters with an intitial batch of data, and re-optimise later if required [61].

Computational complexity One of the challenges when using GPR is the computational complexity. In standard form, GPR computes the inverse of a $n \times n$ matrix, where n is the number of measurements. This makes GPR scale as $\mathcal{O}\left(n^3\right)$ per regression step. Considering magnetometers can take hundreds of measurement every second, the computation becomes infeasible fast. An easy method to reduce the number of required computations is to reduce the number of measurements provided to the GP. In early work, Vallivaara et al. use a grid map, with 5cm × 5cm cells [27]. Each cell has a maximum number measurements, trading accuracy for computational speed.

An alternative is Local GPR [63]. A GP model is trained for several local regions, into which the measurements are divided. The training data is clustered by a distance measure, given by the same covariance function as used in the local GP models. New model centres are created if the distance measure to all other centres exceeds some boundary. Means are predicted by a weighted average over all the local models. The algorithm showed promising result, being significantly faster than standard GPR, with slightly higher errors in the estimation. The experiments were done by learning the dynamics of different robot arms. The complexity scales $\mathcal{O}\left(N^3\right)$ at best, where N is the number of observation for the local model. For large data sets, this could still be too complex for real-time application. The authors suggest to replace older data with new data, after comparing the information gain of the data points.

Akai and Ozaki only use the points nearby for learning [57]. The aim of the study was large scale indoor localisation using GP. Building the map from more than six thousand data points took about one hour. Since this is not fast enough for SLAM, a different approach is needed if GPR is used.

The complexity can also be reduced by using an approximation. Solin and Särkkä proposes a reduced-rank GPR [64], which has been used in [28, 61] for magnetic fields. This approach uses basisfunctions to approximate a full GP. The complexity scales $\mathcal{O}\left(N_m\right)$ per regression step, where N_m is the number of basis functions used, thus not with total number of measurements.

2-2 Magnetic field localisation and mapping

The disturbed indoor magnetic field can be used for localisation. Suksakulchai et al. were the first to use these disturbances as distinctive, recognisable signatures [32]. The authors were able to localise a robot when it returned to such a signature, proving that the disturbed indoor magnetic field is distinct enough for localisation. Several studies exploiting magnetic field anomalies for indoor localisation have followed. Some use robot odometry [34, 45, 52, 65, 66], others use inertial sensors [34, 40, 41, 67] and some approaches just use the magnetic field [36, 39, 45, 51].

These methods first create a map with the same device that is later to be localised. A more recent field of interest is SLAM. In the SLAM problem, a system has to localise itself, without a known map. The map is created and updated during operation time, and the system localises itself within this map. A solution to SLAM would enable localisation relative to the created map in unvisited areas. Indoor SLAM can be done with i.e. WiFi or Bluetooth [see e.g. 68], requiring the necessary infrastructure to be in place. Magnetic Field SLAM relies on anomalies in Earth's magnetic field, therefore no infrastructure is required. Consequently magnetic field SLAM has an advantage over other indoor SLAM methods.

2-2-1 Particle filter based magnetic field SLAM

The first to use the particle filter for magnetic field SLAM were Vallivaara et al. [27]. A RBPF estimates the pose distribution, and GPR is used to model and map the magnetic field. A grid map with limited number of measurements per cell is used. This was done to keep the GPR complexity feasible. The proposal distribution is drawn from the motion model, and resampling is done as suggested in [69]. The GP provides an uncertainty measure of the predicted observation. Predictions with low confidence were discarded. The authors implemented the algorithm for a floor cleaning robot [30], achieving sub decimetre accuracy with a low cost sensor setup.

In their study the authors noticed that their hash-map data structure for storing the map was limiting the number of particles they could use, due to computational demand. In their more recent work [54], they use quadtree-based ancestry trees, based upon distibuted particle SLAM proposed in [70]. The obtained results were better than in their earlier work [27]. They were able to increase the particles from 200 to more than a thousand, while remaining realtime processing time.

In [71], Robertson et al. use the magnetic field to improve their earlier work FootSLAM [16, 17]. FootSLAM localises a pedestrian wearing a foot mounted IMU, using the zero velocity update to make a map of transitions. The approach achieved an accuracy of 1-2m. The authors use layers of hexagonal cells for the magnetic field map, where each layer has differently sized cells. This is done to prevent sparsity in the map, while maintaining the benefits of the smaller cells when they are eventually populated. Adding the magnetic field to FootSLAM improved the accuracy to 10-20cm. Interestingly, the magnetic field SLAM part of the new algorithm performed better without incorporating FootSLAM.

In [72], Vallivaara et al. introduce an extension named Monty Hall Particle Filter. The idea is to group particles based upon their uncertainty. This naturally complements the use of GPR, since it provides a measure of uncertainty. The extension was compared to fixed-deviation particle filters for both simulated as real world magnetic field data. The authors expect it to be well suited for SLAM , because of its implicit loop closing abilities.

In more recent work, magnetic field SLAM has also found its application in outdoor environments like for aviation, where the magnetic field is weak, and lacking the disturbances of indoor environments [59]. The authors use an algorithm similar to [27], resampling only particles that had returned to a location near the trajectory. Kok and Solin use a hexagonal grid map with GPR [28]. The GPR is kept feasible by the reduced-rank GPR from [64], circumventing the cubic complexity of GPR. The method uses a visual-inertial pedestrian dead reckoning from a smartphone instead of robot odometry, and is the first to show results in three dimensions.

2-2-2 Graph based magnetic field SLAM

A common method in SLAM is to match measurements with the map [see e.g. 73]. This approach would fail for magnetic field SLAM, since a single measurement of the magnetic field could match with several locations on the map. Therefore, a sequence of measurements can be used, and match to earlier sequences. This was successfully done by Subbu, Gozick, and Dantu for just localisation, using DTW to match sequences recorded at different movement speeds [42].

SignalSLAM [68] is an algorithm that incorporates an IMU and several RF signals sensors on a smartphone. The authors show an experiment where Wi-Fi was used for a map, though they claim that it could be used on any time independent signal, like magnetic field measurements.

Jung, Oh, and Myung show that using the sequences magnetic field observations in GraphSLAM allows for a improved loop closing [74], claiming a better performance than the RBPF used by [30]. The framework also incorporates robot odometry and some prior knowledge of the magnetic field. The method showed sub meter accuracy, but suffered from false negatives when matching sequences for loop closure.

In [75], Gao and Harle use smartphone based pedestrian dead reckoning for GraphSLAM. The aim of the study is to use MagSLAM as a basis for mapping another indoor signal, like Wi-Fi or Bluetooth. A surveyor walks with the smartphone, generating the map. This map is then used for localisation. In contrast to [74], a matching algorithm with DTW is proposed, so that speed changes of the surveyor can be accounted for. A major disadvantage of this method stems from the assumptions made in the algorithm, like constant smartphone orientation. Therefore the surveyor with knowledge of these assumptions is required to map the magnetic field, instead of running the GraphSLAM for each phone individually. Fingerprint grid maps were created from Gaussian Process Regression. This method too found sub meter accuracy.

Chapter 3

Method

In this chapter, the proposed method is presented. The chapter starts with a description of the notation used throughout this chapter. Then, the data provided by the HMTS is described. A RBPF, in combination with reduced rank GPR to model the magnetic field, is chosen to estimate improve the position estimate of the HMTS. The chapter concludes with a summary of the algorithm. Prerequisite knowledge on coordinate frame and rotations can be found in Appendix A

3-1 Notation

In this chapter a multitude of symbols is introduced. The notation of these symbols follow some consistant rules.

- Vectors and matrices are denoted in **bold**, scalars not. Matrix variables are capitalised, vectors are not. Note that scalars can be capitalised.
- Several values are indexed. For example, the proposed method is capable of using multiple sensors. These sensors values are indexed with a superscript to the right. For the sensors the indexing symbol is k. For the particle filter, each particle is indexed with i. Specific sensors are indicated by replacing the k, for example the position of the 8th sensor is indicated with $\mathbf{p}^8$. The total number of sensors is then indicated with N_k and the total number particles with N_i. The right superscript is always reserved for this index notation. Index symbols are always *cursive.*
- Vectors, quaternions and rotation matrices are expressed in a coordinate frame. The superscript on the left is always reserved for the coordinate frames. In contrast to the indexing symbols, coordinate frame symbols are not cursive. Although there exist $N_k = 17$ sensor frames s, they are not indexed.
- The right subscript often contains the symbol t for time.

As an example, ${}^{\mathrm{s}}\mathbf{z}^{k}$ indicates the observation of the sensor k, expressed in the sensor frame of the k'th sensor. Sensor observations are never expressed in the sensor frame of a *different* sensor, therefore there is no indexing on the frame notation s, although there exist as many sensor frames as sensors. Similarly, the orientation ${}^{\mathrm{ns}}\mathbf{q}^{k}$ denotes the orientation of the k'th sensors (and thus its corresponding frame s) of k, with respect to navigation frame n. Every introduced symbol can be found in the glossary at the end of this thesis.

3-2 MVN Link inertial HMTS

The proposed method for the SLAM problem uses inertial data provided by the commercial HMTS MVN Link [1], provided by Xsens. The suit defines two coordinate frames that are used throughout the thesis:

- The navigation frame n. Since the wearer of the HMTS is not expected to travel long distances, the navigation frame is assumed stationary with respect to the Earth. The origin and direction of the axes are defined at the initialisation of the algorithm.
- The sensor frames s. The MVN Link contains 17 IMUs. Each IMU has its own sensor frame s. For a full list of these sensor locations, see [2]. The axes are aligned along the IMU casing, and its origin is in the centre of the IMU.

The proposed method uses the following data provided by the HMTS:

- ${}^{\mathrm{s}}\mathbf{z}_{t}^{k} \in \mathbb{R}^{3}$, for $k = 1, \ldots, N_k = 17$. The magnetic field observations in atomic units at the origin of the sensor frame s.
- ${}^{\mathrm{n}}\hat{\mathbf{p}}^{k} \in \mathbb{R}^{3}$, for $k = 1, \ldots, N_k = 17$. The position of each sensor of each sensor in the navigation frame n. This position is estimated with the HMTS and its complementary software.
- ${}^{\mathrm{ns}}\hat{\mathbf{q}}^{k} \in \mathbb{R}^{4}$, for $k = 1, \ldots, N_k = 17$. The orientation of the sensor frames s with respect to the navigation frame n in quaternion space. This position is estimated with the HMTS and its complementary software.

N_k denote the number of sensors. The $\hat{}$ on ${}^{\mathrm{n}}\hat{\mathbf{p}}^{k}$ and ${}^{\mathrm{ns}}\hat{\mathbf{q}}^{k}$ is used to distinct the positions and orientation as estimated by the suit and those estimated by the RBPF proposed in Section 3-3. The goal of the particle filter is to improve the estimation of the sensor positions and orientations. The provided observations ${}^{\mathrm{s}}\mathbf{z}_{t}^{k}$ are used the in the magnetic field map estimation explained in Section 3-5.
The suit provides two different qualities of data:

- *Live*, live data indicates that no post processing is done in the Xsens software. This is necessary for real-time SLAM.

- *HD*, HD data is post processed in the Xsens software. This data is post-processed and has less error than the filtered data. Although this data can not be used in real-time SLAM, it can be used to show the capabilities of the SLAM algorithm with better inertial data.

It is important to note that both qualities can be obtained from the same data recording sessions.

3-3 Rao-Blackwellized particle filter

A RBPF is used to estimate the position $\mathbf{p}$, orientation $\mathbf{q}$ and the magnetic field map with mean $\mathbf{m}_t$ and covariance $\mathbf{P}$. A detailed description on the formulation of the magnetic field map can be found in Section 3-5. The estimation problem is described by the following joint posterior.

$$p\left({}^{\mathrm{n}}\mathbf{p}_{1:t}, {}^{\mathrm{ns}}\mathbf{q}_{1:t}, \mathbf{m}_t | {}^{\mathrm{s}}\mathbf{z}_{1:t}, \mathbf{u}_{1:t}\right), \tag{3-1}$$

where subscript $1 : t$ indicates all values op to time t. $\mathbf{u}$ is the input vector used in the odometry model, see Section 3-4. The time index on $\mathbf{m}_t$ does *not* indicate that the map changes over time, but that the estimate of the map changes over time.
A RBPF relies on a separation of the map estimate and the rest of the position and orientation, so that (3-1) becomes

$$p\left({}^{\mathrm{n}}\mathbf{p}_{1:t}, {}^{\mathrm{ns}}\mathbf{q}_{1:t}, \mathbf{m}_t | {}^{\mathrm{s}}\mathbf{z}_{1:t}, \mathbf{u}_{1:t}\right) = \overbrace{p\left(\mathbf{m}_t | {}^{\mathrm{n}}\mathbf{p}_{1:t}, {}^{\mathrm{ns}}\mathbf{q}_{1:t}, {}^{\mathrm{s}}\mathbf{z}_{1:t}\right)}^{\text{Mapping with known poses}} \underbrace{p\left({}^{\mathrm{n}}\mathbf{p}_{1:t}, {}^{\mathrm{ns}}\mathbf{q}_{1:t} | {}^{\mathrm{s}}\mathbf{z}_{1:t}, \mathbf{u}_{1:t}\right)}_{\text{Target Distribution}}. \tag{3-2}$$

This separation forms the basis of almost all recent particle filter SLAM approaches [i.e 27, 28, 30, 73, 76]. The first distribution in (3-2) represents the problem *mapping with known poses*. The approach to solving the mapping problem uses GPR, which is detailed in Section 3-5. The second part, the *target distribution*, is estimated by a particle filter.
The particle filter uses N_i particles to approximate the target distribution in (3-2). Each particle i for $i = 1, \ldots, N_i$, contains a position ${}^{\mathrm{n}}\mathbf{p}_t^i$ and orientation ${}^{\mathrm{ns}}\mathbf{q}_t^i$ at time t, as well as a magnetic field map $\mathbf{m}_t^i$. A more detailed description of the particle state can be found in Section 3-4.

3-3-1 Particle weight

Each particle i has a weight ω_t^i assigned to it. This weight represents the belief that the particle is in the same location in its map, as the actual mobile platform is in the real world. The weight can be sequentially updated, as proposed by Doucet et al. [77].

$$\omega_t^i \propto \frac{p\left({}^{\mathrm{s}}\mathbf{z}_t | \mathbf{m}_{t-1}^i, {}^{\mathrm{n}}\mathbf{p}_t^i, {}^{\mathrm{ns}}\mathbf{q}_t^i\right) p\left({}^{\mathrm{n}}\mathbf{p}_t^i, {}^{\mathrm{ns}}\mathbf{q}_t^i | {}^{\mathrm{n}}\mathbf{p}_{t-1}^i, {}^{\mathrm{ns}}\mathbf{q}_{t-1}^i, \mathbf{u}_{t-1}\right)}{\pi\left({}^{\mathrm{n}}\mathbf{p}_t, {}^{\mathrm{ns}}\mathbf{q}_t | {}^{\mathrm{n}}\mathbf{p}_{t-1}^i, {}^{\mathrm{ns}}\mathbf{q}_{t-1}^i, {}^{\mathrm{s}}\mathbf{z}_{1:t}, \mathbf{u}_{1:t-1}\right)} \omega_{t-1}^i, \tag{3-3}$$

where $p\left({}^{\mathrm{s}}\mathbf{z}_t | \mathbf{m}_{t-1}^i, {}^{\mathrm{n}}\mathbf{p}_t^i, {}^{\mathrm{ns}}\mathbf{q}_t^i\right)$ is the *measurement model*, the confidence in the particles observation, given its map at time $t-1$, and is specified in Section 3-5. The distribution

$p\left({}^{\mathrm{n}}\mathbf{p}_t^i, {}^{\mathrm{ns}}\mathbf{q}_t^i | {}^{\mathrm{n}}\mathbf{p}_{t-1}^i, {}^{\mathrm{ns}}\mathbf{q}_{t-1}^i, \mathbf{u}_{t-1}\right)$ is the *odometry model* of the system. It describes how a system moves from the previous state to the new one, given a certain control input $\mathbf{u}_{t-1}$. The odometry model is described in Section 3-4. The proposal distribution $\pi\left(\cdot\right)$ is chosen equal to the odometry model, as was done in for instance [27, 78]. The weight update (3-3) simplifies significantly to

$$\omega_t^i \propto p\left({}^{\mathrm{s}}\mathbf{z}_t | \mathbf{m}_{t-1}^i, {}^{\mathrm{n}}\mathbf{p}_t^i, {}^{\mathrm{ns}}\mathbf{q}_t^i\right)\omega_{t-1}^i. \tag{3-4}$$

This simplified weight update relies only on the likelihood of an observation ${}^{\mathrm{s}}\mathbf{z}_t$ compared to the magnetic field map, given the current position and orientation of a particle and the previous weight ω_{t-1}^i.

The vector containing the weights of all particles is denoted with $\boldsymbol{\omega}_t$. Particles with high weight, have a magnetic field map for which the observations match, while a lower weight particles observe a magnetic field that mismatches with its magnetic field map.

3-3-2 Sampling

The target distribution in (3-2) is approximated by a finite amount of particles. More particles will cover more of the target distribution, while also increasing the computation time. The RBPF resamples using *systematic resampling* [79].

Particles are only resampled when the effective sample size N_{eff} is low, as proposed in [69] and used in [27, 59]. The effective sample size is introduced in [80]

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N_i}(\omega_t^i)^2}. \tag{3-5}$$

The effective sampling size N_{eff} is low when the weights of the particles differ significantly, implying only a few particles have trajectories with confidence. Only resampling when a sampling threshold N_{resample} is met, such that $N_{\text{eff}} < N_{\text{resample}}$ reduces the risk of discarding particles that are on a likely trajectory.

The weights of the particles are normalised after resampling, such that

$$\sum_{1}^{N_i} \omega_t^i = 1 \quad \forall t. \tag{3-6}$$

3-4 Odometry model

The goal of the RBPF is to estimate a position and orientation of suit. The odometry model describes how the position ${}^{\mathrm{n}}\mathbf{p}_t^i$ and ${}^{\mathrm{ns}}\mathbf{q}_t^i$ change over time, given some control input $\mathbf{u}_t$. Since the suit consists of 17 sensors, there are $N_k = 17$ positions and orientations to estimate. These positions can be expressed in the navigation frame, for each time step t:

$${}^{\mathrm{n}}\mathbf{p}_t^k, \text{ for } k = 1, \ldots, N_k.$$

The orientation of the sensor frames, and thus the sensors itself, can be expressed with respect to the navigation frame, for each time step t:

$$^{\mathrm{ns}}\mathbf{q}_t^k, \text{ for } k = 1, \ldots, N_k.$$

As described in Section 3-2, the HMTS provides $^{\mathrm{s}}\hat{\mathbf{p}}_t^k$ and $^{\mathrm{ns}}\hat{\mathbf{q}}_t^k$. From these values, the change in position and orientation can be computed for $t > 1$ as follows:

$$\begin{aligned} \Delta^{\mathrm{n}}\mathbf{p}_t^k &= {}^{\mathrm{s}}\hat{\mathbf{p}}_t^k - {}^{\mathrm{s}}\hat{\mathbf{p}}_{t-1}^k, && \forall k, \\ \Delta^{\mathrm{ns}}\mathbf{q}_t^k &= {}^{\mathrm{ns}}\hat{\mathbf{q}}_{t-1}^k \odot {}^{\mathrm{ns}}\hat{\mathbf{q}}_t^k, && \forall k, \end{aligned} \tag{3-7}$$

where $\odot$ denotes the quaternion product.
Both $\Delta^{\mathrm{n}}\mathbf{p}_t^k$ and $\Delta^{\mathrm{ns}}\mathbf{q}_t^k$ can be seen as a control input $\mathbf{u}_t$, even though they are measured. This distinction is remarked upon, since the other value provided by the HMTS, $^{\mathrm{s}}\mathbf{z}_t^k$, is considered an observation in the measurement model as explained in Section 3-5.
This description of the control input allows for a similar odometry model as used in [28]. The first novelty of the proposed odometry model is that it allows for the use of more than one sensor. To use multiple magnetometers to construct the magnetic field map, it is necessary to know their positions and orientations relative to each other. These relative positions and orientations are estimated by the HMTS while the RBPF assumes them known.
First, the odometry model for a single sensor is defined. This single single sensor is denoted with $k = \alpha$. This definition is later used to generalise the odometry model for more than one sensor. The RBPF particle filter estimates the position $^{\mathrm{n}}\mathbf{p}_t^{k=\alpha}$ and orientation $^{\mathrm{ns}}\mathbf{q}_t^{k=\alpha}$ of one sensor as follows:

$$\begin{aligned} {}^{\mathrm{n}}\mathbf{p}_t^{k=\alpha} &= {}^{\mathrm{n}}\mathbf{p}_{t-1}^{k=\alpha} + \Delta^{\mathrm{n}}\mathbf{p}_t^{k=\alpha} + \epsilon_{p,t}^{\alpha}, && \epsilon_{p,t}^{\alpha} \sim \mathcal{N}(0, \mathbf{S}_p), \\ {}^{\mathrm{ns}}\mathbf{q}_t^{k=\alpha} &= \Delta^{\mathrm{ns}}\mathbf{q}_t^{k=\alpha} \odot \exp_q\left(\epsilon_{q,t}^{\alpha}\right) \odot {}^{\mathrm{ns}}\mathbf{q}_{t-1}^{k=\alpha}, && \epsilon_{q,t}^{\alpha} \sim \mathcal{N}(0, \mathbf{S}_q), \end{aligned} \tag{3-8}$$

where $\epsilon_{p,t}^{\alpha}$ and $\epsilon_{q,t}^{\alpha}$ are the zero mean Gaussian noise on the position and orientation of the sensor α with covariance matrices $\mathbf{S}_p$ and $\mathbf{S}_q$ respectively. $\exp_q(\cdot)$ is the quaternion exponential function, as defined in [6]. The formulation of (3-8) can be extended to include the remaining sensor positions and orientations. This extended odometry model does not affect the relative positions and orientations of each sensor. This is accomplished by using the same noise for each sensor. If each sensor would have its own noise, even if the distributions from which this noise would be drawn were the same, the sensors would drift from each other. To prevent this, the same values for $\epsilon_{p,t}^{\alpha}$ and $\epsilon_{q,t}^{\alpha}$ are used in the odometry update of each sensors. To compute the translation of each sensor that occurs when rotating sensor α due to $\epsilon_{q,t}^{\alpha}$, the relative position ${}_{\mathrm{rel}}^{\mathrm{n}}\mathbf{p}_t^k$ of each sensor k with respect to sensor α is required and obtained as

$${}_{\mathrm{rel}}^{\mathrm{n}}\mathbf{p}_t^k = {}^{\mathrm{n}}\mathbf{p}_t^k - {}^{\mathrm{n}}\mathbf{p}_t^{k=\alpha}, \qquad \forall k. \tag{3-9}$$

The position and orientation update (3-8) can be generalised for more than one sensor to

$$\begin{aligned} {}^{\mathrm{n}}\mathbf{p}_t^k &= {}^{\mathrm{n}}\mathbf{p}_{t-1}^k + \Delta^{\mathrm{n}}\mathbf{p}_t^k + \epsilon_{p,t}^{\alpha} + \left(\mathbf{R}_t\left(\epsilon_{q,t}^{\alpha}\right)\right) {}_{\mathrm{rel}}^{\mathrm{n}}\mathbf{p}_t^k, && \forall k, \\ {}^{\mathrm{ns}}\mathbf{q}_t^k &= \Delta^{\mathrm{ns}}\mathbf{q}_t^k \odot \exp_q\left(\epsilon_{q,t}^{\alpha}\right) \odot {}^{\mathrm{ns}}\mathbf{q}_{t-1}^k, && \forall k, \end{aligned} \tag{3-10}$$

where $\mathbf{R}_t\left(\epsilon_{q,t}^{\alpha}\right)$ is the rotation matrix representation of the quaternion resulting from $\exp_q\left(\epsilon_{q,t}^{\alpha}\right)$. Note that (3-10) becomes (3-8) for $k = \alpha$, since ${}_{\mathrm{rel}}^{\mathrm{n}}\mathbf{p}_t^{k=\alpha} = \mathbf{0}_3$.

Since each particle computes its own state, a superscript i is added, so that for $i = 1, \ldots, N_i$, (3-10) becomes

$$\begin{aligned} {}^{\mathrm{n}}\mathbf{p}_t^{i,k} &= {}^{\mathrm{n}}\mathbf{p}_{t-1}^{i,k} + \Delta^{\mathrm{n}}\mathbf{p}_t^{k} + \epsilon_{p,t}^{i,\alpha} + \left(\mathbf{R}_t\left(\epsilon_{q,t}^{i,\alpha}\right)\right) {}^{\mathrm{n}}_{\mathrm{rel}}\mathbf{p}_t^{i,k}, && \forall k, \\ {}^{\mathrm{ns}}\mathbf{q}_t^{i,k} &= \Delta^{\mathrm{ns}}\mathbf{q}_t^{k} \odot \exp_q\left(\epsilon_{q,t}^{i,\alpha}\right) \odot {}^{\mathrm{ns}}\mathbf{q}_{t-1}^{k}, && \forall k. \end{aligned} \tag{3-11}$$

Equation (3-11) describes the motion each sensor for each particle. Since the sensor poses are updated using inertial odometry, no absolute position is ever measured. Therefore, the initial positions ${}^{\mathrm{n}}\mathbf{p}_0^{i,k}$ and orientation ${}^{\mathrm{ns}}\mathbf{q}_0^{i,k}$ have to be set for $i = 1, \ldots, N_i$ and $k = 1, \ldots, N_k$. The origin of frame n is set to the position of sensor $k = \alpha$, ${}^{\mathrm{n}}\mathbf{p}_0^{i,k=\alpha}$. Its axes are aligned with its sensor frame, so that ${}^{\mathrm{ns}}\mathbf{q}_0^{i,k=\alpha} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$.
The initial pose of $k = \alpha$ thus becomes

$$ {}^{\mathrm{n}}\mathbf{p}_0^{i,k=\alpha} = \mathbf{0}_3, \qquad \forall i, \tag{3-12}$$

$$ {}^{\mathrm{ns}}\mathbf{q}_0^{i,k=\alpha} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T, \qquad \forall i. \tag{3-13}$$

By choosing the frame n like this, the initial positions and orientations of the other sensors can be directly read from data

$$ {}^{\mathrm{n}}\mathbf{p}_0^{i,k} = {}^{\mathrm{n}}\hat{\mathbf{p}}_0^{k} - {}^{\mathrm{n}}\hat{\mathbf{p}}_0^{k=\alpha}, \qquad \forall i,k, \tag{3-14}$$

$$ {}^{\mathrm{ns}}\mathbf{q}_0^{i,k} = \left({}^{\mathrm{ns}}\hat{\mathbf{q}}_0^{k=\alpha}\right)^{-1} \odot {}^{\mathrm{ns}}\hat{\mathbf{q}}_0^{k}, \qquad \forall i,k, \tag{3-15}$$

Note that (3-14) and (3-15) become (3-12) and (3-13) for $k = \alpha$.
This odometry model updates the positions and orientation of multiple sensors, while maintaining the relative positions and orientations of each sensors, as estimated by the HMTS. Two more novelties to the odometry model are proposed. The first is scaling the positional noise with the input signal and the second is the introduction of a third state to estimate orientation drift.

3-4-1 Positional noise scaling

Scaling the noise results in the following odometry model

$$\begin{aligned} {}^{\mathrm{n}}\mathbf{p}_t^{i,k} &= {}^{\mathrm{n}}\mathbf{p}_{t-1}^{i,k} + \Delta^{\mathrm{n}}\mathbf{p}_t^{k} + \Delta^{\mathrm{n}}\mathbf{p}_t^{k=\alpha} \cdot \epsilon_{p,t}^{i,\alpha} + \left(\mathbf{R}_t\left(\epsilon_{q,t}^{i,\alpha}\right)\right) {}^{\mathrm{n}}_{\mathrm{rel}}\mathbf{p}_t^{i,k} && \forall i,k, \\ {}^{\mathrm{ns}}\mathbf{q}_t^{i,k} &= \Delta^{\mathrm{ns}}\mathbf{q}_t^{k} \odot \exp_q\left(\epsilon_{q,t}^{i,\alpha}\right) \odot {}^{\mathrm{ns}}\mathbf{q}_{t-1}^{k}, && \forall i,k. \end{aligned} \tag{3-16}$$

Scaling the noise signal $\epsilon_{p,t}^{\alpha}$ with the input signal $\Delta^{\mathrm{n}}\mathbf{p}_t^{k=\alpha}$ enables the odometry model to estimate better trajectories by using more information from the HMTS. A common indoor trajectory moves over different floors. The HMTS considers this and adjusts its trajectory estimations so that it does not move perpendicular to a floor. This results an input signal $\Delta^{\mathrm{n}}\mathbf{p}_t^{k=\alpha}$ that moves along these floors. The odometry model (3-11) cannot take this information into account. Forcing the positional noise towards a two dimensional plane can also be achieved by lowering the diagonal entry in the covariance matrix $\mathbf{S}_p$ that corresponds to dimension where the particle cloud should not expand as much. However, since the noise distribution is the same at all times, choosing the covariance such that the particles do not drift trough a floor, but move enough in each direction on a ascending or descending trajectory, is difficult. The scaled noise thus partially removes the otherwise required tuning of $\mathbf{S}_p$. Of course, the covariance matrix $\mathbf{S}_p$ still has to be chosen.

The second benefit is in low motion scenarios. For example, when standing in place, the particle cloud will not expand as much, since $\Delta^{\mathrm{n}}\mathbf{p}_t^{k=\alpha}$ is low. This is especially useful in unvisited locations, since at these locations the magnetic field map might be insufficient to discard the particles that drift away from this stationary position. This is common on startup, where the wearer of the HMTS might stand sill before moving.

3-4-2 Orientational drift

The inertial odometry contains orientational drift. This drift is contained in $\Delta^{\mathrm{ns}}\mathbf{q}_t^k$, but also shows in $\Delta^{\mathrm{n}}\mathbf{p}_t^k$. The odometry models (3-11) and (3-16) are able to account for the drift in $\Delta^{\mathrm{ns}}\mathbf{q}_t^k$ by adding the noise $\exp_q\left(\epsilon_{q,t}^{i,\alpha}\right)$. However, even if the particle filter estimated the the correct orientation, the odometry model (3-10) updates the position with $\Delta^{\mathrm{n}}\mathbf{p}_t^k$ in the navigation frame n and not in the respective sensor frame s. Since this navigation frame is constant, the positional odometry update is independent of the estimated orientation of the particle filter. Depending on the filtering and smoothing approaches used by Xsens, this $\Delta^{\mathrm{n}}\mathbf{p}_t^k$ could be dependant on the orientation estimation used within the suit and later transformed to the navigation frame n. This would result in the orientational drift showing in $\Delta^{\mathrm{n}}\mathbf{p}_t^k$.

The odometry model (3-10) can be adjusted to handle the orientational drift that influences $\Delta^{\mathrm{n}}\mathbf{p}_t^k$. A new state ${}^{\mathrm{ns}}_{\epsilon}\mathbf{q}_t$ is introduced. In this new state, the orientational drift is modelled as a random walk model. Modelling the drift as an additional state increases the dimensionality of the RBPF. The RBPF might require more particles to capture this new dimension. With this new sate and scaled positional noise, the odometry model from (3-16) becomes

$$\begin{aligned}
{}^{\mathrm{ns}}_{\epsilon}\mathbf{q}_t &= {}^{\mathrm{ns}}_{\epsilon}\mathbf{q}_{t-1} \odot \exp_q\left(\epsilon_{q,t}^{i,\alpha}\right), && \forall i,k, \\
{}^{\mathrm{n}}\mathbf{p}_t^{i,k} &= {}^{\mathrm{n}}\mathbf{p}_{t-1}^{i,k} + \left(\mathbf{R}_t\left({}^{\mathrm{ns}}_{\epsilon}\mathbf{q}_t\right)\right)\Delta^{\mathrm{n}}\mathbf{p}_t^k + \Delta^{\mathrm{n}}\mathbf{p}_t^{k=\alpha}\cdot\epsilon_{p,t}^{i,\alpha} + \left(\mathbf{R}_t\left(\epsilon_{q,t}^{i,\alpha}\right)\right){}^{\mathrm{n}}_{\mathrm{rel}}\mathbf{p}_t^{i,k}, && \forall i,k, \\
{}^{\mathrm{ns}}\mathbf{q}_t^{i,k} &= \Delta^{\mathrm{ns}}\mathbf{q}_t^k \odot \exp_q\left(\epsilon_{q,t}^{i,\alpha}\right) \odot {}^{\mathrm{ns}}\mathbf{q}_{t-1}^k, && \forall i,k, \qquad (3\text{-}17)
\end{aligned}$$

where ${}^{\mathrm{ns}}_{\epsilon}\mathbf{q}_t$ is initialised as $= \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$.

Note that the new state ${}^{\mathrm{ns}}_{\epsilon}\mathbf{q}_t$ is updated with the same noise as ${}^{\mathrm{ns}}\mathbf{q}_t^{i,k}$. These are chosen the same, since it is expected that the drift in $\Delta^{\mathrm{n}}\mathbf{p}_t^k$ due to orientational drift and the drift in $\Delta^{\mathrm{ns}}\mathbf{q}_t^k$ originate from the same error in orientation estimate of the HMTS.

3-5 Measurement model

To compute the particle weight ω_t^i in (3-4), the measurement model $p\left({}^{\mathrm{s}}\mathbf{z}_t|\mathbf{m}_{t-1}^i, {}^{\mathrm{n}}\mathbf{p}_t^i, {}^{\mathrm{ns}}\mathbf{q}_t^i\right)$ needs to be evaluated. In this section, the form of $\mathbf{m}_t^i$ is constructed.

As specified in Section 3-2, the HMTS provides the obersvations of the magnetometers in the

IMUs, ${}^{\mathrm{s}}\mathbf{z}_t \in \mathbb{R}^3$, for each sensor. Both the magnitude and direction is used to construct a magnetic field map. The proposed method uses a GP to represent the magnetic field, similar to [e.g 27, 30, 57, 59–61, 64]. Modelling the magnetic field as GP fits nicely with the RBPF, because the GP describes the magnetic field with a mean vector $\mathbf{m}_t$ and covariance matrix $\mathbf{P}_t$. With this representation, the likelihood of observation ${}^{\mathrm{s}}\mathbf{z}_t$ can be estimated by the density function of a multivariate Gaussian distribution

$$p\left({}^{\mathrm{s}}\mathbf{z}_t|\mathbf{m}^i_{t-1}, {}^{\mathrm{n}}\mathbf{p}^i_t, {}^{\mathrm{ns}}\mathbf{q}^i_t\right) = \frac{1}{\sqrt{(2\pi)^{N_k}\left|\mathbf{P}^i_{t-1}\right|}}\exp\left(-\tfrac{1}{2}\left({}^{\mathrm{s}}\mathbf{z}_t - \mathbf{m}^i_{t-1}\right)^T {\mathbf{P}^i_{t-1}}^{-1}\left({}^{\mathrm{s}}\mathbf{z}_t - \mathbf{m}^i_{t-1}\right)\right). \tag{3-18}$$

The exact form of mean $\mathbf{m}_t$ and covariance $\mathbf{P}_t$ is given in Section 3-5-1.
Similar to [28, 61], the GP prior used in this method is

$$\begin{aligned} \varphi(\mathbf{x}) &\sim \mathcal{GP}\left(0, \kappa_{\text{lin}}(\mathbf{x}, \mathbf{x'}) + \kappa_{\text{SE}}(\mathbf{x}, \mathbf{x'})\right), \\ \mathbf{z}_t &= -\nabla\varphi(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_t} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}\left(0, \sigma^2_{\mathbf{z}}\mathbf{I}_3\right), \end{aligned} \tag{3-19}$$

where $\sigma_{\mathbf{z}}$ is the variance on the observations $\mathbf{z}$ and $\varphi(\mathbf{x})$ is the scalar potential of the magnetic field. The linear kernel and SE kernel are defined as

$$\kappa_{\text{lin}}(\mathbf{x}, \mathbf{x'}) = \sigma^2_{\text{lin}}\mathbf{x}^T\mathbf{x}, \tag{3-20}$$

$$\kappa_{\text{SE}}(\mathbf{x}, \mathbf{x'}) = \sigma^2_{\text{SE}}\exp\left(-\tfrac{|\mathbf{x}-\mathbf{x'}|^2}{2l^2_{\text{SE}}}\right), \tag{3-21}$$

where σ_{lin} and σ_{SE} are the magnitude scale hyperparameters and l^2_{SE} is the lengthscale hyperparameter. The indoor magnetic field is assumed to be curl-free. This is ensured by the GP prior (3-19). A detailed derivation of (3-19) is shown in Appendix B.
Constructing a GP through, for instance, the prior (B-7) is computationally expensive. Standard GPR scales with $\mathcal{O}\left(N_z^3\right)$, where N_z is total number of measurements. The problem quickly becomes too large, as the HMTS provides data up to 240Hz. To significantly reduce the computation complexity by making the complexity independent of the number of measurements, an approximation of a full GP is computed.

3-5-1 Reduced rank GPR for multiple sensors

The method uses the GP approximation proposed in [64]. This approximation assumes the scalar potential $\phi(\mathbf{x})$ to be a GP prior, and the observations of the magnetic field to be its gradient, as in (3-19).
Similar to [28, 61], the SE kernel in (3-19) is approximated by an eigendecomposition, so that the kernel in (3-19) becomes

$$\kappa_{\text{approx}}(\mathbf{p}, \mathbf{p'}) \approx \kappa_{\text{lin}}(\mathbf{p}, \mathbf{p'}) + \sum_{j=0}^{N_m}\left(S_{\text{SE}}(\lambda_j)\phi_j(\mathbf{p})\phi_j(\mathbf{p'})\right), \tag{3-22}$$

with N_m number of basis functions $\phi(\mathbf{p})$ and eigenvalues λ. Since the orientation is only applied later in this formulation, the more general $\mathbf{x}$ is replaced by only the position $\mathbf{p}$. $S_{\text{SE}}(\lambda)$ is the spectral density function of $\kappa_{\text{SE}}(\mathbf{x}, \mathbf{x'})$

$$S_{\text{SE}}(\lambda) = \sigma^2_{\text{SE}}\left(2\pi l^2_{\text{SE}}\right)^{\frac{3}{2}}\exp\left(-\tfrac{l^2_{\text{SE}}\lambda^2}{2}\right). \tag{3-23}$$

The eigendecomposition can be solved in a confined domain, denoted with Ω. For rectangular domains, the expressions for $\phi\left(\mathbf{p}\right)$ and λ can be solved in close form. Therefore several domains Ω^d

$$\Omega^d \in [L_1, L_1] \times [L_2, L_2] \times [L_3, L_3] \subset \mathbb{R}^3 \tag{3-24}$$

with centre ${}^{\mathrm{n}}\mathbf{p}^d$ are introduced, where d is the domain index. These domains span the space where measurements are estimated to be taken. The number domains thus depends on the trajectory and can change over time. The expressions for $\phi\left(\mathbf{p}\right)$ and λ are obtained by solving the eigendecomposition of the Laplace operator subject to to Dirichlet boundary conditions of domain Ω

$$\begin{cases} -\nabla^2 \phi_j\left(\mathbf{p}\right) = \lambda_j^2 \phi_j\left(\mathbf{p}\right), & \mathbf{p} \in \Omega, \\ \qquad \phi_j\left(\mathbf{p}\right) = 0, & \mathbf{p} \in \partial\Omega. \end{cases} \tag{3-25}$$

A downside of these boundary conditions, is that it introduces very very low uncertainty near the boundary and no uncertainty exactly on the border. The mean is also fixed on the boundary due to these boundary conditions. Therefore, in the evaluation of the likelihood of a measurement according to (3-4), it will assign either very high weight if a particle is near the border and observes a field similar to the mean, or a very low weight if it does not observe a field close to the mean, while the real magnetic field might not be close to this mean at all. To prevent this from occurring, adaptations are suggested in Section 3-6.
Choosing the domains as done in (3-24), leads to the following analytic expressions for the basis functions and eigenvalues, for $j = 1, \dots, N_m$ and $k = 1, \dots, N_k$

$$\begin{aligned} \phi_j\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) &= \prod_{h=1}^{3} \frac{1}{\sqrt{L_h}} \sin\left(\frac{\pi n_{j,h}\left({}^{\mathrm{n}}p_{h,t}^{i,k} + L_h\right)}{2L_h}\right), \qquad \phi_j\left(\cdot\right) : \mathbb{R}^3 \mapsto \mathbb{R}^1, \\ \lambda_j^2 &= \sum_{h=1}^{3} \left(\frac{\pi n_{j,h}}{2L_h}\right)^2, \end{aligned} \tag{3-26}$$

where matrix $n \in \mathbb{R}^{N_m \times 3}$ consists of an index set of permutations of integers $\{1, 2, \dots, N_m\}$ (i.e. $\{(1,1,1),\ (1,1,2),\ \dots,\ (1,2,1),\ \dots,\ (2,1,1),\ \dots)\}$) for the N_m eigenvalues with the highest spectral density $S_{\mathrm{SE}}\left(\lambda\right)$ as defined (3-23). Subscript h denotes the h'th element, which corresponds to three dimensions. The basis functions, together with the spectral densities, can be used to write the approximation as follows

$$\kappa_{\text{approx}}\left(\mathbf{p}, \mathbf{p'}\right) = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T.$$

$\mathbf{\Phi}\left(\cdot\right)$ is the collection of the eigenfunctions. Let intermediate row block matrix function $\mathrm{X}\left(\cdot\right) : \mathbb{R}^3 \mapsto \mathbb{R}^{N_m+3}$ be defined as

$$\mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) = \begin{bmatrix} \left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right)^T & \phi_1\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) & \phi_2\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) & \cdots & \phi_{N_m}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) \end{bmatrix}. \tag{3-27}$$

$\mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right)$ is computed with a single position, belonging to a single magnetometer. To include multiple sensors, $\mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right)$ is computed for $k = 1, \dots, N_k$ and collected into the column block matrix function $\mathbf{\Phi}\left(\cdot\right): \mathbb{R}^{3N_k} \mapsto \mathbb{R}^{N_k \times (N_m+3)}$

$$\mathbf{\Phi}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,1}, \dots, {}^{\mathrm{n}}\mathbf{p}_t^{i,N_k}\right) = \left[\mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,1}\right)^T \quad \mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,2}\right)^T \quad \dots \quad \mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,N_k}\right)^T\right]^T. \tag{3-28}$$

The block diagonal matrix $\mathbf{\Lambda}$ contains the spectral densities

$$\mathbf{\Lambda} = \mathrm{diag}\left(\left[\sigma_{\mathrm{lin}}^2 \quad \sigma_{\mathrm{lin}}^2 \quad \sigma_{\mathrm{lin}}^2 \quad S_{\mathrm{SE}}\left(\lambda_1\right) \quad S_{\mathrm{SE}}\left(\lambda_2\right) \quad \dots \quad S_{\mathrm{SE}}\left(\lambda_{N_m}\right)\right]\right). \tag{3-29}$$

Note that the first three entries of $\mathbf{\Phi}$ and $\mathbf{\Lambda}$ are the linear kernel of (3-22) with the remaining entries approximating the SE.
To finalise the approximation for the GP prior (3-19), the gradient of $\mathbf{\Phi}\left(\cdot\right)$, $\nabla\mathbf{\Phi}\left(\cdot\right): \mathbb{R}^{3N_k} \mapsto \mathbb{R}^{3N_k \times (N_m+3)}$, has to be computed. The gradient is obtained by concatenating the gradients of $\mathrm{X}\left(\cdot\right)$

$$\nabla\mathbf{\Phi}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,1}, \dots, {}^{\mathrm{n}}\mathbf{p}_t^{i,N_k}\right) = \left[\nabla\mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,1}\right)^T \quad \nabla\mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,2}\right)^T \quad \dots \quad \nabla\mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,N_k}\right)^T\right], \tag{3-30}$$

where $\nabla\mathrm{X}\left(\cdot\right): \mathbb{R}^3 \mapsto \mathbb{R}^{3\times(N_m+3)}$ in turn consists of the gradients of the basis functions

$$\nabla\mathrm{X}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) = \left[\nabla\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right)^T \quad \nabla\phi_1\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) \quad \nabla\phi_2\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) \quad \dots \quad \nabla\phi_{N_m}\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right)\right]. \tag{3-31}$$

The gradients of the basis functions (3-26) can be expressed by defining helper function $g\left({}^{\mathrm{n}}p_{h,t}^{i,k}\right)$, so that $\nabla\phi_j\left(\cdot\right): \mathbb{R}^3 \mapsto \mathbb{R}^3$ becomes

$$g\left({}^{\mathrm{n}}p_{h,t}^{i,k}\right) = \sin\left(\frac{\pi n_{j,h}\left({}^{\mathrm{n}}p_{h,t}^{i,k} + L_h\right)}{2L_h}\right), \tag{3-32}$$

$$\nabla\phi_j\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k}\right) = \begin{bmatrix} \frac{\pi n_{j,1}}{2\sqrt{L_1}L_1}\cos\left(\frac{\pi n_{j,1}\left({}^{\mathrm{n}}p_{1,t}^{i,k}+L_1\right)}{2L_1}\right) g\left({}^{\mathrm{n}}p_{2,t}^{i,k}\right) g\left({}^{\mathrm{n}}p_{3,t}^{i,k}\right) \\ \frac{\pi n_{j,2}}{2\sqrt{L_2}L_2}\cos\left(\frac{\pi n_{j,2}\left({}^{\mathrm{n}}p_{2,t}^{i,k}+L_2\right)}{2L_2}\right) g\left({}^{\mathrm{n}}p_{1,t}^{i,k}\right) g\left({}^{\mathrm{n}}p_{3,t}^{i,k}\right) \\ \frac{\pi n_{j,3}}{2\sqrt{L_3}L_3}\cos\left(\frac{\pi n_{j,3}\left({}^{\mathrm{n}}p_{3,t}^{i,k}+L_3\right)}{2L_3}\right) g\left({}^{\mathrm{n}}p_{1,t}^{i,k}\right) g\left({}^{\mathrm{n}}p_{2,t}^{i,k}\right) \end{bmatrix}. \tag{3-33}$$

The matrix function $\nabla\mathrm{X}$ thus includes the positions of all $k = 1, \dots, N_k$. This differs from existing literature [28, 61], where only a single position is used.

Sequential updating

The sequential solution to the GP becomes a KF. In the update step of the filter uses the observations ${}^{\mathrm{s}}\mathbf{z}_t$ in the sensor frame. To transform the GP to this frame, the following matrix function is defined

$$\mathbf{C}\left(\boldsymbol{\rho}_t^i, \boldsymbol{\psi}_t^i\right) = \begin{bmatrix} {}^{\mathrm{sn}}\mathbf{R}_t^{i,1} \\ {}^{\mathrm{sn}}\mathbf{R}_t^{i,2} \\ \vdots \\ {}^{\mathrm{sn}}\mathbf{R}_t^{i,N_k} \end{bmatrix} \cdot \nabla\mathbf{\Phi}\left(\boldsymbol{\psi}_t^i\right),$$

where the block column matrix with rotation matrices is constructed using the orientations estimated with the particle filter odometry model (3-10). For notational ease, these orientations are collected into intermediate variable $\boldsymbol{\rho}_t^i = \left\{ {}^{\mathrm{sn}}\mathbf{q}_t^{i,1}, \ldots, {}^{\mathrm{sn}}\mathbf{q}_t^{i,N_k} \right\}$. These orientations are used to compute the rotation matrices ${}^{\mathrm{sn}}\mathbf{R}_t^{i,k}$ $\forall k$. $\nabla\boldsymbol{\Phi}\left(\boldsymbol{\psi}_t^i\right)$ is defined in (3-30), with intermediate variable $\boldsymbol{\psi}_t^i = \left\{ {}^{\mathrm{n}}\mathbf{p}_t^{i,1}, \ldots, {}^{\mathrm{n}}\mathbf{p}_t^{i,N_k} \right\}$. The KF is then be written, for $i = 1, \ldots, N_i$, as

$$\begin{aligned}
\mathbf{S}_t^i &= \mathbf{C}\left(\boldsymbol{\rho}_t^i, \boldsymbol{\psi}_t^i\right)\mathbf{P}_{t-1}^i\left(\mathbf{C}\left(\boldsymbol{\rho}_t^i, \boldsymbol{\psi}_t^i\right)\right)^T + \sigma_{\mathbf{z}}^2\mathbf{I}_{3N_k}, && \mathbf{S}_t^i \in \mathbb{R}^{3N_k \times 3N_k},\\
\mathbf{K}_t^i &= \mathbf{P}_t^i\left(\mathbf{C}\left(\boldsymbol{\rho}_t^i, \boldsymbol{\psi}_t^i\right)\right)^T\left(\mathbf{S}_t^i\right)^{-1}, && \mathbf{K}_t^i \in \mathbb{R}^{(N_m+3)\times 3N_k},\\
\mathbf{m}_t^i &= \mathbf{m}_{t-1}^i + \mathbf{K}_t^i\left({}^{\mathrm{s}}\mathbf{z}_t - \mathbf{C}\left(\boldsymbol{\rho}_t^i, \boldsymbol{\psi}_t^i\right)\mathbf{m}_{t-1}^i\right), && \mathbf{m}_t^i \in \mathbb{R}^{(N_m+3)},\\
\mathbf{P}_t^i &= \mathbf{P}_{t-1}^i - \mathbf{K}_t^i\mathbf{S}_t^i\left(\mathbf{K}_t^i\right)^T, && \mathbf{P}_t^i \in \mathbb{R}^{(N_m+3)\times(N_m+3)}.
\end{aligned} \tag{3-34}$$

Note the difference from existing literature [28], where the matrix sizes of $\mathbf{S}_t^i$ and $\mathbf{K}_t^i$ do not scale with the N_k. This is a direct consequence of using mulitple sensors to compute $\nabla\boldsymbol{\Phi}\left(\boldsymbol{\psi}_t^i\right)$. The KF is initialised with mean and covariance

$$\mathbf{m}_0^i = \mathbf{0}_{N_m+3}, \qquad \forall i, \tag{3-35}$$

$$\mathbf{P}_0^i = \boldsymbol{\Lambda}, \qquad \forall i. \tag{3-36}$$

Now, the likelihood of observations ${}^{\mathrm{s}}\mathbf{z}_t$ of multiple sensors with positions $\boldsymbol{\psi}_t^i$ and orientations $\boldsymbol{\rho}_t^i$ is given by (3-18), where the mean and covariance are obtained from the KF (3-34). The mean and covariance do have to be transformed back the sensor frame, so that (3-18) becomes

$$p\left({}^{\mathrm{s}}\mathbf{z}_t|\mathbf{m}_{t-1}^i, \boldsymbol{\psi}_t^i, \boldsymbol{\rho}_t^i\right) = \frac{\exp\left(-\frac{1}{2}\left({}^{\mathrm{s}}\mathbf{z}_t - \mathbf{C}\mathbf{m}_{t-1}^i\right)^T\left(\mathbf{C}\mathbf{P}_{t-1}^i\mathbf{C}^T\right)^{-1}\left({}^{\mathrm{s}}\mathbf{z}_t - \mathbf{C}\mathbf{m}_{t-1}^i\right)\right)}{\sqrt{(2\pi)^{N_k}\left|\mathbf{C}\mathbf{P}_{t-1}^i\mathbf{C}^T\right|}}. \tag{3-37}$$

The argument of matrix function $\mathbf{C}\left(\boldsymbol{\rho}_t^i, \boldsymbol{\psi}_t^i\right)$ is dropped for notational purpose.
The scheme proposed in (3-34) updates a single domain and allows for the use of multiple sensors. It can be expanded to more than one domain, as explained in Section 3-6-2. Similarly, the likelihood evaluation of (3-37) for multiple domains is described in Section 3-6-1.
The complexity of sequential update step of the reduced rank GP scales as $\mathcal{O}\left({N_m}^3\right)$, so not with the number of measurements taken, but only with the number of basisfunctions used to approximate the full GP. This allows the algorithm to construct a large map with many observations.

3-6 Reduced rank GPR with multiple domains and sensors

In addition to the reduced rank GPR, the mapped space is divided to multiple domains to further reduce the computation complexity, as was done in [28]. Since these domains will be smaller than the original space, less eigenfunctions will be required to approximate the full

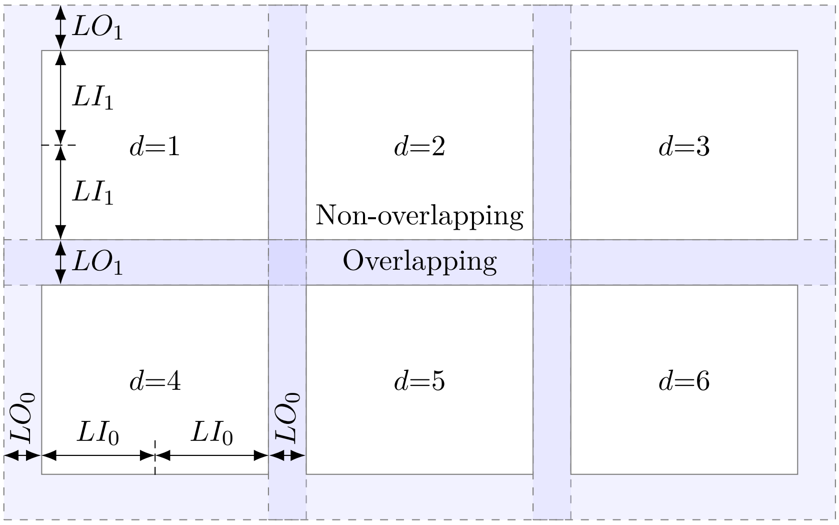


Figure 3-1: Multiple partially overlapping rectangular domains in two dimensions. $L_{1,2}$ indicate the inner lengths and $LO_{1,2}$ indicate the outer lengths.

GP of these individual domains. Each particle has its own domains and each domain will have its own GP, expressed by the mean vector $\mathbf{m}_t^{i,d}$ and covariance matrix $\mathbf{P}_t^{i,d}$, where d is the domain index.

Due to the Dirichlet boundary conditions (3-25), the magntitude of the mean $\mathbf{m}_t^{i,d}$ at each domain boundary is equal to σ_{lin} without any uncertainty. In other words, the algorithm is very certain of its constructed map at the domain boundaries, even though the GP has had no inputs at these locations. As a consequence, particles with sensors located near the domain boundary obtain very high weight if the measured magnetic field happens to coincide with the constructed map.

This overestimation of the certainty can be avoided by dividing each domain into an inner- and outer domain, similar to [28], where the outer parts of the domains overlap with their neighbours, as shown in two dimensions in Figure 3-1. The length of these overlapping areas in each dimension is denoted as $\{LO_0, LO_1, LO_2\}$. The lenghts of the inner domain are denoted with $\{LI_0, LI_1, LI_2\}$, so that $LI_h + LO_h = L_h$. If a particle has sensor positions within the overlapping part of multiple domain boundaries, the weight update step (3-4) can be done with the mean and covariance of the domain for which the boundary is farthest away from the sensor in question. This reduces the influence of the overestimated certainty close the boundary of domain.

The method described in this section is an adaption the the approach used in [28], where only a single sensor was used. Using more than one sensor with multiple domains introduces complexities when updating the particle weight and updating the GP of each domain.

A two dimensional example where these complexities arise is given in Figure 3-2a. Four sensors, of one particle, are located near the boundaries of several domains. Since the sensors are located in four different domain, the observations made by the four sensors can be used to update the GP of each domain, but not all four observations can be used for each domain. For example sensor $k = 1$ is only located in domain $d = 2$, so its observation can only be used to update that specific domain. Section 3-6-1 describes the update steps for multiple domains and sensors in detail.

The same example is used to describe the particle weight update step when using multiple domains and sensors. For each sensor, the likelihood is evaluated in a single domain. This domain is chosen, such that its GP domain is least influenced by the Dirichlet boundary conditions (3-25), as described in Section 3-6-1.

3-6-1 Weight update for multiple domains

To reformulate the weight update equation, several intermediate sets are defined

$$\begin{aligned}
\mathbf{\Psi}_t^{i,d} &= \left\{ \left({}^{\mathrm{n}}\mathbf{p}_t^{i,k} - {}^{\mathrm{n}}\mathbf{p}^{i,d} \right) \mid \min ||{}^{\mathrm{n}}\mathbf{p}_t^{i,k} - {}^{\mathrm{n}}\mathbf{p}^{i,d}||_2, \ \forall d, k \right\}, && \text{(3-38)}\\
\boldsymbol{\varrho}_t^{i,d} &= \{ {}^{\mathrm{ns}}\mathbf{q}_t^{i,k} \mid \min ||{}^{\mathrm{n}}\mathbf{p}_t^{i,k} - {}^{\mathrm{n}}\mathbf{p}^{i,d}||_2, \ \forall d, k \}, && \text{(3-39)}\\
\mathbf{Z}_t^{i,d} &= \{ {}^{\mathrm{s}}\mathbf{z}_t^{k} \mid \min ||{}^{\mathrm{n}}\mathbf{p}_t^{i,k} - {}^{\mathrm{n}}\mathbf{p}^{i,d}||_2, \ \forall d, k \}. && \text{(3-40)}
\end{aligned}$$

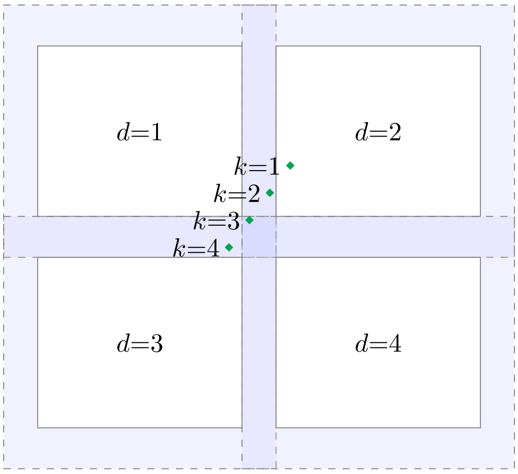
These sets contain, for every domain $\Omega^{i,d}$, 0 up to N_k elements. For each sensor, the domain is selected for which the sensor is closest to its domain center ${}^{\mathrm{n}}\mathbf{p}^{i,d}$. In (3-38) ot (3-40) this is indicated with $\min ||{}^{\mathrm{n}}\mathbf{p}_t^{i,k} - {}^{\mathrm{n}}\mathbf{p}^{i,d}||_2, \ \forall d, k$. Then for each domain, these position of these sensors are collected in $\mathbf{\Psi}_t^{i,d}$, the orientations in $\boldsymbol{\varrho}_t^{i,d}$ and the observations in $\mathbf{Z}_t^{i,d}$. An example of these sets for the sitation in Figure 3-2a is shown in Table 3-2b. Using these sets to update the particle weight ensures the minimal effect of the Dirichlet boundary conditions.

For each particle, a intermediate mean ${}_*\mathbf{m}_t^i$, covariance matrix ${}_*\mathbf{P}_t^i$, transformation function ${}_*\mathbf{C}_t^i$ and observation vector ${}_*\mathbf{Z}_t^i$ is constructed. Then for each particle i these matrices can be constructed for each domain that has at least one sensor closest to its sensor. So for each $d = 0, \dots, N_{d,t}^i \mid \mathbf{\Psi}_t^{id} \neq \emptyset$, where $N_{d,t}^i$ is the number of domains constructed by particle i at time t, these matrices are defined as

$$\begin{aligned}
{}_*\mathbf{m}_t^i &= \left[\left(\mathbf{m}_t^{i,0} \right)^T, \dots, \left(\mathbf{m}_t^{i,N_{d,t}^i} \right)^T \right]^T, && \text{(3-41)}\\
{}_*\mathbf{P}_t^i &= \left[\left(\mathbf{P}_t^{i,0} \right)^T, \dots, \left(\mathbf{P}_t^{i,N_{d,t}^i} \right)^T \right]^T, && \text{(3-42)}\\
{}_*\mathbf{C}_t^i &= \left[\left(\mathbf{C}\left(\boldsymbol{\varrho}_t^{i,0}, \mathbf{\Psi}_t^{i,0} \right) \right)^T, \dots, \left(\mathbf{C}\left(\boldsymbol{\varrho}_t^{i,N_{d,t}^i}, \mathbf{\Psi}_t^{i,N_{d,t}^i} \right) \right)^T \right]^T, && \text{(3-43)}\\
{}_*\mathbf{Z}_t^i &= \left[\left(\mathbf{Z}_t^{i,0} \right)^T, \dots, \left(\mathbf{Z}_t^{i,N_{d,t}^i} \right)^T \right]^T. && \text{(3-44)}\\
{}_*\mathbf{\Psi}_t^i &= \left[\left(\mathbf{\Psi}_t^{i,0} \right)^T, \dots, \left(\mathbf{\Psi}_t^{i,N_{d,t}^i} \right)^T \right]^T, && \text{(3-45)}\\
{}_*\boldsymbol{\varrho}_t^i &= \left[\left(\boldsymbol{\varrho}_t^{i,0} \right)^T, \dots, \left(\boldsymbol{\varrho}_t^{i,N_{d,t}^i} \right)^T \right]^T. && \text{(3-46)}
\end{aligned}$$

An example of these variables is given in Table 3-2c. They are defined, so that the likelihood evaluation (3-37) for multiple domain can be written as

$$p\left({}^{\mathrm{s}}\mathbf{z}_t | {}_*\mathbf{m}_t^i, {}_*\mathbf{\Psi}_t^i, {}_*\boldsymbol{\varrho}_t^i \right) = \frac{\exp\left(-\frac{1}{2} \left({}_*\mathbf{Z}_t^i - {}_*\mathbf{C}_t^i \cdot_* \mathbf{m}_t^i \right)^T \left({}_*\mathbf{C}_t^i \, {}_*\mathbf{P}_t^i \cdot {}_*\mathbf{C}_t^{iT} \right)^{-1} \left({}_*\mathbf{Z}_t^i - {}_*\mathbf{C}_t^i \cdot_* \mathbf{m}_t^i \right) \right)}{\sqrt{(2\pi)^{N_k} |{}_*\mathbf{C}_t^i \, {}_*\mathbf{P}_t^i \cdot {}_*\mathbf{C}_t^{iT}|}}. \tag{3-47}$$



(a) Two dimensional example with four domains and four sensors, indicated by ◆. Used for the example in Table 3-2.

d	k	$\boldsymbol{\Psi}_t^{i,d}$	$\boldsymbol{\varrho}_t^{i,d}$	$\mathbf{Z}_t^{i,d}$
1	3	$\left\{{}^{\mathrm{n}}\mathbf{p}_t^{i,k=3} - {}^{\mathrm{n}}\mathbf{p}^{i,d=1}\right\}$	$\left\{{}^{\mathrm{ns}}\mathbf{q}_t^{i,3}\right\}$	$\left\{{}^{\mathrm{s}}\mathbf{z}_t^{3}\right\}$
2	1, 2	$\left\{{}^{\mathrm{n}}\mathbf{p}_t^{i,k=1} - {}^{\mathrm{n}}\mathbf{p}^{i,d=2}, {}^{\mathrm{n}}\mathbf{p}_t^{i,k=2} - {}^{\mathrm{n}}\mathbf{p}^{i,d=2}\right\}$	$\left\{{}^{\mathrm{ns}}\mathbf{q}_t^{i,1}, {}^{\mathrm{ns}}\mathbf{q}_t^{i,2}\right\}$	$\left\{{}^{\mathrm{s}}\mathbf{z}_t^{1}, {}^{\mathrm{s}}\mathbf{z}_t^{2}\right\}$
3	4	$\left\{{}^{\mathrm{n}}\mathbf{p}_t^{i,k=4} - {}^{\mathrm{n}}\mathbf{p}^{i,d=3}\right\}$	$\left\{{}^{\mathrm{ns}}\mathbf{q}_t^{i,4}\right\}$	$\left\{{}^{\mathrm{s}}\mathbf{z}_t^{4}\right\}$
4	–	$\emptyset$	$\emptyset$	$\emptyset$

(b) Example of the sets defined in (3-38), (3-39) and (3-40) for the situation in Figure 3-2a. The second column k lists the sensors which have the domain's centre in the first column d as closest centre. The positions relative to the corresponding domain, orientations and observations are collected in the sets $\boldsymbol{\Psi}_t^{i,d}$, $\boldsymbol{\varrho}_t^{i,d}$ and $\mathbf{Z}_t^{i,d}$ respectively.

${}_*\mathbf{m}_t^i$	${}_*\mathbf{P}_t^i$	${}_*\mathbf{C}_t^i$	${}_*\mathbf{Z}_t^i$
$\begin{bmatrix} \mathbf{m}_t^{i,d=1} \\ \mathbf{m}_t^{i,d=2} \\ \mathbf{m}_t^{i,d=3} \end{bmatrix}$	$\begin{bmatrix} \mathbf{P}_t^{i,d=1} \\ \mathbf{P}_t^{i,d=2} \\ \mathbf{P}_t^{i,d=3} \end{bmatrix}$	${}_*\mathbf{C}_t^i = \begin{bmatrix} \mathbf{C}\left(\boldsymbol{\varrho}_t^{i,d=3}, \boldsymbol{\Psi}_t^{i,d=3}\right) \\ \begin{bmatrix} \mathbf{C}\left(\boldsymbol{\varrho}_t^{i,d=1}, \boldsymbol{\Psi}_t^{i,d=1}\right) \\ \mathbf{C}\left(\boldsymbol{\varrho}_t^{i,d=2}, \boldsymbol{\Psi}_t^{i,d=2}\right) \end{bmatrix} \\ \mathbf{C}\left(\boldsymbol{\varrho}_t^{i,d=4}, \boldsymbol{\Psi}_t^{i,d=4}\right) \end{bmatrix}$	$\begin{bmatrix} \mathbf{Z}_t^{i,k=3} \\ \begin{bmatrix} \mathbf{Z}_t^{i,k=1} \\ \mathbf{Z}_t^{i,k=2} \end{bmatrix} \\ \mathbf{Z}_t^{i,k=4} \end{bmatrix}$

(c) Example of the variables defined in (3-41), (3-42), (3-43) and (3-44) for the situation in Figure 3-2a. These variables are used to compute the particle weight using (3-47).

Figure 3-2: Two dimensional example of a single particle with four sensors and for domains, as shown in Figure 3-2a The given variables are used to evaluate the particle weight with (3-47).

Updating the weight with multiple domains is summarised in Algorithm 1

Algorithm 1: Particle weight update for multiple domains

```
Input: $\left\{{}^{\mathrm{n}}\mathbf{p}_t^{i,k}, {}^{\mathrm{ns}}\mathbf{q}_t^{i,k}, {}^{\mathrm{s}}\mathbf{z}_t^{k}, \omega_t^i\right\}_{i=1}^{N_i}$ for $k = 1, \ldots, N_k$
Output: $\{\omega_{t+1}^i\}_{i=1}^{N_i}$
for $i = 1$ to $N_i$ do
    Check which domains have a centre ${}^{\mathrm{n}}\mathbf{p}^{i,d}$, for which a sensor position ${}^{\mathrm{n}}\mathbf{p}_t^{i,k}$ is closest
      to this domain's centre. Call these domains $\Omega_0^i, \ldots, \Omega_{N_{d,t}^i}^i$;
    Construct the sets $\boldsymbol{\Psi}_t^{i,d}$, $\boldsymbol{\varrho}_t^{i,d}$, $\mathbf{Z}_t^{i,d}$, using (3-38), (3-39) and (3-40);
    for 0 to $N_{d,t}^i$ do
        Construct ${}_*\mathbf{m}_t^i$, ${}_*\mathbf{P}_t^i$, ${}_*\mathbf{C}_t^i$ and ${}_*\mathbf{Z}_t^i$ using (3-41), (3-42), (3-43) and (3-44);
    end
    Evaluate the likelihood $p\left({}^{\mathrm{s}}\mathbf{z}_t | {}_*\mathbf{m}_{t,*}^i\ \boldsymbol{\Psi}_{t,*}^i\ \boldsymbol{\varrho}_t^i\right)$ using (3-47);
    Update the particle weight $\omega_{t+1}^i = p\left({}^{\mathrm{s}}\mathbf{z}_t | {}_*\mathbf{m}_{t,*}^i\ \boldsymbol{\Psi}_{t,*}^i\ \boldsymbol{\varrho}_t^i\right)\omega_t^i$, according to (3-4);
    Normalise the weight so that the norm of all weights is one, $||\boldsymbol{\omega}_{t+1}||_2 = 1$ holds,
      according to (3-6);
end
return $\boldsymbol{\omega}_{t+1}$
```

3-6-2 Sequential reduced rank GPR for multiple domains

A particle will construct a new domain $\Omega^{i,d}$ when one of its sensors enters a location that would be part of a new domain. This occurs when a sensor enters the 'overlapping' part of two or more domains, while not all the domains that this sensor is located in have been constructed.

Partially overlapping the domains allows for a single sensor to be in multiple domains at once. These domains all need to be updated accordingly. In the three dimensional case, this could mean that a single sensor updates one, two, four or eight different domains.

The KF update step (3-34) becomes dependant on which sensors k are located in what domain d. Suppose a particle has constructed multiple rectangular domains and has multiple sensors, as shown in Figure 3-2a. It is necessary to update the domains only with the observations within that domain.

Several intermediate sets are constructed for $d = 1, \ldots, N_{d,t}^i$

$$\boldsymbol{\psi}_t^{i,d} = \left\{\left({}^{\mathrm{n}}\mathbf{p}_t^{i,k} - {}^{\mathrm{n}}\mathbf{p}^{i,d}\right) \mid {}^{\mathrm{n}}\mathbf{p}_t^{i,k} \in \Omega^{i,d},\ \forall k\right\}, \tag{3-48}$$

$$\boldsymbol{\rho}_t^{i,d} = \left\{{}^{\mathrm{ns}}\mathbf{q}_t^{i,k} \mid {}^{\mathrm{n}}\mathbf{p}_t^{i,k} \in \Omega^{i,d},\ \forall k\right\}, \tag{3-49}$$

$$\boldsymbol{\zeta}_t = \left\{{}^{\mathrm{s}}\mathbf{z}_t^{k} \mid {}^{\mathrm{n}}\mathbf{p}_t^{i,k} \in \Omega^{i,d},\ \forall k\right\}, \tag{3-50}$$

containing the positions with respect to the corresponding domain's centre, orientations or observations of the sensors located within a domain $\Omega^{i,d}$. $N_{d,t}^i$ denotes the number of domains constructed by particle i. Note that his number changes over time and with each particle.

d	$\boldsymbol{\psi}_t^{i,d}$	$N^i_{\boldsymbol{\psi}_t^d}$
1	$\left\{{}^{\mathrm{n}}\mathbf{p}_t^{i,k=2} - {}^{\mathrm{n}}\mathbf{p}^{i,d=1}, {}^{\mathrm{n}}\mathbf{p}_t^{i,k=3} - {}^{\mathrm{n}}\mathbf{p}^{i,d=1}, {}^{\mathrm{n}}\mathbf{p}_t^{i,k=4} - {}^{\mathrm{n}}\mathbf{p}^{i,d=1}\right\}$	3
2	$\left\{{}^{\mathrm{n}}\mathbf{p}_t^{i,k=1} - {}^{\mathrm{n}}\mathbf{p}^{i,d=2}, {}^{\mathrm{n}}\mathbf{p}_t^{i,k=2} - {}^{\mathrm{n}}\mathbf{p}^{i,d=2}, {}^{\mathrm{n}}\mathbf{p}_t^{i,k=3} - {}^{\mathrm{n}}\mathbf{p}^{i,d=2}\right\}$	3
3	$\left\{{}^{\mathrm{n}}\mathbf{p}_t^{i,k=3} - {}^{\mathrm{n}}\mathbf{p}^{i,d=3}, {}^{\mathrm{n}}\mathbf{p}_t^{i,k=4} - {}^{\mathrm{n}}\mathbf{p}^{i,d=3}\right\}$	2
4	$\left\{{}^{\mathrm{n}}\mathbf{p}_t^{i,k=3} - {}^{\mathrm{n}}\mathbf{p}^{i,d=4}\right\}$	1

Table 3-1: Example of the set $\boldsymbol{\psi}_t^{i,d}$ defined in (3-48) and the size of the set for the situation given in Figure 3-2a. Each domain d has a set $\boldsymbol{\psi}_t^{i,d}$, containing the position of each sensor located in that domain. The KF (3-51) uses only these sensors to update the GP of the respective domain.

Let $N^i_{\boldsymbol{\psi}_t^d}$ denote the size of these sets. An example is given in Table 3-1 for the set defined in (3-48).

The KF (3-34) can then be reformulated for each domain $\Omega^{i,d}$

$$\begin{aligned}
\mathbf{S}_t^{i,d} &= \mathbf{C}\left(\boldsymbol{\rho}_t^{i,d}, \boldsymbol{\psi}_t^{i,d}\right)\mathbf{P}_{t-1}^{i,d}\left(\mathbf{C}\left(\boldsymbol{\rho}_t^{i,d}, \boldsymbol{\psi}_t^{i,d}\right)\right)^T + \ \sigma_{\mathbf{z}}\mathbf{I}_{3N^i_{\boldsymbol{\psi}_t^d}}, && \\
& && \mathbf{S}_t^{i,d} \in \mathbb{R}^{3N^i_{\boldsymbol{\psi}_t^d}\times 3N^i_{\boldsymbol{\psi}_t^d}}, \\
\mathbf{K}_t^{i,d} &= \mathbf{P}_t^{i,d}\left(\mathbf{C}\left(\boldsymbol{\rho}_t^{i,d}, \boldsymbol{\psi}_t^{i,d}\right)\right)^T\left(\mathbf{S}_t^{i,d}\right)^{-1}, && \mathbf{K}_t^{i,d} \in \mathbb{R}^{(N_m+3)\times 3N^i_{\boldsymbol{\psi}_t^d}}, \\
\mathbf{m}_t^{i,d} &= \mathbf{m}_{t-1}^{i,d} + \mathbf{K}_t^{i,d}\left({}^{\mathrm{s}}\mathbf{z}_t - \mathbf{C}\left(\boldsymbol{\rho}_t^{i,d}, \boldsymbol{\psi}_t^{i,d}\right)\mathbf{m}_{t-1}^{i,d}\right), && \mathbf{m}_t^{i,d} \in \mathbb{R}^{(N_m+3)}, \\
\mathbf{P}_t^{i} &= \mathbf{P}_{t-1}^{i,d} - \mathbf{K}_t^{i,d}\mathbf{S}_t^{i,d}\left(\mathbf{K}_t^{i,d}\right)^T, && \mathbf{P}_t^{i,d} \in \mathbb{R}^{(N_m+3)\times(N_m+3)}.
\end{aligned} \tag{3-51}$$

Each particle thus gets its own KF matrices for each domain $\Omega^{i,d}$ it has constructed. Note that the mapping function $\mathbf{C}\left(\boldsymbol{\rho}_t^{i,d}, \boldsymbol{\psi}_t^{i,d}\right)$ changes it in- and output size for different domains at different times.

The reformulation of the sequential update step (3-51) increases the computational complexity by running the steps for each domain $\Omega^{i,d}$ for $d = 1, \ldots, N^i_{d,t}$, but not that the inversion of the innovation matrix $\mathbf{S}_t^{i,d}$ can become easier, since $N^i_{\boldsymbol{\psi}_t^d} \leq N_k$. If sensors are located in the overlapping part multiple domains, more computations are required since the same position, orientation and observation are used multiple times. In the worst case there are N_k sets of $N^i_{\boldsymbol{\psi}_t^d} = N_k$ entries. In this case the update step (3-51) is run N_k times, with the largest matrix size possible. However this is very likely to occur of the size of the domains is chosen sufficiently large. Since the sensors are located on the HMTS, they can only be a person's length apart at most.

The algorithm for updating each domain is given in Algorithm 2.

Algorithm 2: Sequential domain update for multiple domains

```
Input: $\left\{ {}^{\mathrm{n}}\mathbf{p}_t^{i,k}, {}^{\mathrm{ns}}\mathbf{q}_t^{i,k}, {}^{\mathrm{s}}\mathbf{z}_t^{k}, \mathbf{m}_{t-1}^{i,d}, \mathbf{P}_{t-1}^{i,d} \right\}_{i=1}^{N_i}$ for $d = 1, \ldots, N_{d,t}^i$
Output: $\left\{ \mathbf{m}_t^{i,d}, \mathbf{P}_t^{i,d} \right\}_{i=1}^{N_i}$ for $d = 1, \ldots, N_{d,t}^i$
for $i = 1$ to $N_i$ do
    for $d$ to $N_{d,t}^i$ do
        Construct the intermediate sets $\boldsymbol{\psi}_t^{i,d}$, $\boldsymbol{\rho}_t^{i,d}$ and $\boldsymbol{\zeta}_t$ using (3-48), (3-49) and (3-50);
        Perform the KF update step (3-51) to obtain $\mathbf{m}_t^{i,d}$ and $\mathbf{P}_t^{i,d}$ ;
    end
end
return $\left\{ \boldsymbol{m}_t^{i,d}, \boldsymbol{P}_t^{i,d} \right\}_{i=1}^{N_i}$
```

3-7 Algorithm

The complete magnetic field SLAM algorithm is summarised in Algorithm 3.

Algorithm 3: Magnetic field SLAM using a HMTS, a RBPF and reduced rank GPR

```
Input: $\left\{ {}^{\mathrm{s}}\mathbf{z}_t^{k}, {}^{\mathrm{n}}\hat{\mathbf{p}}_{,}^{k} {}^{\mathrm{ns}}\hat{\mathbf{q}}^{k} \right\}_{t=1}^{N_{t-1}}$ for $k = 1, \ldots, N_k$
Output: $\left\{ {}^{\mathrm{n}}\bar{\mathbf{p}}_{0:N_{t-1}}^{k=\alpha}, {}^{\mathrm{ns}}\bar{\mathbf{q}}_{0:N_{t-1}}^{k=\alpha} \right\}$
Initialise the $N_i$ particles according to (3-14) and (3-15), and construct domain $\Omega_0^i$ with
  $\mathbf{m}_{0,0}^i$ and $\mathbf{m}_{0,0}^i$ according to (3-35) and (3-36);
for $t = 0$ to $N_{t-1}$ do
    Update the particle weights $\omega_t^i$ according to Algorithm 1;
    Compute $N_{\mathrm{eff}}$;
    if $N_{\mathit{eff}} < N_{\mathit{resample}}$ then
        Resample using systematic resampling [79];
    end
    for $i = 1$ to $N_i$ do
        Update the positions and orientations according to (3-16) or (3-17);
        for $k$ to $N_k$ do
            if ${}^{n}\boldsymbol{p}_t^i$ is in a new subdomain $\Omega^{i,d}$ then
                Create new subdomain $\Omega^{i,d}$ for particle $i$ and initialise $\mathbf{m}_t^{i,d}$ and $\mathbf{P}_t^{i,d}$
                  according to (3-35) and (3-36);
            end
        end
    end
    Update the magnetic field map $\mathbf{m}_t^{i,d}$ and $\mathbf{P}_t^{i,d}$ according to Algorithm 2;
end
Set $\left\{ {}^{\mathrm{n}}\bar{\mathbf{p}}_{0:N_{t-1}}^{k=\alpha}, {}^{\mathrm{ns}}\bar{\mathbf{q}}_{0:N_{t-1}}^{k=\alpha} \right\}$ equal to the trajectories of the particle with the highest weight;
return $\left\{ {}^{n}\bar{\boldsymbol{p}}_{0:N_{t-1}}^{k=\alpha}, {}^{ns}\bar{\boldsymbol{q}}_{0:N_{t-1}}^{k=\alpha} \right\}$
```

Chapter 4

Results

In this chapter, the results of the experiments are presented. In Section 4-1 the proposed method is divided into smaller parts and the performance of these parts are then verified. Section 4-2 shows the results of the SLAM algorithm using a single magnetometer to construct the magnetic field map. Finally, the performance of the SLAM algorithm is shown in Section 4-3.

Throughout this chapter an OTS [81] is used. This system provides very accurate position estimates, which can be used as a groundtruth for comparison. Except for the experiment described in Section 4-1-1 and those where no OTS is used, the OTS tracks the position of the IMU located on the chest. This IMU was chose over others, because it is clearly visible from the cameras mounted on the ceiling.

4-1 Method verification

The proposed method consists of different elements and relies on assumptions. One assumption is that the different magnetometers provide matching observations of the magnetic field. This assumption is tested in Section 4-1-1. In Section 4-1-2, hyperparameters are determined and the performance of the GP is tested, using validation and training data. Section 4-1-3 shows the evolution of the magnetic field map as more observations are added to the GP. In Section 4-1-4 the effect of the domain transitions on the modelled magnetic field is investigated.

4-1-1 Magnetometer calibration

This experiment is performed to ensure that different magnetometers measure the same magnetic field in the same location. The two IMUs on the hands of HMTS were tracked by the Optitrack system [81]. This setup provides accurate optical position estimates for two magnetometers and their magnetic field observations. For the two magnetometers, matching positions are sought, and for these positions the magnetic field observations are compared.

Data set	distance	mean	std
Before calibration	< 1cm	0.027	0.0064
After calibration	< 1cm	0.015	0.0076

Table 4-1: Mean and standard deviation of the difference in magnetic field magnitude observed by the two magnetometers for different data sets and distance for which two magnetometers are considered at the same location.

Magnetometer calibration aims to scale the measurements so that each magnetometer provides observations of the field scaled to the same magnitude. After calibration, the magnetometers are expected to observe the same magnetic field at these matching locations. The experiment is performed before and after all the seventeen magnetometers on the HMTS are manually calibrated using the *magnetic field mapper* [82] tool by Xsens. The magnetometers were calibrated outside, as the magnetic field is expected to not be distorted outside. The difference in measured field magnitude by the two magnetometers before and after calibration is compared. Two magnetic calibration tests were performed. One before magnetometer calibration and one after.For both tests, the magnetometers were considered at the same location, if they were within 1cm of each other, as determined by the OTS. This distance is chosen, because the magnetic field is not expected to differ spatially at these distances. The mean and standard deviation for all tests are shown in Table 4-1. The difference in magnetic field magnitude is lower after calibration. The difference in observed magnetic field could have consequences where multiple sensors are used to construct the magnetic field map, since the GP expects each sensor to have the same characteristics.

4-1-2 Hyperparameters

To obtain the hyperparameters, the max-log-likelihood approach as described in Appendix C was used. The first 2000 data points after downsampling were used. To verify the quality of the GP, the data was split into 80% training data and 20% validation data.

Table 4-2a shows the results, computed with the optimised hyperparameters $\sigma_{\text{lin}} = 0.850$, $\sigma_{\text{SE}} = 1.15$, $l_{\text{SE}} = 0.388$, $\sigma_{\mathbf{z}} = 0.00357$ and $N_m = 500$, $L = \{7.8, 7.8, 2.75\}$. The GP is constructed with the observations of a single sensor. In Table 4-2a, the mean error shows that the GP is close to the measured field. However, the GP overestimates its own accuracy, indicated by the very low $\sigma_{\mathbf{z}}$. The variance is so low no validation observation falls into the 95% confidence interval.

Better results are obtained when choosing hyperparameters characteristic with the Earth's magnetic field. With hyperparameters $\sigma_{\text{lin}} = 1.05$, $\sigma_{\text{SE}} = 0.44$, $l_{\text{SE}} = 0.71$, $\sigma_{\mathbf{z}} = 0.10$ the validation observations fit better into the modelled magnetic field. The results are shown in Table 4-2b. Although the mean error is slightly worse, the confidence of the GP is more realistic, as the fraction of validation observation that fall within the confidence intervals is more realistic.

Constructing the magnetic field with all $N_k = 17$ magnetometers results in a less accurate map if the same hyperparameters are chosen. An explanation could be that the different magnetometers have a slightly different calibration, resulting in different observed magnitudes. The hyperparameters are shared for each sensor, so the magnitude scales σ_{lin} and σ_{SE} cannot

Mean error	0.0104
68% interval	0.0
95% interval	0.0

(a) A single IMU, with hyperparameters obtained trough optimisation [61] $\sigma_{\text{lin}} = 0.850$, $\sigma_{\text{SE}} = 1.15$, $l_{\text{SE}} = 0.388$, $\sigma_{\mathbf{z}} = 0.00357$

Mean error	0.0113
68% interval	0.92
95% interval	1.0

(b) A single IMU, with hyperparameters $\sigma_{\text{lin}} = 1.05$, $\sigma_{\text{SE}} = 0.44$, $l_{\text{SE}} = 0.71$, $\sigma_{\mathbf{z}} = 0.10$.

Mean error	0.0442
68% interval	0.5
95% interval	0.72

(c) All $N_k = 17$ IMUs, with hyperparameters $\sigma_{\text{lin}} = 1.05$, $\sigma_{\text{SE}} = 0.63$, $l_{\text{SE}} = 0.32$, $\sigma_{\mathbf{z}} = 0.39$.

Table 4-2: Results for the magnetic field with different hyperparameters and number of IMUs. The data is split in training (80%) and validation (20%) data. The 68% and 95% interval shows what fraction of validation data was within once or twice standard deviation from the mean of the GP. Error to the mean is defined as $\epsilon = ||\mathbf{C}\left(\boldsymbol{\rho}_t^i, \boldsymbol{\psi}_t^i\right) m\left(\mathbf{x}\right) - \mathbf{z}||$.

be chosen for each sensor separately. Additionally, the orientation of the sensors with respect to each other need to be accurate as well, since the direction of the observed field is vital for GP. The extra difficulties introduced by the additional sensors result in an higher measurement noise $\sigma_{\mathbf{z}}$. The SE magnitude σ_{SE} is also raised so that the higher measurement noise is still smaller than this magnitude. The results of a constructed magnetic field map with hyperparameters, $\sigma_{\text{lin}} = 1.05$, $\sigma_{\text{SE}} = 0.63$, $l_{\text{SE}} = 0.32$, $\sigma_{\mathbf{z}} = 0.39$, is shown in Table 4-2c. The mean error is significantly higher than for the map constructed with only one sensor. Due to the much larger $\sigma_{\mathbf{z}}$, the fraction validation observations that fit into the confidence intervals is still relatively high, considering the large mean error. For multiple sensors, GPs with lower lengthscale l_{SE}^2 seems to perform better. A possible explanation could be that due to the closeness of the different magnetometers, small anomalies can also be observed. A GP with a large lengthscale would have difficulties mapping these smaller anomalies.

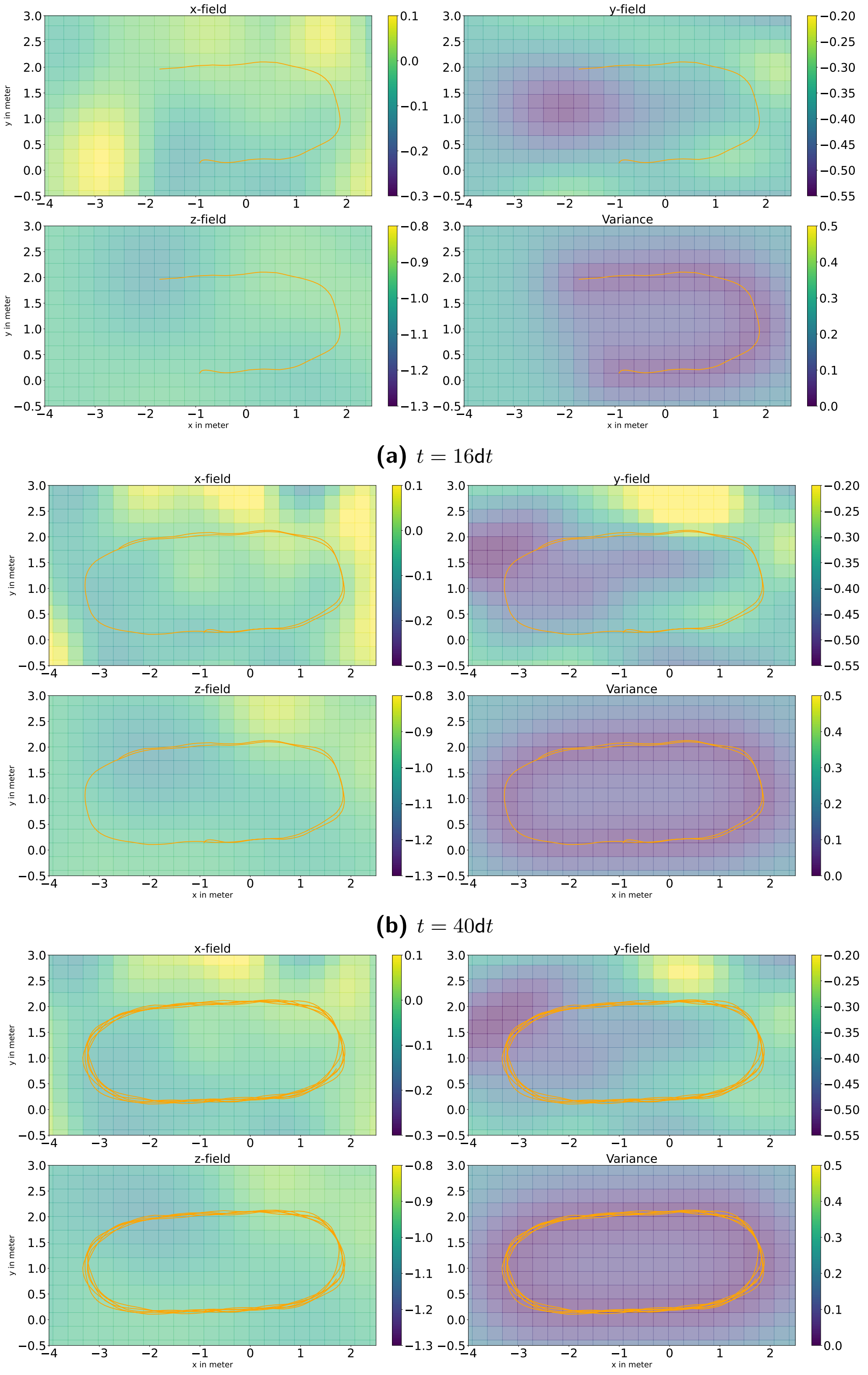
4-1-3 Sequential magnetic field map

The magnetic field maps constructed in Section 4-1-2 are not constructed sequentially. To show the evolution of the map as more observations are made, a magnetic field map is constructed and evaluated at different times. The hyperparameters are $\sigma_{\text{lin}} = 1.05$, $\sigma_{\text{SE}} = 0.63$, $l_{\text{SE}} = 1$, $\sigma_{\mathbf{z}} = 0.05$.

The GP is sampled in a xy-plane, with the height z equal to the height of the sensor at the first measurement. Since no significant height differences are present in the trajectory, this plane is expected to be a good view of the full 3D constructed magnetic field map.

First, sequentially updating the magnetic field map with a single magnetometer is investigated. Figure 4-1 shows the magnetic field map at three different time steps. It shows the estimated field strength in each direction, as well as the variance. The magnetic field changes trough the first circulation, as observations from the newly visited locations are added to the GP. Afterwards, when no new locations are visited, the map does not change significantly near the trajectory. This behaviour is expected, since the magnetic field is assumed to be constant over time, and no ferromagnetic materials were moved during the experiment.

Second, the same experiment is repeated with all $N_k = 17$ IMUs. The same hyperparameters are used. The evolution of the constructed magnetic field is shown in Figure 4-2. The evolution of the magnetic field shows similar behaviour when constructed with a single mag-

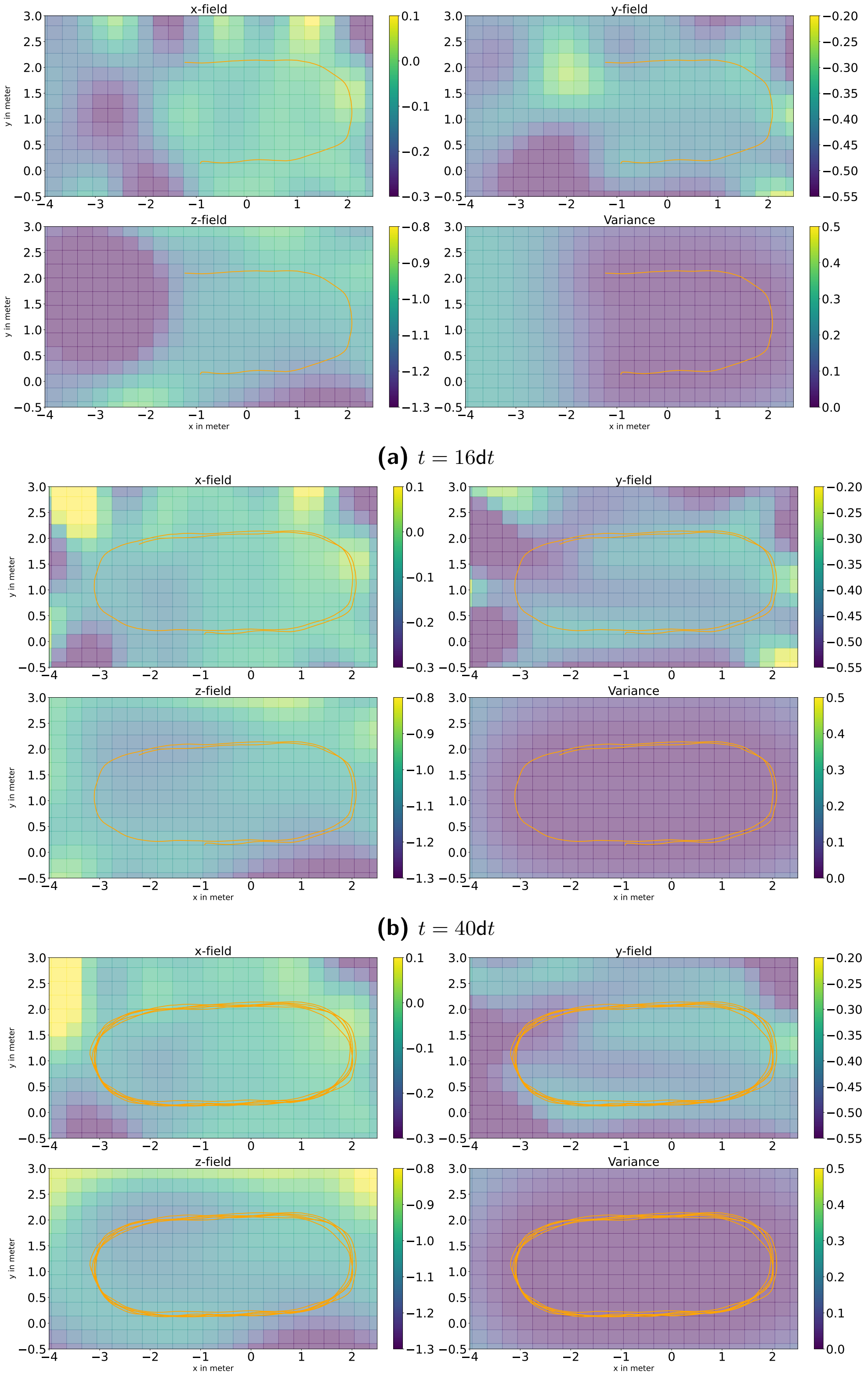


(a) $t = 16\text{d}t$

(b) $t = 40\text{d}t$

(c) $t = 116\text{d}t$

Figure 4-1: Evolution of the magnetic field map over time. The trajectory is shown with the orange line. The colormap represents the field strength. The magnetic field map is constructed using one magnetometer. Hyperparameters: $\sigma_{\text{lin}} = 1.05$, $\sigma_{\text{SE}} = 0.63$, $l_{\text{SE}} = 1$, $\sigma_{\mathbf{z}} = 0.05$



(a) $t = 16\mathrm{d}t$

(b) $t = 40\mathrm{d}t$

(c) $t = 116\mathrm{d}t$

Figure 4-2: Evolution of the magnetic field map over time. The trajectory of the IMU on the pelvis is shown with the orange line. The colormap represents the field strength. The magnetic field map is constructed using $N_k = 17$ magnetometers. Hyperparameters: $\sigma_{\mathsf{lin}} = 1.05$, $\sigma_{\mathsf{SE}} = 0.63$, $l_{\mathsf{SE}} = 1$, $\sigma_{\mathbf{z}} = 0.05$

netometer. An effect of using more magnetometers is that the variance is low further away from the trajectory line shown in the figures, as seen in the variance plots of both Figure 4-1 and Figure 4-2. This occurs because a larger area is covered by multiple IMUs compared to a single one. However, the trajectory line in Figure 4-2 shows only the trajectory of the pelvis sensor.

The magnetic field strength differs more in the x and y direction over short lengths than in the z direction. Unfortunately, the squared exponential covariance kernel $\kappa_{\text{SE}}(\mathbf{x}, \mathbf{x'})$ (3-21) the lengthscale hyperparameter l_{SE} cannot be chosen for each direction individually.

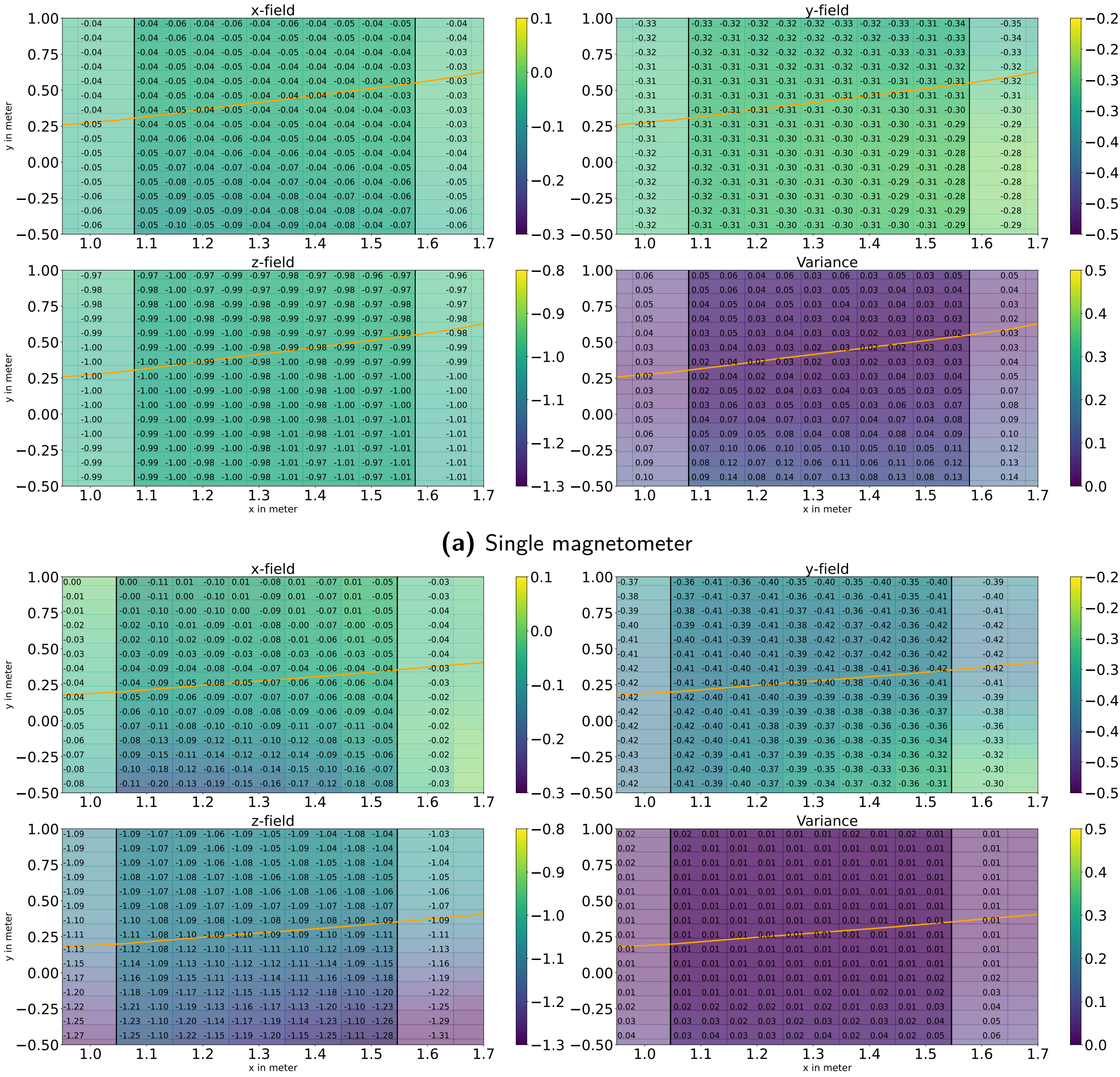
The variance plots in Figure 4-1a and Figure 4-2a shows that the GP is confidant around the trajectory and gradually falling further away from the measurements. After the trajectory is closed in Figure 4-1b and Figure 4-2b, the variance is very similar over the area where the trajectory is. Note that in this experiment the boundaries of the domain are chosen far away from the trajectory, so that they affect the magnetic field map minimally due to the Dirichlet boundary conditions. Therefore, they do not show in the plots of Figure 4-1.

4-1-4 Domain transitions

To verify that no unexpected behaviour occurs when sensors are near domain boundaries and transition to new domains, a scenario with a simple trajectory and small domains is investigated.A close up view of the transition is given in Figure 4-3. The inner domain boundaries are shown with the vertical black lines. Between these two lines is the overlapping part of the domains. The orange line shows the trajectory. Figure 4-3a shows the magnetic field constructed using one IMU and Figure 4-3b shows the field using all $N_k = 17$ IMUs. To closely investigate the difference in the magnetic field map of each domain, the magnetic field is sampled. At each sampled location a coloured rectangle is shown. In this rectangle, a scalar value represents the strength of the magnetic field at that the center of the coloured rectangle. On the overlapping part, two scalar values are shown in each rectangle. The two scalars are computed using the constructed magnetic field from different domains. With these scalars, the behaviour in the overlapping part can be more closely investigated. The values are expected to be close together, since both GPs model the field at the same location.

With a single IMU, the magnetic field in the overlapping areas are very similar in all directions. However, for all IMUs, the overlapping field in the x-direction differ slightly more, while the other direction are modelled much more equal. This slight boundary transition of the x-field could be caused by the difficulty of modelling the x-field for multiple sensors, regardless of domain transitions, as experienced in Section 4-1-2.

For both a single and multiple magnetometers, the two sampled values of the GP are close to each other near the trajectory, while the difference grows further from the trajectory. This is expected behaviour in the case of a single magnetometer, as the GP is less confident in these locations, as the observations are further away. This also shows in the variance. However, for multiple sensors the difference of the two fields starts to show further from the trajectory line. This is also expected, as only the trajectory of the IMU on the pelvis is shown. The other sensors provide observations around the shown trajectory, enabling the GP to model the field better further away from the trajectory line.



(a) Single magnetometer

(b) $N_k = 17$ magnetometers

Figure 4-3: Comparison of the modelled magnetic field at an overlapping part of two domains, constructed with a single and multiple magnetometers. The orange line shows the trajectory of the sensor on the pelvis. The scalars printed within the coloured rectangles give the value of the sampled GP at the centre of the rectangle. On the overlapping part, two scalars are shown, each belonging to another domain. Hyperparameters: $\sigma_{\text{lin}} = 1.05$, $\sigma_{\text{SE}} = .02$, $l_{\text{SE}} = 1$, $\sigma_{\mathbf{z}} = 0.05$

Odometry model	σ_{lin}	l	σ_{SE}	$\sigma_{\mathbf{z}}$	L	LO	N_m
(3-16)	1.10	0.5	0.2	0.05	$\begin{bmatrix}5.5 & 4 & 3\end{bmatrix}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$	500
(3-17)	1.10	1	0.2	0.075	$\begin{bmatrix}5.5 & 4 & 3\end{bmatrix}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$	400

Odometry model	N_i	N_{eff}	$\frac{1}{\text{d}t}\mathbf{S}_p$	$\frac{1}{\text{d}t}\mathbf{S}_q$	N_k
(3-16)	60	0.5	$3\mathbf{I}_3$	$10^{-5}\begin{bmatrix}0.02 & 0 & 0\\ 0 & 0.02 & 0\\ 0 & 0 & 1\end{bmatrix}$	1
(3-17)	150	0.5	$0.01\mathbf{I}_3$	$10^{-5}\begin{bmatrix}0.02 & 0 & 0\\ 0 & 0.02 & 0\\ 0 & 0 & 1\end{bmatrix}$	1

Table 4-3: The used parameters for the comparison of odometry model used in the RBPF.

4-2 SLAM using a single magnetometer

In this section, the performance of the proposed algorithm is evaluated with experiments. This sections shows the results of the algorithm when only one magnetometer is used to construct the magnetic field map. This does not mean that the HMTS suit can be replaced by a single IMU, because the HMTS still uses all IMUs to compute its odometry.

In Section 4-2-1 the results of using different odometry models is shown. Both qualities of data from the HMTS, as explained in Section 3-2, were used to asses the performance of the SLAM algorithm. The results are shown in Section 4-2-2. For both these results, no domain transitions are used. The odometry is covered by a single domain. Finally, a set of stairs is traversed to show the performance in a fully 3D environment, shown in Section 4-2-3. The space covered with the odometry in this experiment does require multiple domains.

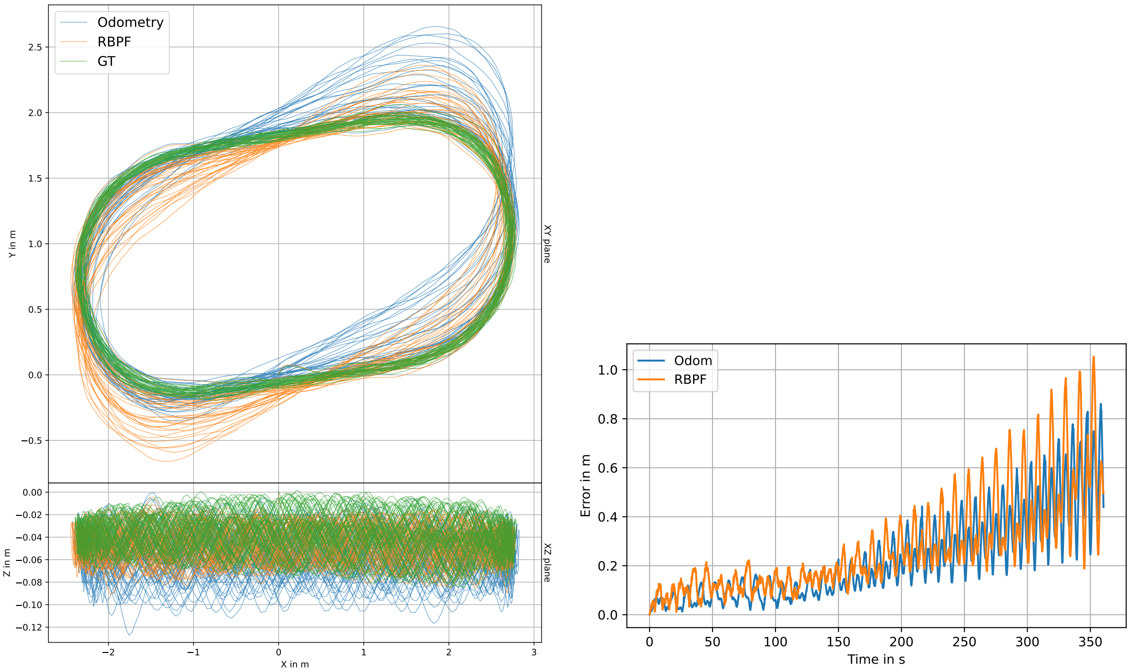
4-2-1 Comparison of the odometry models

In Section 3-4, two odometry models are presented. The first model (3-16) applies uncertainty on both the position and orientation trough added noise scaled with the input $\Delta^{\text{n}}\mathbf{p}_t^{k=\alpha}$. The second model (3-17) builds upon the first with an additional state to account for drift in the orientation, so that it can correct the input signal $\Delta^{\text{n}}\mathbf{p}_t^{k=\alpha}$. Both models can be used seperately in the SLAM algorithm. The results are compared in this section. In both situation, the post-processed HD data was used. Table 4-3 shows the settings used for both results.

Figure 4-4 shows the result of the SLAM algorithm, using the first odometry model.

From Figure 4-4a it can be seen that the odometry suffers from drift in all four corners. The trajectory estimated by the RBPF shows significantly less drift in two corners, but significantly worse performance in the other two corners. The RBPF is able to correct its own large error in these corners, but not able to prevent the error in the two badly performing corners.

In Figure 4-4b the Euclidian distance of both the odometry and the RBPF to groundtruth is shown. It is clear that both errors increase over time with the same slope, although the



(a) Odometry, RBPF estimate and groundtruth

(b) Euclidian distance to groundtruth.

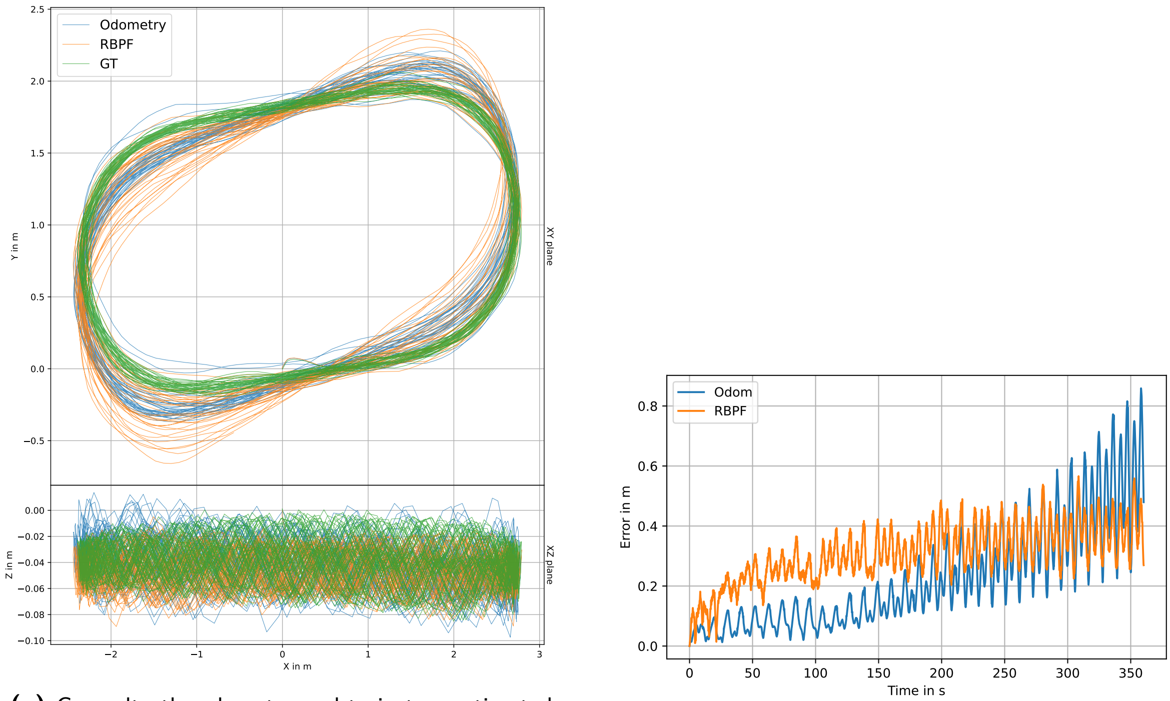
Figure 4-4: Results for first odometry model (3-16). Post-processed odometry was used.

RBPF has a lower minimum and higher maximum error. Through different tunings of the SLAM algorithm, the contrast in performance in the two sets of corners is similar. All test performed significantly worse in the two right corners, and good performance in the two left corners.

The cause of this behaviour is the orientation drift present in $\Delta^{\mathrm{n}}\mathbf{p}_t^{k=\alpha}$. This can verified by running the same experiment, but with the second odometry model. The results are shown in Figure 4-5. For these results, significantly more particles were used, so that the improvement of the second odometry model can be validated.

The esimated trajectory does not suffer from a drifting error and settles after around 150 seconds. After 250 seconds the RBPF starts to outperform the odometry. This lack of dirft in the error shows that the RBPF is capable of using the map to correct for the drift.

The performance of the RBPF is significantly worse compared the odometry early on. A possible cause could be that the first circulation of particles is polluted due to the introduced noise. A map is constructed with this incorrect circulation, resulting in a incorrect map. This map is than used to estimate the position in the further circulations. This would cause a large error appearing early, but settling afterwards. This behaviour is consistent when using lower values of orientational and positional noise. As the portion of the positional noise is shifted to $\mathbf{S}_q$, the covariance matrix $\mathbf{S}_p$ can be significantly lower than for the experiments without this odometry model.



(a) Groundtruth, odometry and trajectory estimated by the RBPF.

(b) Euclidian distance to the groundtruth

Figure 4-5: Results for the second odometry model (3-17). Post-processed odometry was used.

4-2-2 Impact of post-processed odometry

Since the method proposed in Chapter 3 is sequential it is capable of running real-time, assuming the computations can be performed fast enough. Therefore, it is interesting to investigate the impact of using real-time or post-processed odometry.

For these results, the data was collected from the same experiment as performed in Section 4-2-1, although the results shown that section used post-processed odometry. The results of the test with real-time data used the second odometry model (3-17) so that the can be compared to the results in Figure 4-5. The used parameters are shown in Table 4-4.

The results are shown in Figure 4-6. The error shows similar characteristics as the error when using post-processed data. At first, the RBPF has a larger error. Later it outperforms the odometry. However, the RBPF trajectory shows significantly less drift in the corner than the odometry. The RBPF trajectory is slightly rotated in both yaw and pitch, possibly caused by a slightly rotated initial circulation. This is very likely a result of the second odometry model, since a change in the yaw and pitch early on can result in the wrong particle obtaining the highest weight. Since no map is yet build, the RBPF cannot correct for it. This particle then builds the map for this first circulation and is then able to close the loop on this offset circulation. This is also shown in the error, which settles after approximately 100 seconds.This would explain the large error in the start and settling of the error thereafter. The error in pitch shown in the estimation of the RBPF shows a limitation of the second odometry model. The inertial odometry moves along the floor, while the RBPF estimation moves trough the floor twice with each circulation.

The SLAM algorithm is capable of improving both the real-time and smoothed odometry.

Odometry model	σ_{lin}	l	σ_{SE}	$\sigma_{\mathbf{z}}$	L	LO	N_m
(3-17)	1.10	0.5	0.2	0.05	$\begin{bmatrix}5.5 & 4 & 3\end{bmatrix}$	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$	400

Odometry model	N_i	N_{eff}	$\frac{1}{dt}\mathbf{S}_p$	$\frac{1}{\mathrm{d}t}\mathbf{S}_q$	N_k
(3-17)	300	0.5	$0.01\mathbf{I}_3$	$10^{-5}\begin{bmatrix}0.02 & 0 & 0\\ 0 & 0.02 & 0\\ 0 & 0 & 1\end{bmatrix}$	1

Table 4-4: The used parameters for the estimation using real-time odometry.

In both cases, the odometry is outperformed after roughly 250 seconds. The error for the real-time RBPF estimation shows more fluctuations than for the estimation using the post-processed odometry. This corresponds to the odometry signal, where the real-time error in the odometry fluctuates more then the post-processed odometry error.

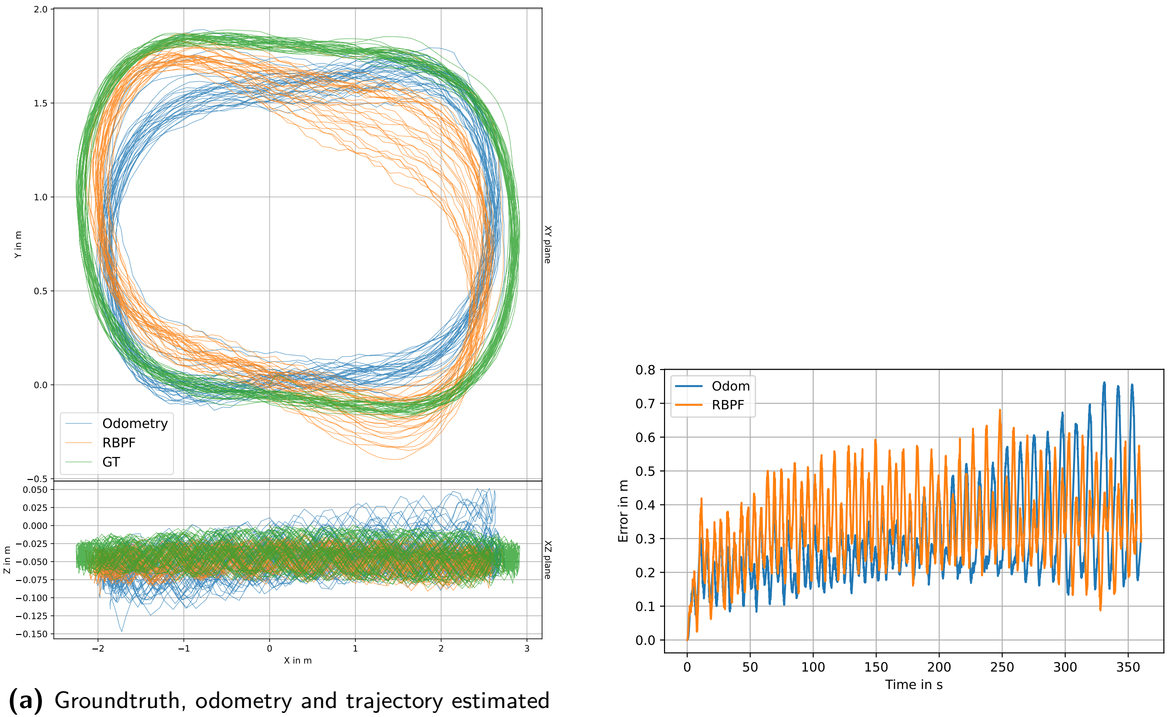
4-2-3 Traversing a set of stairs

To test the performance of the SLAM algorithm in a full three dimensional setting, an odometry that moves trough three dimensions is required. The odometry used in Section 4-2-1 and Section 4-2-2 mostly move trough a two dimensional plane. The trajectories were captured on a single floor. This does not imply that in these situations the three dimensional SLAM approach could be interchanged with a two dimensional one, since all sensors do make movement in three dimensions. Especially the sensors mounted on the arms and feet make motions those three dimensional motions, even when the odometry was captured on a single floor. For these results post-processed odometry data and the first odometry model (3-16) were used. To test the full three dimensional capabilities, this experiments contains a stairs. The scenario starts on the lower floor, the stairs are taken to the second floor. The stairs contain a platform parallel to the floors halfway. On the second floor a small circulation is made before returning down the same stairs. This trajectory is too large for a single domain, therefore the domain transitions occur in this estimation.
For this experiment, no optical groundtruth is available. The performance of the RBPF is

Odometry model	σ_{lin}	l	σ_{SE}	$\sigma_{\mathbf{z}}$	L	LO	N_m
(3-16)	1.10	0.5	0.2	0.075	$\begin{bmatrix}5 & 5 & 5\end{bmatrix}$	$\begin{bmatrix}2 & 2 & 2\end{bmatrix}$	400

N_i	N_{eff}	$\frac{1}{dt}\mathbf{S}_p$	$\frac{1}{\mathrm{d}t}\mathbf{S}_q$	N_k
200	0.5	$10\mathbf{I}_3$	$10^{-4}\begin{bmatrix}0.0001 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 1\end{bmatrix}$	1

Table 4-5: The used parameters for the estimation when traversing the stairs.



(a) Groundtruth, odometry and trajectory estimated by the RBPF

(b) Euclidian distance to the groundtruth

Figure 4-6: Results for the SLAM algorithm with real-time odometry.

evaluated using the start of the stairs. In this experiment, the estimated trajectory is expected to start and return from the stairs end at roughly the same spot.

Figure 4-7 shows that the inertial odometry is able to estimate the correct height after traversing the stairs, but not the x and y coordinates. The inertial odometry places the return point from the stairs approximately one meter away from the starting point. An explanation could be that the post-processing algorithm used correctly assumes that the system returns to same floor, which could be expected to have the same height. Forcing the return point on the same xy-plane could introduce errors in the other dimensions, which results in the large error in a short time span, compared to the error in for example the results in Section 4-2-1. In that experiment the error after 360s appears smaller than the error after traversing the stairs, which only takes approximately 45 seconds.

The trajectory estimated by the RBPF estimates the return point closer to the starting point, as shown in Figure 4-7, although the error is still noticeable. The two passages of the upper half of the stairs is much better than on the lower half. The inability of the RBPF to correct the odometry error could be same as the cause for drift in Section 4-2-1, where where orientational drift caused large drift in the position when using the first odometry model (3-16). For this scenario the yaw drift is very low, as the data set is rather short. However, traversing stairs seems to introduce an temporary error in the pitch of the odometry. On the ascent, the stairs appear steeper than on the descent. The pitch is restored when the system returns to the lower floor. On the small platform halfway the stairs, the pitch is corrected as well. This pitch error explains the distance between the starting and return point a the base of the stairs in the odemetry estimate. The same limitation of the motion model that could not correct for the yaw drift in Section 3-4-2 makes the RBPF unable to correct for the pitch error on

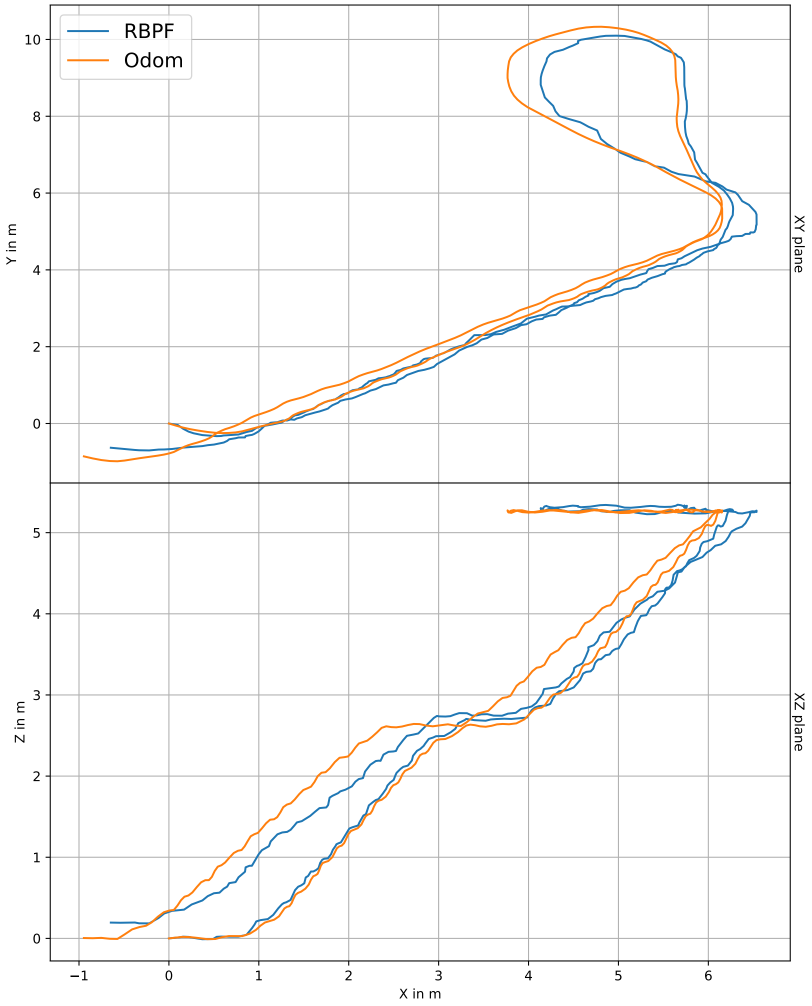


Figure 4-7: Odometry and estimated trajectory by the RBPF with a single IMU in a scenario where a set of stairs is walked up and down once.

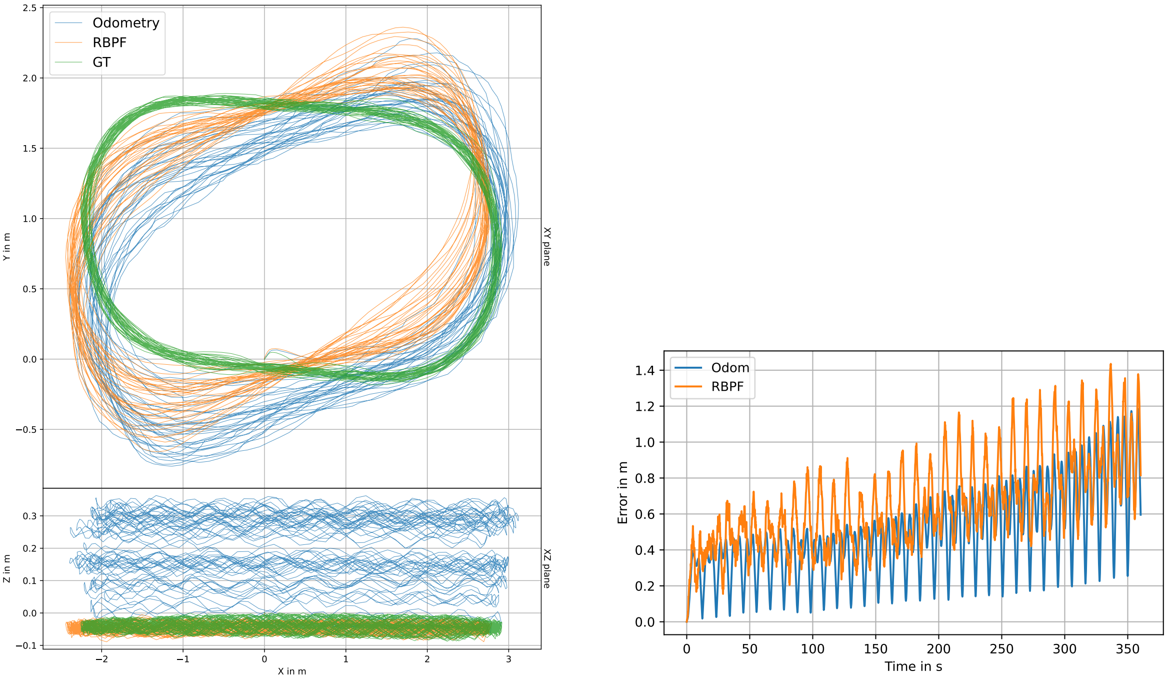
the stairs.
Using the odometry model that estimates the orientational drift (3-17) does not improve the trajectory estimated by the RBPF. This model estimates the drift with a random walk model, while the pitch error in orientation estimate experienced in this test is large at the start of the stairs. While moving along a floor this error is significantly smaller. The random walk model is unable to capture this behaviour, as the difference in error is too large. However, the benefit of scaling the noise with the input signal $\Delta^{\mathrm{n}}\mathbf{p}_t^{k=\alpha}$ does show. The trajectory does not drift in the z direction when the system is on the second floor, but is able to differ from the odometry in the z direction when traversing the stairs.

4-3 SLAM using multiple magnetometers

The HMTS suit contains $N_k = 17$ IMUs. In this section, the results of the SLAM algorithm are shown, when all magnetometers are used to construct the magnetic field map. The experiment in Section 4-2-1 is repeated with multiple magnetometers in Section 4-3-1. The performance of the SLAM algorithm when traversing a stairs and multiple magnetometers is presented in Section 4-3-2

4-3-1 Comparison of the odometry models

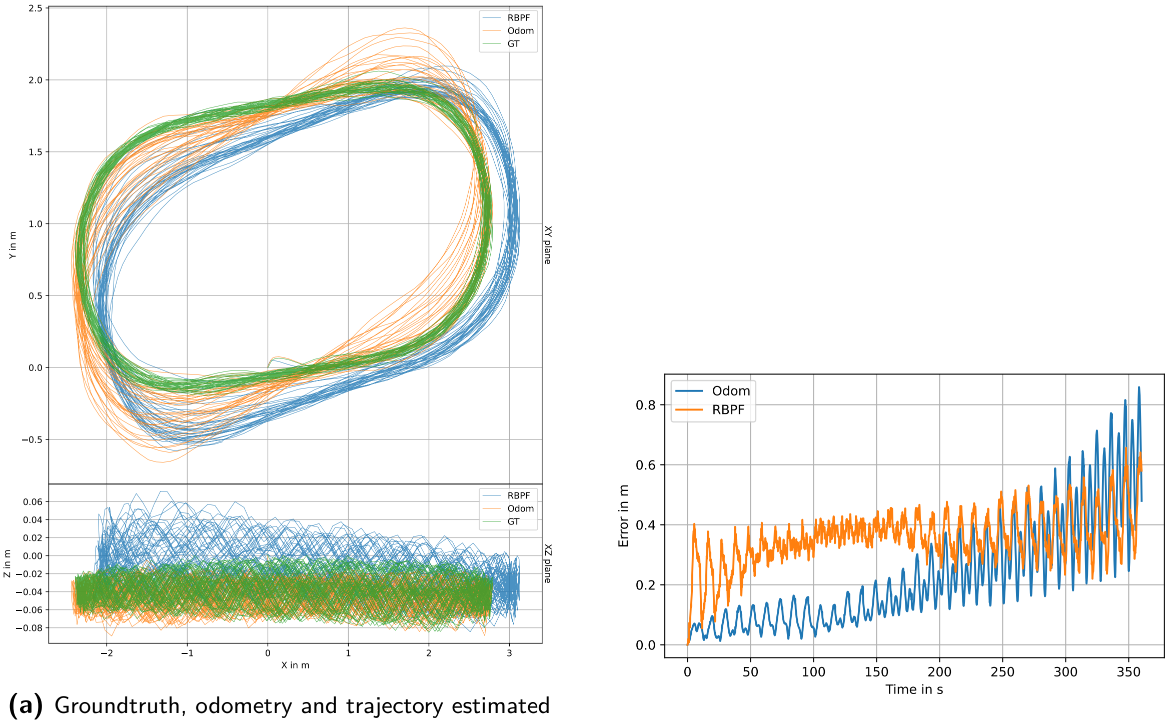
Similarly as was done for one IMU in Section 4-2-1, the two odometry models (3-16) and (3-17) are compared with all $N_k = 17$ IMUs. The estimated trajectory is shown in Figure 4-8. Only the trajectory of one IMU is shown. The RBPF performance worse than both the odometry and the estimation where only one sensor is used. Especially the estimation in the z direction is worse and drifts away from the plane where the sensor is. In contrast to the single IMU case, the map is now constructed with magnetometers located at different heights. This causes the magnetic field map to have more confidence along the third dimension compared to the single sensor case, where the is only confident along a single xy-plane. This could cause the more excessive drift of the RBPF estimation in the z direction.
With the second odometry model (3-17), the RBPF estimation improves, as is shown in Figure 4-9. The RBPF error grows very large in the first circulation, indicating that this estimated circulation is incorrect. The error in this first circulation is much more present than in the single magnetometer case. This error in the first circulation also showed when using real-time odometry and a single magnetometer, as discussed in Section 4-2-2. Similarly to previous results using the second odometry model, the error of the RBPF does not drift over time and performs better than the odometry after 250 seconds. The error is slightly larger than for the single sensor experiment, which is caused by the significantly worse first circulation. The RBPF is never able to restore that initial error. The overall worse performance when using multiple magnetometers, regardless of the odometry model, could be caused by the difficulty the GP has to represent the magnetic field when multiple magnetometers are used, as was observed in Section 4-1-2.



(a) Odometry, RBPF estimate and groundtruth

(b) Euclidean distance to groundtruth.

Figure 4-8: The results when using all $N_k = 17$ IMUs to construct the magnetic field map. The first odometry model $(3\text{-}16)$ was used.



(a) Groundtruth, odometry and trajectory estimated by the RBPF.

(b) Euclidian distance to the groundtruth

Figure 4-9: The results when using all $N_k = 17$ IMUs to construct the magnetic field map. The second odometry model $(3\text{-}16)$ was used.

Odometry model	σ_{lin}	l	σ_{SE}	$\sigma_{\mathbf{z}}$	L	LO	N_m
(3-16)	1.10	0.5	0.2	0.075	$\begin{bmatrix}5 & 5 & 5\end{bmatrix}$	$\begin{bmatrix}2 & 2 & 2\end{bmatrix}$	400

N_i	N_{eff}	$\frac{1}{\text{d}t}\mathbf{S}_p$	$\frac{1}{\text{d}t}\mathbf{S}_q$	N_k
250	0.5	$0.001\mathbf{I}_3$	$10^{-5}\begin{bmatrix}0.0001 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 1\end{bmatrix}$	17

Table 4-6: The used parameters for the estimation when traversing the stairs and using multiple magnetometers to construct the magnetic field map.

4-3-2 Traversing a set of stairs

Similar as was done for a single IMU in Section 4-2-3, the performance of the SLAM algorithm in three dimensional situations is evaluated. The parameters used are shown in Table 4-6. The RBPF does not perform well in this scenario. The return point is too low and further from the starting point of the stairs. Additionally, the upper floor and the platform halway the stairs is slightly tilted compared the lower floor.

The cause for the worse performance compared to the single sensor version, could be due to the domain transitions. As noticed in Section 4-1-4, using multiple magnetometers to construct the magnetic field map creates slightly fields in the overlapping part of the domains. Figure 4-11 shows the trajectory and the domain boundaries. Each red cross represents shows a point on the trajectory where the particles were resampled. If the worse performance is caused by the domain transitions, the RBPF estimate would perform worse in the overlapping part of the domains. The RBPF estimate starts to differ from the odometry on the ascent of the second half of the stairs. At the start of the descent, on the second floor, the RBPF is able to close the loop. However, the estimated descent is not the same as the ascent of the stairs. These parts of the estimation start to differ within a non-overlapping part of the domain, indicating that domain transitions are not responsible for the decreased performance.

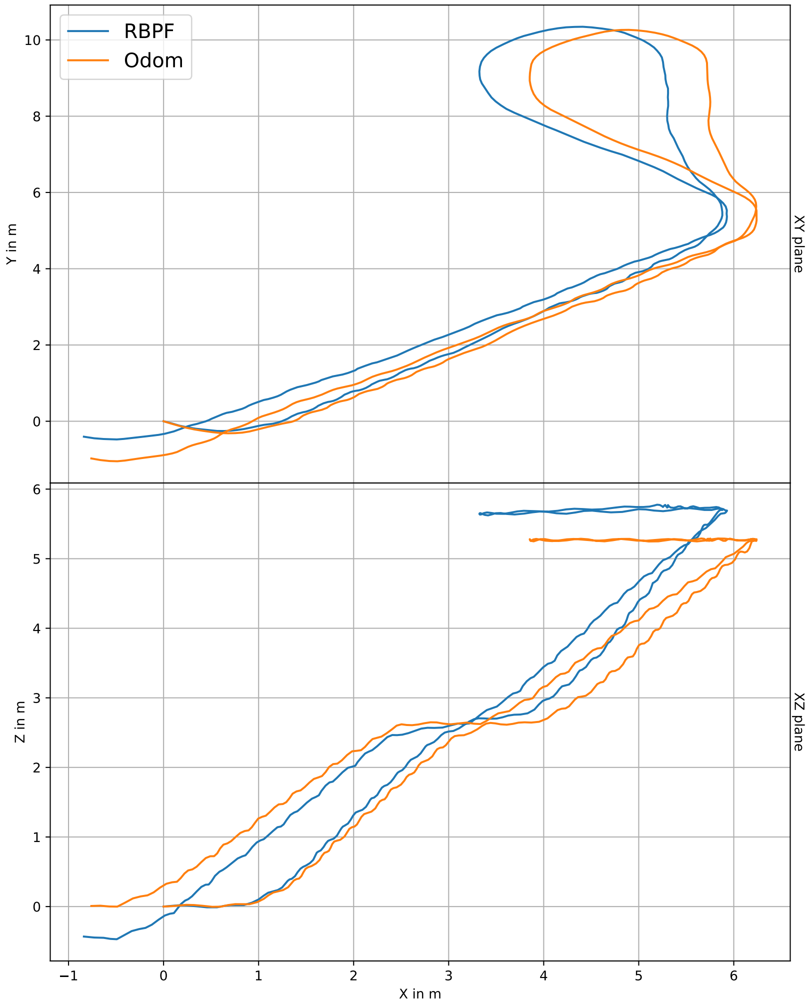


Figure 4-10: Odometry and RBPF estimate when traversing a set of stairs. All $N_k = 17$ IMUs were used.

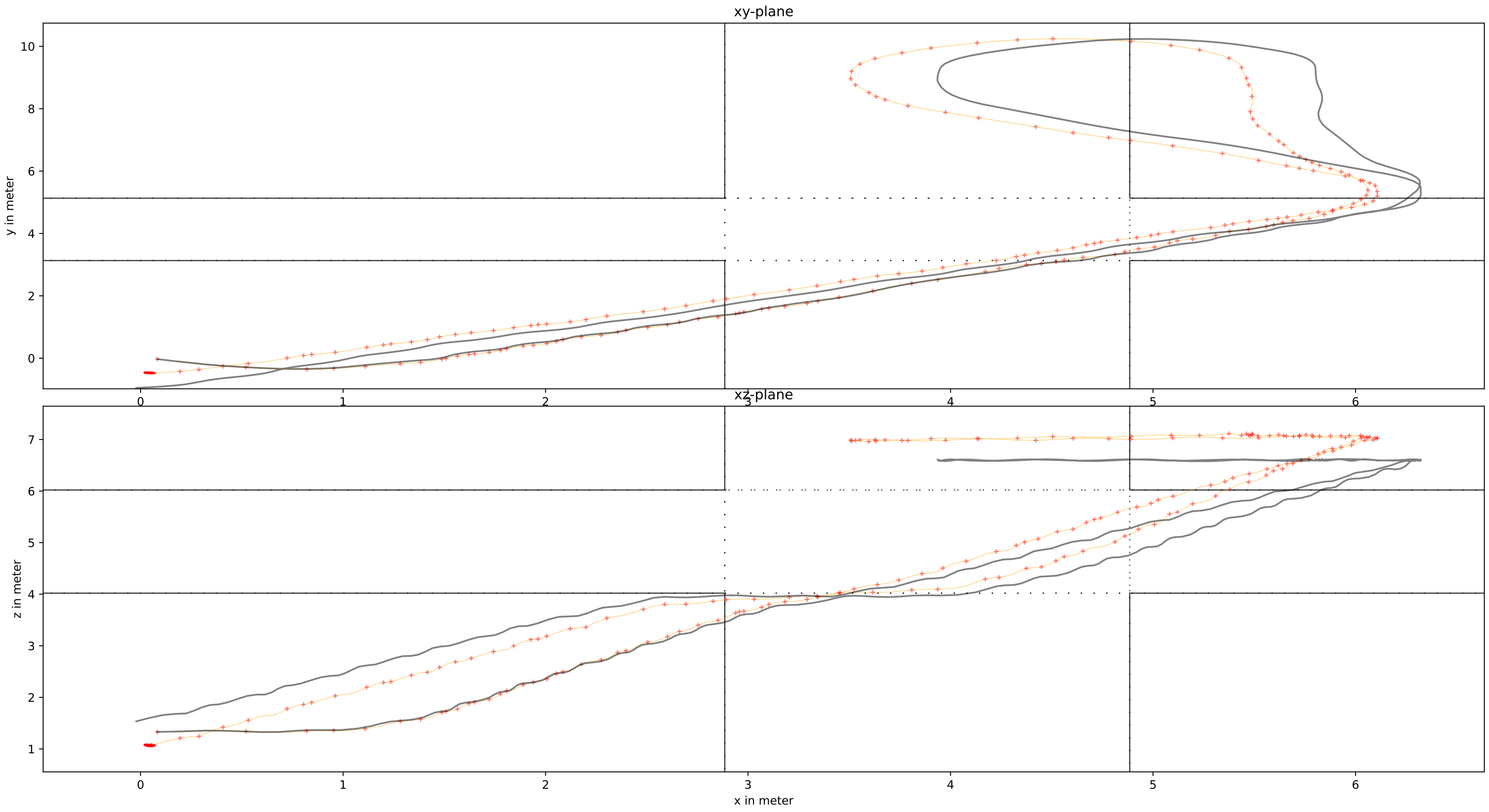


Figure 4-11: The RBPF estimate(orange), odometry (grey) with domain boundaries (black lined rectangles) and particles (red dots). The area between the solid and dotted domain boundaries is the overlapping area. The red crosses represent location where resampling occurred.

Chapter 5

Conclusion

This thesis focuses on using the ambient magnetic field for to tackle SLAM problem. The novelties presented in this thesis are the use of an inertial HMTS for odometry and multiple magnetometers to construct a map of the magnetic field.
In the introduction, the first research question for this thesis was posed:
What are the improvements to the position estimate of an inertial human motion tracking suit when combined with magnetic field simultatenous localistation and mapping?
The results show that using a RBPF and reduced rank GPR can improve the position estimate after roughly 250 seconds. This result was achieved for both the real-time odemetry and the post-processed odometry.
The second research question of this thesis is:
How does using multiple magnetometers to construct a magnetic field map in a magnetic field simultatenous localistation and mapping approach differ compared to a map constructed with a single magnetometer?
The results do not show improved position estimates when multiple magnetometers are used. When traversing the stairs, using multiple magnetometers decreased the location estimate performance compared to using a single magnetometer.

In Chapter 2 related work was discussed. A RBPF and the reduced rank GPR was chosen as best option to improve the inertial position estimate from the HMTS. The reduced rank GPR had already been successfully used in literature for magnetic field SLAM [28]. The RBPF was chosen of other filtering methods since natural fit with the GP.
Chapter 3 shows the derivation of the algorithm. The first novelty is use of multiple sensors in magnetic field SLAM through the use of a HMTS. Two motion models are proposed. Both models are capable of adding noise to the odometry, while using the relative poses of each IMU as estimated by the HMTS. The difference between the models is that the second model introduces an extra state to estimate the orientation drift through a random walk model. This extra state is then used to compensate for orientational drift in the positional odometry. The improved performance of the additional state is shown in Section 4-2-1.
The second novelty is the adaptation of the reduced rank GPR proposed in [61] to model the

magnetic field. The adaptation allows for multiple magnetometers to be used in the construction of the magnetic field map. The third novelty is the adjustment made to splitting the space into multiple domain as done in [28]. The adjustment allows for the use of more than one magnetometer when multiple domains are required by determining which observations are used to update the GP of each domain and which GPs are used to evaluate the weight of the particles.

Both for real-time and post-processed odometry, the RBPF is able to improve the position estimate, but only after around 250 seconds, as shown if Figure 4-5b and Figure 4-6b. In the first 250 seconds, the post-processed odometry is considerably better compared to the RBPF position estimate. This difference in performance between the odometry and the RBPF is less pronounced for the real-time odometry. The spread in error is considerably larger for the real-time data than for the post-processed. For both experiments the error oscillates around 0.4 meter. The real-time RBPF trajectory is rotated, pitch and yaw, with respect to its odometry, while the post-processed RBPF trajectory is not rotated.

With training and validation data, it was shown that the map constructed with a single sensor is considerably better than its multi sensor counterpart. A possible explanation could be that in the multiple sensor case, the magnetometer calibration becomes vital. Different magnetometers at the same location have to observe the same field, otherwise the GP has difficulty discerning magnetic field anomalies with measurement noise. Additionally, the postion and orientation estimate of each magnetometer with respect to each other is important. If these relative position and orientation estimates are not accurate enough, the observations enter the GP in a different coordinate frame, further deteriorating the quality of the magnetic field map.

The difference in the magnetic field map when using multiple sensors is further shown when system crosses a domain boundary. For single sensor constructed maps, the overlapping parts of the domain show similar fields, while for magnetic field map constructed with multiple sensors, the overlapping domains differ slightly.

The results when using a single magnetometer and 17 magnetometers show similar results, although with 17 mangetometers the trajectory estimated by the RBPF is rotated with pitch compared to the odometry. The single sensor version does not have this rotation. In both experiments, the RBPF outperforms the odometry after around 250 seconds.

For larger trajectories, where multiple domains are required, the single sensor version outperforms the multiple sensors variant.

The RBPF performs as good or better with a single sensor compared to using all sensors. Note that the sensors are still used to create the odometry from the HMTS which is used in the RBPF.

This theses has shown that using magnetic field SLAM can improve the position estimate of a inertial HMTS. However, for the best results, not all sensors are used in the construction magnetic field map. In future work, the better use of multiple sensors for the magnetic field mapping could be investigated. More extensive research should be done into the effect of magnetometer calibration on the constructed magnetic field to investigate the reduced performance when using multiple magnetometers. Another interesting research goal would be to quantify the effect of the accuracy of the relative position and orientation estimation of each magnetometer on the quality of the magnetic field map.

Scaling the positional with the positional odometry showed significant improvements. Further research could be done to improve the odometry model. For example, similar to scaling the positional noise with the change in position, the orientational noise could be scaled to the change in orientation. This could improve the performance of the RBPF at the start of a stairs.

In three dimensional situations, the suit seems to be able to detect different floors, and correct the orientation of the system accordingly. The RBPF does not make use of this. An improvement could be to use this detection in either the odometry model or in the weight evaluation, so that particles that move along the floor, are assigned a greater weight.

The estimated trajectory is significantly worse than the odometry in the early stages. Further research could determine if this is a fundamental of the method, or else if it can be improved. If it is fundamental, using the odometry early might improve the overall performance of the estimation.

Appendix A

Rotations

A-1 Coordinate frames

A coordinate frame is a frame in which a position can be expressed. All coordinate frames have an origin and an alignment. All the frames required for the proposed method are described in Section 3-2.

Vectors, like position $\mathbf{p}$ can be expressed in different coordinate frames. The position with respect to the origin of a frame f1 is denotes with the superscript ${}^{\text{f1}}\mathbf{p}$. The same position with expressed in frame f2 is denotes as ${}^{\text{f2}}\mathbf{p}$.

Coordinate frames also have an alignment or orientation. The orientation of a coordinate frame can be expressed with respect to another coordinate frame. The mapping between these two frame orientations is expressed by a rotation matrix.

A-2 Rotation Matrices and Quaternions

$\mathbf{R}$ There exist several representations of orientation, two of which are used in this thesis. The first one is the *rotation matrix*. These matrices are denoted by $\mathbf{R} \in \mathbb{R}^{3\times 3}$ and have the following properties

$$\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}_3, \qquad |\mathbf{R}| = 1,$$

where $|\mathbf{R}|$ is the matrix determinant. The rotation matrix is part of the special orthogonal group $SO(3)$. Rotation matrices rotation of a coordinate frame with respect to a different frame. The rotation from navigation frame n to sensor frame s is denoted with the double superscript ${}^{\text{sn}}\mathbf{R}$. The inverse rotation, from frame s to n, is equal to the transpose

$${}^{\text{ns}}\mathbf{R} = {}^{\text{sn}}\mathbf{R}^T.$$

To transform a vector from one coordinate frame to the new one, the vector can be multiplied with the corresponding rotation matrix on the right

$${}^{\text{n}}\mathbf{p} = {}^{\text{ns}}\mathbf{R}\,{}^{\text{s}}\mathbf{p}.$$

The second parametrisation of orientation is the unit quaternion. Where the rotation matrix requires nine values, the quaternion needs only four. Quaternions are denoted with $\mathbf{q} \in \mathbb{R}^4$. Unit quaternions are always of length one, so that $||\mathbf{q}||_2 = 1$. Quaternions do not describe a rotation uniquely, meaning that one orientation can have multiple quaternion representations. For quaternions $\mathbf{q}$ and $-\mathbf{q}$ describe the same orientation. The rotation between two frames as denoted with the same superscript as for the rotation matrix. The quaternion can be split as

$$\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix}.$$

Transforming a vector to a different frame requires the quaternion $\mathbf{q}^c$, so that

$$^{\mathrm{ns}}\mathbf{q} = (^{\mathrm{sn}}\mathbf{q})^c = \begin{bmatrix} ^{\mathrm{ns}}q_0 & -^{\mathrm{ns}}\mathbf{q}_v{}^T \end{bmatrix}^T.$$

Transforming vector $\bar{\mathbf{p}}$ can then be done as follows

$$^{\mathrm{n}}\bar{\mathbf{p}} = {}^{\mathrm{ns}}\mathbf{q} \odot {}^{\mathrm{s}}\bar{\mathbf{p}} \odot (^{\mathrm{sn}}\mathbf{q})^c \tag{A-1}$$

where $^{\mathrm{n}}\bar{\mathbf{p}}$ is the quaternion representation of $^{\mathrm{n}}\mathbf{p}$ and $\odot$ is the quaternion multiplication operator, defined as

$$\mathbf{q} \odot \mathbf{r} = \begin{bmatrix} q_0 r_0 - \mathbf{q}_v \cdot \mathbf{r}_0 \\ q_0 \mathbf{r}_v + r_0 \mathbf{q}_v + \mathbf{q}_v \times \mathbf{r}_v \end{bmatrix}.$$

Note that the quaternion resulting from (A-1) is not guaranteed to be of unit length. Converting quaternion $\mathbf{q}$ to rotation matrix $\mathbf{R}$ is done as

$$\mathbf{R} = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1 q_2 - 2q_0 q_3 & 2q_1 q_3 + 2q_0 q_2 \\ 2q_1 q_2 + 2q_0 q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2 q_3 - 2q_0 q_1 \\ 2q_1 q_3 - 2q_0 q_2 & 2q_2 q_3 + 2q_0 q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}.$$

Appendix B

Scalar potential Gaussian process prior

GPs can be seen as a probabilistic approach to learning in kernel machines [56]. In machine learning, GPR (Gaussian process regression) is used to learn the input-output mapping from measurements. In the case of modelling the three dimensional magnetic field, the input could be the position and orientation, and the output would be an estimate of the magnetic field at that location.

To embed the most information from the observation, it is necessary to consider not only the magnitude of the magnetic field observation, but also its direction. Therefore it is required to know the orientation of the magnetometer when it makes an observation.

Suppose the magnetic field can be described by a real, unknown process $f(\mathbf{x})$, where $\mathbf{x}$ consists of a position ${}^{\mathrm{n}}\mathbf{p}$ and an orientation ${}^{\mathrm{ns}}\mathbf{q}$. The GP that describes $f(\mathbf{x})$, is defined by a mean function $m(\mathbf{x})$ and covariance function or kernel $\kappa(\mathbf{x}, \mathbf{x'})$

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}\left[f(\mathbf{x})\right], && \text{(B-1)} \\ \kappa(\mathbf{x}, \mathbf{x'}) &= \mathbb{E}\left[\left(f(\mathbf{x}) - m(\mathbf{x})\right)\left(f(\mathbf{x'}) - m(\mathbf{x'})\right])\right]. && \text{(B-2)} \end{aligned}$$

The output of the GP is multidimensional, since the magnetic field consists of three components. One way to obtain this multidimensionality is to consider a GP for each direction, x, y and z individually. The GP prior can then be written as

$$\begin{aligned} \begin{bmatrix} f_x(\mathbf{x}) \\ f_y(\mathbf{x}) \\ f_z(\mathbf{x}) \end{bmatrix} &= f_h(\mathbf{x}) \sim \mathcal{GP}(m_k(\mathbf{x}), \kappa_k(\mathbf{x}, \mathbf{x'})), \\ y_k &= f_h(\mathbf{x}) + \epsilon_h, \qquad \epsilon_h \sim \mathcal{N}\left(0, \sigma^2_{h,\text{ noise}}\right), && \text{(B-3)} \end{aligned}$$

where y_k are noisy measurement for $h = \{x, y, z\}$. The drawback of modelling the three components of the magnetic field vector as separate GPs, is that these GPs are then independent of each other. A real magnetic field follows Maxwell's equations [83].

The indoor magnetic field $\mathbf{H}(\mathbf{x})$ that is to be modelled is assumed to be curl-free, so that

$$\nabla \times \mathbf{H} = \mathbf{0}_3. \tag{B-4}$$

This assumption holds if there is no free current within area where the magnetic field is to be modelled. This curl-free property can be embedded into the kernel, so that the GP will also be curl-free. This curl-free kernel is presented in [84]

$$\kappa_{\text{curl-free}}\left(\mathbf{x},\mathbf{x'}\right)=\kappa_{\text{SE}}\left(\mathbf{x},\mathbf{x'}\right)\left(\mathbf{I}_3-\left(\frac{\mathbf{x}-\mathbf{x'}}{l_{\text{curl-free}}}\right)\left(\frac{\mathbf{x}-\mathbf{x'}}{l_{\text{curl-free}}}\right)^T\right). \tag{B-5}$$

$\kappa_{\text{SE}}\left(\mathbf{x},\mathbf{x'}\right)$ is the SE kernel

$$\kappa_{\text{SE}}\left(\mathbf{x},\mathbf{x'}\right)=\sigma_{\text{SE}}^2\exp\left(-\tfrac{|\mathbf{x}-\mathbf{x'}|^2}{2l_{\text{SE}}^2}\right). \tag{B-6}$$

This kernel guarantees smoothness of the output, which in turn ensures that the modelled magnetic field is continuous. σ_{SE} is the magnitude scale hyperparameter and for both (B-5) and (3-21) l_{SE}^2 are the lengthscale parameters. These hyperparameters can be learned from data, see Appendix C for details. A curl free prior in the mangnetic field $\mathbf{H}\left(\mathbf{x}\right)$ can be written as

$$\mathbf{H}\left(\mathbf{x}\right)\sim\mathcal{GP}\left(\mathbf{0}_3,\sigma_{\text{const}}^2\mathbf{I}_3+\kappa_{\text{curl-free}}\left(\mathbf{x},\mathbf{x'}\right)\right), \tag{B-7}$$

with magnitude scale hyperparameter σ_{const}.
It is possible to model a curl-free GP using the scalar potential $\varphi\left(\mathbf{x}\right)$ This approach assumes $\varphi\left(\mathbf{x}\right)$ to be a GP prior, and the observations of the magnetic field to be its gradient subject to Gaussian noise. This approach is used in the reduced rank GPR detailed in Section 3-5-1. The $\mathbf{H}$-field can be written as the gradient of the scalar potential field

$$\mathbf{H}\left(\mathbf{x}\right)=-\nabla\varphi\left(\mathbf{x}\right). \tag{B-8}$$

The GP prior then becomes

$$\begin{aligned}\varphi\left(\mathbf{x}\right)&\sim\mathcal{GP}\left(0,\kappa_{\text{lin}}\left(\mathbf{x},\mathbf{x'}\right)+\kappa_{\text{SE}}\left(\mathbf{x},\mathbf{x'}\right)\right),\\ \mathbf{z}_t&=-\nabla\varphi\left(\mathbf{x}\right)|_{\mathbf{x}=\mathbf{x}_t}+\epsilon_t,\quad\epsilon_t\sim\mathcal{N}\left(0,\sigma_{\mathbf{z}}^2\mathbf{I}_3\right),\end{aligned} \tag{B-9}$$

where $\sigma_{\mathbf{z}}$ is the variance on the observations $\mathbf{z}$, and

$$\kappa_{\text{lin}}\left(\mathbf{x},\mathbf{x'}\right)=\sigma_{\text{lin}}^2\mathbf{x}^T\mathbf{x}. \tag{B-10}$$

is the linear kernel, with σ_{lin} is the linear magnitude scale hyperparameter. The equivalence of (B-7) and (3-19) is shown in [85].

Appendix C

Hyperparameter optimisation for reduced rank Gaussian processes

The GP the hyperparameters need to be chosen or determined. Hyperparameters are often determined by minimising the log marginal likelihood. For the reduced rank GP, the expression for this likelihood can be written as [64]

$$\mathcal{L}\left(\boldsymbol{\theta}\right) = \frac{1}{2}\log\left|\kappa_{\text{approx}}\left(\mathbf{p},\mathbf{p'}\right)_{\boldsymbol{\theta}} + \sigma_{\mathbf{z}}^{2}\mathbf{I}_{N_z}\right| + \\ \frac{1}{2}\text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right)^{T}\Big(\kappa_{\text{approx}}\left(\mathbf{p},\mathbf{p'}\right)_{\boldsymbol{\theta}} + \sigma_{\mathbf{z}}^{2}\mathbf{I}_{N_z}\Big)^{-1}\text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right) + \frac{3N_z}{2}\log 2\pi,$$

where $\boldsymbol{\theta}$ is a vector containing all hyperparameters and $\text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right)$ is a vector containing all magnetic field observations up to time t. $\kappa_{\text{approx}}\left(\mathbf{p},\mathbf{p'}\right)_{\boldsymbol{\theta}}$ is the kernel from (3-27) with hyperparameters $\boldsymbol{\theta}$. The log likelihood functions can be approximated by the matrices defined in (3-27)

$$\log\left|\kappa_{\text{approx}}\left(\mathbf{p},\mathbf{p'}\right)_{\boldsymbol{\theta}} + \sigma_{\mathbf{z}}^{2}\mathbf{I}_{N_z}\right| \approx \left(3N_z - N_m\right)\log\left(\sigma_{\mathbf{z}}\right) + \sum_{j=1}^{N_m}\boldsymbol{\Lambda}_{\boldsymbol{\theta}_{j,j}} + \\ \log|\sigma_{\mathbf{z}}^{2}|\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1} + \mathbf{C}_{1:t}{}^{T}\mathbf{C}_{1:t}, \tag{C-1}$$

$$\text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right)^{T}\Big(\kappa_{\text{approx}}\left(\mathbf{p},\mathbf{p'}\right)_{\boldsymbol{\theta}} + \sigma_{\mathbf{z}}^{2}\mathbf{I}_{N_z}\Big)^{-1}\text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right) \approx \frac{1}{\sigma_{\mathbf{z}}^{2}}\Big(\text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right)^{T}\text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right) - \\ \text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right)^{T}\mathbf{C}_{1:t}\Big(\sigma_{\mathbf{z}}^{2}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1} + \mathbf{C}_{1:t}{}^{T}\mathbf{C}_{1:t}\Big)^{-1}\mathbf{C}_{1:t}{}^{T}\text{vec}\left({}^{\text{s}}\mathbf{z}_{1:t}\right)\Big), \tag{C-2}$$

where $\mathbf{C}_{1:t} = \mathbf{C}\left(\boldsymbol{\rho}_{1:t}, \boldsymbol{\psi}_{1:t}\right)$. $\boldsymbol{\rho}_{1:t}, \boldsymbol{\psi}_{1:t}$ are all the positions and orientation up to time t respectively.

Bibliography

[1] Daniel Roetenberg, Henk Luinge, and Per Slycke. “Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors”. In: *Xsens Motion Technologies BV, Tech. Rep* 1 (2009).

[2] *MVN User Manual.* Xsens. Enschede.

[3] Jan Stenum et al. “Applications of Pose Estimation in Human Health and Performance across the Lifespan”. In: *Sensors* 21.21 (2021). ISSN: 1424-8220.

[4] Martin Buchheit et al. “Integrating different tracking systems in football: multiple camera semi-automatic system, local position measurement and GPS technologies”. In: *Journal of Sports Sciences* 32.20 (2014), pp. 1844–1857.

[5] Laxman Kumarapu and Prerana Mukherjee. “AnimePose: Multi-person 3D pose estimation and animation”. In: *Pattern Recognition Letters* 147 (2021), pp. 16–24. ISSN: 0167-8655.

[6] Manon Kok et al. “Using Inertial Sensors for Position and Orientation Estimation”. In: *Foundations and Trends R in Signal Processing* 11 (2 2017), pp. 1–153.

[7] Alessandro Filippeschi et al. “Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion”. In: *Sensors* 17.6 (2017), p. 1257.

[8] Benjamin R Hindle, Justin WL Keogh, and Anna V Lorimer. “Inertial-based human motion capture: A technical summary of current processing methodologies for spatiotemporal and kinematic measures”. In: *Applied Bionics and Biomechanics* 2021 (2021).

[9] Daniel Roetenberg. “Inertial and magnetic sensing of human motion”. PhD thesis. Universiteit Twente, Enschede, 2006.

[10] Z Lin et al. “Development of an ultra-miniaturized inertial measurement unit WB-3 for human body motion tracking”. In: *2010 IEEE/SICE International Symposium on System Integration.* IEEE. 2010, pp. 414–419.

[11] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. “An optimization-based approach to human body motion capture using inertial sensors”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 19 (2014), pp. 79–85.

[12] Verne T Inman, Howard D Eberhart, et al. “The major determinants in normal and pathological gait”. In: *JBJS* 35.3 (1953), pp. 543–558.

[13] J. R.W. Morris. “Accelerometry-A technique for the measurement of human body movements”. In: *Journal of Biomechanics* 6 (6 Nov. 1973), pp. 729–736.

[14] Eric R Bachmann. *Inertial and magnetic tracking of limb segment orientation for inserting humans into synthetic environments.* Tech. rep. Naval Postgraduate School Monterey CA, 2000.

[15] Sen Qiu et al. “MEMS Inertial Sensors Based Gait Analysis for Rehabilitation Assessment via Multi-Sensor Fusion”. In: *Micromachines 2018, Vol. 9, Page 442* 9 (9 Sept. 2018), p. 442.

[16] Michael Angermann and Patrick Robertson. “FootSLAM: Pedestrian simultaneous localization and mapping without exteroceptive sensorshitchhiking on human perception and cognition”. In: *Proceedings of the IEEE* 100 (SPL CONTENT May 2012), pp. 1840–1848.

[17] Patrick Robertson, Michael Angermann, and Bernhard Krach. “Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors”. In: *ACM International Conference Proceeding Series* (2009), pp. 93–96.

[18] W Todd Faulkner et al. “GPS-denied pedestrian tracking in indoor environments using an IMU and magnetic compass”. In: *Proceedings of the 2010 international technical meeting of the institute of navigation.* 2010, pp. 198–204.

[19] Chengliang Huang, Zaiyi Liao, and Lian Zhao. “Synergism of INS and PDR in self-contained pedestrian tracking with a miniature sensor module”. In: *IEEE Sensors Journal* 10 (8 2010), pp. 1349–1359.

[20] K. Abdulrahim et al. “On magnetometer heading updates for inertial pedestrian navigation system”. In: *Gyroscopy and Navigation* 5 (3 2014), pp. 145–152.

[21] Soo Suh Young and Park Sangkyung. “Pedestrian inertial navigation with gait phase detection assisted zero velocity updating”. In: *ICARA 2009 - Proceedings of the 4th International Conference on Autonomous Robots and Agents* (2009), pp. 336–341.

[22] Uriel Martinez-Hernandez, Imran Mahmood, and Abbas A. Dehghani-Sanij. “Simultaneous Bayesian Recognition of Locomotion and Gait Phases with Wearable Sensors”. In: *IEEE Sensors Journal* 18 (3 Feb. 2018), pp. 1282–1290.

[23] Matej Supej. “3D measurements of alpine skiing with an inertial sensor motion capture suit and GNSS RTK system”. In: *Journal of Sports Sciences* 28.7 (2010). PMID: 20473823, pp. 759–769.

[24] Jun-Tian Zhang et al. “Concurrent validation of Xsens MVN measurement of lower limb joint angular kinematics”. In: *Physiological Measurement* 34 (8 July 2013), N63.

[25] Jeff Bird and Dale Arden. “Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors”. In: *IEEE Wireless Communications* 18 (2 Apr. 2011), pp. 28–35.

[26] Gerald Glanzer and Ulrich Walder. “Self-contained indoor pedestrian navigation by means of human motion analysis and magnetic field mapping”. In: *2010 7th Workshop on Positioning, Navigation and Communication.* IEEE. 2010, pp. 303–307.

[27] Ilari Vallivaara et al. “Simultaneous localization and mapping using ambient magnetic field”. In: *2010 IEEE Conference on Multisensor Fusion and Integration*. IEEE. 2010, pp. 14–19.

[28] Manon Kok and Arno Solin. “Scalable Magnetic Field SLAM in 3D Using Gaussian Process Maps”. In: *2018 21st International Conference on Information Fusion (FUSION)*. 2018, pp. 1353–1360.

[29] Chao Gao and Robert Harle. “MSGD: Scalable back-end for indoor magnetic field-based GraphSLAM”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (July 2017), pp. 3855–3862.

[30] Ilari Vallivaara et al. “Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task”. In: *2011 15th international conference on advanced robotics (ICAR)*. IEEE. 2011, pp. 198–203.

[31] Dieter Fox et al. “Monte Carlo Localization: Efficient Position Estimation for Mobile Robots”. In: (1999).

[32] S. Suksakulchai et al. “Mobile robot localization using an electronic compass for corridor environment”. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* 5 (2000), pp. 3354–3359.

[33] Xavier Robert-Lachaine et al. “Effect of local magnetic field disturbances on inertial measurement units accuracy”. In: *Applied Ergonomics* 63 (2017), pp. 123–132. ISSN: 0003-6870. URL: https://www.sciencedirect.com/science/article/pii/S0003687017300972.

[34] Martin Frassl et al. “Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion”. In: *IEEE International Conference on Intelligent Robots and Systems* (2013), pp. 913–920.

[35] Binghao Li et al. “How feasible is the use of magnetic field alone for indoor positioning?” In: *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings* (2012).

[36] Jaewoo Chung et al. “Indoor Location Sensing Using Geo-Magnetism”. In: *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*. MobiSys ’11. Association for Computing Machinery, 2011, pp. 141–154.

[37] Yuanchao Shu et al. “Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing”. In: *IEEE Journal on Selected Areas in Communications* 33 (7 July 2015), pp. 1443–1457.

[38] Ilari Vallivaara. “Simlutaneous Localization and Mspping Using The Indoor Magnetic Field”. University of Oulu, Faculty of Information Technology, Electrical Engineering, Computer Science, and Engineering, 2018.

[39] William Storms, Jeremiah Shockley, and John Raquet. “Magnetic field navigation in an indoor environment”. In: *2010 Ubiquitous Positioning Indoor Navigation and Location Based Service*. IEEE. 2010, pp. 1–10.

[40] Etienne Le Grand and Sebastian Thrun. “3-Axis magnetic field mapping and fusion for indoor localization”. In: *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. Sept. 2012, pp. 358–364.

[41] Manon Kok et al. “MEMS-based inertial navigation based on a magnetic field map”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing.* May 2013, pp. 6466–6470.

[42] Kalyan Pathapati Subbu, Brandon Gozick, and Ram Dantu. “LocateMe: Magnetic-Fields-Based Indoor Localization Using Smartphones”. In: *ACM Trans. Intell. Syst. Technol.* 4.4 (Oct. 2013).

[43] Seong Eun Kim et al. “Indoor positioning system using geomagnetic anomalies for smartphones”. In: *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings* (2012).

[44] Haiyang Zhang and Fred Martin. “Robotic mapping assisted by local magnetic field anomalies”. In: *2011 IEEE Conference on Technologies for Practical Robot Applications, TePRA 2011* (2011), pp. 25–30.

[45] Janne Haverinen and Anssi Kemppainen. “Global indoor self-localization based on the ambient magnetic field”. In: *Robotics and Autonomous Systems* 57 (10 Oct. 2009), pp. 1028–1035.

[46] Charles Vercauteren, Maarten Weyn, and Dries Vandermeulen. “Indoor localization Using a Magnetic Flux Density Map of a Building Indoor localization Using a Magnetic Flux Density Map of a Building Feasibility study of geomagnetic indoor localization”. In: *The Third International Conference on Ambient Computing,Applications, Services and Technologies* (2013), pp. 42–49.

[47] Abhinav Saxena and MacIej Zawodniok. “Indoor positioning system using geo-magnetic field”. In: *Conference Record - IEEE Instrumentation and Measurement Technology Conference* (2014), pp. 572–577.

[48] Kalyan Pathapati Subbu, Brandon Gozick, and Ram Dantu. “Indoor localization through dynamic time warping”. In: *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* (2011), pp. 1639–1644.

[49] Brandon Gozick et al. “Magnetic maps for indoor navigation”. In: *IEEE Transactions on Instrumentation and Measurement* 60 (12 Dec. 2011), pp. 3883–3891.

[50] Carlos E Galván-Tejada, Juan Pablo García-Vázquez, and Ramon F Brena. “Magnetic Field Feature Extraction and Selection for Indoor Location Estimation”. In: *Sensors* 14 (2014), pp. 11001–11015.

[51] Daniel Carrillo et al. “MagicFinger: 3D Magnetic Fingerprints for Indoor Location”. In: *Sensors 2015* 15 (7 July 2015), pp. 17168–17194.

[52] Danilo Navarro and Gines Benet. “Magnetic map building for mobile robot localization purpose”. In: *ETFA 2009 - 2009 IEEE Conference on Emerging Technologies and Factory Automation* (2009).

[53] Seung-Mok Lee, Jongdae Jung, and Hyun Myung. “Geomagnetic field-based localization with bicubic interpolation for mobile robots”. In: *International Journal of Control, Automation and Systems 2015 13:4* 13 (4 May 2015), pp. 967–977.

[54] Ilari Vallivaara et al. “Quadtree-based ancestry tree maps for 2D scattered data SLAM”. In: *Advanced Robotics* 32.5 (2018), pp. 215–230.

[55] A. O’Hagan. “Curve Fitting and Optimal Design for Prediction”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 40.1 (1978), pp. 1–24.

[56] C E Rasmussen and C K I Williams. *Gaussian Processes for Machine Learning.* 2006.

[57] Naoki Akai and Koichi Ozaki. “Gaussian processes for magnetic map-based localization in large-scale indoor environments”. In: *IEEE International Conference on Intelligent Robots and Systems* 2015-December (Dec. 2015), pp. 4459–4464.

[58] Yuqi Li et al. “Using Wi-Fi/magnetometers for indoor location and personal navigation”. In: *2015 International Conference on Indoor Positioning and Indoor Navigation* (Dec. 2015).

[59] Taylor N. Lee and Aaron J. Canciani. “MagSLAM: Aerial simultaneous localization and mapping using Earth’s magnetic anomaly field”. In: *Navigation* 67 (1 Mar. 2020), pp. 95–107.

[60] Niklas Wahlström et al. “Modeling magnetic fields using Gaussian processes”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing.* May 2013, pp. 3522–3526.

[61] Arno Solin et al. “Modeling and Interpolation of the Ambient Magnetic Field by Gaussian Processes”. In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1112–1127.

[62] Dirk Hähnel, Brian Ferris, and Dieter Fox. “Gaussian processes for signal strength-based location estimation”. In: *Proceeding of robotics: science and systems.* Citeseer. 2006.

[63] Duy Nguyen-Tuong, Jan Peters, and Matthias Seeger. “Local Gaussian Process Regression for Real Time Online Model Learning and Control”. In: *Proceedings of the 21st International Conference on Neural Information Processing Systems.* NIPS’08. Curran Associates Inc., 2008, pp. 1193–1200.

[64] Arno Solin and Simo Särkkä. “Hilbert space methods for reduced-rank Gaussian process regression”. In: *Statistics and Computing* 30.2 (2020), pp. 419–446.

[65] Evangelos Georgiou and Jian Dai. “Self-Localization of an Autonomous Maneuverable Nonholonomic Mobile Robot using a Hybrid Double-Compass Configuration”. In: *7th International Symposium on Mechatronics and its Applications* (2010).

[66] Han-Sol Kim, Woojin Seo, and Kwang-Ryul Baek. “Indoor Positioning System Using Magnetic Field Map Navigation and an Encoder System”. In: *Sensors 2017, Vol. 17, Page 651* 17 (3 Mar. 2017), p. 651.

[67] Arno Solin et al. “Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning”. In: *2016 European Navigation Conference (ENC).* May 2016, pp. 1–9.

[68] Piotr Mirowski et al. “SignalSLAM: Simultaneous localization and mapping with mixed WiFi, Bluetooth, LTE and magnetic signals”. In: *International Conference on Indoor Positioning and Indoor Navigation.* Oct. 2013, pp. 1–10.

[69] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. “Improved techniques for grid mapping with Rao-Blackwellized particle filters”. In: *IEEE Transactions on Robotics* 23 (1 Feb. 2007), pp. 34–46.

[70] Austin Eliazar and Ronald Parr. “DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks”. In: *International Joint Conferences on Artificial Intelligence* 3 (2003), pp. 1135–1142.

[71] Patrick Robertson et al. "Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments". In: *2013 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013* (2013).

[72] Ilari Vallivaara et al. "Monty Hall Particle filter: A new method to tackle predictive model uncertainties". In: *2013 16th International Conference on Advanced Robotics, ICAR 2013* (2013).

[73] Dirk Hähnel et al. "An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements". In: *IEEE International Conference on Intelligent Robots and Systems* 1 (2003), pp. 206–211.

[74] Jongdae Jung, Taekjun Oh, and Hyun Myung. "Magnetic field constraints and sequence-based matching for indoor pose graph SLAM". In: *Robotics and Autonomous Systems* 70 (Aug. 2015), pp. 92–105.

[75] Chao Gao and Robert Harle. "Sequence-based magnetic loop closures for automated signal surveying". In: *2015 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2015* (Dec. 2015).

[76] Stephen Barkby et al. "A featureless approach to efficient bathymetric SLAM using distributed particle mapping". In: *Journal of Field Robotics* 28 (1 Jan. 2011), pp. 19–39.

[77] Arnaud Doucet et al. "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks". In: *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence* (Jan. 2000), pp. 499–515.

[78] Michael Montemerlo et al. "FastSLAM: A factored solution to the simultaneous localization and mapping problem". In: *Aaai/iaai* 593598 (2002).

[79] M Sanjeev Arulampalam et al. "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking; A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking". In: *IEEE Transactions on Signal Processing* 50 (2 2002), pp. 174–188.

[80] Jun S. Liu. "Metropolized independent sampling with comparisons to rejection sampling and importance sampling". In: *Statistics and Computing 1996 6:2* 6 (2 1996), pp. 113–119.

[81] *OptiTrack optical tracking system.* Corballis, OR: OptiTrack.

[82] *Magnetic Calibration Manual.* Xsens. Enschede.

[83] James Clerk Maxwell. "VIII. A dynamical theory of the electromagnetic field". In: *Philosophical transactions of the Royal Society of London* 155 (1865), pp. 459–512.

[84] Edward J Fuselier Jr. "Refined error estimates for matrix-valued radial basis functions". PhD thesis. Texas A&M University, 2007.

[85] Niklas Wahlström. "Modeling of Magnetic Fields and Extended Objects for Localization Applications". In the electronic version figure 2.2a is corrected. PhD thesis. Linköping University, Faculty of Science & Engineering, 2015, p. 236.

Glossary

Abbreviations

DTW	dynamic time warping
GP	Gaussian process
GPR	Gaussian process regression
GPS	Global posioning system
HMTS	Human motion tracking suit
IMU	Inertial measurement unit
KF	Kalman filter
OTS	Optical tracking system
RBPF	A particle filter where, in case of SLAM, the conditional on the map and pose are seperated
SE	Squared exponential covariance function or kernel
SLAM	simultaneous localisation and mapping

Miscellaneous

$\nabla\times$	Curl operator
$\lvert\cdot\rvert$	Matrix determinant of $\cdot$
$\exp()$	Exponential function
∇	Gradient operator
$\odot$	Quaternion product
$\mathbf{0}_n$	Vector of zeros of length n

$\mathbb{E}[\cdot]$	Expectation of random variable $\cdot$
h	Index variable that indicates the dimension of a vector in $\mathbb{R}^3$
$\mathbf{I}_n$	Identy matrix of size $n \times n$
$\mathcal{N}\left(0, \sigma^2\right)$	Gaussian distribution with mean 0 and standard deviation σ^2
${}^{\mathrm{ns}}\mathbf{R}_t$	Rotations matrix representation of the orientation at t. Upper-left superscript denotes the orientation of frame s with rescpect to frame n
N_{t-1}	Total time steps
$\mathcal{O}\left(\cdot\right)$	Computational scaling by $\cdot$
$\exp_q\left(\right)$	Exponential function
$\log_q\left(\right)$	Logarithm function
$\mathbb{R}$	Real numbers
t	Time index
$\mathcal{U}$	Uniform distribution
$\mathbb{V}[\cdot]$	Variation of random variable $\cdot$
ϵ	Noise

Particle filter

b	Body coordinate frame. The coordinate frames of the 23 bodies as defined by the MVN Link [2]
dt	Single timestep
i	Particle index variable
k	Sensor index
n	Navigation coordiate frame. Assumed static of time
N_{eff}	Effective sample size
N_i	Number of particles
$N^i_{d,t}$	Number of domains Ω^d constructed by particle i at time t
${}^{\mathrm{n}}\bar{\mathbf{p}}^{k=\alpha}_{0:N_{t-1}}$	Position trajectory outputted by the RBPF
${}^{n}\mathbf{p}^k_t$	Position of each sensor expressed in the navigation frame relative to sensor $k = \alpha$
N_{resample}	Resampling threshold
N_k	Number of sensors
${}^{\mathrm{ns}}\bar{\mathbf{q}}^{k=\alpha}_{0:N_{t-1}}$	Orientation trajectory outputted by the RBPF

${}^{\mathrm{ns}}\mathbf{q}_t$	Unit quaternion representation of the orientation at t. Upper-left superscript denotes the orientation of frame s with rescpect to frame n
${}^{\mathrm{ns}}\hat{\mathbf{q}}_t$	Unit quaternion representation of the orientation at t. Upper-left superscript denotes the orientation of frame s with rescpect to frame n. The hat indicates that this value comes directly from the HMTS
N_z	Number of measurements
$\mathbf{p}$	Position vector
$\hat{\mathbf{p}}$	Position vector. The hat indicates that this value comes directly from the HMTS
$p\left(A\vert B\right)$	Conditional distrubtion, where random variable A is conditioned on B
${}^{\mathrm{ns}}_{\epsilon}\mathbf{q}_t$	Orientational drift at time t, accumulated upto time t. Represented by a unit quaternion.
s	Sensor coordinate frame. The coordinate frames of the 17 sensors as defined by the MVN Link [2]
$\mathbf{S}_p$	Covariance matrix of the position update step in the odometry model
$\mathbf{S}_q$	Covariance matrix of the orientation update step in the odometry model
$\mathbf{u}_t$	Input vector at time t
$\mathbf{x}^i$	The state containing a position and orientation for particle i
$\mathbf{z}$	Observation, measurement vector
$\mathrm{\alpha}$	Sensor frame of sensor $k = \alpha$
$\Delta\mathbf{p}$	Change in position, used in as control input $\mathbf{u}$
$\Delta\mathbf{q}$	Change in orientation, used in as control input $\mathbf{u}$
α	Sensor index of the sensor for which the RBPF is estimating the position and orientation
$\pi\left(\cdot\right)$	Proposal distribution used in a particle filter
$\boldsymbol{\omega}_t$	Vector containing the weight of all particles at time t
ω_t^i	Weight of particle i at time t

Magnetic field

$\mathbf{C}$	Matrix functions that transforms the GP to the sensor frame s
d	Domain index
$f\left(\mathbf{p}\right)$	Unkown process described by a GP, depending on variable $\mathbf{p}$
$\mathcal{GP}$	Gaussian process prior

$\nabla\mathbf{\Phi}$	Matrix containing the gradients of $(N_m - 1)$ eigenfunctions $\phi\left(\right)$ at time t
$\mathbf{H}\left(\cdot\right)$	Magnetic field
$\mathbf{K}_t$	Kalman gain matrix in the KF at t
L	Cuboid lenghts
l	Lenghtscale hyperparameter for a kernel
$\mathcal{L}\left(\boldsymbol{\theta}\right)$	Log marginal likelihood function used to optimise the hyperparameters $\boldsymbol{\theta}$
LI	Cuboid inner, non-overlapping lengths for mulitple domain scenarios
LO	Cuboid overlapping lenghts for mulitple domain scenarios
$m\left(\mathbf{x}\right)$	Mean function of a GP at $\mathbf{x}$
$\mathbf{m}_t$	GPR mean at t
$\mathbf{n}$	Index set of permutation integers for the largest N_m eigenvalues
N_d	Size of the sets $\boldsymbol{\psi}_t^{i,d}$, $\boldsymbol{\rho}_t^{i,d}$ and $\boldsymbol{\zeta}_t$
N_m	Number of eigenvalues λ and eigenfunctions $\phi\left(\right)$ used in the reduced rank GP to approximate a full GP
${}^{\mathrm{n}}\mathbf{p}^d$	Center of domain Ω^d, expressed in the navigation frame
$\mathbf{P}_{d,t}$	Covariance of a GP representing the magnetic field within domain d at time t
s	Sensor Index
$S_{\mathrm{SE}}\left(\lambda\right)$	Spectral density function of $\kappa_{\mathrm{SE}}\left(\mathbf{x}, \mathbf{x'}\right)$ as function of λ
$\mathbf{S}_t$	Innovation matrix in the KF at t
$\mathrm{vec}\left(\mathbf{z}\right)$	Column vector containing observations $\mathbf{z}$
$\mathbf{x}$	State in the general description of the GP
y_k	Output in one direction of the multidimensional GP
$\mathbf{Z}_d^i$	Intermediate variable containing the observations of different sensors. Each domain has this set and it contains the sensor observations of the sensors within this domain, if the domain boundary of this set is further away from the sensor than the other domains in which this sensor is located
$\mathbf{\Lambda}$	Matrix containing the spectral densities
$\mathbf{\Phi}$	Matrix containing (N_m) eigenfunctions $\phi\left(\right)$
X	Intermediate matrix containing N_m eigenfunctions $\phi\left(\right)$

$\mathbf{\Psi}_d^i$	Intermediate variable containing the positions of different sensors. Each domain has this set and it contains the sensor positions of the sensors within this domain, if the domain boundary of this set is further away from the sensor than the other domains in which this sensor is located
Ω	Compact subset of $\mathbb{R}^3$
$\boldsymbol{\zeta}_d$	Intermediate variable containing the observations for the sensor located in d
$\boldsymbol{\theta}$	Vector containing the hyperparameters
$\kappa\left(\mathbf{x}, \mathbf{x'}\right)$	Kernel function of a GP
$\kappa_{\text{approx}}\left(\mathbf{p}, \mathbf{p'}\right)$	Approximation kernel where the SE is opproximated with basisfunctions and their eigenvalues λ_j
$\kappa_{\text{curl-free}}\left(\mathbf{x}, \mathbf{x'}\right)$	Curl free covariance function, kernel
$\kappa_{\text{lin}}\left(\mathbf{x}, \mathbf{x'}\right)$	Linear covariance function, kernel
$\kappa_{\text{SE}}\left(\mathbf{x}, \mathbf{x'}\right)$	Squared exponential covariance function, kernel
λ	Eigenvalue
$\boldsymbol{\rho}^d$	Intermediate variable containing the orientations of different sensors. Used as argument to matrix function $\mathbf{C}$. Optional subscript d indicates that the variable only contains orientations of sensors located in domain d
ς	The set of sensor indices, for which the corresponding sensor are located in a certain domain
σ_{const}	Magnitude scale for the GP prior on the magnetic field
σ_{lin}	Linear magnitude hyperparameter
$\sigma_{\mathbf{z}}$	Noise on the observations
σ_{SE}	SE magnitude hyperparameter
$\varphi\left(\cdot\right)$	Scalar potential
$\boldsymbol{\chi}$	Matrix in which eigenfunctions $\phi\left(\right)$ are collected
$\boldsymbol{\psi}^{i,d}$	Intermediate variable containing the positions of different sensors. Used as argument to matrix function $\nabla\mathbf{\Phi}$. Optional subscript d indicates that the variable only contains positions of sensors located in domain d
$\phi\left(\mathbf{p}\right)$	Eigenfunction evaluated at $\mathbf{p}$
$\boldsymbol{\varrho}_d^i$	Intermediate variable containing the orientations of different sensors. Each domain has this set and it contains the sensor orientations of the sensors within this domain, if the domain boundary of this set is further away from the sensor than the other domains in which this sensor is located