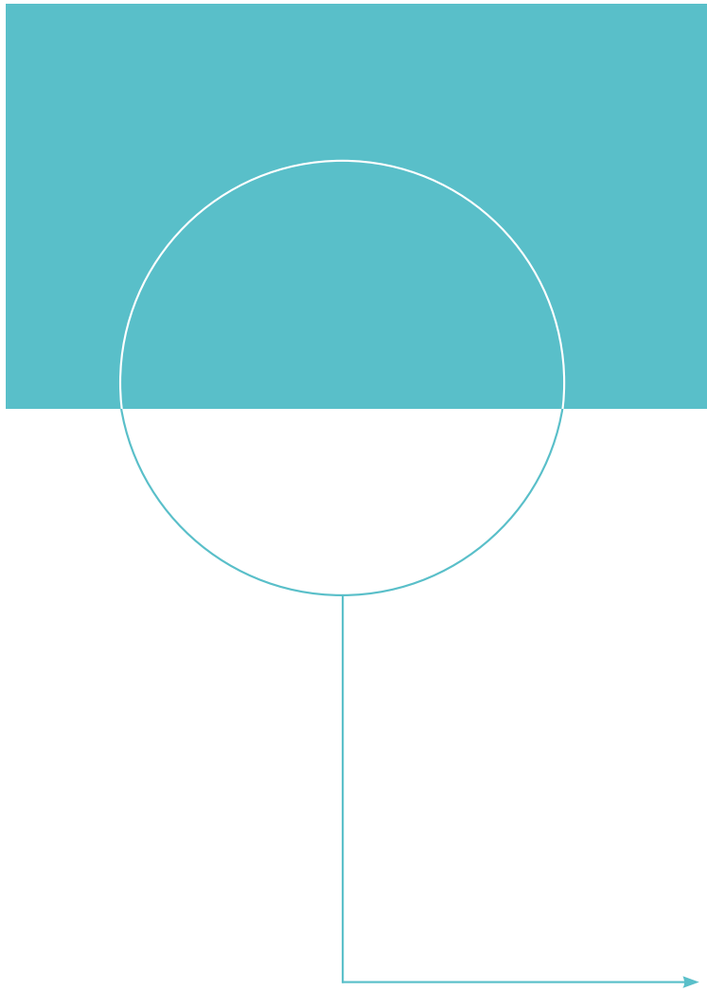


Master's thesis

2020

Frida Viset

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Engineering Cybernetics

Frida Viset

Tightly and loosely coupled sensor fusion of inertial and magnetic field norm measurements in a foot-mounted sensor

July 2020

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Tightly and loosely coupled sensor fusion of inertial and magnetic field norm measurements in a foot-mounted sensor

Frida Viset

Cybernetics and Robotics
Submission date: July 2020
Supervisor: Manon Kok
Co-supervisor: Jan Tommy Gravdahl

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Acknowledgements

The creation of this thesis began with a great discovery in an unexpected place. I arrived in Delft at the beginning of August 2018, planning to stay for a year as an exchange student at the Technical University of Delft. During the first few weeks, I was introduced to the research of Dr Manon Kok in the field of sensor fusion through a leaflet for master thesis topics that someone had left in a printer. Sensor fusion combined topics from all my favourite subjects NTNU - linear systems, optimisation, modelling and simulation, and it even had a touch of functional analysis. I composed a mail asking for a thesis project, which I then edited in multiple iterations before finally sending it.

One year in Delft became two when she agreed to supervise my thesis. Her research inspired my interest in sensor fusion, and her enthusiasm and support have made this thesis possible. I want to thank her for inspiring me to set ambitious goals, guiding me to approach the problems in the right way, and patiently helping with many details. I would also like to thank professor Jan Tommy Gravdahl, for being my supervisor at NTNU, for helping organise the practicalities around doing the master abroad, and for answering questions at any hours.

I want to give my thanks to my friend Silvia, for providing a customised LaTeX-template for this thesis and for teaching me to make pasta like the Italians do it. I also want to thank my lovely boyfriend Niels, for always believing in me, and his mother Marjanne, for making us apple pie when the study nights got long.

Contents

Abstract

This thesis proposes and compares experimentally two algorithms for indoor pedestrian navigation using a foot-mounted tri-axis gyroscope, accelerometer and magnetometer. The algorithms estimate the pedestrians' position using magnetic field simultaneous localisation and mapping (Mag-SLAM).

The zero-velocity aided Extended Kalman filter (the ZUPT-aided EKF) for navigation using a foot-mounted accelerometer, magnetometer and gyroscope were proposed for indoor pedestrian navigation by [1]. Recently, a computationally tractable algorithm using a Rao-Blackwellized particle filter to perform Mag-SLAM on the three-component magnetic field was proposed by [2]. The first proposed algorithm in this thesis uses the output of the ZUPT-aided EKF as an input to a Rao-Blackwellized particle filter to perform Mag-SLAM using magnetic field norm measurements from a foot-mounted sensor. This is an example of loosely coupled sensor fusion. The second algorithm uses a joint stochastic state-space model for the system and applies a single Rao-Blackwellised particle filter to the joint estimation problem. This is an example of tightly coupled sensor fusion.

Both algorithms are tested experimentally. The experimental results demonstrate that the algorithms remedies drift in the position estimate compared to the ZUPT-aided EKF. The performance of the two algorithms also illustrates how tightly coupled sensor fusion can be beneficial for the estimation of Mag-SLAM in a foot-mounted sensor.

Sammendrag

I denne oppgaven utformes to algoritmer for innendørs personavigasjon ved hjelp av fotfestet akselerometer, gyroskop og magnetometer. Algoritmene konstruerer et kart over variasjoner i magnetfeltet innendørs, samtidig som de bruker magnetfeltmålinger til å navigere i kartet.

Den første algoritmen bruker to tilgjengelige metoder som byggesteiner. Den første metoden er et Kalmanfilter for estimering av innendørs posisjon ved hjelp av fotfestet akselerometer og gyroskop. Den andre metoden er et marginalisert partikkelfilter for samtidig å lage et kart over magnetfeltanomalier og navigere i kartet. Metoden bruker gaussiske prosesser, tilnærmet med metoder fra funksjonalanalyse for å unngå at kjøretid skaleres med antall målinger. Ved å modelere posisjonsestimatet fra Kalmanfilteret som en usikker informasjonskilde om endring i posisjon, anvendes et marginalisert partikkelfilter for å oppnå samtidig navigasjon og kart-konstruksjon av variasjoner i magnetfeltet. Den første algoritmen er et eksempel på løst koblet sensorfusjon. Den andre algoritmen bruker de matematiske modellene som byggesteinene i den første algoritmen er basert på. Disse modellene gir til sammen et estimeringsproblem som kan løses av et enkelt marginalisert partikkelfilter. Den andre algorithmen er et eksempel på tett koblet sensorfusjon.

Begge algoritmene testes eksperimentelt, og vises å fjerne effekten av drift i eksisterende personnavigasjon ved bruk av fot-montert akselerometer ved passering av tidligere kartlagte områder. Algoritmene oppnår sammenlignbar presisjon i tre dimensjoner som eksisterende metoder for navigasjon ved hjelp av fotfestet magnetometer, akselerometer og gyroskop oppnår i to dimensjoner. Resultatene illustrerer også hvordan tett koblet sensorfusjon kan være fordelaktig for samtidig magnetfeltkartlegging og navigasjon ved bruk av fotfestet sensor.

Chapter 1

Introduction

This chapter first presents the motivation for the research done in this thesis. Secondly, previous research indoor pedestrian navigation using foot-mounted accelerometer and gyroscope is briefly discussed. Thirdly, related research into pedestrian navigation using accelerometer, gyroscope and magnetometer measurements is summarised. At the end of the chapter, the contributions of this thesis are introduced and compared with existing work.

1.1 Motivation for research

Firefighters and other first responders have reported that disorientation and being lost inside a building is a common reason for traumatic injuries, as well as a contributing factor to firefighter deaths [3]. Currently, most common navigation methods for firefighters are a combination of burning flashlights to mark exits, ropes to mark safe ways out of the dangerous area, and alarms connected to wearable sensors that go off if a firefighter does not move for a specific amount of time [3]. Location and navigation is essential in emergency response scenarios [4]. In most of these situations, complete building plans are not available [5].

A common way to estimate the position of a pedestrian outdoors is to combine accelerometer, gyroscope and global navigation satellite system (GNSS) signals. Indoors, the GNSS signals are usually not strong enough to improve the position estimate [6]. Accelerometers and gyroscopes can still be used to give information about the acceleration and angular velocity of a pedestrian. The use of only on-board accelerometer and gyroscope data to estimate the position and orientation of an object is called inertial navigation [7]. For sensors that are rigidly attached to some part of a pedestrian, the most straightforward approach to inertial navigation is called dead reckoning integration. Dead reckoning integration estimates orientation and position by the integration of the gyroscope measurements and acceleration measurements corrected for gravity [7]. Dead reckoning integration is not a sufficient approach alone for indoor pedestrian navigation, because the orientation and position estimates from dead reckoning have errors that grow over time (drift) [8]. More accurate orientation estimates can be obtained by using extended Kalman filter methods. The extended Kalman filters typically incorporate the assumption that the measured acceleration is mostly caused by the gravitational force and therefore can be used to reduce drift in orientation estimate by indicating which way is down [8]. When this assumption holds, the extended Kalman filter methods can eliminate orientation drift except for orientation deviations parallel to the gravity field [8]. The variance of the velocity estimate obtained from integrating the acceleration from the extended Kalman filter will increase linearly with time. The variance of the position estimate given by integrating this velocity will increase quadratically with time [7].

For a sensor that is placed on the foot, more accurate position and orientation estimates can be obtained [1]. These sensors require no external infrastructure, and they are cheap, lightweight and precise [9]. The position and orientation estimates from a foot-mounted sensor are more accurate because it is possible to use the raw accelerometer and gyroscope measurements to detect the stationary phase of a step, and then incorporate the assumption that velocity of the foot is zero during this phase [1]. This is called the zero-velocity assumption [1]. By using the zero-velocity assumption, orientation drift that is not parallel with the gravity field can be eliminated [3]. In addition, acceleration drift is eliminated, velocity drift is eliminated, and the of the position estimation error will increase more or less linearly with time [10]. As a foot-mounted sensor can be integrated into the heel of a large shoe, foot-mounted sensors can be used to retrieve measurements for indoor localisation of firefighters [11]. However, a survey [3] on Location and Navigation Support for Emergency Responders concludes that pedestrian inertial navigation using a foot-mounted sensor and the zero-velocity assumption still has a too large position drift to be a feasible strategy in emergency scenarios.

Recent work has shown that the indoor magnetic field is perturbed by furniture and metallic structures, perturbing the underlying earth magnetic field [12] and that recognising patterns in these anomalies can be used for indoor navigation [13]. A scalable and computationally tractable approach for removing drift in three-dimensional pedestrian navigation by fusing magnetometer readings and a drifting position estimate from a smartphone was proposed and implemented by [2]. The paper simultaneously creates a three-dimensional map of the magnetic field potential and uses the map to remove drift in pedestrian position estimation. Simultaneous localisation and magnetic field mapping on a foot-mounted sensor have been proposed by [14] to remove drift in two-dimensional pedestrian navigation using a foot-mounted sensor. The method in [14] removes drift from the two-dimensional position estimate and achieves high accuracy. This thesis investigates how magnetic field simultaneous localisation and mapping can be implemented using measurements from a foot-mounted sensor. The research is motivated by the need for drift reduction of indoor position estimates based on data from a foot-mounted sensor and the reported high performance of magnetic field simultaneous localisation and mapping for drift reduction.

1.2 Existing work indoor pedestrian navigation using foot-mounted accelerometer and gyroscope

The zero-velocity aided Extended Kalman filter, proposed by [1], is a computationally efficient method to obtain accurate, low-drift pedestrian position estimates from a foot-mounted sensor. The filter fuses measurements from a tri-axis accelerometer and a tri-axis gyroscope to give an estimate of the foot's orientation, position and velocity. Also, the filter uses detection of the stationary phase of the step to compensate for drift in velocity and orientation, and reduce drift in the position estimate. The assumption is that when the sensor has low angular velocity, low acceleration, low variance in acceleration and low variance in angular velocity over a short timeframe, it is an indication that the velocity of the sensor is zero [9].

The zero-velocity detector is defined as active whenever the sum of the variance and magnitude of the accelerometer and gyroscope measurements over a short time frame is below a threshold, and inactive when the sum exceeds the threshold [10]. The foot is typically stationary when the zero velocity detector of a foot-mounted sensor is active [1]. The zero-velocity aided Kalman filter includes a zero pseudo-measurement of the foot's velocity when the zero velocity detector is active. The difference between the expected velocity and the detection of the velocity being zero is close to linearly affected by the orientation error. The orientation error changes the estimated direction of the gravitational force, which changes the estimated acceleration vector, which changes the velocity estimate [15].

The extended Kalman filter uses the difference between the estimated velocity and the assumed zero-velocity to correct the orientation relative to the gravity vector. The zero-velocity aided extended Kalman filter is, therefore, able to compensate for orientation drift that is not parallel with the gravity field [15]. The drift of the zero-velocity aided extended Kalman filter was reported by [1] for one experiment to be 2 meters for a 0.15 meter long walk, using the RMSE of one checkpoint. The same method was later implemented and tested on multiple datasets by [9], with an approximate observed position estimation error of 0.15 meters at the end of a 50 meter long walk. The method was also tested by [16], who observed an estimation error of $0.4 - 0.6$ meters after a 65 meter long walk. The drift of the position estimate can increase when the sensor is mounted on the front of the foot (as is done for the experiments in this thesis) rather than integrated into the heal [3].

The odometry from a foot-mounted accelerometer and gyroscope has also been used to perform simultaneous localisation and mapping in two dimensions of frequently traversed paths in an office environment to compensate for drift [14]. The RMSE of this method was reported in a later publication [17] by the same research team to be in the range of 20-50 cm.

1.3 Existing work pedestrian navigation using accelerometer, gyroscope and magnetic field measurements

The research team in [17] also performs simultaneous localisation and mapping in two dimensions using the magnetic field and odometry from a foot-mounted sensor. By using the magnetic field anomalies, they achieve RMSE errors in the range of 9-22 cm. These errors were reported on a walk that had a duration of 17 minutes. Given a walking speed of 1.5 m/s, this corresponds to a distance of approximately 1.4 km. Simultaneous localisation and mapping using magnetic field anomalies have also been used in combination with various sources of odometry for indoor navigation purposes.

Previous research has investigated multiple approaches to estimate position and orientation indoors in the presence of magnetic field anomalies. Most of the research acknowledges the significant impact of magnetic field anomalies indoors. Magnetic field anomalies indoors occur because of the presence of steel structures and furniture with metallic components [12]. The magnetic field anomalies are rich in spatial variation [18] and are stable over long periods of time [19]. Some of the previous research attempts to detect and discard data with significant disturbances and uses the remaining magnetic field measurements to estimate the yaw angle ([20], [21], [22]). In [22], the magnetic field data is fused with gyroscope measurements using convex optimisation. Convex optimisation gives a robust heading estimate despite magnetic field disturbances after approximately 75 seconds for a hand-held device [22]. Other research uses maps of the magnetic field anomalies to improve position estimates. Navigation in two dimensions in a known magnetic field map stored as discrete values in a square grid with odometry and magnetic field measurements from a foot-mounted sensor was found by [19] to give an average pedestrian localisation error of 7.95 cm. As mentioned in the previous section, a similar approach using Simultaneous Localisation and mapping with a hexagonal grid for localisation in a two-dimensional plane and odometry from a foot-mounted sensor has been found by [14] to give a localisation error of 9-22 cm upon re-visitation of a single checkpoint.

In [2], an algorithm is developed for creating a map of the magnetic field anomalies and the underlying magnetic field direction on the fly, which makes it possible to both correct the orientation estimate and the position estimate using the magnetic field map. The researchers used odometry from a smartphone generated with the phone's accelerometer, gyroscope and camera in combination with magnetic field measurements to perform simultaneous localisation and magnetic field mapping. They used reduced rank Gaussian process

regression to create the magnetic field map, and a Rao-Blackwellised particle filter to do simultaneous localisation and mapping. The reduced rank Gaussian process regression makes the mapping procedure computationally tractable and is possible in a finite region. To estimate the position in an arbitrarily large region, the research team build a larger map from a combination of many hexagonal tiles. The method is, therefore, computationally tractable and scalable. Combined gyroscope, camera and accelerometer measurements give a very accurate position estimate, but it requires that the phone is held in hand and used to film the surrounding environment. Other sources of pedestrian odometry are usually less accurate. The researchers added a heuristic filter that removed any discontinuous corrections for drift provided by the visual information. The position estimate after applying this filter was still considerably more accurate compared to dead-reckoning of accelerometer and gyroscope measurements in a hand-held device. The drift in their final position estimate was comparable to the reported drift from using foot-mounted accelerometer and gyroscopes. They then showed that their method was able to eliminate the effect of this drift, indicating that simultaneous localisation and mapping can compensate for drift in pedestrian navigation methods.

1.4 Contributions

This thesis proposes two algorithms for drift-free three-dimensional position estimate using only magnetometer, accelerometer and gyroscope data from a foot-mounted sensor. One algorithm has a more straightforward design and is modular, and the other is more accurate. The algorithms extend on the open-source Matlab implementation of the ZUPT-aided EKF published by [23], to overcome the inherent estimation drift present in the filter. Both algorithms use a magnetic field norm map which reduces the computational complexity compared to creating a map of all three magnetic field components. The map is built within the boundaries of a fixed area, but in [2] it was shown that the approach is scalable to an arbitrarily large area. Additionally, in contrast to the application in [2], the odometry used in this thesis is only based on an accelerometer, gyroscope and magnetic field data, without the use of a camera to reduce odometry drift.

It is demonstrated through an experiment on real data collected by revisiting checkpoints that the proposed methods for Magnetic field SLAM compensate for drift when positions are revisited. The estimation problem solved in this thesis is similar to the estimation problem that is solved in [14]. The researchers in [14] combines odometry from a foot-mounted sensor with magnetic field norm measurements to achieve position estimates in two dimensions and uses a different method for creating the magnetic field map. This thesis estimates both the map and position in three dimensions and achieves similar position estimation accuracy as [14] achieves in two dimensions.

Chapter 2

Localisation filters

This chapter introduces Bayesian recursive estimation, the linear Kalman filter, the extended Kalman filter, and the particle filter. The Kalman filter and the Particle filter are both examples of recursive Bayesian estimation [24], and they will be presented in this chapter as instances of a general recursive Bayesian filtering algorithm. The ZUPT-aided EKF is an example of an extended Kalman filter for indoor pedestrian navigation. Particle filters are a suitable approach to incorporate the magnetic field measurements in the position estimate [25]. This is because the indoor magnetic field tends to have strong perturbations [12], and particle filters are a suitable approach when the measurement is a highly nonlinear function of the state [26]. Many publications also demonstrate how map information can be used to perform localisation with a particle filter ([27], [19], [28]).

2.1 Recursive Bayesian estimation

Recursive Bayesian estimation refers to the estimation of a state $\mathbf{x}(t)$ using a history of measurements $\mathbf{y}(t)$ from a model of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{w}(t), \mathbf{u}(t)), \tag{2.1}$$
$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{v}(t), \mathbf{u}(t)), \tag{2.2}$$

where $\mathbf{f}$ and $\mathbf{h}$ are known functions, $\mathbf{u}(t)$ is a known input function, and $\mathbf{v}(t)$ and $\mathbf{w}(t)$ are stochastic processes with known distributions [24]. This chapter will investigate Bayesian recursive estimation methods for a discrete state-space model

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{w}_{t-1}, \mathbf{u}_{t-1}), \tag{2.3}$$
$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{v}_t, \tag{2.4}$$

where the functions $\mathbf{f}$ and $\mathbf{h}$ are time-independent, the input $\mathbf{u}_t$ does not affect the measurement directly, the measurement noise $\mathbf{v}_t$ is additive to the measurement. Also, the distribution of $\mathbf{u}_t$ and $\mathbf{v}_t$ is assumed known and Gaussian, and the two processes are assumed to be uncorrelated. In the remainder of the thesis, a collection of measurements $\{\mathbf{y}_t\}_{t=1}^{N} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N)$ will be denoted $\mathbf{y}_{1:N}$ for simplicity. It is not generally the case that the noise processes are uncorrelated (see for example the derivation of the Kalman filter in [29]), but it holds for all the models used in this thesis. The filtering algorithms for localisation used in the fields of indoor pedestrian navigation using a foot-mounted sensor and magnetic field anomaly navigation commonly includes Kalman filters and Particle filters [2].

Algorithm 1: A general Bayesian recursive estimation algorithm

Data: A measurement sequence $\mathbf{y}_{0:N}$ produced by the stochastic state-space model in equations 2.3 and 2.4, Process noise $\mathbf{Q}_k$, and measurement noise $\mathbf{R}_k$. Initial state estimate $\hat{\mathbf{x}}_{0|-1}$ normally distributed around the true initial state with covariance $\mathbf{P}_{0|-1}$

Result: Estimates of the prior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\}_{t=0}^{N}$ and the posterior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t})\}_{t=0}^{N}$

for *t=0:N* **do**

1. Compute or estimate the posterior distribution using Bayes theorem

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{0:t-1})p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1})}, \tag{2.5}$$

   where $p(\mathbf{y}_t|\mathbf{y}_{0:t-1})$ can be computed as a marginalisation of known distributions according to $p(\mathbf{y}_t|\mathbf{y}_{0:t-1}) = \int p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{0:t-1})p(\mathbf{x}_t|\mathbf{y}_{0:t-1})d\mathbf{x}_t$, and the density $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{0:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t)$ is defined implicitly by equation 2.4.

2. Compute or estimate the prior distribution using marginalisation

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{0:t}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{y}_{0:t})p(\mathbf{x}_t|\mathbf{y}_{0:t})d\mathbf{x}_t, \tag{2.6}$$

   where the term $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{y}_{0:t}) = p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ is defined implicitly by equation 2.3.

end

The probability density $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{0:t-1})$ is defined implicitly by equation 2.4. As the stochastic term $\mathbf{v}_t$ in this equation is additive and Gaussian, the probability density $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{0:t-1})$ will also be Gaussian. The probability density $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ is defined implicitly through equation 2.3. This probability density is not necessarily Gaussian, as the possibly nonlinear function $\mathbf{f}$ takes the noise term $\mathbf{w}_t$ as an argument. The probability density can not always be determined analytically, which is an example of how the general Bayesian recursive estimation problem is an ill-posed problem [24]. Multiple approaches have been developed to estimate this probability density, which each are useful under different circumstances. The linear Kalman filter is derived for a special case of this model where the probability density is Gaussian and therefore can be determined analytically. The extended Kalman filter approximates the probability density with a best-fit Gaussian, and the Particle filter uses Monte-Carlo sampling to approximate this probability density.

2.2 The linear Kalman filter

The linear Kalman filter estimates the state $\mathbf{x}_t$ based on all measurements $\mathbf{y}_{1:t}$ up until and including the current timestep, assuming a linear stochastic state-space model

$$\mathbf{x}_t = \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{G}_t\mathbf{w}_{t-1} + \mathbf{B}_t\mathbf{u}_{t-1} \tag{2.7}$$

$$\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{L}_t\mathbf{v}_t, \tag{2.8}$$

where the system matrices $\mathbf{F}_t$, $\mathbf{G}_t$, $\mathbf{B}_t$, $\mathbf{H}_t$ and $\mathbf{L}_t$ are known. The process and measurement noise processes $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ and $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ have known distributions and have zero correlation to each other. Their joint distribution is

$$\begin{bmatrix}\mathbf{w}_t \\ \mathbf{v}_t\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0 \\ 0\end{bmatrix}, \begin{bmatrix}\mathbf{Q}_t & 0 \\ 0 & \mathbf{R}_t\end{bmatrix}\right), \tag{2.9}$$

where the zero-matrices in the upper right and the lower left corner of the joint covariance matrix indicate the noise processes are uncorrelated.

The version of the Kalman filter that will be described in this section assumes that there exists an estimate $\hat{\mathbf{x}}_{0|-1}$ of the initial value $\mathbf{x}_0$, and that the estimate relates to the initial state according to

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_{0|-1}, \mathbf{P}_{0|-1}), \tag{2.10}$$

where $\mathbf{P}_{0|-1}$ is the variance of the estimate. The subscript $0|-1$ indicates that no measurements are used to construct the estimate. This can alternatively be expressed as a Gaussian probability density function for the initial state

$$p(\mathbf{x}_0) = \frac{1}{\sqrt{(2\pi)^k \left|\mathbf{P}_{0|-1}\right|}} \exp\left(-\frac{1}{2}(\mathbf{x}_0 - \hat{\mathbf{x}}_{0|-1})^T \mathbf{P}_{0|-1}^{-1}(\mathbf{x}_0 - \hat{\mathbf{x}}_{0|-1})\right), \tag{2.11}$$

where k is the state dimension of $\mathbf{x}_0$, and $\left|\mathbf{P}_{0|-1}\right|$ is the determinant of the matrix $\mathbf{P}_{0|-1}$. In the remainder of this thesis, the multivariate Gaussian distribution will be denoted using the notation

$$\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^k \left|\boldsymbol{\Sigma}\right|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right), \tag{2.12}$$

where $\boldsymbol{\mu}$ is the expected value of the normal distribution, k is the state dimension of $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ is the covariance matrix, and $\mathbf{y}$ is the point where the distribution is evaluated. The measurement $\mathbf{y}_0$ is therefore a linear combination of stochastic variables with Gaussian distributions, and therefore itself normally distributed. The probability density function can be computed as

$$p(\mathbf{y}_0|\mathbf{x}_0) = \mathcal{N}(\mathbf{y}_0; \mathbf{H}\mathbf{x}_0, \mathbf{L}_0^T \mathbf{R}_0 \mathbf{L}_0), \tag{2.13}$$

by using the property that if $\mathbf{z} \sim \mathcal{N}(\bar{\mathbf{z}}, \boldsymbol{\Sigma})$, then $\mathbf{G}\mathbf{z} \sim \mathcal{N}(\mathbf{G}\bar{\mathbf{z}}, \mathbf{G}^T\boldsymbol{\Sigma}\mathbf{G})$. The posterior density $p(\mathbf{x}_0|\mathbf{y}_0)$ can be computed using Bayes' theorem as

$$p(\mathbf{x}_0|\mathbf{y}_0) = \frac{p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{x}_0)}{p(\mathbf{y}_0)}, \tag{2.14}$$

where $p(\mathbf{y}_0)$ can be computed by integration of known densities according to $p(\mathbf{y}_0) = \int p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{x}_0)d\mathbf{x}_0$. After insertion of all known terms into equation 2.14, the expression for the posterior probability density is given by a Gaussian distribution

$$p(\mathbf{x}_0|\mathbf{y}_0) = \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_{0|0}, \mathbf{P}_{0|0}) \tag{2.15}$$

$$\hat{\mathbf{x}}_{0|0} = \hat{\mathbf{x}}_{0|-1} + (\mathbf{P}_{0|-1}\mathbf{H}_0^T(\mathbf{L}_0\mathbf{R}_0\mathbf{L}_0^T + \mathbf{H}_0\mathbf{P}_{0|-1}\mathbf{H}_0^T)^{-1})(\mathbf{y}_0 - \mathbf{H}_0\hat{\mathbf{x}}_{0|0-1}) \tag{2.16}$$

$$\mathbf{P}_{0|0} = \mathbf{P}_{0|0-1} - (\mathbf{P}_{0|-1}\mathbf{H}_0^T(\mathbf{L}_0\mathbf{R}_0\mathbf{L}_0^T + \mathbf{H}_0\mathbf{P}_{0|-1}\mathbf{H}_0^T)^{-1})\mathbf{H}_0\mathbf{P}_{0|-1}, \tag{2.17}$$

with mean and covariance describing the posterior state estimate. The prior can then be calculated by marginalizing out the previous state through integration

$$p(\mathbf{x}_1|\mathbf{y}_0) = \int p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{y}_0)p(\mathbf{x}_0|\mathbf{y}_0)d\mathbf{x}_t, \tag{2.18}$$

where the probability $p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{y}_0) = p(\mathbf{x}_1|\mathbf{x}_0)$ because of the Markov property of state-space models [26]. Furthermore, the density $p(\mathbf{x}_1|\mathbf{x}_0)$ is defined implicitly by equation 2.8 as

$$p(\mathbf{x}_1|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_1; \mathbf{F}\mathbf{x}_0 + \mathbf{B}_0\mathbf{u}_0, \mathbf{G}_0^T\mathbf{Q}_0\mathbf{G}_0) \tag{2.19}$$

The marginalised probability density of a Gaussian is another Gaussian, and by calculating the expected value and the covariance of the prior estimate we can find an explicit solution to equation 2.18

$$\begin{aligned} &p(\mathbf{x}_1|\mathbf{y}_0) = \mathcal{N}(\mathbf{x}_1; \hat{\mathbf{x}}_{1|0}, \mathbf{P}_{1|0}) \\ &\hat{\mathbf{x}}_{1|0} = \mathbf{F}_0\hat{\mathbf{x}}_{0|0} + \mathbf{B}_0\mathbf{u}_0 \\ &\mathbf{P}_{0|0} = \mathbf{F}_0^T\mathbf{P}_{0|0}\mathbf{F}_0 + \mathbf{Q}_0. \end{aligned} \tag{2.20}$$

When these computational steps are repeated for the general transition from $p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})$ and $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ to $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ and $p(\mathbf{x}_{t+1}|\mathbf{y}_{0:t-1})$, the result is the linear Kalman filter described in Algorithm 2.

Algorithm 2: A linear Kalman filtering algorithm

Data: A measurement sequence $\mathbf{y}_{0:N}$ produced by the stochastic state-space model in equations 2.7 and 2.8, Process noise $\mathbf{Q}_k$, and measurement noise $\mathbf{R}_k$. Initial state estimate $\hat{\mathbf{x}}_{0|-1}$ normally distributed around the true initial state with covariance $\mathbf{P}_{0|-1}$

Result: Estimates of the prior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\}_{t=0}^{N}$ and the posterior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t})\}_{t=0}^{N}$

for *t=0:N* **do**

1. Compute the posterior distribution using Bayes theorem

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}) \tag{2.21}$$
$$\mathbf{S}_t = (\mathbf{L}_t\mathbf{R}_t\mathbf{L}_t^T + \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T) \tag{2.22}$$
$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1} \tag{2.23}$$
$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}) \tag{2.24}$$
$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{H}_t\mathbf{P}_{t|t-1} \tag{2.25}$$

2. Compute the prior distribution using marginalisation as

$$p(\mathbf{x}_{t+1}|\mathbf{y}_t) = \mathcal{N}(\mathbf{x}_{t+1}; \hat{\mathbf{x}}_{t+1|t}, \mathbf{P}_{t+1|t}) \tag{2.26}$$
$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{F}_t\hat{\mathbf{x}}_{t|t} + \mathbf{B}_t\mathbf{u}_t \tag{2.27}$$
$$\mathbf{P}_{t+1|t} = \mathbf{F}_t\mathbf{P}_{t|t}\mathbf{F}_t^T + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^T. \tag{2.28}$$

end

The linear Kalman filter solves the filtering problem for a model that is a special case of the model in equations 2.3 and 2.4, and the algorithm itself is also a special case of Algorithm 1. As both the posterior probability density and prior probability density functions are Gaussians, it is natural to choose their expected values $\hat{\mathbf{x}}_{t|t}$ and $\hat{\mathbf{x}}_{t|t-1}$ as filtered and predictive state estimates respectively. The estimate $\hat{\mathbf{x}}_{t|t}$ corresponds to the maximum a-posteriori estimate of the state $\hat{\mathbf{x}}_t$, in addition to being the expected value of the state.

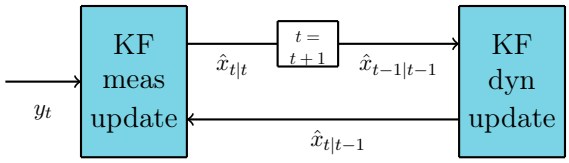


Figure 2.1: *The linear Kalman filter with uncorrelated process and measurement noise*

The working mechanism of the linear Kalman filter can be summarized with the block diagram in Figure 2.1. The figure shows how the prior is updated using the measurement to give the posterior estimate, and how the posterior is updated using the dynamic model to give the prior.

2.3 The extended Kalman filter

The extended Kalman filter described in this section estimates the state $\mathbf{x}_t$ based on all measurements $\mathbf{y}_{1:t}$ up until and including the current timestep, assuming that the state is a result of the nonlinear state space model in equations 2.3 and 2.4, and with $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ and $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ being known distributions with zero correlation as described in equation

2.9. As for the linear Kalman filter, it is assumed that an initial state estimate $\hat{\mathbf{x}}_{0|-1}$ of the initial state $\mathbf{x}_0$ is available, where $\mathbf{x}_0 \sim \mathbf{N}(\hat{\mathbf{x}}_{0|-1}, \mathbf{P}_{0|-1})$. When the measurement model in 2.4 gives the measurement $\mathbf{y}_0$, the distribution $p(\mathbf{y}_0)$ is not necessarily Gaussian. Instead of using the property that if $\mathbf{z} \sim \mathbf{N}(\bar{\mathbf{z}}, \mathbf{\Sigma})$, then $\mathbf{Gz} \sim \mathcal{N}(\mathbf{G}\bar{z}, \mathbf{G}^T\mathbf{\Sigma}\mathbf{G})$, the extended Kalman filter uses the approximation that if $\mathbf{z} \sim \mathbf{N}(\bar{\mathbf{z}}, \mathbf{\Sigma})$, then $\mathbf{g}(\mathbf{z})$ is approximately distributed according to

$$\mathbf{g}(\mathbf{z}) \sim \mathcal{N}\left(\mathbf{g}(\bar{z}), \left.\frac{\mathrm{d}\mathbf{g}}{\mathrm{d}\mathbf{z}}\right|_{\mathbf{z}=\bar{\mathbf{z}}}^{T} \mathbf{\Sigma} \left.\frac{\mathrm{d}\mathbf{g}}{\mathrm{d}\mathbf{z}}\right|_{\mathbf{z}=\bar{\mathbf{z}}}\right). \tag{2.29}$$

The covariance of the Gaussian is defined as the covariance of the input argument multiplied on either end with the gradient of the nonlinear function [26]. To simplify the gradient notation, the system matrices

$$\hat{\mathbf{F}}_t = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_t}\right|_{\substack{\mathbf{w}_t=\mathbf{0} \\ \mathbf{x}_t=\hat{\mathbf{x}}_{t|t}}}, \; \hat{\mathbf{G}}_t = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}_t} =\right|_{\substack{\mathbf{w}_t=\mathbf{0} \\ \mathbf{x}_t=\hat{\mathbf{x}}_{t|t}}}, \tag{2.30}$$

$$\hat{\mathbf{H}}_t = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_t}\right|_{\substack{\mathbf{v}_t=\mathbf{0} \\ \mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1}}}, \; \hat{\mathbf{L}}_t = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{v}_t}\right|_{\substack{\mathbf{v}_t=\mathbf{0} \\ \mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1}}}, \tag{2.31}$$

defined based on equations 2.3 and 2.4 will be used to describe the function gradients. Using these definitions and the approximation in equation 2.29, we can approximate the measurement probability density as

$$p(\mathbf{y}_0|\mathbf{x}_0) = \mathcal{N}(\mathbf{y}; \mathbf{g}(\mathbf{x}_0, \mathbf{0}), \hat{\mathbf{L}}_t^T \mathbf{R}_0 \hat{\mathbf{L}}_t), \tag{2.32}$$

Based on this approximation, the posterior density $p(\mathbf{x}_0|\mathbf{y}_0)$ can be computed in the same way as in equations 2.15 - 2.16, replacing the matrices $\mathbf{H}_0$ and $\mathbf{L}_0$ with the matrices $\hat{\mathbf{H}}_0$ and $\hat{\mathbf{L}}_0$, and replacing the term $\mathbf{H}_0\hat{\mathbf{x}}_{0|-1}$ with the term $\mathbf{h}(\hat{\mathbf{x}}_{0|-1})$. The probability density $p(\mathbf{x}_1|\mathbf{x}_0)$ can also be estimated using the approximation in equation 2.29, resulting in the prior probability density estimate

$$p(\mathbf{x}_1|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_1; \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0, \mathbf{0}), \hat{\mathbf{G}}_0^T \mathbf{Q}_0 \hat{\mathbf{G}}_0), \tag{2.33}$$

that can be used in equation 2.18 to calculate the prior probability density $p(\mathbf{x}_1|\mathbf{y}_0)$.

When these computational steps are repeated for the general transition from $p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})$ and $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ to $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ and $p(\mathbf{x}_{t+1}|\mathbf{y}_{0:t-1})$, the result is the extended Kalman filter described in Algorithm 3.

Algorithm 3: An extended Kalman filtering algorithm

Data: A measurement sequence $\mathbf{y}_{0:N}$ produced by the stochastic state-space model in equations 2.3 and 2.4, process noise $\mathbf{Q}_k$, and measurement noise $\mathbf{R}_k$. Initial state estimate $\hat{\mathbf{x}}_{0|-1}$ normally distributed around the true initial state with covariance $\mathbf{P}_{0|-1}$

Result: Estimates of the prior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\}_{t=0}^{N}$ and the posterior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t})\}_{t=0}^{N}$

for *t=0:N* **do**

1. Compute the posterior distribution using Bayes theorem

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}) \tag{2.34}$$
$$\mathbf{S}_t = (\hat{\mathbf{L}}_t \mathbf{R}_t \hat{\mathbf{L}_t}^T + \hat{\mathbf{H}}_t \mathbf{P}_{t|t-1} \hat{\mathbf{H}}_t^T) \tag{2.35}$$
$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \hat{\mathbf{H}}_t^T \mathbf{S}_t^{-1} \tag{2.36}$$
$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{h}(\hat{\mathbf{x}}_{t|t-1}, \mathbf{0})) \tag{2.37}$$
$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \hat{\mathbf{H}}_t \mathbf{P}_{t|-1} \tag{2.38}$$

2. Compute the prior distribution using marginalisation

$$p(\mathbf{x}_{t+1}|\mathbf{y}_t) = \mathcal{N}(\mathbf{x}_{t+1}; \hat{\mathbf{x}}_{t+1|t}, \mathbf{P}_{t+1|t}) \tag{2.39}$$
$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{f}(\hat{\mathbf{x}}_{t|t}, \mathbf{u}_t, \mathbf{0}) \tag{2.40}$$
$$\mathbf{P}_{t+1|t} = \hat{\mathbf{F}}_t \mathbf{P}_{t|t} \hat{\mathbf{F}}_t^T + \hat{\mathbf{G}}_t \mathbf{Q}_t \hat{\mathbf{G}}_t^T. \tag{2.41}$$

end

This extended Kalman filter algorithm solves the stochastic nonlinear filtering problem in equations 2.3-2.4, but it relies on the assumption that the local linearisations of the model will give a good approximation of the necessary probability densities. The algorithm itself is a special case of Algorithm 1. The expected values $\hat{\mathbf{x}}_{t|t}$ and $\hat{\mathbf{x}}_{t+1|t}$ are used as filtered and predictive state estimates, respectively. These values will also correspond to the maximal likelihood arguments of the posterior and prior probability densities.

2.4 The particle filter

The particle filter, like the extended Kalman filter, estimates the state $\mathbf{x}_t$ based on all measurements $\mathbf{y}_{0:t}$ up until and including the current timestep. The particle filter described in this section is based on the assumption that the state is a result of the nonlinear stochastic state-space model in equations 2.3 and 2.4. Additionally, it is based on the assumption that the process and measurement noise are white noise processes with covariances $\mathbf{Q}_t$ and $\mathbf{R}_t$, and zero covariance between the two noise processes. It is assumed that the state has a known initial distribution of $p(\mathbf{x}_0)$. However, this distribution does not necessarily have to be Gaussian.

The initialisation of the particle filter is performed by creating a particle cloud $\left\{\mathbf{x}_{t|t}^i\right\}_{i=1}^{M}$ with M corresponding weights $\left\{\mathbf{w}_t^i\right\}_{i=1}^{M}$, where M is the number of particles. A particle cloud can be described as a set of guesses at what the state is. The weights give information about how likely each estimate is, given the incoming measurements at each timestep. The cloud is generated by Monte-Carlo sampling from the initial known distribution $p(\mathbf{x}_0)$, so that the initial particles follow the distribution $\mathbf{x}_{0|-1}^i \sim p(\mathbf{x}_0)$. For each of the particles,

the probability $p(\mathbf{y}_0|\mathbf{x}^i_{0|-1})$ can be computed as an Gaussian with the expected value from equation 2.4 and covariance $\mathbf{R}_t$. This probability density is used to calculate the weights according to

$$w^i_0 = p(\mathbf{y}_0|\mathbf{x}^i_{0|-1}), \tag{2.42}$$

so that each weight reflects the likelihood of the particle being the state to produce the observed measurement. In addition, the weights are normalised so that $\sum_{i=1}^{n} w^i_{-1} = 1$ is true. The particle filter again uses sequential Monte-Carlo sampling to estimate the posterior probability density. This is called the resampling step [30], as it samples a set of new particles based on a constructed posterior probability density function. The posterior probability density is often approximated by a discrete distribution, where the probability of the new particle is equal to any of the old particle values is proportional to the corresponding weight [31]. This reflects the decomposition of the posterior according to Bayes' theorem in equation 2.14. After resampling, the particles will be instances of the distribution

$$\mathbf{x}^i_{0|0} \sim \frac{p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{x}_0)}{p(\mathbf{y}_0)}, \tag{2.43}$$

because the particles are instances of the distribution

$$\mathbf{x}^i_{0|0} \sim w^i_0(\mathbf{x}^i_{0|-1}) \propto p(\mathbf{y}_0|\mathbf{x}^i_{0|-1})p(\mathbf{x}^i_{0|-1}) \propto \frac{p(\mathbf{y}_0|\mathbf{x}_0)p(\mathbf{x}_0)}{p(\mathbf{y}_0)}, \tag{2.44}$$

where the normalisation of the weights reflects the scaling $\frac{1}{p(\mathbf{y}_0)}$. This can be observed from the notion that $\sum_{i=1}^{M} w^i_0$ is an approximation of $p(\mathbf{y}_0)$, as $\sum_{i=1}^{M} w^i_0$ is a Riemann-sum that approximates the integral $\int p(\mathbf{y}_0, \mathbf{x}^i_{0|-1}) d\mathbf{x}^i_{0|-1}$ as M tends to infinity.

Resampling in the particle filter makes sure that less likely particles are discarded, and that more particles are generated in positions of the most likely particles. If resampling is performed when the weights are very similar, it can over time unnecessarily destroy information in the particle cloud [26]. A typical adaptation of the resampling step to avoid premature resampling of the particle cloud is so-called selective resampling [26]. Selective resampling is performed by only executing the resampling when the number of samples with a high particle weight becomes small. As the weight difference between the particles increases, a coefficient M_{eff} called the effective number of samples, defined as

$$M_{eff} = \frac{1}{\sum_{i=1}^{M}(w^i_{t|t})^2}, \tag{2.45}$$

will decrease [26]. Selective resampling is implemented by only resampling when the coefficient M_{eff} is smaller than some threshold $M_{eff,t}$ (typically chosen as some constant between 0 and 1 multiplied with the true number of particles M) [26].

After the particle filter measurement update, the prior density is estimated using Monte-Carlo sampling, according to the distribution

$$\mathbf{x}^i_{1|0} \sim p(\mathbf{x}^i_{1|0}|\mathbf{x}^i_{0|0}), \tag{2.46}$$

which is performed by simulating a white-noise process with the same distribution as $\mathbf{w}_0$, and evaluate the function

$$\mathbf{x}^i_{1|0} = \mathbf{f}(\mathbf{x}^i_{0|0}, \mathbf{u}_t, \mathbf{w}_0) \tag{2.47}$$

from the dynamic model in equation 2.3. Repeating these steps for the entire measurement sequence results in Algorithm 4.

Algorithm 4: A particle filter

Data: A measurement sequence $\mathbf{y}_{0:N}$ produced by the stochastic state-space model in equations 2.3 and 2.4, process noise $\mathbf{Q}_k$, and measurement noise $\mathbf{R}_k$. An initial probability density for the state $p(\mathbf{x}_0)$

Result: Estimates of the prior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\}_{t=0}^N$ and the posterior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t})\}_{t=0}^N$

for *t=0:N* **do**

Initialise the particle cloud by sampling $\mathbf{x}^i_{0|-1} \sim p(\mathbf{x}_0)$

1. Estimate the posterior distribution through the particle cloud by using Bayes theorem. Start by calculating the particle weights

$$w^i_t = p(\mathbf{y}_t|\mathbf{x}^i_{t|t-1}), \tag{2.48}$$

then normalise the particle weights so that $\sum_{i=1}^n w^i_t = 1$, and perform re-sampling of the particles according to

$$\mathbf{x}^i_{t|t} \sim w^i_t(\mathbf{x}^i_{t|t-1}). \tag{2.49}$$

Each particle is now a sample from the posterior density $\mathbf{x}^i_{t|t} \sim p(\mathbf{x}_t|\mathbf{y}_{0:t})$.

2. Estimate the prior distribution through the particle cloud by using Monte-Carlo samples of the noise $\mathbf{w}_t$ and calculate the effect on each particle according to

$$\mathbf{x}^i_{t+1|t} = \mathbf{f}(\mathbf{x}^i_{t|t}, \mathbf{u}_t, \mathbf{w}_t). \tag{2.50}$$

Each particle is now a sample from the prior density $\mathbf{x}^i_{t+1|t} \sim p(\mathbf{x}_{t+1}|\mathbf{y}_{0:t})$.

end

As the particle filter allows for prior and posterior probabilities that are non-Gaussian, it is normal to use the highest-weight particle before resampling as the posterior state estimate $\hat{\mathbf{x}}_{t|t}$ [32]. Picking the particle with the highest corresponding weight corresponds to picking the prior estimate of the previous step that has the highest likelihood given the last observed measurement. As the evaluation of the particle weights combined with the resampling transforms the particle cloud from an estimate of the prior density to an estimate of the posterior density, it will be referred to in this thesis as the "Particle filter measurement update". These steps are illustrated in the leftmost block of the diagram showing the Particle filter function in Figure 2.2.

The working mechanism of the particle filter can be summarised by the block diagram in Figure 2.2. Similarly to the linear Kalman filter in Figure 2.1, the prior particles are updated using the measurement to give the posterior particle cloud, and the posterior

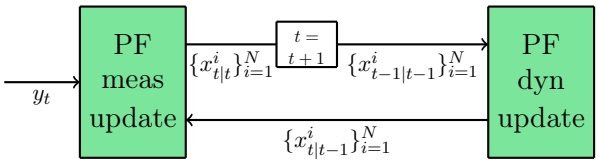


Figure 2.2: *Working mechanism of the particle filter when process and measurement noise are uncorrelated*

particles are then updated using the dynamic model of the system to give the new prior particle cloud.

A range of the research into pedestrian navigating in a known magnetic field uses a particle filter to estimate the pedestrians' position, see for example [33], [19], [25] and [28]. The particle state is usually given by the pedestrians pose [33], [19], [25]. The weights in the particle filter can be evaluated by comparing information about the magnetic field and its distribution in the particles' positions with the magnetic field measurement [33]. The function for evaluating the weights is a normal distribution with a known expected value and variance if the magnetic field map is estimated using Gaussian process regression [33].

Chapter 3

Gaussian process regression for magnetic field norm mapping

Gaussian process regression is a popular tool for estimating the indoor magnetic field for robot and pedestrian navigation ([13], [33], [2]). Gaussian process regression is generally regarded as powerful tools for accurate prediction given little knowledge about the nature of the function that is predicted [32]. They are, however, computationally demanding. The computational complexity grows cubically with the number of available magnetic field measurements [27]. By using Hilbert-space methods for reduced-rank Gaussian process regression to create a map of the magnetic field, the computational complexity can be reduced to a constant [27]. This chapter summarises how the magnetic field norm can be interpolated and predicted in new locations using Gaussian process regression with a squared exponential kernel. It also introduces how the predictions can be made computationally tractable by using Hilbert-space methods for Reduced Rank Gaussian process regression.

3.1 Gaussian process regression with a squared exponential kernel for magnetic field norm mapping

The magnetic field norm $\|\mathbf{H}(\mathbf{p})\|$ is a map from a three dimensional position to a one-dimensional field, $\|\mathbf{H}(\mathbf{p})\| : \mathbb{R}^3 \to \mathbb{R}$. Gaussian process regression is a maximum a posteriori estimate based on the assumption that all measurement data arises from a measurement model on the form

$$\begin{aligned} y &= f(\mathbf{x}) + v_k \\ v_k &\sim \mathcal{N}(0, \sigma_y^2) \\ f(\mathbf{x}) &\sim \mathcal{N}(0, \kappa(\cdot,\cdot)), \end{aligned} \tag{3.1}$$

where $y \in \mathbb{R}$ is the measurement, $x \in \mathbb{R}^m$ is a known input associated with the measurement, $\kappa(\cdot,\cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a known kernel function defining and $f : \mathbb{R}^m \to \mathbb{R}$ is an unknown linear function. Although the value of f is not know, it is an infinite-dimensional stochastic variable with a known distribution. The kernel function is a map from each input pair $\mathbf{x}_1, \mathbf{x}_2$ to the covariance between the corresponding function evaluations $f(\mathbf{x}_1), f(\mathbf{x}_2)$. A common choice for modeling magnetic field anomalies is the the squared exponential kernel function

$$\kappa_{SE}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_{SE}^2 \exp\left\{ \frac{-\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2l_{SE}^2} \right\}, \tag{3.2}$$

where σ_{SE} and l_{SE} are hyperparameters of the Gaussian process prior [34]. The shape of kernel function for all possible combinations of inputs from a piece of the real number line is illustrated in Figure 3.1. This figure shows that the kernel function is higher when the inputs are closer on the real number line. The squared exponential kernel causes prediction at a new point in input space to be more similar to magnetic field measurements in positions that are close.

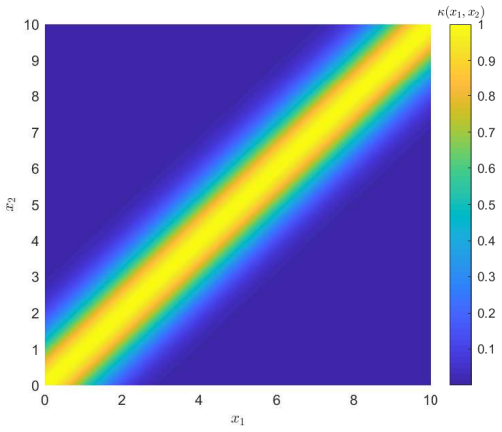


Figure 3.1: *The squared exponential kernel function, illustrated with a one-dimensional input vector as the input argument. The hyperparameters of the kernel function approximation is* $\sigma_{SE} = 1$ *and* $l_{SE} = 1$

For a set of observed measurements $\{y_i\}_{i=1}^N$ at positions $\{\mathbf{x}_i\}_{i=1}^N$, from here on denoted by the vectors $\mathbf{y}$ and $\mathbf{x}$ respectively, Gaussian process regression can be used to predict an unknown output y^* given a known input $\mathbf{x}^*$. The distribution of the stochastic variable y^* can be derived by taking the marginal distribution of the Gaussian process in equation 3.1. As the marginal distribution of a Gaussian is another Gaussian, it is uniquely defined by it's expected value and covariance

$$E[y^*] = \mathbf{K}(\mathbf{x}^*, \mathbf{x}_{1:N})(\mathbf{K}(\mathbf{x}_{1:N}, \mathbf{x}_{1:N}) + \sigma_y^2 I_N)^{-1}\mathbf{y}_{1:N}, \tag{3.3}$$

$$E[(y^* - E[y^*])^2] = \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) \tag{3.4}$$

$$- \mathbf{K}(\mathbf{x}^*, \mathbf{x}_{1:N})(\mathbf{K}(\mathbf{x}_{1:N}, \mathbf{x}_{1:N}) + \sigma_y^2 \mathbf{I}_N)^{-1}\mathbf{K}(\mathbf{x}_{1:N}, \mathbf{x}^*) + \sigma_y^2, \tag{3.5}$$

where $\mathbf{K}(\mathbf{x}_{1:N}, \mathbf{x}_{1:N})$ denotes the matrix of evaluations of the kernel function at all combinations of input points in the input vectors. The Gaussian process regression can be used to predict the magnetic field given a set of positions $\{\mathbf{p}_t\}_{t=1}^N$ as input values, and a corresponding set of magnetic field measurements $\{y_{m,t}\}_{t=1}^N$ as output values. The predicted magnetic field is then given by equation 3.3, and the variance is given by equation 3.5.

3.2 Reduced Rank Gaussian process regression with the squared exponential kernel using Hilbert space Methods

The matrix sizes involved in the predictive calculations in traditional Gaussian process regression grows with $O(N^2)$ with the number of observed samples, which means that the

required matrix inversion operations has a computational complexity of order $O(N^3)$ [27]. Reduced Rank Gaussian Process regression reduces the computational complexity of traditional Gaussian process regression. By approximating the Kernel function as a linear combination of Hilbert space basis functions, the computational complexity can be reduced to $O(N_m^3)$ [2].

The Gaussian process regression presented in the previous section aims to estimate a function $f : \mathbb{R}^n \to \mathbb{R}$ that is a realisation of the Gaussian process prior in equation 3.1. If we instead limit the problem to estimating a function $f : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^n$ is a compact domain, and where $f(\mathbf{x}) = 0$, $\forall \mathbf{x} \in \delta\Omega$, the assumed underlying Gaussian process regression for all $x \in \Omega$ can be rewritten as

$$f(\mathbf{x}) \sim \mathcal{N}(0, \kappa_{SE}(\cdot,\cdot)|f(\mathbf{x}) = 0, \forall \mathbf{x} \in \delta\Omega), \tag{3.6}$$

where $\delta\Omega$ is the edge of the domain. The utility of this assumption is that this Gaussian process regression is on a form that can be approximated using Hilbert space methods [27]. The Hilbert-space methods can be used to derive that the Gaussian process prior in equation 3.6 is equivalent to the Gaussian process prior

$$f(\mathbf{x}) \sim \mathcal{N}(0, \sum_{k=1}^{\infty} S_{SE}(\lambda_k)\phi_k(\mathbf{x}_1)\phi_k(\mathbf{x}_2)), \tag{3.7}$$

where the functions $\phi_k(\mathbf{x})$ are the ordered eigenfunctions of the negative Laplace operator, and $S_{SE}(\lambda_k)$ is the spectral density of the squared exponential kernel evaluated at the corresponding eigenvalues. In other words $\phi_k(\mathbf{x})$ are solutions to the equation

$$\begin{cases} -\nabla^2\phi_k(\mathbf{x}) = \lambda_k\phi_k(\mathbf{x}), & \mathbf{x} \in \Omega \\ \phi_k(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega \end{cases}, \tag{3.8}$$

ordered by decreasing magnitude of their corresponding eigenvalue λ_k [27]. S_{SE} is the Fourier transform of the squared exponential kernel [27]

$$S_{SE}(\omega) = \int k_{SE}(\mathbf{x}_1 - \mathbf{x}_2)e^{-i\omega^T(\mathbf{x}_1-\mathbf{x}_2)}d(\mathbf{x}_1 - \mathbf{x}_2), \tag{3.9}$$

which in three dimensions evaluates to

$$S_{SE}(\omega) = \sigma_{SE}^2\left(2\pi l_{SE}^2\right)^{\frac{3}{2}}\exp\left(-\frac{\omega^2 l_{SE}^2}{2}\right), \tag{3.10}$$

using the hyperparameters from equation 3.2 [35].

At points $\mathbf{x}_1$, $\mathbf{x}_2$ not too close to the domain edge $\delta\Omega$, the kernel function $k(\mathbf{x}_1, \mathbf{x}_2)$ can be approximated by a finite sum

$$k_{SE}(\mathbf{x}_1, \mathbf{x}_2) \approx \sum_{k=1}^{N_m} S(\lambda_k)\phi_k(\mathbf{x}_1)\phi_k(\mathbf{x}_2), \tag{3.11}$$

where the approximation becomes more accurate the more terms N_m are included in the series [27]. This convergence property of the kernel function is illustrated with an example domain Ω in one dimension in Figure 3.2. Similarly to the illustration in Figure 3.1, the kernel function approximation is evaluated for all combinations of inputs from the one-dimensional domain $\Omega = [0\,10]$. The eigenfunctions $\phi(\mathbf{x})$ are defined generally by equation 3.8, independent of the dimension of the state vector $\mathbf{x}$, and the chosen domain. The consequence is that for a one-dimensional state space vector, all possible domains will be some finite piece of the real number line. In two dimensions, arbitrary shapes can be chosen [36], as long as the domain is connected [37].

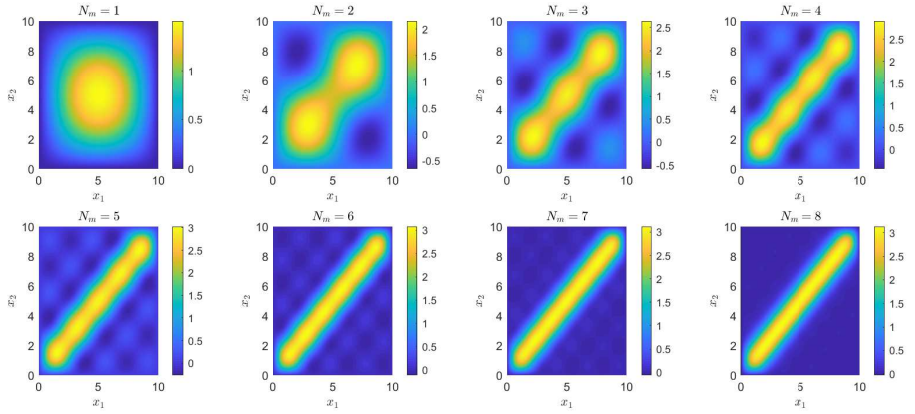


Figure 3.2: *The squared exponential kernel function approximation* $\sum_{k=1}^{N_m} S(\lambda_k)\phi_k(x_i)\phi_k(x_j)$ *approximating the kernel function in Figure 3.1. The kernel function is displayed with increasing number of basis functions* N_m*. The hyperparameters of the kernel function approximation is* $\sigma_{SE} = 1$ *and* $l_{SE} = 1$

When this approximation is used, the Gaussian process regression prior reduces to

$$f(\mathbf{x}) \sim \mathcal{N}(0, \sum_{k=1}^{N_m} S_{SE}(\lambda_k)\phi_k(\mathbf{x}_1)\phi_k(\mathbf{x}_2)),, \tag{3.12}$$

for inputs in the domain Ω. This is the prior for reduced-rank Gaussian process regression using Hilbert space basis functions. As the covariance matrix $\mathbf{K}$ is defined by

$$\mathbf{K}(\{\mathbf{x}_{1,i}\}_{i=1}^{n}, \{\mathbf{x}_{2,j}\}_{j=1}^{m}) = \begin{bmatrix} \kappa(\mathbf{x}_{1,1}, \mathbf{x}_{2,1}) & \cdots & \kappa(\mathbf{x}_{1,1}, \mathbf{x}_{2,m}) \\ \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_{1,n}, \mathbf{x}_{2,1}) & \cdots & \kappa(\mathbf{x}_{1,n}, \mathbf{x}_{2,m}) \end{bmatrix}, \tag{3.13}$$

it can be factorised as

$$\begin{aligned} &\mathbf{K}(\{\mathbf{x}_{1,i}\}_{i=1}^{n}, \{\mathbf{x}_{2,j}\}_{j=1}^{m}) \\ &= \begin{bmatrix} \sum_{k=1}^{N_m} S(\lambda_k)\phi_k(\mathbf{x}_{1,1})\phi_k(\mathbf{x}_{2,1}) & \cdots & \sum_{k=1}^{N_m} S(\lambda_k)\phi_k(\mathbf{x}_{1,1})\phi_k(\mathbf{x}_{2,m}) \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{N_m} S(\lambda_k)\phi_k(\mathbf{x}_{1,n})\phi_k(\mathbf{x}_{2,1}) & \cdots & \sum_{k=1}^{N_m} S(\lambda_k)\phi_k(\mathbf{x}_{1,n})\phi_k(\mathbf{x}_{2,m}) \end{bmatrix} \\ &= \begin{bmatrix} \phi_1(\mathbf{x}_{1,1}) & \cdots & \phi_{N_m}(\mathbf{x}_{1,1}) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_{1,n}) & \cdots & \phi_{N_m}(\mathbf{x}_{1,n}) \end{bmatrix} \begin{bmatrix} S(\lambda_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S(\lambda_{N_m}) \end{bmatrix} \begin{bmatrix} \phi_1(\mathbf{x}_{2,1}) & \cdots & \phi_1(\mathbf{x}_{2,m}) \\ \vdots & \ddots & \vdots \\ \phi_{N_m}(\mathbf{x}_{2,1}) & \cdots & \phi_{N_m}(\mathbf{x}_{2,m}) \end{bmatrix} \end{aligned}.$$

(3.14)

This matrix can be expressed more compactly by defining the operator

$$\mathbf{\Phi}(\mathbf{x}_{1:n}) = \mathbf{\Phi}(\{\mathbf{x}_i\}_{i=1}^n) = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_{N_m}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_n) & \cdots & \phi_{N_m}(\mathbf{x}_n) \end{bmatrix}, \tag{3.15}$$

and the matrix

$$\Lambda = \begin{bmatrix} S(\sqrt{\lambda_1}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S(\sqrt{\lambda_{N_m}}) \end{bmatrix}, \tag{3.16}$$

which results in

$$\mathbf{K}(\mathbf{x}_{1,1:n}, \mathbf{x}_{2,1:m}) = \mathbf{\Phi}(\mathbf{x}_{1,1:n})\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}_{2,1:m})^T \tag{3.17}$$

being a factorisation of the matrix $\mathbf{K}(x_{1,1:n}, x_{2,1:m})$. As the GP prediction is given by equations 3.3-3.5, inserting the approximation of $\mathbf{K}(x_{1,1:n}, x_{2,1:m})$ gives the following reduced-rank Gaussian process predictions

$$E[y^*] = \mathbf{\Phi}(\mathbf{x}^*)(\sigma_y^2\mathbf{\Lambda}^{-1} + \mathbf{\Phi}(\mathbf{x}_{1:N})^T\mathbf{\Phi}(\mathbf{x}_{1:N}))^{-1}\mathbf{\Phi}(\mathbf{x}_{1:N})^T\mathbf{y}_{1:N} \tag{3.18}$$

$$E[(y^* - E[y^*])^2] = \sigma_y^2\mathbf{\Phi}(\mathbf{x}^*)\left(\mathbf{\Phi}(\mathbf{x}_{1:N})^T\mathbf{\Phi}(\mathbf{x}_{1:N}) + \sigma_y^2\mathbf{\Lambda}^{-1}\right)^{-1}\mathbf{\Phi}(\mathbf{x}^*)^T + \sigma_y^2 \tag{3.19}$$

The reduced-rank Gaussian process regression can be expressed as a series of recursive Kalman filter measurement updates of a state $\mathbf{m}_t$ with a covariance $\mathbf{P}_m$ [2]. The magnetic field prediction in any given point is then given as

$$E[y^*] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{m}_t \tag{3.20}$$

$$E[(y^* - E[y^*])^2] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{P}_t\mathbf{\Phi}(\mathbf{x}^*)^T + \sigma_y^2 \tag{3.21}$$

These recursive Kalman filter update equations are described in Appendix A.2. The Kalman filter estimating the magnetic field map corresponds to the estimation of

$$\mathbf{m}_{t+1} = \mathbf{m}_t, \tag{3.22}$$

$$y_t = \mathbf{\Phi}(\mathbf{x}_t)\mathbf{m}_t + v_t, \qquad v_t \sim \mathcal{N}(0, \sigma_y^2), \tag{3.23}$$

by considering the combination of magnetic field basis functions as a state vector [2]. As the state vector of a linear stochastic model can be estimated with a Kalman filter, the magnetic field map state m and its covariance P_m can also be estimated with a Kalman filter (see section A.2 for a detailed description of the Kalman filter). To reconstruct the magnetic field map and its covariance from this linear state vector and its covariance, some

algebraic manipulation to the expressions in equations A.10-A.10 outlined in section A.2 gives that

$$E[y^*] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{m}_t \tag{3.24}$$

$$E[(y^* - E[y^*])^2] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{P}_t\mathbf{\Phi}(\mathbf{x}^*)^T + \sigma_y^2 \tag{3.25}$$

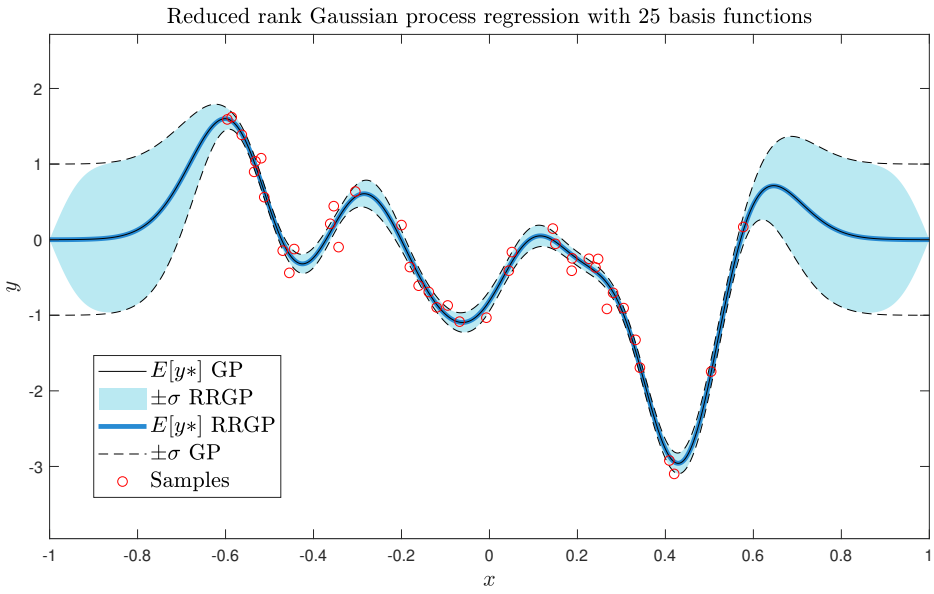
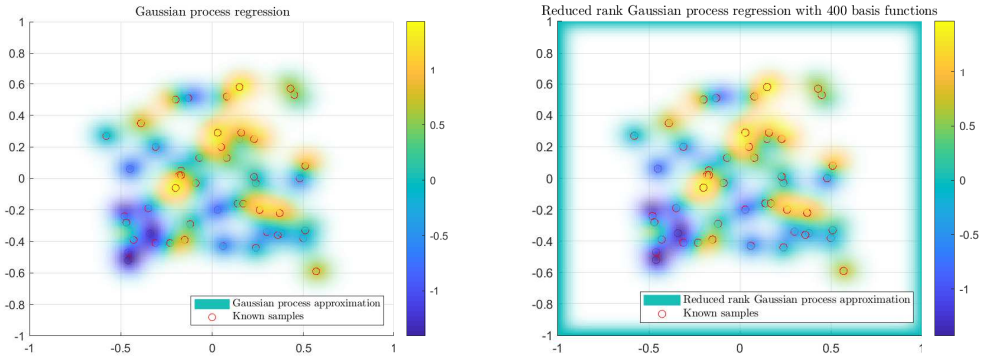


Figure 3.3: *A reproduction of an illustrative example of reduced rank Gaussian process regression estimating a function $f : \mathbb{R} \to \mathbb{R}$ similar to an illustration made by Solin and Särkkä [27]. The example shows the estimation of a nonlinear function sampled from a Gaussian process prior with a squared exponential function with parameters $\sigma_{SE}^2 = 1$, $l_{SE} = 0.1$, comparing the mean and covariance from a standard and a reduced-rank Gaussian process with 25 basis functions. The domain of the reduced rank Gaussian process regressions is defined as $\mathbf{\Omega} = [-1, 1]$.*



(a) Standard GP estimate

(b) Reduced-rank GP estimate

Figure 3.4: *Comparing the estimates of standard and reduced-rank Gaussian process regression estimating a function $f : \mathbb{R}^2 \to \mathbb{R}$ based on 100 samples drawn from a Gaussian prior defined by a squared exponential kernel with parameters $\sigma_{SE} = 1$ and $l_{SE} = 0.1$. The positions of the measurements in $\mathbb{R}^2$ are marked by red circles. The color corresponds to the estimated function value, and the opacity is inversely proportional with the standard deviation of the function value estimate at each location. The domain of the reduced rank Gaussian process regression here is $\mathbf{\Omega} = [-1, 1] \times [-1, 1]$.*

In Figures 3.3 and 3.4 simulation results comparing the estimated function value of a reduced-rank Gaussian process and a standard Gaussian process are displayed. The results

illustrate how the reduced rank Gaussian process estimate goes to zero at the boundaries, and the variance of the reduced rank GP also goes to zero near the boundaries. At the same time, in the middle of the domain, it approximates the result of the standard Gaussian process. As long as the domain is large enough, a reduced-rank Gaussian process may be used in place of a standard Gaussian process, which will reduce the computational complexity from $O(N^3)$ to $O(N_m^3)$ [2]. It is necessary to pick a large enough amount of basis functions to use a reduced-rank Gaussian process approximation with a squared exponential kernel to approximate the actual Gaussian process predictions successfully. The necessary number of basis functions depends on the size and dimensions of the region to be mapped and on the length scale parameter l_{SE} of the Gaussian process.

Chapter 4

Magnetic field simultaneous localisation and mapping

Magnetic field simultaneous localisation and mapping (Mag-SLAM) uses the magnetic field map to improve the position estimate, and the position estimates to build a magnetic field map [13]. The process is illustrated in Figure 4.1. The input called odometry in the top left corner of the figure is a measurement of the change in position. The trajectory estimate is used together with the magnetic field measurements to create a magnetic field map. The magnetic field map estimate is then used to improve the estimate of the position trajectory. As was explained in Chapter 2, a magnetic field map can be used to improve position trajectory estimates by using a particle filter. In chapter 3, it was explained how a magnetic field map can be estimated using Kalman filter measurement updates. It is a popular strategy in magnetic field simultaneous localisation and mapping to use a Rao-Blackwellized particle filter to perform both localisation and magnetic field mapping in one filter [13], [38], [2]. A Rao-Blackwellized particle filter is a combination of a particle filter and a Kalman filter. It uses a particle filter to estimate part of the states and uses a Kalman filter to estimate the remaining states for each particle. For the purposes of simultaneous localisation and mapping, the Rao-Blackwellized particle filter is usually applied by using a particle filter to estimate the trajectory and to estimate the magnetic field map for each particle with a Kalman filter [13], [38], [2]. In particular, [2] uses reduced-rank Gaussian process regression to estimate the magnetic field map's of each particle with a linear Kalman filter.

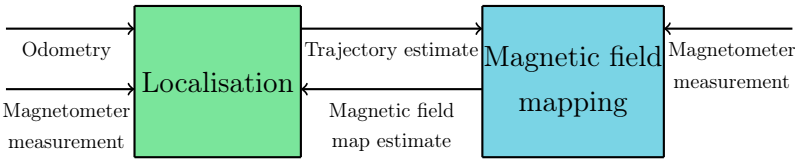


Figure 4.1: *Principle of magnetic field simultaneous localisation and mapping*

This chapter gives an introduction to the working mechanisms of two variants of the Rao-Blackwellized particle filter. Both of these filters are special-cases of filters described in [31]. The first filter uses a mathematical model on a diagonal form. The second uses a mathematical model on a triangular form. Following the convention in [31], the two filters are therefore referred to as the diagonal Rao-Blackwellized particle filter and the triangular Rao-Blackwellized particle filter respectively in the remainder of this thesis.

4.1 The diagonal Rao-Blackwellized particle filter

Rao-Blackwellized particle filters aim to increase the accuracy of an estimate by using the expected value of a state rather than the Monte Carlo estimate wherever it is possible [39]. This corresponds to using the linear Kalman filter to estimate the linear states, and a particle filter to estimate states with nonlinear dynamics [31]. The Rao-Blackwellized particle filter can be used to perform simultaneous localisation and magnetic field mapping using Gaussian process regression [13].

The Rao-Blackwellized particle filter presented in this section will estimate a state $\mathbf{x}_t$ that can be split into a linear and a nonlinear part according to

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_t^n \\ \mathbf{x}_t^l \end{bmatrix}, \tag{4.1}$$

where the states $\mathbf{x}_t^n$ encompasses all states that have nonlinear dynamics, or that affect the measurement nonlinearly. These states are estimated with nonlinear particle filter techniques. The states $\mathbf{x}_t^l$ describes the states that both have linear dynamics and that linearly affect the measurement. These states are estimated using a linear Kalman filter for each particle. The Rao-Blackwellized particle filter presented in this section can be applied to systems that have a model that can be expressed as

$$\mathbf{x}_{t+1}^n = \mathbf{f}_t(\mathbf{x}_t^n) + \mathbf{G}_t^n \mathbf{w}_t^n, \tag{4.2}$$
$$\mathbf{x}_{t+1}^l = \mathbf{F}_t \mathbf{x}_t^l + \mathbf{G}_t^l \mathbf{w}_t^l, \tag{4.3}$$
$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t^n) + \mathbf{H}_t \mathbf{x}_t^l + \mathbf{v}_t, \tag{4.4}$$

where the white noise processes $\mathbf{w}_t^n$, $\mathbf{w}_t^l$ arise from a white noise process with distribution

$$\begin{bmatrix} \mathbf{w}_t^n \\ \mathbf{w}_t^l \\ \mathbf{e}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{Q}_t^n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_t^l & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_t \end{bmatrix} \right), \tag{4.5}$$

where $\mathbf{Q}_t^n$, $\mathbf{Q}_t^l$ and $\mathbf{R}_t$ are known [31].

The prior and posterior probability density functions of the states in equations 4.2-4.4 can be estimated with the particle filter in Algorithm 4. If the state vector has many components; this solution quickly becomes computationally intractable, due to the large number of particles required to represent a many-dimensional probability density function [31]. This motivates the desire to use Kalman filter methods to estimate the linear part of the state vector, and Particle filter updates to estimate the nonlinear part. By using Bayes' theorem, the joint probability density is then simply the product of the linear and nonlinear probability density function

$$p(\begin{bmatrix} \mathbf{x}_t^n \\ \mathbf{x}_t^l \end{bmatrix} \mid \mathbf{y}_{1:t}) = p(\mathbf{x}_t^n \mid \mathbf{y}_{1:t}) p(\mathbf{x}_t^l \mid \mathbf{x}_t^n, \mathbf{y}_{1:t}) \tag{4.6}$$

The prior and posterior nonlinear state distributions can, in this case, be derived analytically in the same way as for the linear Kalman filter. To prior and posterior nonlinear

state distributions can be estimated using the particle filter. Joining these two methods gives the Rao-Blackwellized particle filter described in Algorithm 5.

Algorithm 5: The triangular Rao-Blackwellized particle filter

Data: A measurement sequence $\mathbf{y}_{0:N}$ produced by the stochastic state-space model in equations 2.3 and 2.4, process noise $\mathbf{Q}_k$, and measurement noise $\mathbf{R}_k$. An initial probability density for the state $p(\mathbf{x}_0)$

Result: Estimates of the prior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\}_{t=0}^N$ and the posterior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t})\}_{t=0}^N$ in the form of marginalized particle clouds, represented by a set of particles $\{\mathbf{x}_t^{i,n}\}_{i=1}^N$ with corresponding linear states $\{\hat{\mathbf{x}}_t^{i,l}\}_{i=1}^N$ and covariances for each linear state $\{\mathbf{P}_t^i\}_{i=1}^N$

for *t=0:N* **do**

Initialise the particle cloud by sampling $\mathbf{x}_{0|-1}^{i,n} \sim p(\mathbf{x}_0^n)$, and initalise all states and covariances to $\mathbf{x}_{0|-1}^{i,l} = \mathbf{x}_0$, $\mathbf{P}_{0|-1}^i = \mathbf{P}_0$

1. **PF measurement update:** Evaluate the particle weights

$$\begin{aligned} w_t^i &= p(\mathbf{y}_t|\mathbf{x}_{t|t-1}^{i,n}, \hat{\mathbf{x}}_{t|t-1}^{i,l}, \mathbf{P}_{t|t-1}^i) \\ &= \mathcal{N}(\mathbf{h}_t(\mathbf{x}_{t|t-1}^{i,n}) + \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}, \mathbf{H}_t\mathbf{P}_{t|t-1}^i\mathbf{H}_t^T + \mathbf{R}_t)\Big|_{\mathbf{y}_t}, \end{aligned} \tag{4.7}$$

   then normalise the particle weights so that $\sum_{i=1}^n w_t^i = 1$. Perform re-sampling of the particles according to equation 2.49.

2. **KF measurement update:** For each particle, calculate the posterior state $\hat{\mathbf{x}}_{t|t}^{i,l}$ and covariances $\mathbf{P}_{t|t}^i$ using

$$\mathbf{S}_t^i = \mathbf{R}_t + \mathbf{H}_t\mathbf{P}_{t|t-1}^i\mathbf{H}_t^T) \tag{4.8}$$
$$\mathbf{K}_t^i = \mathbf{P}_{t|t-1}^i\mathbf{H}_t^T(\mathbf{S}_t^i)^{-1} \tag{4.9}$$
$$\hat{\mathbf{x}}_{t|t}^{i,l} = \hat{\mathbf{x}}_{t|t-1}^{i,l} + \mathbf{K}_t^i(\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_{t|t}^{i,n}) - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}^{i,l}) \tag{4.10}$$
$$\mathbf{P}_{t|t}^i = \mathbf{P}_{t|t-1}^i - \mathbf{K}_t^i\mathbf{H}_t\mathbf{P}_{t|t-1}^i \tag{4.11}$$

   The state and covariance uniquely describes the Gaussian marginal posterior distribution $p(\mathbf{x}_t^l|\mathbf{y}_{0:t}, \mathbf{x}_{1:t}^{n,i}) = \mathcal{N}(\hat{\mathbf{x}}_{t|t}^{i,l}, \mathbf{P}_{t|t}^i)\Big|_{\mathbf{x}_t^l}$

3. **PF dynamic update:** Calculate the prior particle cloud particle cloud $\{\hat{\mathbf{x}}_{t+1|t}^{i,l}\}_{i=1}^N$ by using Monte-Carlo samples of the noise $\mathbf{w}_t^{i,n} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t^n)$ and calculate the effect on each particle according to equation 4.2. The prior particle cloud are then samples from the distribution $\mathbf{x}_{t+1|t}^{i,n} \sim p(\mathbf{x}_{t+1}^n|\mathbf{y}_{0:t})$

4. **KF dynamic update:** For each particle, calculate the prior state $\hat{\mathbf{x}}_{t+1|t}^{i,l}$ and covariance $\mathbf{P}_{t+1|t}^{i,n}$ using

$$\hat{\mathbf{x}}_{t+1|t}^{i,l} = \mathbf{F}_t\hat{\mathbf{x}}_{t|t}^{i,l} \tag{4.12}$$
$$\mathbf{P}_{t+1|t}^i = \mathbf{F}_t\mathbf{P}_{t|t}^i\mathbf{F}_t^T + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^T. \tag{4.13}$$

   The prior state and covariance uniquely describes the Gaussian marginal prior probability density $p(\mathbf{x}_{t+1}^l|\mathbf{y}_{0:t}, \mathbf{x}_{0:t}^{i,n}) = \mathcal{N}(\hat{\mathbf{x}}_{t+1|t}^{i,l}, \mathbf{P}_{t+1|t}^i)\Big|_{\mathbf{x}_{t+1}^l}$

end

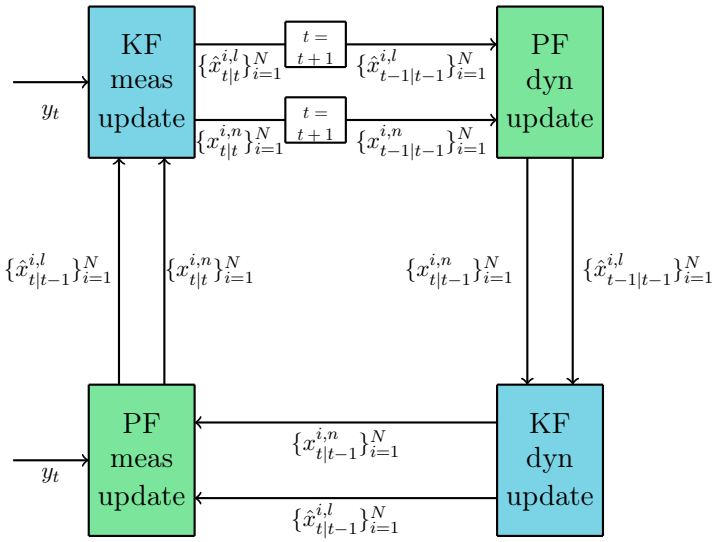


Figure 4.2: *The triangular Rao-Blackwellized particle filter*

The working mechanism of the Rao-Blackwellized particle filter is illustrated in Figure 4.2. The illustration shows the interplay between the particle filter and the linear Kalman filter component of the Rao-Blackwellized particle filter. The Rao-Blackwellized particle filter uses the particle filter estimate as a parameter in the linear Kalman filter updates, and the linear Kalman filter estimates as parameters in the particle filter updates. Magnetic field simultaneous localisation and mapping uses the position estimate to create a map of the indoor magnetic field anomalies and uses the map of the indoor magnetic field anomalies in the estimation of the position. For indoor pedestrian simultaneous localisation and mapping using the magnetic field, the Rao-Blackwellized particle filter has previously been used by letting the particle filter estimate the position and orientation as the nonlinear states [2]. The magnetic field map was estimated in [2] as conditional linear states given the position and orientation of each particle. The use of the map as the linear state is only possible when the magnetic field map can indeed be estimated with a linear model. The magnetic field map can be estimated with recursive Bayesian estimation when the reduced-rank Gaussian process regression introduced in the previous chapter is used to create a map of the magnetic field [27].

4.2 The triangular Rao-Blackwellized particle filter

Bayes' theorem can be used to derive combinations of the Rao-Blackwellized particle filter that apply to much more general models compared to the model in equations 4.2-4.4 [31]. For this thesis, it is desirable to consider a model that also includes the possibility of the linear state being a (linear) argument to the dynamic equation of the nonlinear state

$$\mathbf{x}^n_{t+1} = \mathbf{f}_t(\mathbf{x}^n_t) + \mathbf{F}^n_t \mathbf{x}^l_t + \mathbf{w}^n_t \tag{4.14}$$

$$\mathbf{x}^l_{t+1} = \mathbf{F}^l_t \mathbf{x}^l_t + \mathbf{G}_t \mathbf{w}^l_t \tag{4.15}$$

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t) + \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t \tag{4.16}$$

Augmenting the model in this way means that the prior and posterior values of a particle $\mathbf{x}^{i,n}_{t+1|t}$ and $\mathbf{x}^{i,n}_{t|t}$ can be combined in equation 4.4 to give a perceived measurement of the state $\mathbf{x}^{i,l}_t$ [31]. This additional information can be included by an extra Kalman filter measurement update, after the Kalman filter dynamic update in step 4 in Algorithm 5 [26]. In addition, the state estimate $\mathbf{x}^{i,l}_{t|t}$ is considered an input noise with known distribution

$\mathcal{N}(\mathbf{x}_{t|t}^{i,l}, \mathbf{P}_{t|t}^{i,l})$ in step three of Algorithm 5. This means that in addition to using Monte-Carlo samples of the noise $\mathbf{w}_t^n \sim \mathcal{N}(0, \mathbf{Q}_t^n)$, the particle filter dynamic update step uses Monte-Carlo samples of the linear state $\mathbf{x}_t^{i,l} = \mathcal{N}(\mathbf{x}_{t|t}^{i,l}, \mathbf{P}_{t|t}^{i,l})$. The full particle filter algorithm including this generalisation is defined in Appendix A.5. This filter is a special case of the triangular Rao-Blackwellized particle filter described in [31].

4.3 Rao-Blackwellised particle filters applied to Mag-SLAM

Previous research applies the diagonal Rao-Blacwellised particle filters using a range of magnetic field mapping techniques and sources of position odometry ([40], [38], [2]).

The magnetic field map is generally stored on a format that makes it appropriate to estimate it with Kalman filter updates [38], [40], [2]. The magnetic field map, therefore, is estimated as the linear state $\mathbf{x}^l$ in the Rao-Blackwellized particle filter. The specific implementations of this linear update vary between different sources. The variant that is used in this thesis most closely resembles the version in [2], where the magnetic field map is also created using reduced-rank Gaussian process regression. An important difference between the implementation in this thesis and the one in [2] is that this thesis only makes a map of the magnetic field norm, while [2] makes a map of the magnetic field potential, and estimates all three components of the magnetic field as the gradient of this potential. This approach is based on knowledge about the physics of the magnetic field described in detail in [34]. The magnetic field reduced rank Gaussian process regression can, in both cases, be performed recursively as linear Kalman filter measurement updates. The Kalman filter measurement updates estimate a magnetic field state vector with dimension equal to the number of basis functions. This vector is equal to the linear states in the Rao-Blackwellized particle filter in [2]. In [38], the linear states are defined as the three components of the magnetic field in the corners of a grid. The values in the corners are updated with Kalman-filter measurement updates when a magnetic field measurement is made in a square tile adjacent to a corner [38].

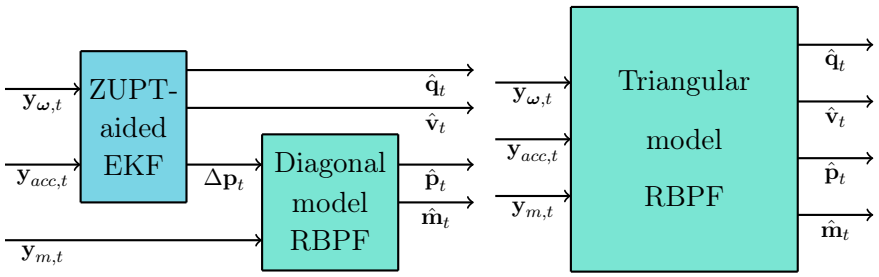
Both [38] and [40] uses position and orientation of a robot as the nonlinear states $\mathbf{x}^n$ in a Rao-Blackwellized particle filter to perform Mag-SLAM.In [2], odometry from a phone's accelerometer, gyroscope and camera are used to estimate pedestrian position with Mag-SLAM. In [2], the diagonal Rao-Blackwellized particle filter in Algorithm 5 is used to estimate the position and orientation as the nonlinear states. The researchers evaluate the weights of the particles by comparing the gradient of the magnetic field potential with the measured magnetic field in body frame. Because the orientation of the sensor affects the magnetic field measurement, the likelihood of the orientation can be evaluated from the magnetic field measurement. If only a magnetic field norm map is used, the orientation of the particle will not influence the weight, as it is not an argument to the measurement function.

Chapter 5

Methods

In this chapter, the previously proposed ZUPT-aided EKF is summarised, and two algorithms for magnetic field simultaneous localisation and mapping on a foot-mounted sensor are proposed. The first algorithm estimates the position of the foot-mounted sensor using a loosely coupled sensor fusion algorithm. The second algorithm estimates the position with a tightly coupled sensor fusion algorithm.

The first algorithm proposed in this thesis applies the diagonal Rao-Blackwellized particle filter in a way that closely resembles previous research using a Rao-Blackwellized particle filter for Mag-SLAM ([2], [40], [38]). The second algorithm proposed in this thesis applies the triangular Rao-Blackwellized particle filter to perform Mag-SLAM on the foot-mounted sensor. The next chapter is devoted to describing these algorithms.



(a) Loosely coupled algorithm (b) Tightly coupled algorithm

Figure 5.1: *Simultaneous localisation of foot-mounted sensor and magnetic field mapping*

Loosely coupled sensor fusion is the practice of using multiple filters in cascade or parallel to estimate the state. In contrast, tightly coupled sensor fusion aims to use a single filter on data that is minimally processed [41]. The loosely coupled algorithm presented in this section uses the ZUPT-aided EKF implementation by [23] to calculate position odometry and then uses the diagonal Rao-Blackwellized particle filter to perform Mag-SLAM. The tightly coupled algorithm is designed by viewing all the models of the foot-mounted sensor and magnetic field used by [23] and [2] as a single state-space model, with a state that can be estimated with a single filter. The full state-space model is a version of the triangular the Rao-Blackwellized particle filter in section 4.2. The tightly coupled sensor algorithm applies Algorithm 10 to measurements from the foot-mounted sensor directly, and from that information producing a magnetic field map and position estimate. Both algorithms use reduced-rank Gaussian process regression for the magnetic field maps in cubical domains. The analytic solutions for the basis functions for a cubical domain centred at the origin are given in [27]. In Appendix A.6, it is explained how the analytic basis functions can be shifted to an arbitrary cubical domain, and the ordering of the analytic basis functions and eigenvalues are discussed.

In this chapter, both algorithms will be presented in detail. The first algorithm has the advantage that it is more modular. Hence, the Mag-SLAM filter presented can also be applied to position odometry from other sources. On the other hand, tightly coupled sensor fusion typically gives more accurate state estimates [41]. The second algorithm has the advantage of giving position estimates that more closely resemble realistic movement of the foot, as is demonstrated by the experimental results in chapter 6.

5.1 The zero-velocity aided extended Kalman filter

The zero-velocity aided extended Kalman filter was proposed by [1], and later implemented as an open-source project by [23]. The filter is a version of the extended Kalman filter presented in section 2.3. This section describes the model [23] used to derive the zero-velocity aided extended Kalman filter for indoor localisation.

The first-order discretisation of the foot-mounted sensor the zero-velocity aided extended Kalman filter in [23] is based on a kinematic motion model in the navigation frame. The model can be expressed as

$$\dot{\mathbf{p}}^w(t) = \mathbf{v}^w(t) \tag{5.1}$$

$$\dot{\mathbf{v}}^w(t) = \left(\mathbf{R}(\mathbf{q}^{wb}(t))\mathbf{f}^b(t) - \mathbf{g}^w\right) \tag{5.2}$$

$$\dot{\mathbf{q}}^{wb}(t) = \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}(t) \end{bmatrix} \odot \mathbf{q}^{wb}(t), \tag{5.3}$$

where $\mathbf{p}^w$ is the position displacement from the initial position in world frame w, $\mathbf{v}^w$ is the velocity, $\mathbf{q}^{wb}(t)$ is the unit quaternion encoding the rotation from world frame to the body frame b, $\mathbf{R}(\cdot)$ is the operator mapping a rotation represented by a unit quaternion to the corresponding rotation matrix, $\mathbf{g}^w$ is the gravity vector in world frame, $\boldsymbol{\omega}$ is the rotational velocity, and $\mathbf{f}^b(t)$ is the specific acceleration [42]. A first-order discretization of this equation gives [9]

$$\begin{bmatrix} \mathbf{p}_t^w \\ \mathbf{v}_t^w \\ \mathbf{q}^{wb}{}_t \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{t-1}^w + T\mathbf{v}_{t-1}^w \\ \mathbf{v}_{t-1}^w + T(\mathbf{R}(\mathbf{q}_{t-1}^{wb})\mathbf{f}_t^b - \mathbf{g}^w) \\ \mathbf{q}_{t-1}^{wb} \odot \exp_q(\frac{\boldsymbol{\omega}_t T}{2}) \end{bmatrix}. \tag{5.4}$$

See page 22 in [43] for a definition of the $\exp_q()$ operator, and the chapter 3.2 in [43] for a detailed introduction to the rotation parametrisation notation used in this thesis. The gyroscope measurement model used in [9] is defined as

$$\mathbf{y}_{\boldsymbol{\omega},t}^b = \boldsymbol{\omega}_t + \mathbf{e}_{\boldsymbol{\omega},t}, \tag{5.5}$$

where $\mathbf{y}_{\boldsymbol{\omega},t}^b$ is the gyroscope measurement, $\boldsymbol{\omega}_t$ is the angular velocity, $\mathbf{e}_{\boldsymbol{\omega},t}$ is an additive measurement noise arising from the white noise process $\mathbf{e}_{\boldsymbol{\omega},t} \sim \mathcal{N}(0, \mathbf{R}_{\boldsymbol{\omega},t})$. The accelerometer measurement model is defined as

$$\mathbf{y}_{acc,t}^b = \mathbf{a}_t^b - \mathbf{g}^b + \mathbf{e}_{acc,t}, \tag{5.6}$$

where $\mathbf{y}_{acc,t}^b$ is the accelerometer measurement, $\boldsymbol{a}_t$ is the acceleration of the sensor, $\mathbf{e}_{\boldsymbol{\omega},t}$ is an additive measurement noise arising from the white noise process $\mathbf{e}_{acc,t} \sim \mathcal{N}(0, \mathbf{R}_{acc,t})$.

Inserting the measurement functions into the motion model gives

$$\begin{bmatrix} \mathbf{p}_t^w \\ \mathbf{v}_t^w \\ \mathbf{q}^{wb}{}_t \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{t-1}^w + T\mathbf{v}_{t-1}^w \\ \mathbf{v}_{t-1}^w + T(\mathbf{R}(\mathbf{q}_{t-1}^{wb})(\mathbf{y}_{acc,t}^b - \mathbf{e}_{acc,t}) - \mathbf{g}^w) \\ \mathbf{q}_{t-1}^{wb} \odot \exp_q(\frac{(\mathbf{y}_{\boldsymbol{\omega},t}^b - \mathbf{e}_{\boldsymbol{\omega},t})T}{2}) \end{bmatrix}, \tag{5.7}$$

which is a nonlinear process on the same form as the dynamic model in equation 2.3, if we consider the position $\mathbf{p}_t^w$, the velocity $\mathbf{v}_t^w$ and the orientation $\mathbf{q}_t^{wb}$ as the state vector $\mathbf{x}_t$, and the measurements $\mathbf{y}_{acc,t}^b$ and $\mathbf{y}_{\boldsymbol{\omega},t}^b$ as the input vector $\mathbf{u}_t$, and the measurement noise $\mathbf{e}_{\boldsymbol{\omega},t}$ and $\mathbf{e}_{acc,t}$ as the process noise. This model is therefore used as the dynamic model in the extended Kalman filter.

The measurement model for the zero-velocity aided EKF depends on the activation of the zero-velocity detector. If the detector is inactive, it is treated as if no measurement is available. If the detector is active, the measurement model is defined as

$$\mathbf{y}_{v,t}^w = \mathbf{0} = \mathbf{v}_t^w + \mathbf{e}_{\mathbf{v},t}, \tag{5.8}$$

that is equal to zero and affected by a measurement noise from the distribution $\mathbf{e}_{\mathbf{v},t} \sim \mathcal{N}(0, \mathbf{R}_{\mathbf{v},t})$. The zero-velocity aided extended Kalman filter is given by applying the extended Kalman filter described in Algorithm 3 on the state-space model given by equation 5.7 when the zero velocity detector is inactive, and on the state-space model given by equations 5.7 and 5.8 when the zero velocity detector is active. The details of the algorithm can be found in section A.3.

5.2 Loosely coupled Mag-SLAM for foot-mounted sensor

The loosely coupled approach to Mag-SLAM for the foot-mounted sensor is inspired by implementations in [2] and [38]. The zero-velocity aided EKF gives estimates of position, orientation and velocity [9]. The magnetic field norm is a function of position only, meaning that only an estimate of the particles' positions is necessary to meaningfully perform a particle filter measurement update based on the magnetic field norm measurement. It is possible to use the position estimate from the ZUPT-EKF to calculate the estimated perturbation in position from each timestep to the next. The variance of the position estimate from the ZUPT-aided EKF has been shown to increase linearly in time [23]. The variance from integrating odometry with an additive white noise will also increase linearly in time [7]. This motivates the choice to model the estimated position displacement $\Delta\hat{\mathbf{p}}_t^{\mathrm{w}}$ as a sum of a true position displacements $\Delta\mathbf{p}_t^w = \mathbf{p}_t^{\mathrm{w}} - \mathbf{p}_{t-1}^{\mathrm{w}}$ and an additive white noise $\mathbf{w}_t$. The model is given on recursive form in equation 5.9. Combining this dynamic odometry model with the magnetic field linear stochastic model presented in Appendix A.2 gives the full state-space model for the loosely coupled Mag-SLAM

$$\mathbf{p}_t^{\mathrm{w}} = \mathbf{p}_{t-1}^{\mathrm{w}} + \Delta\hat{\mathbf{p}}_t^{\mathrm{w}} + \mathbf{w}_t, \tag{5.9}$$

$$\mathbf{m}_{t+1} = \mathbf{m}_t, \tag{5.10}$$

$$y_{m,t} = \boldsymbol{\Phi}(\mathbf{p}_t)\mathbf{m}_t + e_{m,t}, \tag{5.11}$$

where $\mathbf{p}_t^{\mathrm{w}}$ is the position, $\mathbf{m}_{t+1}$ is the magnetic field vector, $y_{m,t}$ is the magnetic field norm measurement, and the noise has joint distribution

$$\begin{bmatrix} \mathbf{w}_t \\ e_{m,t} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{Q}_t & \mathbf{0} \\ \mathbf{0} & \sigma_y^2 \end{bmatrix}), \tag{5.12}$$

where $\mathbf{Q}_t$ and σ_y^2 are assumed to be known. A similar odometry model has previously been used on measurements from a smartphone, for simultaneous localisation and mapping of all three components of a three-dimensional magnetic field [2]. The approach presented in this thesis does not gain the benefit in accuracy and heading drift correction obtained from an approach that models all three components of the magnetic field [2]. On the other hand, it is less computationally demanding, as the nonlinear state only has three dimensions (to describe the position) as opposed to seven (to describe the position and orientation using a quaternion). The combined structure of the ZUPT-aided extended Kalman filter, and the Rao-Blackwellized particle filter is displayed in figure 5.1 (a).

The RBPF in algorithm 5 can be used as it is in this case, as the model in equations 5.9-5.11 is a special case of the model in equations 4.2-4.4. The resulting loosely coupled Mag-SLAM method is described in algorithm 6.

Algorithm 6: Loosely coupled Mag-SLAM on foot-mounted sensor

Data: Noisy position odometry $\{\Delta\hat{\mathbf{p}}_t^{\mathrm{w}}\}_{t=0}^N$, a sequence of magnetic field measurements $\{y_{m,t}\}_{t=0}^N$

Result: Estimates of the prior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\}_{t=0}^N$ and the posterior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t})\}_{t=0}^N$ in the form of marginalized particle clouds, represented by a set of particles representing the positions $\{\hat{\mathbf{p}}_t^{\mathrm{w},i}\}_{i=1}^N$ with corresponding linear states $\{\hat{\mathbf{m}}_t^i\}_{i=1}^M$ and covariances for each linear state $\{\mathbf{P}_t^i\}_{i=1}^M$ representing the magnetic field map of each particle.

for *t=0:N* **do**

Initialise the particle cloud by setting $\mathbf{p}_{0|-1}^{w,i} = \mathbf{0}$, $\mathbf{m}_{0|-1}^i = \mathbf{0}$, $\mathbf{P}_{0|-1}^i = \Lambda$

1. **PF measurement update:** Use the evaluation of the particle weights according to

$$\begin{aligned} w_t^i &= p(y_{m,t}|\hat{\mathbf{p}}_{t|t-1}^{\mathrm{w},i}, \hat{\mathbf{m}}_{t|t-1}^i, \mathbf{P}_{t|t-1}^i) w_{t-1}^i \\ &= \mathcal{N}(y_{m,t}; \Phi(\mathbf{p}_{t|t-1}^{\mathrm{w},i})\mathbf{m}_{t|t-1}^i, \sigma_y^2), \end{aligned} \tag{5.13}$$

   then normalise the particle weights so that $\sum_{i=1}^M w_t^i = 1$. Perform selective resampling of the particles according to equation 2.49, with criterium $M_{eff} < \frac{2}{3}M$. If particles are resampled, set weights $w_t^i = \frac{1}{M}$.

2. **KF measurement update:** For each particle, calculate the posterior state $\hat{\mathbf{m}}_{t|t}^i$ and covariances $\mathbf{P}_{t|t}^i$ using

$$\begin{aligned} \mathbf{S}_t^i &= \Phi(\mathbf{p}_{t|t}^i)\mathbf{P}_{t|t-1}^i\Phi(\mathbf{p}_{t|t}^{w,i})^T + \sigma_y^2 \\ \mathbf{K}_t^i &= \mathbf{P}_{t|t-1}\Phi(\mathbf{p}_{t|t}^{\mathrm{w},i})^T\mathbf{S}_t^{iT} \\ \hat{\mathbf{m}}_{t|t}^i &= \hat{\mathbf{m}}_{t|t-1}^i + \mathbf{K}_t^{\mathrm{w},i}(\mathbf{y}_{m,t} - \Phi(\mathbf{p}_{t|t}^i)\hat{\mathbf{m}}_{t|t-1}^i) \\ \mathbf{P}_{t|t}^i &= \mathbf{P}_{t|t-1}^i - \mathbf{K}_t^i\Phi(\mathbf{p}_{t|t}^i)\mathbf{P}_{t|t-1}^i \end{aligned} \tag{5.14}$$

   The state and covariance uniquely describes the Gaussian marginal posterior distribution $p(\mathbf{m}_t|y_{m,0:t}, \mathbf{p}_{1:t}^{\mathrm{w},i}) = \mathcal{N}(\hat{\mathbf{m}}_{t|t}^i, \mathbf{P}_{t|t}^{w,i})$

3. **PF dynamic update:** Calculate the prior particles $\{\hat{\mathbf{p}}_{t+1|t}^{\mathrm{w},i}\}_{i=1}^N$ by using Monte-Carlo samples of the noise $\hat{\mathbf{w}}_t^i \sim \mathcal{N}(0, \mathbf{Q}_t)$ and calculate the effect on each particle according to equation 5.9. The prior particle cloud are then samples from the distribution $\hat{\mathbf{p}}_{t+1|t}^{\mathrm{w}} \sim p(\mathbf{p}_{t+1}^{\mathrm{w}}|y_{m,0:t})$.

end

The Kalman filter dynamic update has been omitted because the map is assumed constant [35]. Equation 5.10 shows that the magnetic field is assumed to be stationary, making the dynamic matrix $\mathbf{F}_t$ in step 2 in algorithm 2 equal to the identity matrix, and the process noise term $\mathbf{G}^T\mathbf{Q}_t\mathbf{G}$ in the same equation equal to zero. cancelling out the effects of the Kalman filter dynamic update on the state and covariance.

As the target of the particle filter is to do simultaneous localisation and mapping, it does not make any difference where the first position of the pedestrian is on the map. Any set of position and map estimates that are shifted relative to each other from the starting point

will be equivalent, as the map is only defined relative to the position of the pedestrian, and the position of the pedestrian is only defined relative to the map. All the initial position particles of the pedestrian are therefore defined to be $\mathbf{p}_0^{\mathrm{w},i} = \mathbf{0}$.

The reduced-rank Gaussian process regression gives estimates of the expected value and variance of the magnetic field measurements in the location of the particle. If the particle has position $\mathbf{p}_{t|t-1}^{\mathrm{w},i}$, then the estimated magnetic field in this position is given by

$$\mu_{i,t} = E[y_{m,t}] = \mathbf{\Phi}(\mathbf{p}_{t|t-1}^{\mathrm{w},i})\mathbf{m}_t^i, \tag{5.15}$$

$$\sigma_{i,t}^2 = E[(y_{m,t} - E[y_{m,t}])^2] = \mathbf{\Phi}(\mathbf{p}_{t|t-1}^{\mathrm{w},i})\mathbf{P}_t^i\mathbf{\Phi}(\mathbf{p}_{t|t-1}^{\mathrm{w},i})^T + \sigma_y^2, \tag{5.16}$$

where m_t^i, P_t^i are the linear states estimated by the Kalman filter formulation of the reduced-rank Gaussian process regression for particle i at time t. What this implies is that each particle estimates a magnetic field map, and evaluates the likelihood of future observations based on this map [2]. Based on the prediction, the probability density of the observed measurement can be evaluated from a Gaussian distribution

$$p(y_{m,t}|\mathbf{p}_{1:t}^{\mathrm{w},i}, y_{m,1:t}) = \frac{1}{(\sqrt{2\pi})\sigma_{i,t}} \exp\left(-\frac{(y_{m,t} - \mu_{i,t})^2}{2\sigma_{i,t}^2}\right), \tag{5.17}$$

where $\mathbf{p}_{1:t}^{\mathrm{w},i}$ is the position trajectory of the particle, and $y_{m,1:t}$ is the history of magnetic field measurements that were used to build the magnetic field map. This probability is proportional to the likelihood of the measurements. Evaluating the probability at the position of each particle gives the weight of each particle.

5.3 Tightly coupled Mag-SLAM for foot-mounted sensor

The zero-velocity assumption was previously integrated using an extended Kalman filter [1], while magnetic field SLAM has been used as part of a Rao-Blackwellized particle filter. This section combines the models used by both approaches and applies a triangular Rao-Blackwellized particle filter to perform tightly coupled sensor fusion of the available measurements. The tightly coupled algorithm uses the accelerometer, magnetometer and gyroscope measurements and gives estimates of the position $\mathbf{p}_t^w$, the velocity $\mathbf{v}_t^w$, the orientation $\mathbf{q}_t^{w,i}$ and the magnetic field map represented by its state vector $\mathbf{m}_t$. The interface of the filter is illustrated in figure 5.1 (b).

By assuming that the magnetic field measurement noise is uncorrelated with the accelerometer and gyroscope measurement noise, the joint nonlinear stochastic state-space model for the sensor dynamics, the pseudo zero-velocity measurement and the magnetic field representations become

$$\mathbf{p}_t^w = \mathbf{p}_{t-1}^w + T\mathbf{v}_{t-1}^w, \tag{5.18}$$

$$\mathbf{v}_t^w = \mathbf{v}_{t-1}^w + T(\mathbf{R}(\mathbf{q}_{t-1}^{wb})(\mathbf{y}_{acc,t}^b - \mathbf{e}_{acc,t}) - \mathbf{g}^w), \tag{5.19}$$

$$\mathbf{q}^{wb}{}_t = \mathbf{q}_{t-1}^{wb} \odot \exp_q(\frac{(\mathbf{y}_{\boldsymbol{\omega},t}^b - \mathbf{e}_{\boldsymbol{\omega},t})T}{2}), \tag{5.20}$$

$$\mathbf{m}_{t+1} = \mathbf{m}_t, \tag{5.21}$$

$$\mathbf{y}_{\mathbf{v},t} = \mathbf{v}_t^w + \mathbf{e}_{\mathbf{v},t} \tag{5.22}$$

$$y_{\mathbf{m},t} = \mathbf{\Phi}(\mathbf{p}_t^w)\mathbf{m}_t + e_{\mathbf{m},t}, \tag{5.23}$$

with a joint noise distribution

$$\begin{bmatrix} \mathbf{e}_{acc,t} \\ \mathbf{e}_{\boldsymbol{\omega},t} \\ \mathbf{e}_{v,t} \\ e_{m,t} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{R}_{acc,t} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\boldsymbol{\omega},t} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{v,t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma^2_{m,t} \end{bmatrix} \right), \tag{5.24}$$

where $\mathbf{R}_{acc,t}$, $\mathbf{R}_{\boldsymbol{\omega},t}$, $\mathbf{R}_{v,t}$ and $\sigma^2_{m,t}$ are assumed to be known quantities.

The zero-velocity aided extended Kalman filter as implemented in [23] estimates position, orientation and velocity. As was explained in section 2.3, the extended Kalman filter applies a linear Kalman filter to the linearised state-space model. The linearisation is performed around an orientation deviation state $\boldsymbol{\eta}^w_t$. The orientation deviation is defined as the difference between the prior estimate and the true orientation [43]

$$\mathbf{q}^{wb}_t = \exp_q\left(\frac{\boldsymbol{\eta}^w_t}{2}\right) \odot \hat{\mathbf{q}}^{wb}_{t|t-1}. \tag{5.25}$$

To achieve the accurate velocity and orientation estimates that are accomplished by the Kalman filter updates to the linearised model in [9], the dynamic models of the velocity and orientation in equations 5.19-5.20 are linearised. The linearisation makes it possible to include the orientation and the velocity as parts of the linear state in a Rao-Blackwellized particle filter. The lower the state-dimension of the nonlinear part of the particle filter, the fewer particles are required for the filter to converge [26]. When fewer particles are needed, the computational complexity of each filter iteration is lower [26]. By taking the first-order Taylor expansion of equations 5.19-5.20 about the orientation deviation, the proposed tightly coupled system model is given by

$$\mathbf{p}^w_t = \mathbf{p}^w_{t-1} + T\mathbf{v}^w_{k-1} \tag{5.26}$$

$$\begin{bmatrix} \mathbf{v}^w_t \\ \boldsymbol{\eta}^w_t \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{v}}^w_{t-1|t-1} + T(\mathbf{R}(\hat{\mathbf{q}}^{wb}_{t-1|t-1})\mathbf{y}^b_{acc,t} - \mathbf{g}^w) \\ \hat{\boldsymbol{\eta}}^w_{t-1|t-1} \end{bmatrix} \tag{5.27}$$

$$+ \mathbf{F}_t \left(\begin{bmatrix} \mathbf{v}^w_{t-1} \\ \boldsymbol{\eta}^w_{t-1} \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{v}}^w_{t-1|t-1} \\ \hat{\boldsymbol{\eta}}^w_{t-1|t-1} \end{bmatrix} \right) + \mathbf{G}_t \left(\begin{bmatrix} \mathbf{e}_{acc,t} \\ \mathbf{e}_{\omega,t} \end{bmatrix} \right), \tag{5.28}$$

$$\mathbf{q}^{nb}_t = \exp_q\left(\frac{\boldsymbol{\eta}^w_t}{2}\right) \odot \hat{\mathbf{q}}^{wb}_{t|t},, \tag{5.29}$$

$$\mathbf{y}_{\mathbf{v},t} = \mathbf{H}_t \begin{bmatrix} \mathbf{v}^w_t \\ \boldsymbol{\eta}^w_t \end{bmatrix} + \mathbf{e}_{\mathbf{v},t}, \tag{5.30}$$

$$y_t = \boldsymbol{\Phi}(\mathbf{p}^w_t)\mathbf{m}_t + v_t, \tag{5.31}$$

with system matrices

$$\mathbf{F}_t = \begin{bmatrix} \mathbf{I}_3 & T[\mathbf{y}^w_{acc,t}\times] \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix}, \tag{5.32}$$

$$\mathbf{G}_t = \begin{bmatrix} \mathbf{R}(\hat{\mathbf{q}}^{wb}_{t-1|t-1})T & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}(\hat{\mathbf{q}}^{wb}_{t-1|t-1})T \end{bmatrix}, \tag{5.33}$$

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \end{bmatrix}, \tag{5.34}$$

and joint noise distribution defined in equations 5.24. The tightly coupled system model is a special case of the general system model used by the Rao-Blackwellized particle filter with a triangular state-space model described in section 4.2, if the nonlinear states are chosen as $\mathbf{x}^n_t = \mathbf{p}^w_t$, and the linear states are chosen as $\mathbf{x}^l_t = \begin{bmatrix} \mathbf{v}^{wb}_t & \mathbf{q}^{wb}_t & \mathbf{m}_t \end{bmatrix}$. The tightly coupled Mag-SLAM algorithm resulting from applying the triangular Rao-Blackwellized particle to the model in equations 5.26-5.32 is defined in algorithm 7.

By choosing the velocity and orientation as linear states, the Kalman filter measurement update when the zero-velocity detectors become active are equivalent to the measurement updates in the ZUPT-EKF applied only to the velocity and orientation. By reducing the state-space vector, an equivalent filter can be designed for estimating the orientation and velocity only. The details of a zero-velocity aided EKF estimating only orientation and velocity are given in Appendix A.4.

In applying algorithm 5 to the state-space model in equations 5.26-5.34, a modification has been made to give an algorithm that can perform convergent and computationally tractable SLAM on data from a foot-mounted sensor. The particle filter measurement update given the pseudo zero-velocity measurements are omitted. The particle filter measurements were found to reduce the spread of the particle cloud so much that the filter no longer converged. What this means for the final algorithm, is that the zero-velocity pseudo measurement is not used to distinguish which particles are more likely. The weights of the particles are only evaluated by investigating the magnetic field. The zero-velocity pseudo measurement is still used to update the velocity and orientation of each particle, using a Kalman filter measurement update on the linearized model.

Algorithm 7: Tightly coupled Mag-SLAM for foot mounted sensor

for *t=0:N* **do**

Initialise the particle cloud by setting all initial positions $\mathbf{p}^{w,i}_{0|-1} = \mathbf{0}$, velocities to $\hat{\mathbf{v}}^{w,i}_{0|-1} = \mathbf{0}$, orientations to $\hat{\mathbf{q}}^{wb,i}_{0|-1} = \mathbf{q}_0$, velocity+orientation covariances to $\mathbf{P}^{i}_{0|-1} = \mathbf{P}_0$, initial magnetic field vector $\hat{\mathbf{m}}^{i}_{0|-1} = \mathbf{0}$, and the magnetic field covariance $\mathbf{P}^{i}_{m,0|-1} = \mathbf{\Lambda}$.

if *zero velocity detector is active* **then**

KF measurement update: For each particle, perform the measurement update as defined in step 1 in algorithm 9 to calculate the posterior states $\hat{\mathbf{q}}^{wb,i}_{t|t}$, $\hat{\mathbf{v}}^{w,i}_{t|t}$ and covariances $\mathbf{P}^{i}_{t|t}$.

end

1. **PF measurement update:** Identical to step 1 in algorithm 6.
2. **KF measurement update:** Identical to step 2 in algorithm 6.
3. **PF dynamic update:** Draw Monte-Carlo samples of the velocity, using the distribution of the Kalman filter estimate

$$\hat{\mathbf{v}}^{w,i}_{t} \sim \mathcal{N}(\hat{\mathbf{v}}^{w,i}_{t|t}, \mathbf{P}_{\mathbf{v},\mathbf{v},t|t-1}), \tag{5.35}$$

Use the estimated velocity to update the position of the particle according to

$$\mathbf{p}^{w,i}_{t+1|t} = \mathbf{p}^{i}_{t|t} + T\hat{\mathbf{v}}^{w,i}_{t|t} \tag{5.36}$$

4. **KF dynamic measurement update:** Use the prior position estimate of the particle to update the posterior linear state estimates according to

$$\mathbf{L}^{i}_{t} = T^2\mathbf{H}_t\mathbf{P}^{i}_{t|t}\mathbf{H}^{T}_{t} \tag{5.37}$$

$$\mathbf{K}^{i}_{t} \leftarrow \mathbf{P}^{i}_{t|t}\mathbf{H}^{T}_{t}T{\mathbf{L}^{i}_{t}}^{-1} \tag{5.38}$$

$$\begin{bmatrix} \hat{\mathbf{v}}^{w,i}_{t|t} \\ \hat{\boldsymbol{\eta}}^{w,i}_{t|t} \end{bmatrix} \leftarrow \begin{bmatrix} \hat{\mathbf{v}}^{w,i}_{t|t} \\ \hat{\boldsymbol{\eta}}^{w,i}_{t|t} \end{bmatrix} + \mathbf{K}^{i}_{t}\left(\mathbf{p}^{w,i}_{t+1|t} - \mathbf{p}^{w,i}_{t|t} - \hat{\mathbf{v}}^{w,i}_{t|t}\right) \tag{5.39}$$

$$\mathbf{P}^{i}_{t|t} \leftarrow \mathbf{P}^{i}_{t|t} - \mathbf{K}^{i}_{t}\mathbf{L}^{i}_{t}\mathbf{K}^{i}_{t} \tag{5.40}$$

Relinearise using

$$\hat{\mathbf{q}}^{wb,i}_{t|t} \leftarrow \exp_q\left(\frac{\hat{\boldsymbol{\eta}}^{w,i}_{t|t}}{2}\right) \odot \hat{\mathbf{q}}^{wb,i}_{t|t} \tag{5.41}$$

and by setting $\hat{\boldsymbol{\eta}}^{w,i}_{t|t} \leftarrow \mathbf{0}$.

5. **KF dynamic update:** For each particle, perform the dynamic update as defined in step 2 in algorithm 9 to calculate the prior states $\hat{\mathbf{q}}^{w,i}_{t+1|t}$, $\hat{\mathbf{v}}^{w,i}_{t+1|t}$ and covariances $\mathbf{P}^{i}_{t+1|t}$.

end

Chapter 6

Results

6.1 Simulation results Mag-SLAM for pedestrian navigation

This section shows simulation results for Mag-SLAM using Reduced-Rank Gaussian process regression and a Rao-Blackwellized particle filter on pedestrian odometry with magnetic norm field measurements.

The movement of a pedestrian is simulated inside a room of dimensions $8m \times 10m \times 2m$. The pedestrian is assumed to move in the shape of a rectangle at a constant speed $v = 1.4ms^{-1}$. The position vector is then sampled at a frequency of $10Hz$. Simple odometry is simulated by assuming that a noisy and biased measurement of the displacement from each timestep to the next is available according to

$$\Delta \mathbf{p}_t^w = \mathbf{p}_{t+1}^w - \mathbf{p}_t^w + \mathbf{w}_t + \mathbf{b}, \tag{6.1}$$

where the bias is $\mathbf{b} = [0,\ 0.005,\ 0]$, and the noise $\mathbf{w}_t$ is normally distributed as $\mathbf{w}_t \sim \mathcal{N}(0, \sigma_p^2 \mathbf{I}_3)$, with a variance $\sigma_p = 0.01$. The dead-reckoning integration of this odometry leads to the drifting position estimate displayed together with the simulated ground truth trajectory in figure 6.3.

The magnetic field is assumed to be a realisation of a Gaussian process prior with a squared exponential kernel. This simulation uses the hyperparameters $\sigma_{SE} = 1$, $l_{SE} = 0.3$, and $\sigma_y = 0.1$. The magnetic field can then be simulated as a Monte-Carlo sample from the multivariate distribution

$$\|\mathbf{H}(\mathbf{p}_{1:t})\| \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}(\mathbf{p}_{1:t}, \mathbf{p}_{1:t})\right), \tag{6.2}$$

and the magnetic field norm measurements can be simulated by adding a Monte-Carlo sampled measurement noise to the magnetic field norm in each position

$$y_{m,t} = \|\mathbf{H}(\mathbf{p}_{1:t})\| + e_{m,t}, \quad e_{m,t} \sim \mathcal{N}(0, \sigma_y^2). \tag{6.3}$$

The simulated magnetic field in the sampling locations is plotted in Figure 6.1, together with the simulated magnetic field measurements. As the simulated pedestrian motion revisiting the same positions in four laps, the magnetic field measurements in this plot have a pattern that repeats for each of the four simulated laps.

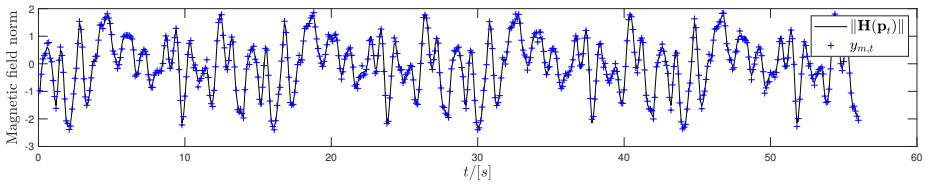
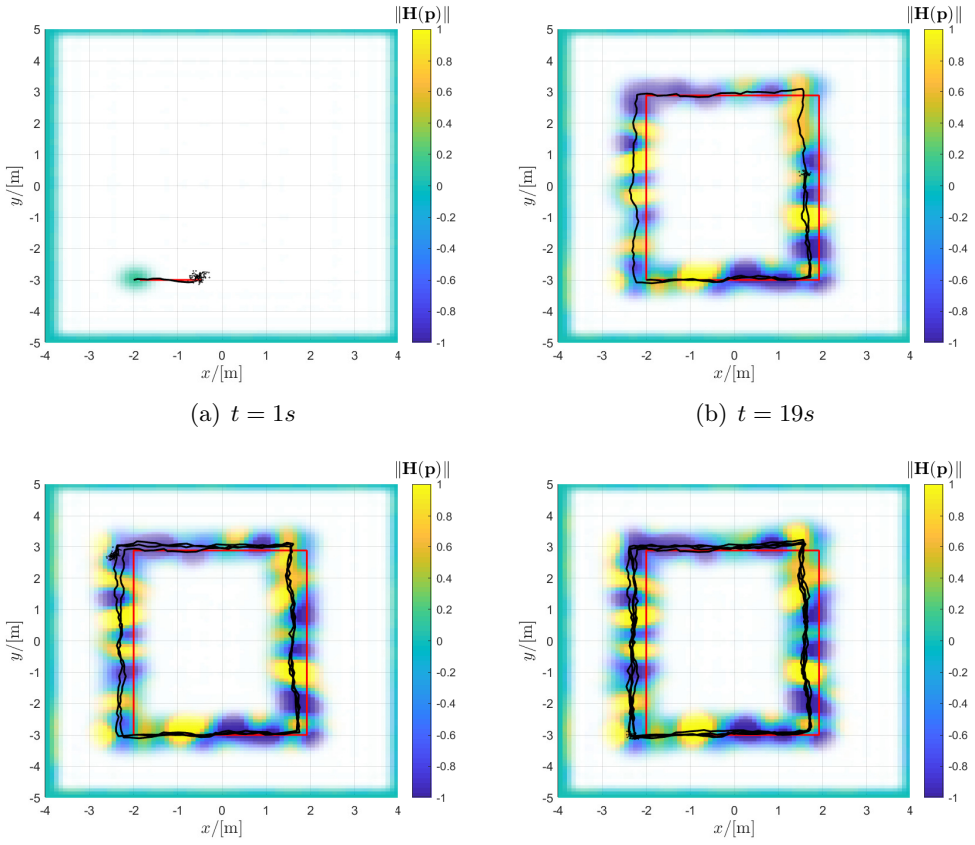


Figure 6.1: *The black line shows the simulated magnetic field norm in the position of the pedestrian at time t, and the blue crosses show the simulated magnetic field norm measurements*



(a) $t = 1s$ (b) $t = 19s$

(c) $t = 38s$ (d) $t = 56s$

Figure 6.2: *The back trajectory show the current highest weight particle, and the positions of all the particle are marked with black dots. The red line show the ground truth trajectory. The colour correspond to the magnetic field norm estimate of the highest weight particle, and the opacity is inversely proportionate with the variance of the estimate.*

The domain of the reduced-rank Gaussian process regression is chosen as the smallest cube with edges that are at least 2 meters away from any position in the correct trajectory. This is a larger margin than two times the length-scale l_{SE}, and the domain is therefore sufficiently large [27]. The domain is also small enough so that 2000 basis functions successfully approximate the true GP regression results.

The performance of the RBPF on these simulated data shows that drift removal is likely to be possible using only the magnetic field norm. In Figure 6.2, the estimated trajectories at the end of each of the four laps is displayed together with the trajectory of the highest weight particle. The RMSE of the position estimate of the noisy odometry is 0.8, while the RMSE of the position estimate from the RBPF is 0.2. To avoid premature resampling of the particle cloud, selective resampling with a criterion that $M_{eff} > 2/3M$ is included.

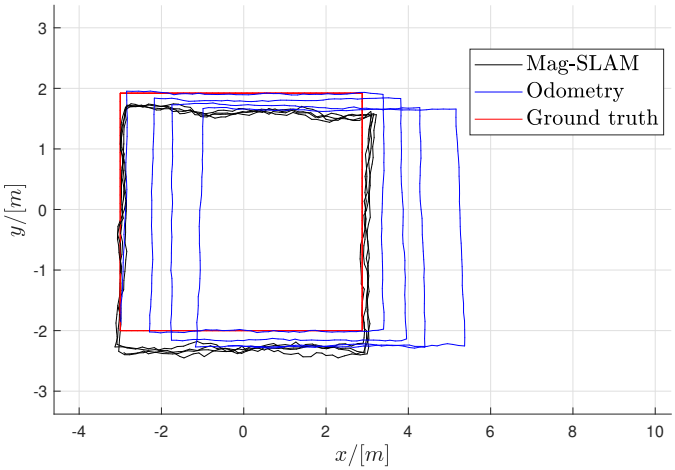
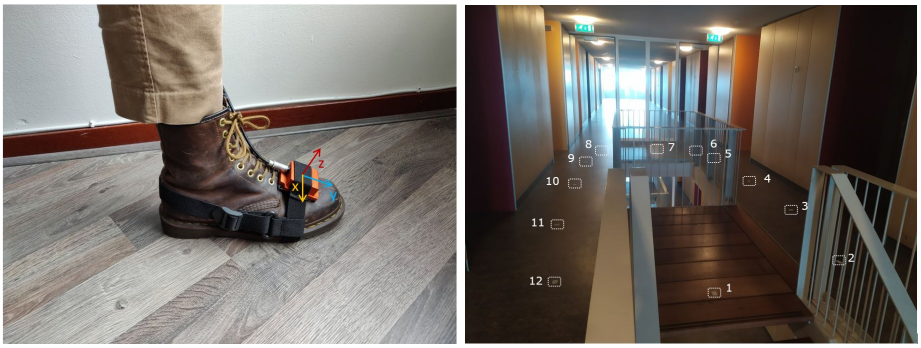


Figure 6.3: *Comparing the estimated trajectories using simulated noisy odometry (blue line) and Magnetic field simultaneous localisation and mapping (black line) with the ground truth (red line)*

6.2 Experimental Setup

Measurements were collected using an Xsens MTi 10 inertial sensor. Figure 6.4 (a) shows the experimental setup on the foot. Checkpoints were marked with tape on the floor in the indoor environment where the measurements were collected, as shown in Figure 6.4 (b). The root mean squared error of the position estimate was then calculated by detecting the difference in position estimates when revisiting the checkpoints.

The position estimation results are compared with the position estimation from an open-source implementation of the ZUPT-aided EKF [23] run on the same measurements. The parameters for the ZUPT-aided EKF were chosen to give as good position estimates as possible given the test subjects walking patterns. As the sensor is front-mounted, the position estimate drifts more compared to what is reported for heel-mounted sensors [10]. This drift can be used to demonstrate visibly in a small experiment how the Mag-SLAM algorithm can compensate for drift. Front-mounted sensors are also easily available off-the-shelf compared to sensors integrated in the heel of a boot. The parameters for the zero-velocity detector were kept to their original values from the open-source implementation, except for the threshold γ, that was set to $3 \cdot 10^5$. The accelerometer and gyroscope covariance matrices were set to $\mathbf{R}_{acc,t} = \sigma_a^2 T\mathbf{I}$ with $\sigma_a = 0.12ms^{-3/2}$ and $\mathbf{R}_{gyr,t} = \sigma_\omega^2 T\mathbf{I}$ with $\sigma_\omega = 0.006°s^{-1/2}$, respectively. The initial covariance for the position, velocity and orientation was set to $\mathbf{P}_0 = 0.12T\mathbf{I}$. The zero-velocity parameters, the gyroscope and the accelerometer measurement noise, were given the same values in the proposed Mag-SLAM algorithm.



(a) Sensor Mount (b) Checkpoint markers

Figure 6.4: *Experimental setup*

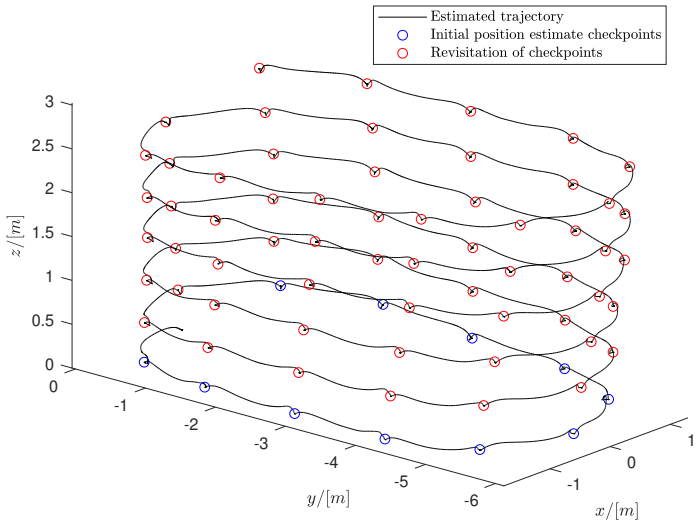


Figure 6.5: *Position estimate using the ZUPT-aided extended Kalman filter. The estimated positions at the checkpoints are marked with red circles.*

When the ZUPT-aided extended Kalman filter is used to estimate the position, the position estimate at the known checkpoints drifts over time, as can be seen in Figure 6.5. The root mean squared error of the ZUPT-aided EKF position estimate at the checkpoints is 1.17 meters for the collected measurements.

6.3 3D magnetic field norm map

The magnetic field norm map in both of the proposed algorithms is created as a function of the three spatial coordinates. This section aims at isolating and comparing the performance of the reduced rank Gaussian process regression compared to the ordinary Gaussian process regression given a set of position estimates and magnetic field measurements.

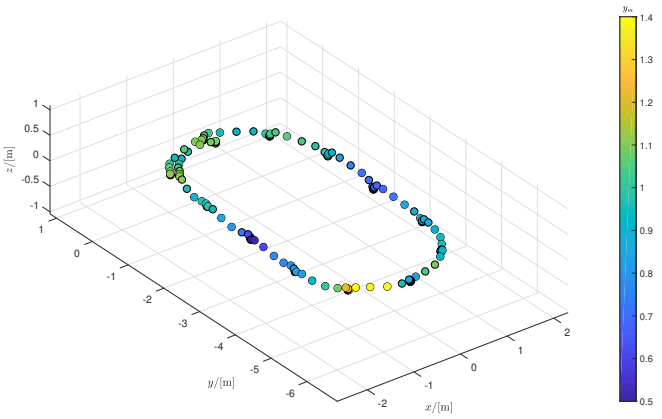


Figure 6.6: *The measured magnetic field norm y_m plotted in the estimated trajectory, by taking magnetic field norm measurements at a sampling rate of 10 Hz*

In simultaneous localisation and mapping, previous measurements are not available. Despite this, the proposed algorithms use two preliminary estimates of the magnetic field. The first estimate is of the average magnetic field norm of the entire measurement series. The average is subtracted from all incoming magnetic field measurements before using gaussian process regression to estimate the magnetic field anomalies. As the Xsens MTi

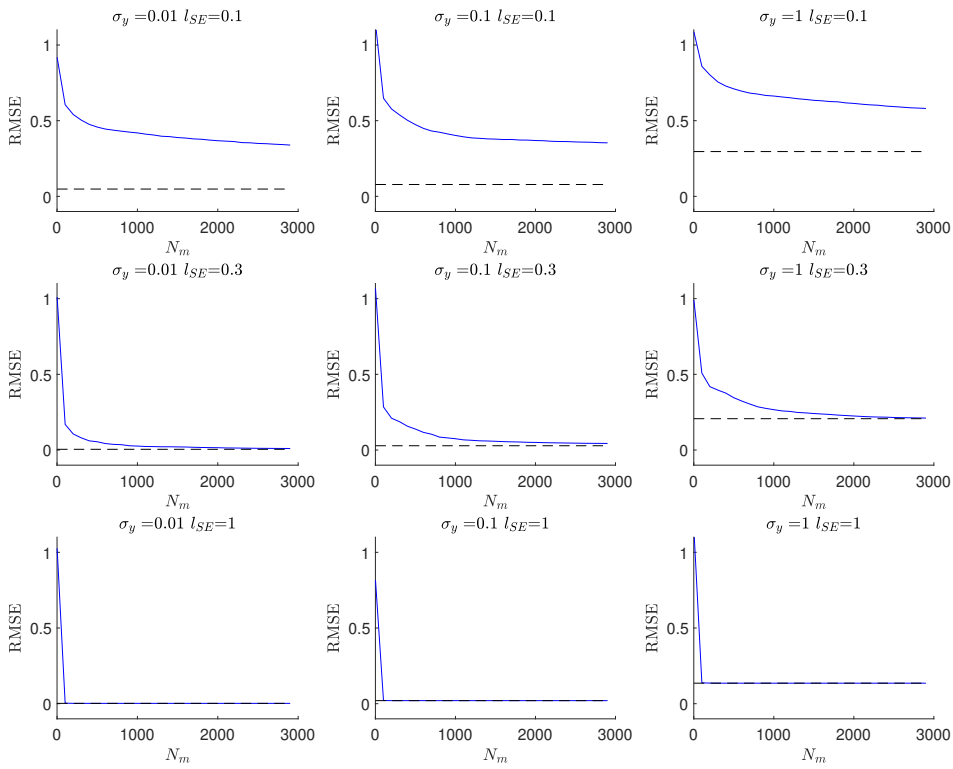
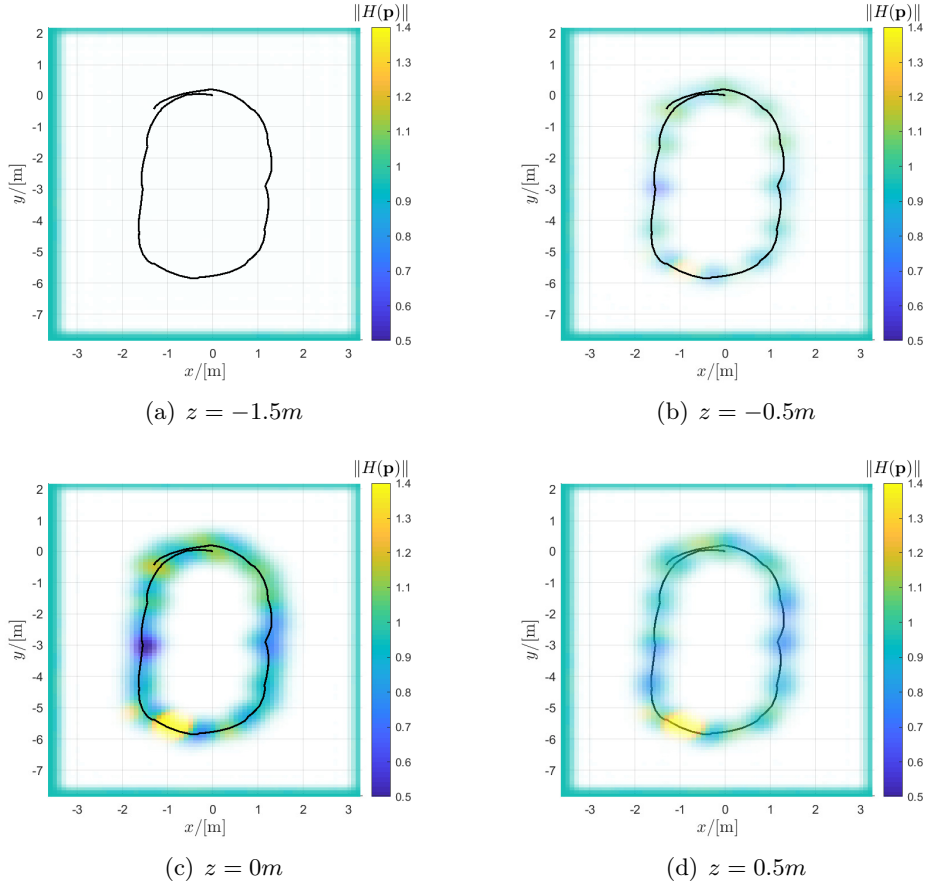


Figure 6.7: *The blue lines show the root mean squared error of Reduced-Rank Gaussian Process predictions on 100 test points of a Monte-Carlo sampled function from the magnetic field GP Prior, for a varying amount of basis functions. The prediction is based on 1000 noisy measurements. The stipulated black line shows the RMSE of Gaussian process regression fit to the same measurements and predicting the values in the same points. The experiment is repeated for different hyperparameters σ_y and l_{SE}, and the parameter σ_{SE} was set to 1 for all experiments.*

10 gives normalised measurements of the magnetic field, this estimate is likely to be usable for the same sensor in different locations. This average aims to estimate how large the underlying earth magnetic field norm is in terms of the sensors unitless measurements. The second estimate is a selection of somewhat appropriate hyperparameters for the kernel by visually inspecting the available magnetic field measurements in a preliminary position estimate. The visual inspection aims to estimate reasonable hyperparameters for indoor magnetic field anomalies near the floor in general. The visualisation in Figure 6.6 shows that the measured magnetic field norm anomalies in the initial position trajectory have a similar spatial variation as the more thorough measurements in a different indoor environment made by [18]. The selection of hyperparameters usually occurs by maximising the log-likelihood of the observed measurements [44]. This method is considered too computationally demanding to be a viable part of the online implementation proposed in this thesis.

This section displays the magnetic field norm map found by applying reduced-rank Gaussian process regression to magnetic field norm measurements from the foot-mounted sensor. A ZUPT-aided EKF estimated the position of each measurement, and manually corrected for the visible drift. The drift was corrected by adding a correction to each position estimate that increases linearly from zero with time. The magnitude of the increase was chosen so that the trajectory position estimate was correct at the first checkpoint. The drift correction was done to give a position estimate that was as accurate as possible so that the magnetic field maps are constructed based on a trajectory close to the ground truth position.



(a) $z = -1.5m$ (b) $z = -0.5m$

(c) $z = 0m$ (d) $z = 0.5m$

Figure 6.8: *Magnetic field norm predictions from a reduced-rank Gaussian process regression on measurements from a foot-mounted sensor in layers of a three-dimensional domain. The colour correspond to the predicted magnetic field norm value, and the opacity is inversely proportional with the variance of the estimate.*

The magnetic field map measurement are plotted in the estimated trajectory in Figure 6.6. In this figure, it can be seen that the magnetic field norm in the indoor environment where the measurements were collected vary rapidly with changing position. The length-scale of the Gaussian process regression used in this section is $l_{SE} = 0.3m$ to reflect the length scale of the magnetic field anomalies in Figure 6.6. The variance parameter is chosen to be $\sigma^2_{SE} = 1$. The chosen variance parameter reflects the magnitude of the same anomalies. As the measurements noise of the magnetic field sensor is quite low compared to the magnetic field anomalies, it was chosen as $\sigma^2_y = 0.01$. The resulting magnetic field norm map in three dimensions is illustrated using four layers in Figure 6.8.

The layered predicted magnetic field maps in Figure 6.8 illustrate how the Gaussian process regression interpolate and predict based on the magnetic field norm measurement in Figure 6.6. The confidence of the prediction decreases the further away the position is from any measurements. In subfigures (b) and (d), it is also possible to see that in the locations where the foot was positioned for longer (the stationary point of the step), more magnetic field measurements were collected. Hence, the prediction in and near these locations are mode confident. To obtain a reasonable magnetic field estimate, 2000 basis functions were needed. Simulation investigating the necessary amount of basis functions for the reduced-rank Gaussian to converge to the Gaussian process prediction is shown in Figure 6.7. The position samples are drawn from the position trajectory estimated by the zero-velocity aided extended Kalman filter. The cubical domain for the Reduced-Rank Gaussian process regression is chosen as the smallest cube where the edges are no closer than 2 meters to any of the positions in the trajectory.

The results of the simulation show that the Reduced-Rank Gaussian process regression requires more basis functions the smaller the length-scale, as the Reduced-Rank GP predictions converge more slowly to the GP prediction for smaller length-scales. The Reduced-Rank GP regression also required more basis functions when the measurement noise is higher. The plot on the second row, the second column tests how many basis functions are needed for the Reduced-Rank GP prediction to converge to the GP prediction for the hyper-parameters chosen for this magnetic field. The Reduced-Rank GP prediction converges at approximately 2000 basis functions, so in the remainder of this thesis, $N_m = 2000$ basis functions are used to create the magnetic field map.

It is worth mentioning that using 2000 basis functions to store the magnetic field in the particle filter is computationally demanding. This is because each of the M particles has a magnetic field state vector estimate $\hat{m}_t^i$ that has dimension N_m, and a magnetic field state covariance P_m^i that has dimension $N_m \times N_m$. As can be seen from Figure 6.7, the number of required basis functions reduces dramatically when the length scale of the anomalies l_{SE} relative to the size of the domain Ω increases. A way to reduce the number of necessary basis functions is, therefore, to use multiple tiles that each map a smaller domain. This method was implemented by [2], and the results they present indicate that executing the Reduced-Rank GP regression for magnetic field mapping as it is done in this thesis in multiple tiles rather than a single domain covering all of the estimated positions is feasible. The researchers needed 256 basis functions to map the area in each tile, where they use a length scale that is about 1/5 the cross-section of each tile.

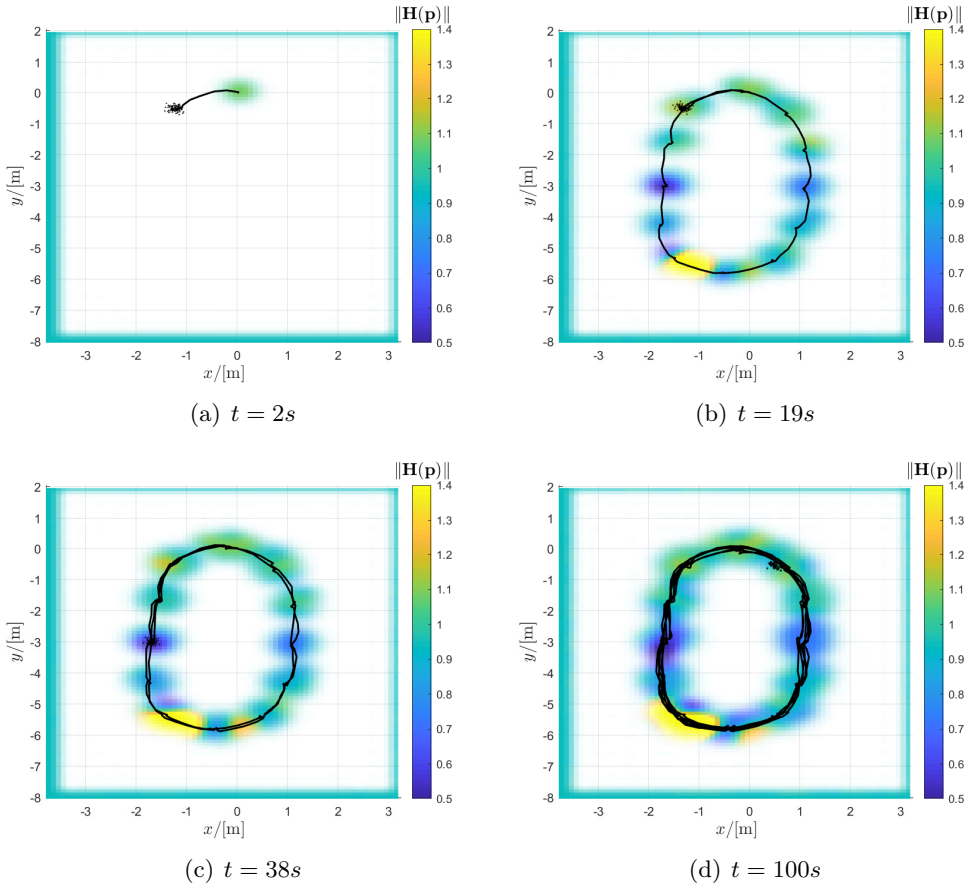
6.4 Experimental results loosely coupled Mag-SLAM for foot-mounted sensor

This section presents the results from Algorithm 6 on a set of measurements collected from a foot-mounted sensor during a walk between marked checkpoints. The algorithm was run using the same initial orientation estimate $\mathbf{q}_0$ as the open-source implementation of the ZUPT-aided EKF in [23]. The covariance of the additive white noise was chosen as

$$\mathbf{Q}_t = \begin{bmatrix} 0.0005 & 0 & 0 \\ 0 & 0.0005 & 0 \\ 0 & 0 & 0.001, \end{bmatrix} \tag{6.4}$$

with a slightly larger noise along the z-direction to compensate for the tendency of the odometry to drift along the z-direction. The results were obtained using 100 particles.

The trajectory estimated with the loosely coupled Mag-SLAM is based on an abstracted model with an additive white noise, which makes it less accurate compared to the zero-velocity aided extended Kalman filter at the beginning of the trajectory. The noisy, but drift-free trajectory from the loosely coupled Mag-SLAM algorithm is compared with the smoother, but drifting trajectory from the zero-velocity aided extended Kalman filter in Figure 6.10. The Mag-SLAM algorithm can, however, be seen to compensate for drift. Over time the trajectory estimated from Algorithm 6 is closer to the correct trajectory, as can be seen, by the fact that the estimated checkpoint locations are much closer in the trajectory estimated by Algorithm 6 compared to the zero-velocity aided extended Kalman filter. The RMSE of the revisitation of the checkpoints for the zero-velocity aided extended Kalman filter for this experiment is 1.68 m. In comparison, the RMSE of the checkpoint position estimates for the loosely coupled Mag-SLAM is 0.20 m.



(a) $t = 2s$ (b) $t = 19s$

(c) $t = 38s$ (d) $t = 100s$

Figure 6.9: *The blue trajectory show the current highest weight particle, and the positions of all the particle are marked with black dots. The colour correspond to the predicted magnetic field norm value, and the opacity is inversely proportionate with the variance of the estimate.*

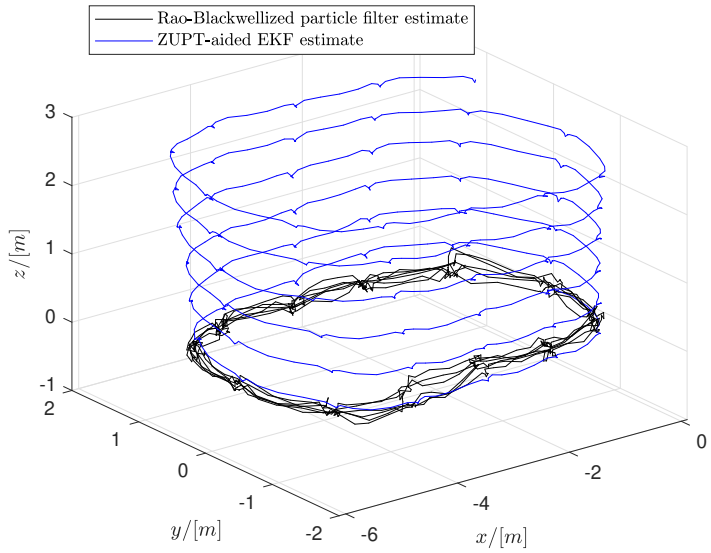


Figure 6.10: *Comparison of the position trajectories obtained from the zero-velocity aided extended Kalman filter and the proposed loosely coupled Mag-SLAM for foot-mounted sensor*

6.5 Experimental results tightly coupled Mag-SLAM for foot-mounted sensor

This section presents the results from running Algorithm 7 on a set of measurements collected from a foot-mounted sensor during a walk between marked checkpoints.

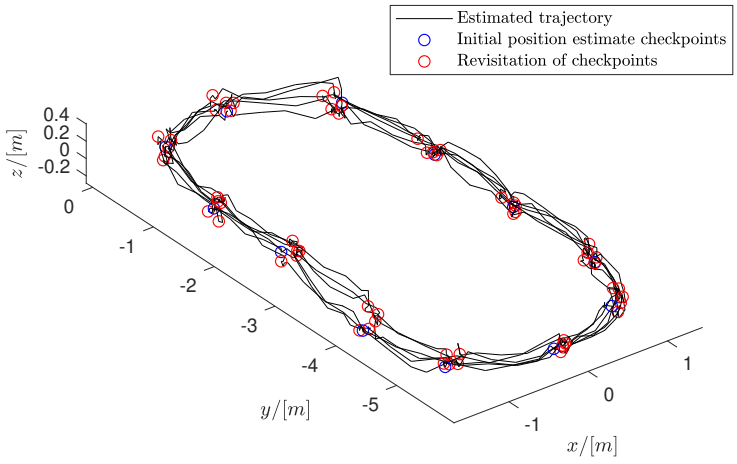


Figure 6.11: *Estimated positions of checkpoints in the estimated trajectory using loosely coupled Mag-SLAM*

The initial velocity and orientation covariance in the proposed Mag-SLAM algorithm was set to $\mathbf{P}_0 = 0.12T\mathbf{I}$, to reflect the initial covariance in the ZUPT-aided EKF.

The estimated trajectory from the tightly coupled Mag-SLAM algorithm is displayed together with the trajectory from the zero-velocity aided extended Kalman filter in Figure 6.14. The RMSE of the revisitation of the checkpoints for the zero-velocity aided extended Kalman filter for this experiment is 1.68 m. In comparison, the RMSE of the estimated checkpoint positions for the tightly coupled Mag-SLAM is 0.16 m. The estimated locations of the checkpoints are displayed in Figure 6.15. The tightly coupled Mag-SLAM algorithm can both remove drift, and give realistic estimates of the foot's movement locally. The tightly coupled Mag-SLAM is based on a full state-space model, so the information about the motion model of each particle is integrated into the particle filter algorithm, rather than abstracted away as it is in the loosely coupled approach. Part of the explanation why the trajectory looks smoother can, therefore, be found by investigating how the two algorithms search for viable positions - by looking at how the trajectories of the particle clouds spread without any incoming magnetic field measurements. The trajectories of 50 particles are displayed for both the loosely coupled and the tightly coupled approach in Figure 6.12. The spread of the trajectories is similar for both approaches. Still, the trajectory of each particle in the tightly coupled algorithm gives a trajectory that is a more realistic description of the movement for the foot of a pedestrian.

The loosely coupled approach creates hypothetical new positions for each particle by following the simple dynamic model in equation 5.9. This corresponds to adding a white-noise to the estimated position perturbation. The trajectories of each particle in the loosely coupled Mag-SLAM algorithm will frequently change position abruptly. This causes the small, but frequent discontinuities in the loosely coupled Mag-SLAM particle trajectories in Figure 6.12. The tightly coupled position trajectories are created by first perturbing the positions of each trajectory, but then also including a correction on the velocity and orientation of the particle given the new particle position, in step 4 in Algorithm 7. The initial perturbations are much smaller compared to the loosely coupled algorithm because the variance of the velocity is small. This causes the spread of the particle cloud to be smaller initially compared to the loosely coupled approach, as can be seen in Figure 6.12. By giving a correction to the velocity and orientation of the particle with step 4 in Algorithm 7, the velocity and orientation of each particle will be changed so that they are

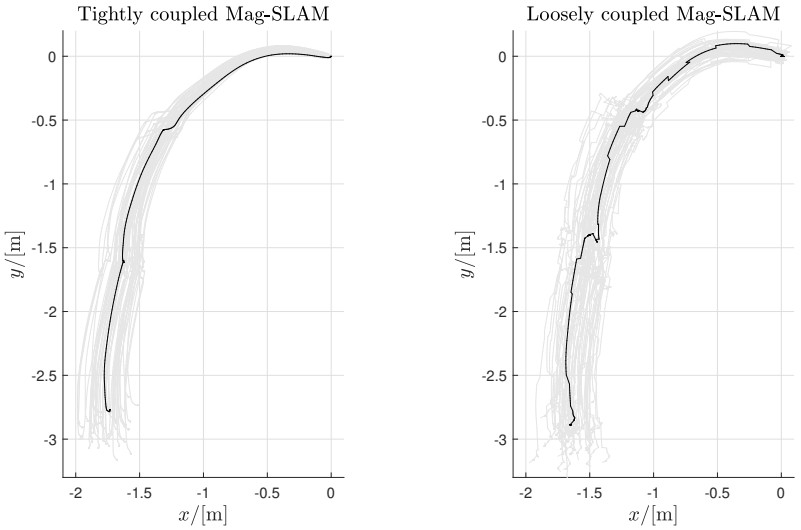
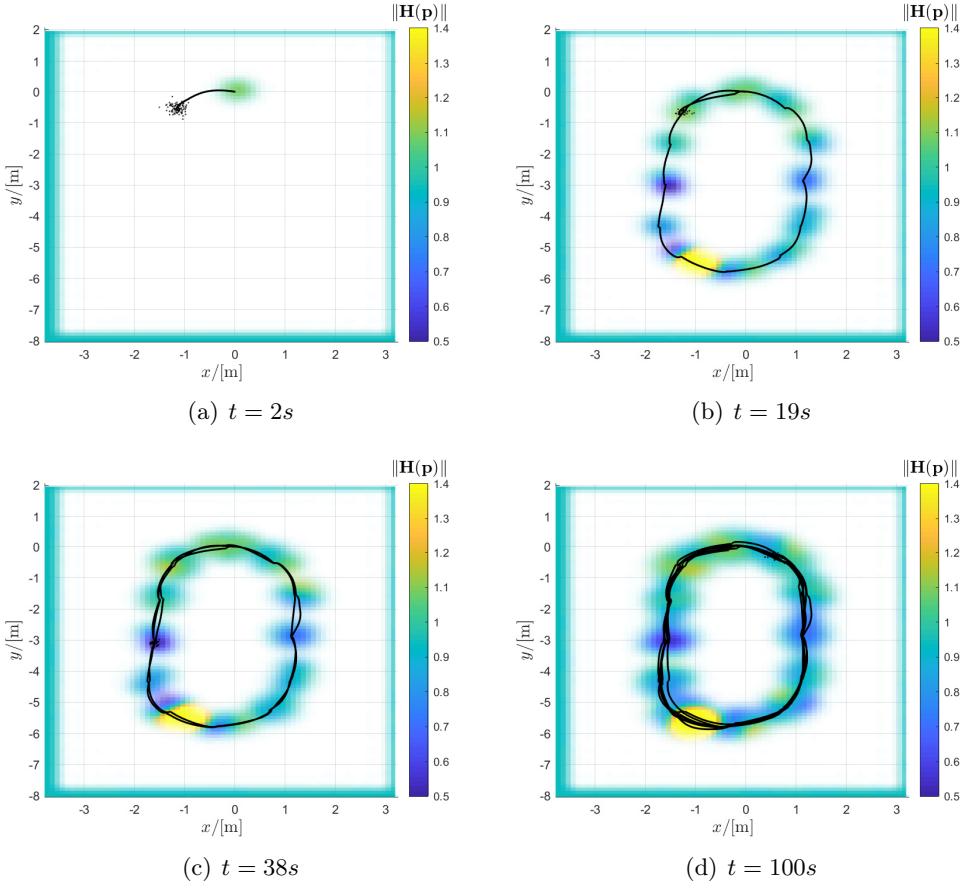


Figure 6.12: *This figure illustrates the difference in the dynamic updates of the loosely coupled and tightly coupled Mag-SLAM algorithms. The grey lines show the beginning of the position trajectories of 50 particles for each of the two proposed algorithms. The black lines highlight the trajectory of a single, randomly selected particle trajectory. The position trajectories of the particles in this figure are estimated without using any incoming magnetic field measurements.*

further away from the ZUPT-aided EKF velocity and orientation estimates, but better explain the position of the particle. On the next timestep, the position of each particle is affected by its own orientation and velocity and will be moved in a direction that is further away from the position estimate given by the zero-velocity aided extended Kalman filter. The perturbations in orientation and velocity then cause an increased spread in the particle cloud. The trajectories of each particle resemble realistic movement, because they follow the dynamic model in equation 5.26-5.29 more closely.



(a) $t = 2s$ (b) $t = 19s$

(c) $t = 38s$ (d) $t = 100s$

Figure 6.13: *The blue trajectory show the current highest weight particle, and the positions of all the particle are marked with black dots. The colour correspond to the predicted magnetic field norm value, and the opacity is inversely proportionate with the variance of the estimate.*

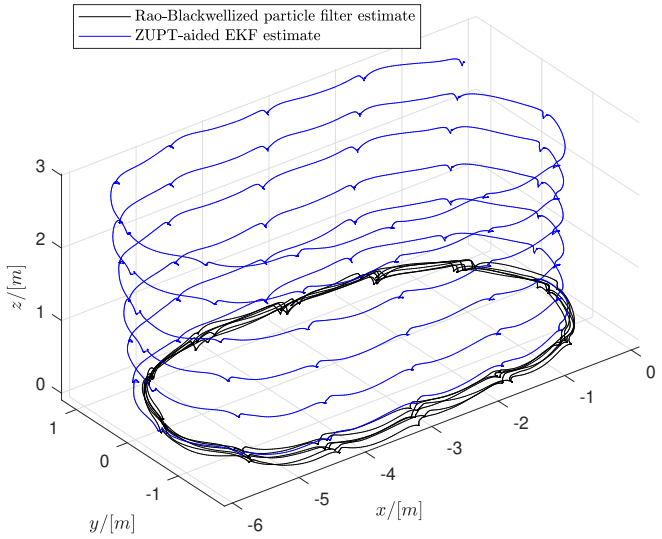


Figure 6.14: *Comparison of the position trajectories obtained from the zero-velocity aided extended Kalman filter and the proposed tightly coupled Mag-SLAM for foot-mounted sensor*

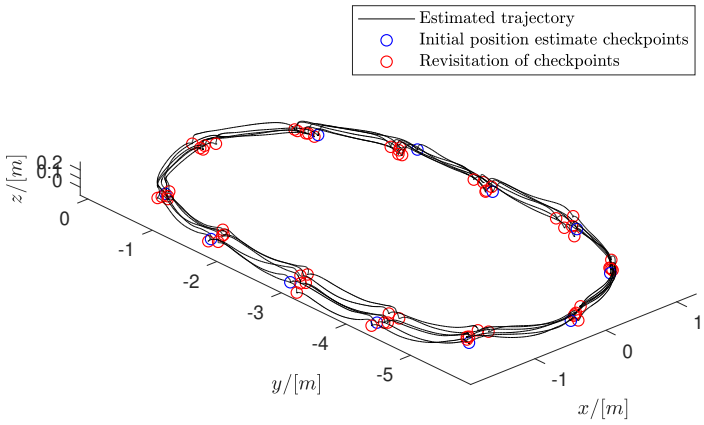


Figure 6.15: *Estimated positions of checkpoints in the estimated trajectory using loosely coupled Mag-SLAM*

Chapter 7

Conclusion and further research

Two algorithms for Mag-SLAM estimating three-dimensional position using Gaussian process regression on a foot-mounted sensor has been proposed in this thesis. The loosely coupled method is more modular, and the tightly coupled method gives more realistic trajectory estimates. Both methods give position estimates in three dimensions with comparable accuracy to a previous method for Mag-SLAM estimating two-dimensional position in a foot-mounted sensor. The results illustrate how tightly coupled sensor fusion can be beneficial compared to loosely coupled sensor fusion for improved position estimation in Mag-SLAM for pedestrian navigation.

Further research could be in the direction of making a map of all three magnetic field components instead of only the magnetic field norm. It could also investigate whether it is possible to include the zero-velocity particle filter measurement update in the tightly coupled sensor fusion algorithm using separate, stricter selective resampling criteria. Alternatively, using custom variances in the weight evaluation can be used to avoid premature loss of information by over-aggressive resampling, as is done in [40]. To make the algorithm more applicable to realistic scenarios, it can be investigated if the method can be made tolerant to changes in the magnetic field due to building collapse or extreme temperatures. Temperatures up 60 degrees Celcius are considered normal working conditions for firefighters, while temperatures in extreme conditions sometimes reach 300 degrees Celcius [45]. Steel starts changing magnetic properties around 200 degrees Celcius, but the changes don't become significant until around 400 degrees Celcius [46]. The effects of temperature on the magnetic field are generally not raised as a concern in articles that discuss the potential use of magnetic field navigation by firefighters ([3], [47]). As the magnetic field is caused by metal structures and furniture inside buildings [48], it is likely to change if parts of the building were to collapse in an emergency scenario. This possibility needs to be taken into account in further research for a fully integrated positioning system.

Further research could also investigate if maps from multiple sources could be fused to one map so that in a team of search and rescue workers, the measurements from one worker can benefit the whole team. The robustness of the algorithms can potentially be improved by investigating how different movement patterns can be taken into account, such as crawling, jumping or running. The accuracy of the algorithms can potentially be improved by using measurements from a heel-integrated sensor, as it would give higher accuracy in the initial position estimate.

Appendix A

Appendix

A.1 Using an approximated kernel function to obtain reduced rank Gaussian process predictions

This section contains an elaboration of the derivation described in lecture slides from a lecture on Magnetic Field SLAM given by Manon Kok in May, 2018 [49].

By inserting the approximation in equation 3.17 into the Gaussian process predictions defined in equations 3.3-3.5, the Gaussian process prediction can be expressed as

$$E[y^*] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}_{1:t})^T(\mathbf{\Phi}(\mathbf{x}_{1:t})\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}_{1:t})^T + \sigma_y I_N)^{-1} y \tag{A.1}$$

$$\begin{aligned} E[(y^* - E[y^*])^2] = {} & \mathbf{\Phi}(\mathbf{x}^*)\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}^*)^T \\ & - \mathbf{\Phi}(\mathbf{x}^*)\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}_{1:t})^T(\mathbf{\Phi}(\mathbf{x}_{1:t})\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}_{1:t})^T \\ & + \sigma_y \mathbf{I}_N)^{-1}\mathbf{\Phi}(\mathbf{x}_{1:t})\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}^*)^T + \sigma_y^2. \end{aligned} \tag{A.2}$$

As $\mathbf{\Lambda}$ is a matrix where all elements are zero except for elements on the diagonal, a matrix $\mathbf{\Lambda}^{\frac{1}{2}}$ can easily be computed by using the positive square-root of each element on the diagonal. This matrix has the property that $\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}} = \mathbf{\Lambda}$. Using this property, the equations can be factorized as

$$E[y^*] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}_{1:t})^T(\mathbf{\Phi}(\mathbf{x}_{1:t})\mathbf{\Lambda}\mathbf{\Phi}(\mathbf{x}_{1:t}) + \sigma_y I_N)^{-1}\mathbf{y}_{1:t} \tag{A.3}$$

$$\begin{aligned} E[(y^* - E[y^*])^2] = {} & \mathbf{\Phi}(\mathbf{x}^*)\mathbf{\Lambda}^{\frac{1}{2}}\Big[\mathbf{I}_m - \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Phi}(\mathbf{x}_{1:t})(\mathbf{\Phi}(\mathbf{x}_{1:t})\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Phi}(\mathbf{x}_{1:t})^T \\ & + \sigma_y \mathbf{I}_N)^{-1}\mathbf{\Phi}(\mathbf{x}_{1:t})^T\mathbf{\Lambda}^{\frac{1}{2}}\Big]\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Phi}(\mathbf{x}^*)^T + \sigma_y^2. \end{aligned} \tag{A.4}$$

The matrix inversion lemma

$$(\mathbf{P}^{-1} + \mathbf{B}^T\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{R}^{-1} = \mathbf{P}\mathbf{B}^T(\mathbf{B}\mathbf{P}\mathbf{B}^T + \mathbf{R})^{-1} \tag{A.5}$$

holds when where $\mathbf{P}$ and $\mathbf{R}$ are positive definite [50]. Using this for $\mathbf{P} = \mathbf{\Lambda}$, $\mathbf{B} = \mathbf{\Phi}(\mathbf{x}_{1:t})$, $\mathbf{R} = \sigma_y^2\mathbf{I}$ gives a reduced-rank expression for the expected value of the Gaussian process prediction

$$E[y^*] = \mathbf{\Phi}(\mathbf{x}^*)(\sigma_y^2\mathbf{\Lambda}^{-1} + \mathbf{\Phi}(\mathbf{x}_{1:t})^T\mathbf{\Phi}(\mathbf{x}_{1:t}))^{-1}\mathbf{\Phi}(\mathbf{x}_{1:t})^T\mathbf{y}_{1:t} \tag{A.6}$$

Using the matrix inversion lemma [51] in equation A.7

$$\sigma^2(\mathbf{Q}\mathbf{Q}^T + \sigma^2 I_n)^{-1} = \mathbf{I}_n - \mathbf{Q}(\sigma^2 I_q + \mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T, \tag{A.7}$$

with $\mathbf{Q} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Phi}(\mathbf{x}_{1:t})^T$, $n = m$, $q = N$, and $\sigma = \sigma_y$ gives a reduced-rank expression for the variance

$$E[(y^* - E[y^*])^2] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{\Lambda}^{\frac{1}{2}}\left[\sigma_y^2(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Phi}^T(\mathbf{x}_{1:t})\mathbf{\Phi}(\mathbf{x}_{1:t})\mathbf{\Lambda}^{\frac{1}{2}} + \sigma_y^2\mathbf{I}_m)^{-1}\right]\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Phi}(\mathbf{x}^*)^T + \sigma_y^2 \tag{A.8}$$

which can be further rewritten to the final reduced-rank formulation

$$E[(y^* - E[y^*])^2] = \sigma_y^2\mathbf{\Phi}(\mathbf{x}^*)\left(\mathbf{\Phi}(\mathbf{x}_{1:t})^T\mathbf{\Phi}(\mathbf{x}_{1:t}) + \sigma_y^2\mathbf{\Lambda}^{-1}\right)^{-1}\mathbf{\Phi}(\mathbf{x}^*)^T + \sigma_y^2 \tag{A.9}$$

A.2 Linear Model from Gaussian process regression

This section performs equivalent derivations defined in lecture slides from a guest lecture given by Manon Kok in May 2018 [49] for the magnetic field norm measurement model, in place of the magnetic field potential model. This derivation is not strictly necessary to achieve the final result, as it is a special case of the results in the lecture slides. However, the derivation in this appendix may help clarify the equivalence between the reduced rank gaussian regression and the linear stochastic model used in this thesis.

Given the reduced-rank Gaussian process model

$$E[y^*] = \mathbf{\Phi}(\mathbf{x}^*)(\sigma_y^2\mathbf{\Lambda}^{-1} + \mathbf{\Phi}(\mathbf{x})^T\mathbf{\Phi}(\mathbf{x}))^{-1}\mathbf{\Phi}(\mathbf{x})^T\mathbf{y} \tag{A.10}$$

$$E[(y^* - E[y^*])^2] = \sigma_y^2\mathbf{\Phi}(\mathbf{x}^*)\left(\mathbf{\Phi}(\mathbf{x})^T\mathbf{\Phi}(\mathbf{x}) + \sigma_y^2\mathbf{\Lambda}^{-1}\right)^{-1}\mathbf{\Phi}(\mathbf{x}^*)^T \tag{A.11}$$

It can be observed that the expected value $E[y^*]$ can be factorised into a term $\mathbf{\Phi}(\mathbf{x}^*)$ that is only a function of the prediction position, and the term $(\sigma_y^2\mathbf{\Lambda}^{-1} + \mathbf{\Phi}(\mathbf{x})^T\mathbf{\Phi}(\mathbf{x}))^{-1}\mathbf{\Phi}(\mathbf{x})^T\mathbf{y}$ which is only a function of the observed positions and measurements. This term can be represented by a state

$$\mathbf{m}_t = (\sigma_y^2\mathbf{\Lambda}^{-1} + \mathbf{\Phi}(\mathbf{x})^T\mathbf{\Phi}(\mathbf{x}))^{-1}\mathbf{\Phi}(\mathbf{x})^T\mathbf{y} \tag{A.12}$$

and a covariance

$$\mathbf{P}_t = \sigma_y^2\left(\mathbf{\Phi}(\mathbf{x})^T\mathbf{\Phi}(\mathbf{x}) + \sigma_y^2\mathbf{\Lambda}^{-1}\right)^{-1}, \tag{A.13}$$

that is the result of recursively applying the Kalman filter measurement update

$$\mathbf{S}_t = \mathbf{\Phi}(\mathbf{x}_t)\mathbf{P}_{t-1}\mathbf{\Phi}(\mathbf{x}_t)^T + \sigma_y^2\mathbf{I}, \tag{A.14}$$

$$\mathbf{K}_t = \mathbf{P}_{t-1}\mathbf{\Phi}(\mathbf{x}_t)^T\mathbf{S}_t^{-1}, \tag{A.15}$$

$$\mathbf{m}_t = \mathbf{m}_{t-1} + \mathbf{K}_t(y_t - \mathbf{\Phi}(\mathbf{x}_t)\mathbf{m}_{t-1}), \tag{A.16}$$

$$\mathbf{P}_t = \mathbf{P}_{t-1} - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T, \tag{A.17}$$

for each state $\mathbf{x}_t \in \mathbf{x}$ and $y_t \in \mathbf{y}$, using the initial states $\mathbf{m}_0 = 0$, and $\mathbf{P}_0 = \mathbf{\Lambda}$. It is useful to keep track of this state and covariance matrix, because combining equations A.10, A.11, A.12 and A.13, gives

$$E[y^*] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{m}_t \tag{A.18}$$
$$E[(y^* - E[y^*])^2] = \mathbf{\Phi}(\mathbf{x}^*)\mathbf{P}_t\mathbf{\Phi}(\mathbf{x}^*)^T + \sigma_y^2 \tag{A.19}$$

This expression shows that the reduced rank GP prediction reduces to a Kalman filter measurement prediction of a linear-stochastic state space model of the form

$$\mathbf{m}_{t+1} = \mathbf{m}_t, \tag{A.20}$$
$$y_t = \mathbf{\Phi}(\mathbf{x}_t)\mathbf{m}_t + v_t, \qquad v_t \sim \mathcal{N}(0, \sigma_y^2) \tag{A.21}$$

where $\mathbf{x}_t$ is not estimated by the Kalman filter measurement update, but assumed to be a known input. In the full state-space formulation, the state x_t is modelled as the nonlinear state in a Rao-Blackwellized particle filter.

A.3 The ZUPT aided EKF interpreted as a Kalman filter applied to a linearized model

The ZUPT-aided EKF as defined by Skog et al [9] applies the Kalman filter to the linearized state-space model

$$\begin{aligned}\begin{bmatrix}\mathbf{p}_t^w \\ \mathbf{v}_t^w \\ \boldsymbol{\eta}_t^w\end{bmatrix} &= \begin{bmatrix}\hat{\mathbf{p}}_{t-1|t-1}^w + T\hat{\mathbf{v}}_{t-1|t-1}^w \\ \hat{\mathbf{v}}_{t-1|t-1}^w + T(\mathbf{R}(\hat{\mathbf{q}}_{t-1|t-1}^{nb})\mathbf{y}_{acc,t}^b - \mathbf{g}) \\ \hat{\boldsymbol{\eta}}_{t-1|t-1}^w\end{bmatrix} \\ &\quad + \mathbf{F}_t\left(\begin{bmatrix}\mathbf{p}_{t-1}^w \\ \mathbf{v}_{t-1}^w \\ \boldsymbol{\eta}_{t-1}^w\end{bmatrix} - \begin{bmatrix}\hat{\mathbf{p}}_{t-1|t-1}^w \\ \hat{\mathbf{v}}_{t-1|t-1}^w \\ \hat{\boldsymbol{\eta}}_{t-1|t-1}^w\end{bmatrix}\right) + \mathbf{G}_t\left(\begin{bmatrix}\mathbf{e}_{acc,t} \\ \mathbf{e}_{\omega,t}\end{bmatrix}\right),\end{aligned} \tag{A.22}$$

$$\mathbf{q}_t^{nb} = \exp_q\left(\frac{\boldsymbol{\eta}_t^w}{2}\right) \odot \hat{\mathbf{q}}_{t|t}^{nb}, \tag{A.23}$$

$$\mathbf{y}_{\mathbf{v},t} = \mathbf{H}_t\begin{bmatrix}\mathbf{p}_t^w \\ \mathbf{v}_t^w \\ \boldsymbol{\eta}_t^b\end{bmatrix} + \mathbf{e}_{\mathbf{v},t}, \tag{A.24}$$

with system matrices

$$\mathbf{F}_k = \begin{bmatrix}\mathbf{I}_3 & T\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & Ts[\mathbf{y}_{acc}\times] \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3\end{bmatrix}, \tag{A.25}$$
$$\mathbf{G}_k = \begin{bmatrix}\mathbf{0} & \mathbf{0} \\ \mathbf{R}(\hat{\mathbf{q}}_{k|j}^{nb})T & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}(\hat{\mathbf{q}}_{k|j}^{nb})T\end{bmatrix}, \tag{A.26}$$
$$\mathbf{H}_t = \begin{bmatrix}\mathbf{0} & \mathbf{I}_3 & \mathbf{0}\end{bmatrix}, \tag{A.27}$$

and noise distribution

$$\begin{bmatrix} \mathbf{e}_{acc,t} \\ \mathbf{e}_{\omega,t} \\ \mathbf{e}_{v,t} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{R}_{acc,t} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\boldsymbol{\omega},t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{\mathbf{v},t} \end{bmatrix} \right), \tag{A.28}$$

which results in the algorithm

Algorithm 8: The ZUPT-aided extended Kalman filter

Data: A series of inputs $\mathbf{y}_{acc,t}$, $\mathbf{y}_{\boldsymbol{\omega},t}$ and $\mathbf{y}_{\mathbf{v},t}$

Result: Position, velocity and orientation estimates $\hat{\mathbf{p}}^w_{t|t}$, $\hat{\mathbf{v}}^w_{t|t}$, $\hat{\mathbf{q}}^{nb}_{t|t}$.

for *t=0:N* **do**

1. **EKF measurement update:** Calculate the posterior state using

$$\mathbf{S}_t = \mathbf{R}_{\mathbf{v},t} + \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T \tag{A.29}$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^T \mathbf{S}_t^{-1}, \tag{A.30}$$

$$\begin{bmatrix} \hat{\mathbf{p}}^w_{t|t} \\ \hat{\mathbf{v}}^w_{t|t} \\ \hat{\boldsymbol{\eta}}^w_{t|t} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{p}}^w_{t|t-1} \\ \hat{\mathbf{v}}^w_{t|t-1} \\ \hat{\boldsymbol{\eta}}^w_{t|t-1} \end{bmatrix} + \mathbf{K}_t \left(\mathbf{y}_{\mathbf{v},t} - \hat{\mathbf{v}}^w_{t|t-1} \right), \tag{A.31}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1}, \tag{A.32}$$

   using the matrix $\mathbf{H}_t$ defined in equation A.27.
   Relinearise using

$$\hat{\mathbf{q}}^{nb}_{t|t} = \exp_q \left(\frac{\hat{\boldsymbol{\eta}}_{t|t}}{2} \right) \odot \hat{\mathbf{q}}^{nb}_{t|t-1} \tag{A.33}$$

   and by setting $\hat{\boldsymbol{\eta}}_{t|t} \leftarrow \mathbf{0}$.

2. **EKF dynamic update:** Calculate the prior state using

$$\hat{\mathbf{p}}^w_{t+1|t} = \hat{\mathbf{p}}^w_{t|t} + T \hat{\mathbf{v}}^w_{t|t}, \tag{A.34}$$

$$\hat{\mathbf{v}}^w_{t+1|t} = \hat{\mathbf{p}}^w_{t|t} + T(\mathbf{R}(\hat{\mathbf{q}}^{nb}_{t|t}) \mathbf{y}^b_{acc,t} - \mathbf{g}^w), \tag{A.35}$$

$$\hat{\boldsymbol{\eta}}^w_{t+1|t} = \hat{\boldsymbol{\eta}}^w_{t|t}, \tag{A.36}$$

$$\mathbf{P}_{t+1|t} = \mathbf{F}_t \mathbf{P}_{t|t} \mathbf{F}_t^T + \mathbf{G}_t \begin{bmatrix} \mathbf{R}_{acc,t} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\boldsymbol{\omega},t} \end{bmatrix} \mathbf{G}_t^T, \tag{A.37}$$

   with the system matrices $\mathbf{F}_t$ and $\mathbf{G}_t$ defined in equations A.25-A.26.

end

A.4 The ZUPT aided EKF for only orientation and velocity estimation

In this section, a similar filter to the ZUPT-aided EKF that only estimates velocity and orientation using a subset of the linearised state-space model in equation A.22-A.24 is defined. After the definition, this section also shows that the velocity and orientation estimate compared to the standard formulation of the ZUPT-aided EKF are equivalent.

Starting from the subset of the linearised statespace model for velocity and orientation

$$\begin{aligned}\begin{bmatrix}\mathbf{v}_t^w \\ \boldsymbol{\eta}_t^w\end{bmatrix} &= \begin{bmatrix}\hat{\mathbf{v}}_{t-1|t-1}^w + T(\mathbf{R}(\hat{\mathbf{q}}_{t-1|t-1}^{nb})\mathbf{y}_{acc,t}^b - \mathbf{g}^w) \\ \hat{\boldsymbol{\eta}}_{t-1|t-1}^w\end{bmatrix} \\ &+ \mathbf{F}_t\left(\begin{bmatrix}\mathbf{v}_{t-1}^w \\ \boldsymbol{\eta}_{t-1}^w\end{bmatrix} - \begin{bmatrix}\hat{\mathbf{v}}_{t-1|t-1}^w \\ \hat{\boldsymbol{\eta}}_{t-1|t-1}^w\end{bmatrix}\right) + \mathbf{G}_t\left(\begin{bmatrix}\mathbf{e}_{acc,t} \\ \mathbf{e}_{\omega,t}\end{bmatrix}\right),\end{aligned} \tag{A.38}$$

$$\mathbf{q}_t^{wb} = \exp_q\left(\frac{\boldsymbol{\eta}_t^w}{2}\right) \odot \hat{\mathbf{q}}_{t|t}^{wb}, \tag{A.39}$$

$$\mathbf{y}_{\mathbf{v},t} = \mathbf{H}_t\begin{bmatrix}\mathbf{v}_t^w \\ \boldsymbol{\eta}_t^w\end{bmatrix} + \mathbf{e}_{\mathbf{v},t}, \tag{A.40}$$

with system matrices

$$\mathbf{F}_t = \begin{bmatrix}\mathbf{I}_3 & Ts[\mathbf{y}_{acc,t}^w\times] \\ \mathbf{0} & \mathbf{I}_3\end{bmatrix}, \tag{A.41}$$

$$\mathbf{G}_t = \begin{bmatrix}\mathbf{R}(\hat{\mathbf{q}}_{t|t}^{wb})T & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}(\hat{\mathbf{q}}_{t|t}^{wb})T\end{bmatrix}, \tag{A.42}$$

$$\mathbf{H}_t = \begin{bmatrix}\mathbf{I}_3 & \mathbf{0}\end{bmatrix}, \tag{A.43}$$

and the same noise distribution as in equation A.28, the ZUPT-aided EKF for velocity and orientation estimation is defined in Algorithm 9.

Algorithm 9: The ZUPT-aided extended Kalman filter for only orientation and velocity estimation

Data: A series of inputs $\mathbf{y}_{acc,t}$, $\mathbf{y}_{\boldsymbol{\omega},t}$ and $\mathbf{y}_{\mathbf{v},t}$

Result: Velocity and orientation estimates $\hat{\mathbf{v}}^w_{t|t}$, $\hat{\mathbf{q}}^{nb}_{t|t}$.

for *t=0:N* **do**

1. **EKF measurement update:** Calculate the posterior state using

$$\mathbf{S}_t = \mathbf{R}_{\mathbf{v},t} + \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T, \tag{A.44}$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^T \mathbf{S}_t^{-1}, \tag{A.45}$$

$$\begin{bmatrix} \hat{\mathbf{v}}^w_{t|t} \\ \hat{\boldsymbol{\eta}}^w_{t|t} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{v}}^w_{t|t-1} \\ \hat{\boldsymbol{\eta}}^w_{t|t-1} \end{bmatrix} + \mathbf{K}_t \left(\mathbf{y}_{\mathbf{v},t} - \hat{\mathbf{v}}^w_{t|t-1} \right), \tag{A.46}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1}, \tag{A.47}$$

   using the matrix $\mathbf{H}_t$ defined in equation A.43.
   Relinearise using

$$\hat{\mathbf{q}}^{wb}_{t|t} = \exp_q \left(\frac{\hat{\boldsymbol{\eta}}_{t|t}}{2} \right) \odot \hat{\mathbf{q}}^{wb}_{t|t-1} \tag{A.48}$$

   and by setting $\hat{\boldsymbol{\eta}}_{t|t} \leftarrow \mathbf{0}$.

2. **EKF dynamic update:** Calculate the prior state using

$$\hat{\mathbf{v}}^w_{t+1|t} = \hat{\mathbf{v}}^w_{t|t} + T(\mathbf{R}(\hat{\mathbf{q}}^{wb}_{t|t}) \mathbf{y}^b_{acc,t} - \mathbf{g}^w), \tag{A.49}$$

$$\hat{\boldsymbol{\eta}}^w_{t+1|t} = \hat{\boldsymbol{\eta}}^w_{t|t}, \tag{A.50}$$

$$\mathbf{P}_{t+1|t} = \mathbf{F}_t \mathbf{P}_{t|t} \mathbf{F}_t^T + \mathbf{G}_t \begin{bmatrix} \mathbf{R}_{acc,t} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\boldsymbol{\omega},t} \end{bmatrix} \mathbf{G}_t^T, \tag{A.51}$$

   with the system matrices $\mathbf{F}_t$ and $\mathbf{G}_t$ defined in equations A.41-A.41.

end

A.5 Rao-Blackwellized particle filter algorithm for triangular state-space model

Algorithm 10: A Rao-Blackwellized particle filter using a triangular model

Data: A measurement sequence $\mathbf{y}_{0:N}$ produced by the stochastic state-space model in equations 2.3 and 2.4, process noise $\mathbf{Q}_k$, and measurement noise $\mathbf{R}_k$. An initial probability density for the state $p(\mathbf{x}_0)$

Result: Estimates of the prior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\}_{t=0}^N$ and the posterior distributions $\{p(\mathbf{x}_t|\mathbf{y}_{0:t})\}_{t=0}^N$ in the form of marginalized particle clouds, represented by a set of particles $\{\hat{\mathbf{x}}_t^{i,n}\}_{i=1}^N$ with corresponding linear states $\{\hat{\mathbf{x}}_t^{i,l}\}_{i=1}^N$ and covariances for each linear state $\{\mathbf{P}_t^i\}_{i=1}^N$

for *t=0:N* **do**

Initialise the particle cloud by sampling $\mathbf{x}_{0|-1}^{i,n} \sim p(\mathbf{x}_0^n)$, and initalise all states and covariances to $\mathbf{x}_{0|-1}^{i,l} = \mathbf{x}_0$, $\mathbf{P}_{0|-1}^i = \mathbf{P}_0$

1. **PF measurement update:** Identical to step 1 in Algorithm 5
2. **KF measurement update:** Identical to step 2 in Algorithm 5
3. **PF dynamic update:** Calculate the prior particle cloud particle cloud $\{\hat{\mathbf{x}}_{t+1|t}^{i,l}\}_{i=1}^N$ by using Monte-Carlo samples of the noise $\mathbf{w}_t^{i,n} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t^n)$, Monte-Carlo samples of the state $\hat{\mathbf{x}}_t^{i,l} \sim \mathcal{N}(\hat{\mathbf{x}}_{t|t}^{i,l}, \mathbf{P}_{t|t}^i)$ and calculate the effect on each particle according to equation 4.14. The prior particle cloud are then samples from the distribution $\hat{\mathbf{x}}_{t+1|t}^{i,n} \sim p(\mathbf{x}_{t+1}^n|\mathbf{y}_{0:t})$
4. **KF dynamic measurement update:**
   For each particle, perform a pseudo-measurement update on the posterior states $\hat{\mathbf{x}}_{t|t}^{i,l}$ and covariances $\mathbf{P}_{t|t}^i$ using

$$\mathbf{L}_t^i = \mathbf{G}_t^n \mathbf{Q}_t^n \mathbf{G}_t^{nT} + \mathbf{F}_t^n \mathbf{P}_{t|t-1}^i \mathbf{F}_t^{nT} \tag{A.52}$$

$$\tilde{\mathbf{K}}_t^i = \mathbf{P}_{t|t-1}^i \mathbf{F}_t^{nT} (\mathbf{L}_t^i)^{-1} \tag{A.53}$$

$$\tilde{\mathbf{x}}_{t|t}^{i,l} = \hat{\mathbf{x}}_{t|t}^{i,l} + \tilde{\mathbf{K}}_t^i (\hat{\mathbf{x}}_{t+1|t}^{i,n} - \mathbf{f}_t(\hat{\mathbf{x}}_{t|t}^{i,n}) - \mathbf{F}_t^n \hat{\mathbf{x}}_{t|t-1}^{i,l}) \tag{A.54}$$

$$\tilde{\mathbf{P}}_{t|t}^i = \mathbf{P}_{t|t}^i - \mathbf{K}_t^i \mathbf{F}_t^n \mathbf{P}_{t|t}^i \tag{A.55}$$

   The state and covariance uniquely describes the Gaussian marginal posterior distribution $p(\mathbf{x}_t^l|\mathbf{y}_{0:t}, \mathbf{x}_{1:t+1}^{n,i}) = \mathcal{N}(\hat{\mathbf{x}}_{t|t}^{i,l}, \mathbf{P}_{t|t}^i)\Big|_{\mathbf{x}_t^l}$
5. **KF dynamic update:** For each particle, calculate the prior state $\hat{\mathbf{x}}_{t+1|t}^{i,l}$ and covariance $\mathbf{P}_{t+1|t}^{i,n}$ using

$$\hat{\mathbf{x}}_{t+1|t}^{i,l} = \mathbf{F}_t \tilde{\mathbf{x}}_{t|t}^{i,l} \tag{A.56}$$

$$\mathbf{P}_{t+1|t}^i = \mathbf{F}_t \tilde{\mathbf{P}}_{t|t}^i \mathbf{F}_t^T + \mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^T. \tag{A.57}$$

   The prior state and covariance uniquely describes the Gaussian marginal prior probability density $p(\mathbf{x}_{t+1}^l|\mathbf{y}_{0:t}, \hat{\mathbf{x}}_{0:t+1}^{i,n}) = \mathcal{N}(\hat{\mathbf{x}}_{t+1|t}^{i,l}, \mathbf{P}_{t+1|t}^i)\Big|_{\mathbf{x}_{t+1}^l}$

end

A.6 Using Hilbert-space based method with analytical basis functions for reduced-rank Gaussian process regression in a cubical domain

When Hilbert space methods for reduced-rank Gaussian process regression are used in a cubical domain, the set of basis functions $\{\phi_j(x)\}_{j=1}^{N_m}$ and eigenvalues $\{\lambda_j\}_{j=1}^{N_m}$ can be found analytically [27]. The analytical solution for a 3-dimensional Cartesian rectangular domain $\Omega = [-L_1, L_1] \times [-L_2, L_2] \times [-L_3, L_3]$ are

$$\phi_{j_1,j_2,j_3}(\mathbf{x}) = \prod_{k=1}^{3} \frac{1}{\sqrt{L_k}} \sin\left(\frac{\pi j_k (x_k + L_k)}{2L_k}\right) \tag{A.58}$$

$$\lambda_{j_1,j_2,j_3} = \sum_{k=1}^{3} \left(\frac{\pi j_k}{2L_k}\right)^2 \tag{A.59}$$

Subtracting an input vector $\mathbf{a}$ to input argument $\mathbf{x}$ corresponds to shifting the domain to $\Omega = [-L_1 + a_1, L_1 + a_1] \times [-L_2 + a_2, L_2 + a_2] \times [-L_3 + a_3, L_3 + a_3]$. This gives the basis functions

$$\phi_{j_1,j_2,j_3}(\mathbf{x}) = \prod_{k=1}^{3} \frac{1}{\sqrt{L_k}} \sin\left(\frac{\pi j_k (x_k - a_k + L_k)}{2L_k}\right), \tag{A.60}$$

and the equations for the eigenvalues remain the same as in A.59. A domain with arbitrary limits $\Omega = [L_{l,1}, L_{u,1}] \times [L_{l,2}, L_{u,2}] \times [L_{l,3}, L_{u,3}]$ can always be rewritten on the form $\Omega = [-L_1 + a_1, L_1 + a_1] \times [-L_2 + a_2, L_2 + a_2] \times [-L_3 + a_3, L_3 + a_3]$, by selecting $a_k = \frac{1}{2}(L_{l,k} + L_{u,k})$ and $L_k = \frac{1}{2}(L_{u,k} - L_{l,k})$. Inserting these expressions into A.60 and A.59 gives

$$\phi_{j_1,j_2,j_3}(\mathbf{x}) = \prod_{k=1}^{3} \frac{1}{\sqrt{\frac{1}{2}(L_{u,k} - L_{l,k})}} \sin\left(\frac{\pi j_k (x_k - L_{l,k})}{L_{u,k} - L_{l,k}}\right), \tag{A.61}$$

$$\lambda_{j_1,j_2,j_3} = \sum_{k=1}^{3} \left(\frac{\pi j_k}{L_{u,k} - L_{l,k}}\right)^2, \tag{A.62}$$

which therefore solves the Laplace equations for a rectangular domain with arbitrary upper and lower limits along each axis. A corresponding argument can be carried out for the n-dimensional case.

As these basis functions are used in the reduced-rank approximation by decreasing order of spectral density of the corresponding eigenvalue, it is necessary to compute the N_m basisfunctions which eigenvalues have the highest spectral density. The eigenvalues are indexed by three indices, $j_1 \in \mathbb{N}$, $j_2 \in \mathbb{N}$ and $j_3 \in \mathbb{N}$. All possible eigenvalues are determined by all possible permutations of these indices. To determine which eigenvalue has the highest spectral density, it is necessary to analytically prove that there is no other possible permutation of indices that produces an eigenvalue with a higher spectral density. This is considered outside the scope of this thesis. Instead, it is possible to take advantage of the

observation that combinations of lower-valued indices generally produce eigenvalues with higher-valued spectral densities.

By looking at all possible permutations of indices beyond an upper limit N_{search} and sorting them according to the spectral density of their corresponding eigenvalues, the analytical solutions can be demonstrated to correspond to numerical solutions in a two-dimensional domain $\Omega = [L_{l,1}, L_{u,1}] \times [L_{l,2}, L_{u,2}]$. The result of this approach for varios values of N_{search} can be seen in Figure A.1. This figure compares the analytical solutions with a numerical solution. The numerical solution automatically finds the 16 combination of modes that has the highest corresponding eigenvalues. In equation A.61 it can be seen that the wave frequency of the sinusoid along axis x_k is multiplied proportional with the index j_k. The index can therefore be seen by the number of wavetops and bottoms along the vertical and horisontal direction in Figure A.1 respectively. It is therefore possible to visually inspect the similarity of the numerical solution with the analytical ones to determine weather a sufficient amount of index permutations have been investigated.

For this small example it is apparent that both $N_{search} = 2$ and $N_{search} = 3$ is insufficient, as the permutations do not investigate enough possible index combinations to suggest 16 different basis functions. By picking $N_{search} = 4$, enough permutations are investigated to suggest 16 basis functions. However, by comparing basis function number 14-16 with the numeric solution, it can be seen that they do not match. This is because basis function number 14 has 5 wavetops and bottoms along the vertical dimension, and it is therefore necessary to investigate at least up until $N_{search} = 5$ to find the correct basis functions. The analytic basis functions for both $N_{search} = 5$ and $N_{search} = 6$ can be seen to correspond to the numeric basis functions. It is necessary to select N_{search} large enough to investigate all the needed basis functions, but selecting it too large is neither beneficial nor harmful, except for the increased computational complexity.

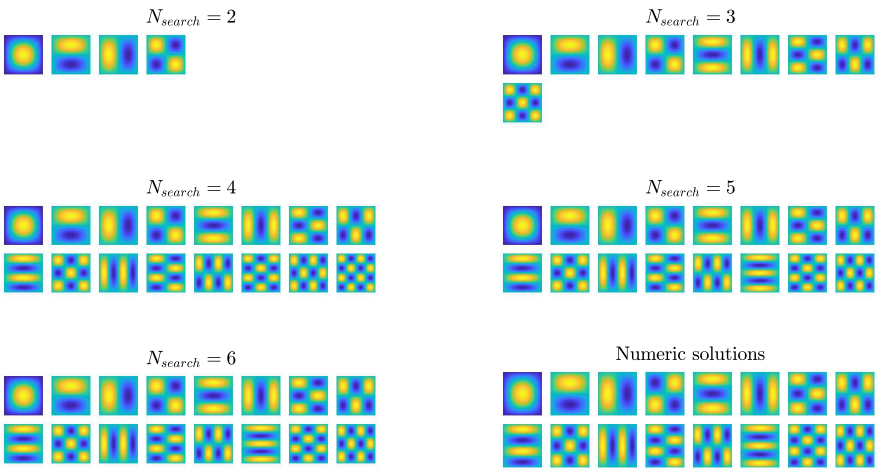


Figure A.1: *A comparison of the first 16 eigenfunctions for a rectangular domain* $\Omega = [-0.9,\ 0.9] \times [-1,\ 1]$. *The first 5 sub-figures display the first eigenfunctions found by looking for modes up until* N_{search}. *The bottom right figure show the shape of the true eigenfunctions found from a numerical solution.*

A.7 Draft publication

Magnetic field norm SLAM using Gaussian process maps in foot-mounted sensor

Frida Viset
Norwegian University of Science and Technology
frida.viset@gmail.com

Jan Tommy Gravdahl
Norwegian University of Science and Technology
jan.tommy.gravdahl@ntnu.no

Manon Kok
Delft University of Technology
m.kok-1@tudelft.nl

***Abstract*—Magnetic field Simultaneous localisation and mapping (Mag-SLAM) for indoor pedestrian navigation simultaneously creates a map of indoor magnetic field anomalies and estimate the position of the pedestrian inside this map. This paper proposes an algorithm for Mag-SLAM on a foot-mounted sensor. The algorithm uses a model of the movement of the sensors, together with a zero-velocity detector and magnetic field norm measurements. This gives an accurate position estimate that compensates for drift when positions are revisited. The algorithm is tested on measurements from a foot-mounted sensor collected from a walk around a room, and the estimated position trajectories are compared with estimated position trajectories from an existing method for position estimation using a foot-mounted sensor.**

I. Introduction

Indoor location and navigation is essential in emergency response scenarios [1]. Firefighters and other first responders have reported that disorientation and being lost inside a building is a common reason for traumatic injuries, as well as a contributing factor to firefighter deaths [2]. Currently, most common navigation methods for firefighters are a combination of burning flashlights to mark exits, ropes to mark safe ways out of the dangerous area, and alarms connected to wearable sensors that go off if a firefighter does not move for a specific amount of time [2].

A foot-mounted combined tri-axis gyroscope and accelerometer has previously been used to obtain accurate position estimates by using the zero-velocity update aided extended Kalman filter (ZUPT-aided EKF), proposed by [3]. A tri-axis accelerometer, gyroscope and magnetometer can be integrated into the shoe of substantial footwear [4]. A team of researchers from Sweden implemented the zero-velocity aided extended Kalman filter in open-source software [5]. The proposed algorithms build on the open-source software in [5]. The derivation of the zero-velocity aided extended Kalman filter and the specifics of the implementation is described in an accompanying publication, see [6]. The position estimate from the ZUPT-aided EKF is highly accurate for indoor position estimation [6].

An accelerometer measures the sensor's acceleration, while a gyroscope measures the sensors angular velocity. All state estimation that relies only on integration of noisy measurements will have a variance that increases linearly with time [7]. This phenomenon is often referred to as drift [8]. As accelerometers and gyroscopes measure acceleration and rotational velocity, respectively, integration of the measurements is a common approach to achieving position and orientation estimates [7]. This is commonly referred to as strap-down inertia navigation, and the variance of this position estimate increases more than cubically in time if both the accelerometer and gyroscope measurements are corrupted by white noise [7].

Although the ZUPT-aided EKF is highly accurate compared to strap-down inertial integration, it still suffers from a small drift [3]. The A survey [2] on Location and Navigation Support for Emergency Responders conclude that pedestrian inertial navigation using a foot-mounted sensor and the zero-velocity assumption still has a too large position drift to be a feasible strategy in emergency scenarios.

Mag-SLAM has previously been used to achieve drift-reduction for foot-mounted position estimates using magnetic field measurements by [9]. The researchers in [9] combine odometry from a foot-mounted sensor with magnetic field norm measurements to achieve drift-free position estimates in two dimensions and uses a different method for creating the magnetic field map. Computationally tractable Mag-SLAM using Gaussian process regression in a Rao-Blackwellized particle filter was proposed by [10], and shows how on-board magnetic field measurements can be used to compensate for position estimation drift in three dimensions.

While the method proposed by [11] is limited to only two dimensions, the method proposed by [10] is not, but relies on an accelerometer, a gyroscope and a camera in a hand-held smartphone for obtaining the underlying pedestrian odometry.

We propose an algorithm for indoor position pedestrian estimation using a foot-mounted tri-axis gyroscope, accelerometer and magnetometer through magnetic field norm SLAM. We use the same assumptions that were used by [6] to create a low-drift position estimate on a foot-mounted sensor and a similar method for magnetic field mapping that was proposed by [10] to formulate a state-space model for the simultaneous estimation of the foot-mounted sensors position, velocity orientation and the magnetic field map, that is able to compensate for drift in the position estimate when positions are revisited. The new filter has the advantage of being accurate for shorter trajectories in a comparable way to the filter implemented by [6], and it compensates for drift in longer trajectories by recognising patterns in the magnetic field anomalies in a comparable way

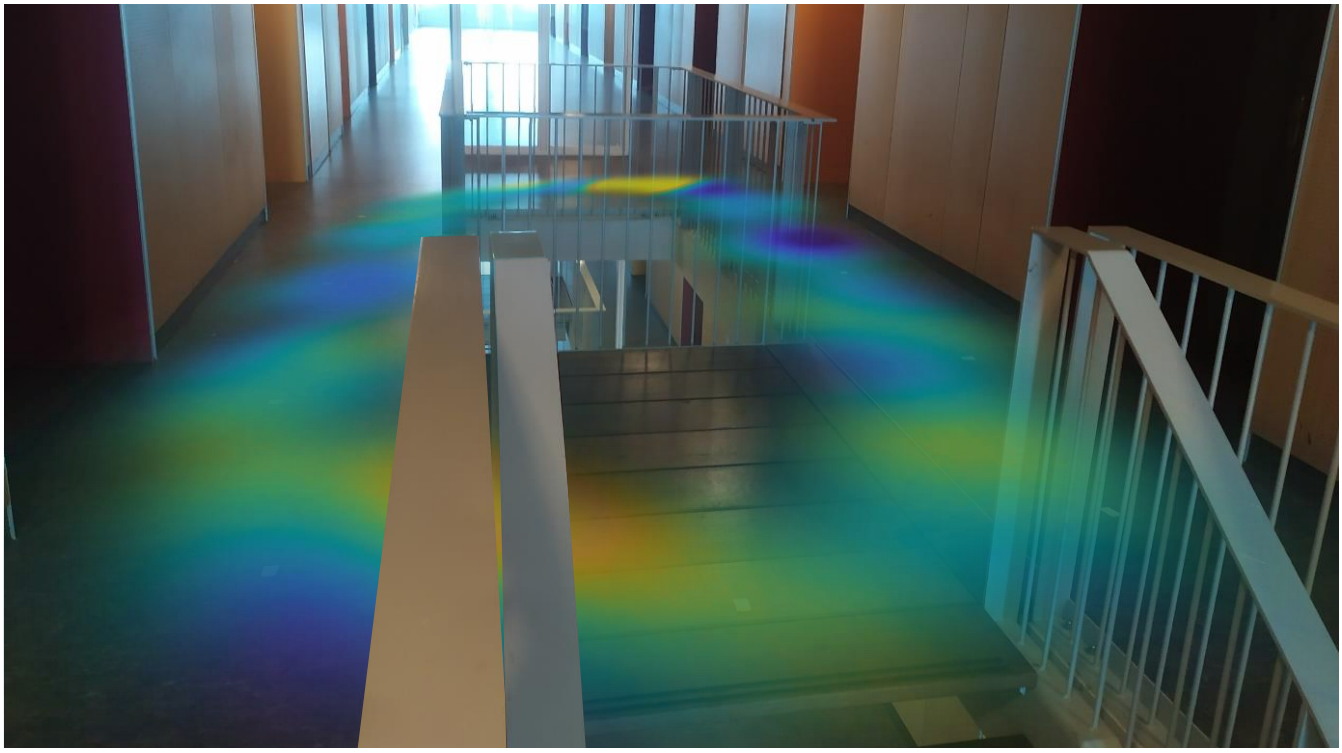


Fig. 1. Estimated magnetic field norm indoors from measurements from foot-mounted sensor

to the filter implemented by [10]. Our proposed algorithm finds a position estimate in three dimensions, and achieves similar accuracy as [9], while only using an accelerometer, gyroscope and magnetometer.

II. Motion model

Fig. 2. Foot-mounted sensor

The proposed algorithms estimate the position and orientation transformation from a world frame w to a sensor frame s. The sensor frame is attached to the foot-mounted sensor, with the origin at the sensors centre of mass, and directions following the orientation of the aligned tri-axis gyroscope and accelerometer. The sensor frame is illustrated in Figure 2. The origin of the world frame is equal to the origin of the sensor frame at time $t = 0$. The orientation of the sensor frame is chosen such that the gravitational force on the sensor in the world frame points down, i.e. $\mathbf{g}^w = [00 - 9.81]$, where 9.81 is assumed to be the strength of the local gravitational field. Otherwise, the orientation is chosen such that the yaw angle of the initial rotation in Euler angles from sensor to world frame is zero. The angular velocity of the sensor is the derivative of the change in orientation between the sensor and world frame. Assuming that the gyroscope provides a measurement of the angular velocity around each of the three axes, corrupted by additive white noise, the first-order discretisation of the kinematic motion model gives the following recursive expression for the orientation of the sensor at timestep t

$$\mathbf{q}^{\mathrm{ws}}{}_t = \mathbf{q}^{\mathrm{ws}}_{t-1} \odot \exp_q(\frac{(\mathbf{y}^{\mathrm{s}}_{gyr,t} - \mathbf{e}_{gyr,t})T}{2}), \tag{1}$$

where $\mathbf{q}^{\mathrm{ws}}{}_t$ is the rotation from world frame to sensor frame on quaternion form, $\mathbf{y}^{\mathrm{s}}_{gyr,t}$ is the gyroscope measurement, $\mathbf{e}_{gyr,t} \sim \mathcal{N}(0, \mathbf{\Sigma}_{gyr,t})$ is the white noise corrupting the gyroscope measurement, $\mathbf{R}_{gyr,t}$ is assumed to be a known covariance, $\odot$ is the quaternion product, $\exp_q$ is an operator transforming rotations on orientation-deviation form to a quaternion (see [8] for details). The acceleration of the sensor can be found by subtracting the gravity vector in body frame, according to

$$\mathbf{a}^{\mathrm{w}}_t = \mathbf{R}(\mathbf{q}^{\mathrm{ws}}{}_t)(\mathbf{y}^{\mathrm{w}}_{acc,t} - \mathbf{e}_{acc,t}) - \mathbf{g}^w, \tag{2}$$

where $\mathbf{y}_{acc,t}$ is the acceleration measurement, $\mathbf{e}_{acc,t} \sim \mathcal{N}(0, \mathbf{\Sigma}_{acc,t})$ is the white noise process corrupting the acceleration measurement, $\mathbf{R}(\cdot)$ is an operator transforming the rotation quaternion to a rotation matrix. The integral of acceleration gives the velocity, and discretising this equation gives that

$$\mathbf{v}_t^{\mathrm{w}} = \mathbf{v}_{t-1}^{\mathrm{w}} + T\mathbf{a}_t^{\mathrm{w}}, \tag{3}$$

where $\mathbf{v}_t^{\mathrm{w}}$ is the velocity of the sensor in world frame at time t, and T is the timestep of the discrete-time approximation. Similarly, the discrete integral of velocity is given by

$$\mathbf{p}_t^{\mathrm{w}} = \mathbf{p}_{t-1}^{\mathrm{w}} + T\mathbf{v}_t^{\mathrm{w}} \tag{4}$$

We will refer to Equations 1-4 as the dynamic model. By substituting $\mathbf{e}_{gyr,t} = \mathbf{0}$ and $\mathbf{e}_{acc,t} = \mathbf{0}$ into these equations we can find the expected value of the position at time t using the information from the dynamic model. This estimation technique is commonly referred to as dead-reckoning integration [7]. Simply the effect of integrating the white-noise of the acceleration measurement causes an expected variance of the dead-reckoning position estimate that grows cubically in time [7]. Ever-growing expected errors are commonly referred to as drift [8]. The gyroscope measurement will also make the orientation estimate drift, causing the estimated direction of the gravitational force to be wrong, which further increases the variance of the dead-reckoning velocity and position estimates.

During normal walk, approximately 40% of the time, the front of the foot is stationary [12]. When the front of the foot is stationary, the acceleration measurement is close to zero with low variance, and the gyroscope measurement is also close to zero with low variance [4]. The stance phase of the step can be detected by investigating when the function

$$\begin{aligned} &D(\mathbf{y}^{\mathrm{s}}_{acc,t-W-1:t}, \mathbf{y}^{\mathrm{s}}_{gyr,t-W-1:t}) \\ &= \frac{1}{W}\sum_{k=t-1}^{t}\left(\frac{1}{\sigma^2_{acc}}\left\|\mathbf{y}^{\mathrm{s}}_{acc,k} - g\frac{\bar{\mathbf{y}}^{\mathrm{s}}_{acc,t-W-1:t}}{\left\|\bar{\mathbf{y}}^{\mathrm{s}}_{acc,t-W-1:t}\right\|_2}\right\|_2^2\right. \\ &\left.+\frac{1}{\sigma^2_{gyr}}\left\|y^{\mathrm{s}}_{gyr,k}\right\|_2^2\right) \end{aligned} \tag{5}$$

is below a selected threshold γ [4]. The parameter g is the magnitude of the gravity field, the ration between σ^2_{gyr} and σ^2_{acc} needs to reflect the ratio between the gyroscope and accelerometer variance, and the parameter γ is a tunable detection threshold, while W is a tunable detection window [4]. The ZUPT-aided EKF improves upon the dead-reckoning position, velocity and orientation estimates by incorporating a pseudo-measurement every time the stance phase of the step is detected. The pseudo-measurement is modelled by [3] as

$$\mathbf{y}^w_{v,t} = 0 = \mathbf{v}^w_t + \mathbf{e}_{v,t}, \tag{6}$$

where $\mathbf{y}^w_{v,t}$ denotes the measurement that always is zero, and is assumed to be equal to the velocity of the foot $\mathbf{v}^w_t$ plus an additive white noise $\mathbf{e}_{v,t} \sim \mathcal{N}(0, \mathbf{\Sigma}_v)$ with a known covariance.

III. Magnetic field model

The magnetic field norm has rich spatial variation in indoor environments, which makes it possible to recognise previously visited areas based on the similarities in the magnetic field norm anomalies [10]. The magnetic field model used by [10] to perform simultaneous localisation and mapping use Reduced-Rank Gaussian process regression for estimation of a nonlinear function. The magnetic field is a three-component vector field, often denoted $\mathbf{H}(\mathbf{p}) : \mathbb{R}^3 \to \mathbb{R}^3$, where $\mathbf{p}$ is the position vector [13]. The magnetic field norm can therefore be considered a nonlinear function $\|\mathbf{H}(\mathbf{p})\| : \mathbb{R} \to \mathbb{R}^3$, mapping the position $\mathbf{p}$ to a magnetic field norm. The Gaussian process regression can be implemented with a squared exponential kernel function

$$\kappa_{SE}(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2_{SE}\exp\frac{-\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2l^2_{SE}}, \tag{7}$$

to include the assumption that the magnetic field norm is similar to the magnetic field norm variations in nearby locations, and variations typically have the magnitude σ_{SE} and length-scale l_{SE} [10].

The magnetic field is measured by a tri-axis magnetometer, so the magnetic field norm measurement can be found by taking the euclidean norm of this measurement. We assume the magnetic field norm measurement is normally distributed about the true magnetic field norm in the position of the measurement, according to

$$y_{m,t} = \|\mathbf{H}(\mathbf{p}^{\mathrm{w}}_t)\| + e_{m,t}, \tag{8}$$

where $e_{m,t} \sim \mathcal{N}(0, \sigma^2_m)$.

Recursive Reduced-Rank Gaussian process regression using Hilbert-Space methods can be used to estimate the magnetic field norm as a scaled sum of basis functions in a finite domain. The basis functions are defined as solutions to the negative Laplace operator that is subject to the Dirichlet boundary condition [14]. The basis functions are denoted $\phi_k : \mathbb{R}^3 \to \mathbb{R}$, and they solve

$$\begin{cases} -\nabla^2\phi_k(\mathbf{p}^{\mathrm{w}}) = \lambda_k\phi_k(\mathbf{p}^{\mathrm{w}}), & \mathbf{p}^{\mathrm{w}} \in \Omega \\ \phi_k(\mathbf{p}^{\mathrm{w}}) = 0, & \mathbf{p}^{\mathrm{w}} \in \partial\Omega \end{cases}. \tag{9}$$

The Gaussian process regression prediction in a position $\mathbf{p}^{\mathrm{w}}_t$ can be approximated using a linear sum of these basis functions according to

$$E[y_{m,t}] = \mathbf{\Phi}(\mathbf{p}^{\mathrm{w}}_t)\mathbf{m}_{t-d} \tag{10a}$$

$$E[(y_{m,t} - E[y_{m,t}])^2] = \mathbf{\Phi}(\mathbf{p}^{\mathrm{w}}_t)\mathbf{P}_{t-d}\mathbf{\Phi}(\mathbf{p}^{\mathrm{w}}_t)^T + \sigma^2_m, \tag{10b}$$

where $\mathbf{m}_{t-d}$ and $\mathbf{P}_{t-d}$ are computed recursively using all available measurements up until time $t-d$ where d is a delay, $\{y_{m,k}\}_{k=0}^{t-d}$ in initial position estimates $\{\tilde{\mathbf{p}}^{\mathrm{w}}_k\}_{k=0}^{t-d}$ according to

$$\mathbf{m}_0 = \mathbf{0} \tag{11a}$$
$$\mathbf{P}_{m,0} = \Lambda \tag{11b}$$
$$\mathbf{S}_k = \mathbf{\Phi}(\tilde{\mathbf{p}}_k^{\mathrm{w}})\mathbf{P}_{k-1}\mathbf{\Phi}(\tilde{\mathbf{p}}_k^{\mathrm{w}})^T + \sigma_m^2\mathbf{I}, \tag{11c}$$
$$\mathbf{K}_k = \mathbf{P}_{k-1}\mathbf{\Phi}(\tilde{\mathbf{p}}_k^{\mathrm{w}})^T\mathbf{S}_k^{-1}, \tag{11d}$$
$$\mathbf{m}_k = \mathbf{m}_{k-1} + \mathbf{K}_k(y_{m,k} - \mathbf{\Phi}(\tilde{\mathbf{p}}_k^{\mathrm{w}})\mathbf{m}_{k-1}), \tag{11e}$$
$$\mathbf{P}_{m,k} = \mathbf{P}_{m,k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T, \tag{11f}$$

where $\mathbf{\Phi}(\tilde{\mathbf{p}}_k^{\mathrm{w}})$ is the matrix

$$\mathbf{\Phi}(\mathbf{x}_{1:n}) = \mathbf{\Phi}(\{\mathbf{x}_i\}_{i=1}^n) = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_{N_m}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_n) & \cdots & \phi_{N_m}(\mathbf{x}_n) \end{bmatrix}, \tag{12}$$

and Λ is the matrix

$$\Lambda = \begin{bmatrix} S(\sqrt{\lambda_1}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S(\sqrt{\lambda_{N_m}}) \end{bmatrix}, \tag{13}$$

and the function $S(\sqrt{\lambda_k})$ is the spectral density of the squared exponential kernel [13]

$$S_{SE}(\sqrt{\lambda_k}) = \sigma_{SE}^2\left(2\pi l_{SE}^2\right)^{\frac{3}{2}}\exp(-\frac{\lambda_k l_{SE}^2}{2}), \tag{14}$$

with the same parameters σ_{SE} and l_{SE} as the Gaussian process kernel in equation 7. See [15] for details about magnetic field mapping using reduced-rank Gaussian process regression.

IV. Algorithm

Equations 1-3 are commonly linearised around the current estimate in order to apply the extended Kalman filter to include the zero-velocity pseudo measurements [16]. The high accuracy of the ZUPT-aided EKF confirms that equations 1-3 can safely be approximated by linearisation. The state of a nonlinear stochastic process with some linear substructures can be estimated using a Rao-Blackwellized particle filter [17]. A Rao-Blackwellized particle filter splits the state-space vector into a linear and a nonlinear part. The linear part of each particle's state is then estimated with a conditional Kalman filter, given the corresponding nonlinear part of the particle's state [18]. Mag-SLAM, in general, is often based on state-space models with both linear and nonlinear dynamics, and apply Rao-Blackwellized particle filters to estimate the pose and the magnetic field map ([19], [20], [10]). For the foot-mounted sensor, we consider both the linear and the close to linear structures as linear substructures in the state-space model and estimate the magnetic field, the orientation and the velocity as conditionally linear states given the position of each particle.

Algorithm 1 Mag-SLAM for foot-mounted sensor

Input: $\left\{\mathbf{y}_{acc,t}, \mathbf{y}_{gyr,t}, y_{m,t}\right\}_{t=1}^N$

Output: $\left\{\hat{\mathbf{p}}_{t|t}\right\}_{t=1}^N$

Initialisation :$\mathbf{p}_{0|-1}^{\mathrm{w},i} = \mathbf{0}$, $\hat{\mathbf{v}}_{0|-1}^{\mathrm{w},i} = \mathbf{0}$, $\hat{\mathbf{q}}_{0|-1}^{\mathrm{wb},i} = \mathbf{q}_0$, $\mathbf{P}_{0|-1}^i = \mathbf{P}_0$, $\hat{\mathbf{m}}_{0|-1}^i = \mathbf{0}$, $\mathbf{P}_{m,0|-1}^i = \mathbf{\Lambda}$, w_t^i

for $t = 1$ to N **do**

  if $(T(\mathbf{y}_{acc,(t-5):t}, \mathbf{y}_{gyr,(t-5):t}) < \gamma)$ **then**

    KF MEASUREMENT UPDATE:

$$\mathbf{S}_t^i = \mathbf{R}_{\mathbf{v},t} + \mathbf{H}_t\mathbf{P}_{t|t-1}^i\mathbf{H}_t^T \tag{15a}$$
$$\mathbf{K}_t^i = \mathbf{P}_{t|t-1}^i\mathbf{H}_t^T(\mathbf{S}_t^i)^{-1} \tag{15b}$$
$$\begin{bmatrix} \hat{\mathbf{v}}_{t|t}^{\mathrm{w},i} \\ \hat{\boldsymbol{\eta}}_{t|t}^{\mathrm{w},i} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{v}}_{t|t-1}^{\mathrm{w},i} \\ \hat{\boldsymbol{\eta}}_{t|t-1}^{\mathrm{w},i} \end{bmatrix} + \mathbf{K}_t^i(\mathbf{y}_{v,t} - \hat{\mathbf{v}}_{t|t-1}^{\mathrm{w},i}) \tag{15c}$$
$$\mathbf{P}_{t|t}^i = \mathbf{P}_{t|t-1}^i - \mathbf{K}_t^i\mathbf{H}_t\mathbf{P}_{t|t-1}^i \tag{15d}$$

    Relinearise using

$$\hat{\mathbf{q}}_{t|t}^{\mathrm{wb},i} = \exp_q(\frac{\hat{\boldsymbol{\eta}}_{t|t}^{\mathrm{w},i}}{2}) \odot \hat{\mathbf{q}}_{t|t-1}^{\mathrm{wb},i} \tag{16a}$$
$$\hat{\boldsymbol{\eta}}_{t|t}^{\mathrm{w},i} \leftarrow \mathbf{0} \tag{16b}$$

  end if

  PF MEASUREMENT UPDATE

$$w_t^i = p(y_{m,t} \mid \mathbf{p}_{t|t-1}^{\mathrm{w},i}, \hat{\mathbf{m}}_{t|t-1}, \mathbf{P}_{m,t|t-1}^i)w_{t-1}^i \tag{17}$$

  Normalise weights such that $\sum_i^N w_t^i = 1$.

  if $M_{eff} > \frac{2}{3}M$ **then**

    Resample, set $w_t^i = \frac{1}{M}$

  end if

  KF MEASUREMENT UPDATE

$$\mathbf{S}_t^i = \Phi(\mathbf{p}_{t|t}^{\mathrm{w},i})\mathbf{P}_{m,t|t-1}^i\Phi(\mathbf{p}_{t|t}^{\mathrm{w},i})^T + \sigma_m^2 \tag{18a}$$
$$\mathbf{K}_t^i = \mathbf{P}_{t|t-1}^i\Phi(\mathbf{p}_{t|t}^{\mathrm{w},i})^T(\mathbf{S}_t^i)^T \tag{18b}$$

  PF DYNAMIC UPDATE

$$\hat{\mathbf{v}}_t^{\mathrm{w},i} \sim \mathcal{N}(\hat{\mathbf{v}}_{t|t}^{\mathrm{w},i}, \mathbf{H}_t\mathbf{P}_{t|t-1}^i\mathbf{H}_t^T) \tag{19a}$$
$$\mathbf{p}_{t+1|t}^{\mathrm{w},i} = \mathbf{p}_{t|t}^{\mathrm{w},i} + T\hat{\mathbf{v}}_t^{\mathrm{w},i} \tag{19b}$$

  KF DYNAMIC MEASUREMENT UPDATE

$$\mathbf{L}_t^i = T^2\mathbf{H}_t\mathbf{P}_{t|t}^i\mathbf{H}_t^T \tag{20a}$$
$$\mathbf{K}_t^i \leftarrow \mathbf{P}_{t|t}^i\mathbf{H}_t^T T(\mathbf{L}_t^i)^{-1} \tag{20b}$$
$$\begin{bmatrix} \hat{\mathbf{v}}_{t|t}^{\mathrm{w},i} \\ \hat{\boldsymbol{\eta}}_{t|t}^{\mathrm{w},i} \end{bmatrix} \leftarrow \begin{bmatrix} \hat{\mathbf{v}}_{t|t}^{\mathrm{w},i} \\ \hat{\boldsymbol{\eta}}_{t|t}^{\mathrm{w},i} \end{bmatrix} + \mathbf{K}_t^i(\mathbf{p}_{t+1|t}^{\mathrm{w},i} - \mathbf{p}_{t|t}^{\mathrm{w},i} - T\hat{\mathbf{v}}_{t|t}^{\mathrm{w},i}) \tag{20c}$$
$$\mathbf{P}_{t|t}^i \leftarrow \mathbf{P}_{t|t}^i - T\mathbf{K}_t^i\mathbf{H}_t\mathbf{P}_{t|t}^i \tag{20d}$$

  KF DYNAMIC UPDATE

$$\hat{\mathbf{v}}_{t+1|t} = \hat{\mathbf{v}}_{t|t}^{\mathrm{w},i} + T(\mathbf{R}(\hat{\mathbf{q}}^{\mathrm{w},i})\mathbf{y}_{acc,t}^{\mathrm{b}} - \mathbf{g}^{\mathrm{w}}) \tag{21a}$$
$$\hat{\boldsymbol{\eta}}_{t+1|t}^{\mathrm{w},i} = \hat{\boldsymbol{\eta}}_{t|t}^{\mathrm{w},i} \tag{21b}$$
$$\mathbf{P}_{t+1|t}^i = \mathbf{F}_t\mathbf{P}_{t|t}^i\mathbf{F}_t^T + \mathbf{G}_t^i\begin{bmatrix} \mathbf{R}_{acc,t} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{gyr,t} \end{bmatrix}\mathbf{G}_t^{i,T} \tag{21c}$$

end for

The state-space model with equations 1-3 linearised about each estimate and the magnetic field measurement model, is a subset of the triangular state-space model that [17] uses to formulate a Rao-Blackwellized. By applying this Rao-Blackwellized particle filter algorithm IV is obtained. The matrices $\mathbf{H}_t$ and $\mathbf{F}_t$are given by

$$\mathbf{F}_t = \begin{bmatrix} \mathbf{I}_3 & T[\mathbf{y}^w_{acc,t}\times] \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix}, \tag{22}$$

$$\mathbf{G}_t = \begin{bmatrix} \mathbf{R}(\hat{\mathbf{q}}^{wb,i}_{t-1|t-1})T & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}(\hat{\mathbf{q}}^{wb,i}_{t-1|t-1})T \end{bmatrix}, \tag{23}$$

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \end{bmatrix}, \tag{24}$$

where $\hat{\mathbf{q}}^{wb,i}_{t-1|t-1}$ is the orientation estimate of each particle.

In applying algorithm the Rao-Blackwellized particle filter to the state-space model in equations 1-3 and the magnetic field measurement model, we made one modification to give an algorithm that can perform convergent and computationally tractable SLAM on data from a foot-mounted sensor. The particle filter measurement update given the pseudo zero-velocity measurements are omitted. The particle filter measurements were attempted included but found to reduce the spread of the particle cloud so much that all the particles positions were the same at the end of each footstep. What this means for the final algorithm, is that the zero-velocity pseudo measurement is not used to distinguish which particles are more likely. The weights of the particles are only evaluated by investigating the magnetic field. This ensures that the spread of the particle cloud is large enough to discover revisitation of previous positions.

V. Experimental Results

The experimental results demonstrate how the proposed Mag-SLAM algorithm for a foot-mounted sensor compensates for drift by recognising patterns in the magnetic field norm anomalies in previously visited positions. The results focus on illustrating the removal of drift in a short trajectory in a single tile, although the method scales to arbitrary long trajectories, and it was demonstrated by [10] that the method can be implemented using multiple tiles to scale to arbitrarily large areas.

The position estimation results are compared with the position estimation from an open-source implementation of the ZUPT-aided EKF [5] run on the same measurements. The parameters for the ZUPT-aided EKF were chosen to give as good position estimates as possible given the test subjects walking patterns. As the sensor is front-mounted, the position estimate drifts more compared to what is reported for heel-mounted sensors [4]. This drift can be used to demonstrate visibly in a small experiment how the Mag-SLAM algorithm can compensate for drift. Front-mounted sensors are also easily available off-the-shelf compared to sensors integrated into the heel of a boot. The parameters for the zero-velocity detector were kept to their original values from the open-source implementation, except for the threshold γ, that was set to $3 \cdot 10^5$. The accelerometer and gyroscope covariance matrices were set to $\mathbf{R}_{acc,t} = \sigma_a^2 T\mathbf{I}$ with $\sigma_a = 0.12ms^{-3/2}$ and $\mathbf{R}_{gyr,t} = \sigma_\omega^2 T\mathbf{I}$ with $\sigma_\omega = 0.006^\circ s^{-1/2}$, respectively. The initial covariance for the position, velocity and orientation was set to $\mathbf{P}_0 = 0.12T\mathbf{I}$. The zero-velocity parameters, the gyroscope and the accelerometer measurement noise, were given the same values in the proposed Mag-SLAM algorithm. The initial velocity and orientation covariance in the proposed Mag-SLAM algorithm was set to $\mathbf{P}_0 = 0.12T\mathbf{I}$, to reflect the initial covariance in the ZUPT-aided EKF.

The domain Ω is chosen as the smallest possible cuboid where each point is no closer than $2m$ of an initial position trajectory estimate. The initial position trajectory estimate is obtained by running the ZUPT-aided EKF implemented by [6], and visually correcting for position drift by adding an incrementally larger position correction to each position estimate. In simultaneous localisation and mapping, there are no measurements available to optimise Gaussian process hyperparameters [10]. We, therefore, set the hyper-parameters for the Gaussian process a-priory. The magnetic field measurements were normalised, so we expected magnetic field measurements to have a magnitude of around $\sigma^2_{SE} = 1$. As the magnetic field anomalies tend to rapidly change along the z-direction close to the floor [21], we set the length-scale $l_{SE} = 0.3$ quite small to accurately estimate the details of the magnetic field norm variations in the 3D space the foot-mounted sensor moves through. The measurement noise was set to $\sigma_m = 0.01$, slightly higher than the expected measurement noise in the sensors data-sheet, to compensate for errors in the model. The magnetic field measurements were down-sampled to $10Hz$. This reduces the number of measurement updates that need to be executed while still collecting enough measurements to create a magnetic field map.

Measurements were collected using an Xsens MTi 10 inertial sensor. Figure 2 shows the experimental setup on a foot. To evaluate the performance of the estimation, checkpoints were marked with tape on the floor in the indoor environment where the measurements were collected, as shown in Figure 4. To evaluate the accuracy of the position estimates, the root mean squared error of the position estimate is calculated by detecting the difference in position estimates when revisiting the checkpoints.

To investigate how many basis functions were necessary to accurately approximate the GP prediction, 200 simulated magnetic field measurements were drawn from the GP prior with the squared-exponential kernel with the chosen hyper-parameters, and the prediction accuracy of the standard GP in 100 non-overlapping test-positions were compared with the accuracy of the Reduced-Rank GP prediction in the same 100 test positions. The result of this experiment is displayed in Figure 3. The Reduced-Rank GP prediction converges to the true GP prediction at approximately 2000 basis functions, so $N_m = 2000$ basis functions is used to create the magnetic field map. It is worth mentioning that using 2000 basis functions to store the magnetic field in the particle filter is computationally demanding. This is because each of the M

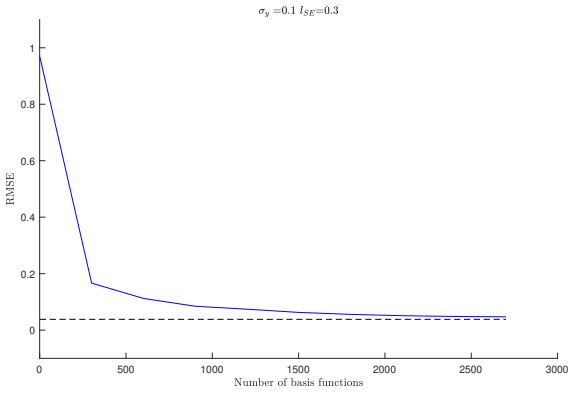


Fig. 3. The blue line shows the RMSE of a reduced-rank GP prediction against the number of basis functions N_m that was used to approximate the true GP prediction. The black dotted line shows the RMSE of the true GP prediciton.

Fig. 4. The twelve labelled checkpoints used for collecting measurements

particles has a magnetic field state vector estimate $\hat{m}_t^i$ that has dimension N_m, and a magnetic field state covariance P_m^i that has dimension $N_m \times N_m$. The number of required basis functions reduces when the length scale of the anomalies l_{SE} relative to the size of the domain Ω increases [10]. A way to reduce the number of necessary basis functions is, therefore, to use multiple tiles that each map a smaller domain. This method was implemented by [10], and the results they present indicate that executing the Reduced-Rank GP regression for magnetic field mapping as it is done in this thesis in multiple tiles rather than a single domain covering all of the estimated positions is feasible.

Mag-SLAM has previously been demonstrated a usable tool for drift removal in combination with drifting position estimates from hand-held cellphones [10], and cleaning robots [22], that both have smoothly varying movement patterns. The results in Figure 7 and Figure 6 shows that Mag-SLAM is also possible for 3D position estimation using a drifting position estimate from a foot-mounted sensor.

When the ZUPT-aided extended Kalman filter is used to estimate the position, the position estimate at the known checkpoints drifts over time, as can be seen in Figure 5. The root mean squared error of the ZUPT-aided EKF position estimate at the checkpoints is 1.68 meters for the dataset used in this figure.

The trajectory and magnetic field norm map of the current highest-weight particle is displayed for four different times in figure 6. The magnetic field estimation certainty is visibly higher near the checkpoints. This is because the checkpoints mark the location of the stance phase of the foot. In these locations, more magnetic field measurements are available, which gives a more confident prediction. The details of the magnetic field anomalies become more apparent when the area is visited multiple times, as can be seen by comparing Figure 6, that shows the magnetic field map estimate after one lap, with Figure 6 (d), that shows the magnetic field map at the end of the six laps around the 12 checkpoints. This fits with the fact that Gaussian process regression with a squared exponential kernel predicts the function value more confidently close to previous measurements [23]. The magnetic field is visibly rich in spatial variation, which makes it possible for the particle filter to compensate for drift in the position estimate, as can be seen in Figure 7. The figure compares the position trajectory for six laps around the marked checkpoints with the positions estimated with the ZUPT-aided EKF with the estimate using the proposed Mag-SLAM algorithm. The RMSE of the revisitation of the checkpoints for the zero-velocity aided extended Kalman filter for this experiment is 1.68 m, while the RMSE of the checkpoint position estimates for the Mag-SLAM method is 0.16 m.

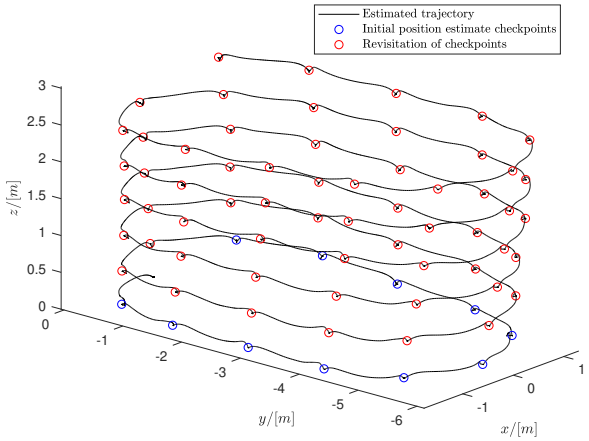
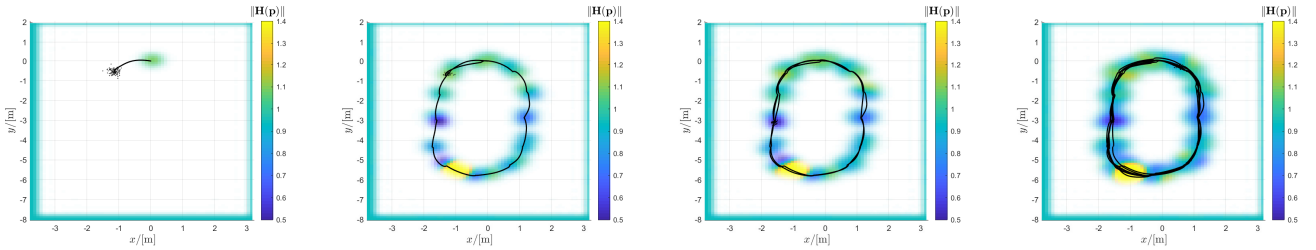


Fig. 5. Position estimate using the ZUPT-aided extended Kalman filter. The estimated positions at the checkpoints are marked with red circles.



(a) $t = 2s$ (b) $t = 19s$ (c) $t = 38s$ (d) $t = 100s$

Fig. 6. The blue trajectory show the current highest weight particle, and the positions of all the particle are marked with black dots. The colour correspond to the predicted magnetic field norm value, and the opacity is inversely proportionate with the variance of the estimate.

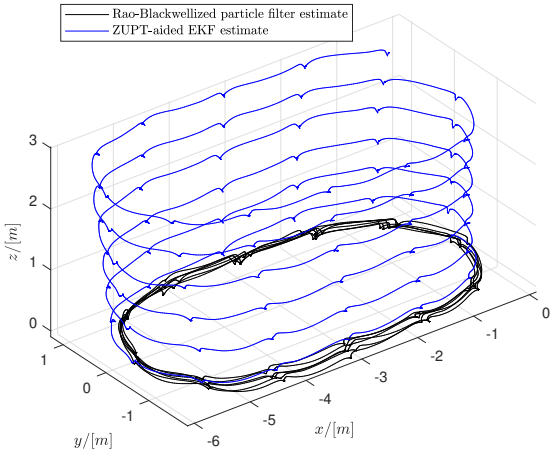


Fig. 7. Comparison of the position trajectories obtained from the zero-velocity aided extended Kalman filter and the proposed Mag-SLAM algorithm for foot-mounted sensor

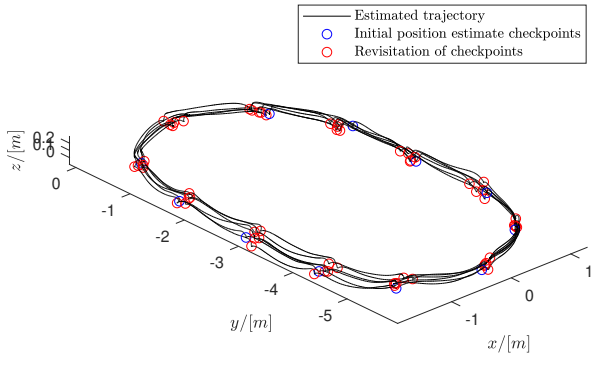


Fig. 8. Estimated positions of checkpoints in the estimated trajectory using Mag-SLAM on a foot-mounted sensor

VI. Conclusion

The proposed method for Mag-SLAM on a foot-mounted sensor remedies the drift in the position estimate from the ZUPT-aided EKF. It can be extended to scale to an arbitrarily large area by using multiple tiles, as was demonstrated by [10]. Further research could be in the direction of making a map of all three magnetic field components instead of only the magnetic field norm or investigate if the method can be made tolerant to changes in the magnetic field due to building collapse or extreme temperatures. Further research could also investigate if maps from multiple sources could be fused to one map so that in a team of search and rescue workers, the measurements from one worker can benefit the whole team. Another natural extension is to investigate how different movement patterns can be taken into account, such as crawling, jumping or running.

References

[1] V. Nunavath, A. Prinz, and T. Comes, "Identifying first responders information needs: Supporting search and rescue operations for fire emergency response," *International Journal of Information Systems for Crisis Response and Management*, vol. 8, pp. 25–46, Jan. 2016.

[2] C. Fischer and H. Gellersen, "Location and navigation support for emergency responders: A survey," *IEEE Pervasive Computing*, vol. 9, pp. 38–47, Jan. 2010.

[3] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, pp. 38–46, Nov. 2005.

[4] I. Skog, J. Nilsson, and P. Händel, "Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, (Zürich, Switzerland), pp. 1–6, Sept. 2010.

[5] I. Skog, "Openshoe matlab framework." http://www.openshoe.org/?page_id=362.

[6] J. Nilsson, I. Skog, P. Händel, and K. V. S. Hari, "Foot-mounted INS for everybody - an open-source embedded implementation," in *Proceedings of IEEE/ION Position, Location and Navigation Symposium*, (Myrtle Beach, SC, USA), pp. 140–145, April 2012.

[7] O. J. Woodman, "An introduction to inertial navigation," tech. rep., Computer laboratory University of Cambridge, 2007.

[8] M. Kok and T. Schön, "Magnetometer calibration using inertial sensors," *IEEE Sensors Journal*, vol. 16, Jan. 2016.

[9] P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. Julian, M. Puyol, M. Khider, M. Lichtenstern, and L. Bruno, "Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, (Montbeliard-Belfort, France), Oct. 2013.

[10] M. Kok and A. Solin, "Scalable magnetic field slam in 3D using Gaussian process maps," in *Proceedings of the 20th International Conference on Information Fusion*, (Cambridge, UK), pp. 1353–1360, July 2018.

[11] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, (Orlando, Florida), Sept. 2009.

[12] S. Park and Y. Suh, "A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems," *Sensors (Basel, Switzerland)*, vol. 10, pp. 9163–78, Oct. 2010.

[13] N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson, "Modeling magnetic fields using Gaussian processes," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, (Vancouver, Canada), pp. 3522–3526, May 2013.

[14] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," *Statistics and Computing*, vol. 30, pp. 419–446, Jan. 2014.

[15] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä, "Modeling and interpolation of the ambient magnetic field by gaussian processes," *IEEE Transactions on Robotics*, vol. 34, pp. 1112–1127, Aug 2018.

[16] C. Fischer, P. Sukumar, and M. Hazas, "Tutorial: Implementing a pedestrian tracker using inertial sensors," *Pervasive Computing, IEEE*, vol. 12, pp. 17–27, Apr. 2013.

[17] T. B. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, pp. 2279–2289, July 2005.

[18] F. Gustafsson, "Statistical sensor fusion," 2013. Studentlitteratur AB.

[19] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, "Simultaneous localization and mapping using ambient magnetic field," in *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, (Utah, USA), pp. 14–19, 2010.

[20] J. Jung, S.-M. Lee, and H. Myung, "Indoor mobile robot localization and mapping based on ambient magnetic fields and aiding radio sources," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, pp. 1922–1934, Nov. 2015.

[21] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavi, and M. Wiseman, "Indoor location sensing using geo-magnetism," pp. 141–154, Jan. 2011.

[22] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, "Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task," in *IEEE 15th International Conference on Advanced Robotics: New Boundaries for Robotics*, (Tallin, Estonia), pp. 198 – 203, July 2011.

[23] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 425–437, Mar. 2002.

Bibliography

[1] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, pp. 38–46, Nov. 2005.

[2] M. Kok and A. Solin, "Scalable magnetic field slam in 3D using Gaussian process maps," in *Proceedings of the 20th International Conference on Information Fusion*, (Cambridge, UK), pp. 1353–1360, Jul. 2018.

[3] C. Fischer and H. Gellersen, "Location and navigation support for emergency responders: A survey," *IEEE Pervasive Computing*, vol. 9, pp. 38–47, Jan. 2010.

[4] V. Nunavath, A. Prinz, and T. Comes, "Identifying first responders information needs: Supporting search and rescue operations for fire emergency response," *International Journal of Information Systems for Crisis Response and Management*, vol. 8, pp. 25–46, Jan. 2016.

[5] S. Beauregard, Widyawan, and M. Klepal, "Indoor PDR performance enhancement using minimal map information and particle filters," in *Proceedings of IEEE/ION Position, Location and Navigation Symposium*, (Monterey, CA), pp. 141–147, May 2008.

[6] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," in *Proceedings of Ubiquitous Positioning Indoor Navigation and Location Based Service*, (Kirkkonummi, Finland), pp. 1–10, Oct. 2010.

[7] O. J. Woodman, "An introduction to inertial navigation," tech. rep., Computer laboratory University of Cambridge, 2007.

[8] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Foundations and Trends in Signal Processing*, vol. 11, no. 1-2, 2017.

[9] J. Nilsson, I. Skog, P. Händel, and K. V. S. Hari, "Foot-mounted INS for everybody - an open-source embedded implementation," in *Proceedings of IEEE/ION Position, Location and Navigation Symposium*, (Myrtle Beach, SC, USA), pp. 140–145, April 2012.

[10] I. Skog, J. Nilsson, and P. Händel, "Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, (Zürich, Switzerland), pp. 1–6, Sept. 2010.

[11] J.-O. Nilsson, D. Zachariah, I. Skog, and P. Händel, "Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging," *EURASIP Journal on Advances in Signal Processing*, vol. 164, Apr. 2013.

[12] K. Yamazaki, K. Kato, K. Ono, H. Saegusa, K. Tokunaga, Y. Iida, S. Yamamoto, K. Ashiho, K. Fujiwara, and N. Takahashi, "Analysis of magnetic disturbance due to buildings," *IEEE Transactions on Magnetics*, vol. 39, pp. 3226–3228, Sep 2003.

[13] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, "Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning

task," in *IEEE 15th International Conference on Advanced Robotics: New Boundaries for Robotics*, (Tallin, Estonia), pp. 198 – 203, July 2011.

[14] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, (Orlando, Florida), Sept. 2009.

[15] C. Fischer, P. Sukumar, and M. Hazas, "Tutorial: Implementing a pedestrian tracker using inertial sensors," *Pervasive Computing, IEEE*, vol. 12, pp. 17–27, Apr. 2013.

[16] Lauro Ojeda and Johann Borenstein, "Non-GPS navigation for security personnel and first responders," *Journal of Navigation*, vol. 60, Aug. 2007.

[17] P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. Julian, M. Puyol, M. Khider, M. Lichtenstern, and L. Bruno, "Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, (Montbeliard-Belfort, France), Oct. 2013.

[18] M. Angermann, M. Frassl, M. Doniec, B. J. Julian, and P. Robertson, "Characterization of the indoor magnetic field for applications in localization and mapping," in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, (Sydney, Australia), pp. 1–9, Nov 2012.

[19] M. Frassl, M. Angermann, M. Lichtenstern, P. Robertson, B. J. Julian, and M. Doniec, "Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Tokyo, Japan), pp. 913–920, Nov 2013.

[20] Hao Lu, Jiyin Zhao, Xiaomeng Li, and Jianpo Li, "A new method of double electric compass for localization," in *Proceedings of World Congress on Intelligent Control and Automation*, vol. 2, (Dalian, China), pp. 5277–5281, 2006.

[21] Ho-Duck Kim, Sang-Wook Seo, In-hun Jang, and Kwee-Bo Sim, "SLAM of mobile robot in the indoor environment with digital magnetic compass and ultrasonic sensors," in *Proceedings of International Conference on Control, Automation and Systems*, (Seoul, South Korea), pp. 87–90, Oct. 2007.

[22] J. Callmer, D. Törnqvist, and F. Gustafsson, "Robust heading estimation indoors using convex optimization," in *Proceedings of the 16th International Conference on Information Fusion*, (Istanbul, Turkey), pp. 1173–1179, July 2013.

[23] I. Skog, "Openshoe matlab framework." `http://www.openshoe.org/?page_id=362`. KTH Royal Institute of Technology, Stockholm, Sweden and Indian Institute of Science, Bangalore, India.

[24] Z. Chen, "Bayesian filtering: From kalman filters to particle filters, and beyond," *Statistics*, vol. 182, Jan. 2003.

[25] J. Haverinen and A. Kemppainen, "A global self-localization technique utilizing local anomalies of the ambient magnetic field," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Kobe, Japan), pp. 3142–3147, May 2009.

[26] F. Gustafsson, "Statistical sensor fusion," 2013. Studentlitteratur AB.

[27] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," *Statistics and Computing*, vol. 30, pp. 419–446, Jan. 2014.

[28] H.-S. Kim, W. Seo, and K. Baek, "Indoor positioning system using magnetic field map navigation and an encoder system," *Sensors*, vol. 17, p. 651, Mar. 2017.

[29] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, 2007.

[30] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[31] T. B. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, pp. 2279–2289, July 2005.

[32] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 425–437, Mar. 2002.

[33] A. Solin, S. Särkkä, J. Kannala, and E. Rahtu, "Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning," in *Proceedings of European Navigation Conference*, (Helsinki, Finland), pp. 1–9, May 2016.

[34] N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson, "Modeling magnetic fields using Gaussian processes," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, (Vancouver, Canada), pp. 3522–3526, May 2013.

[35] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä, "Modeling and interpolation of the ambient magnetic field by gaussian processes," *IEEE Transactions on Robotics*, vol. 34, pp. 1112–1127, Aug 2018.

[36] A. Solin and M. Kok, "Know your boundaries: Constraining gaussian processes by variational harmonic features," in *Proceedings of International Conference on Artificial Intelligence and Statistics*, (Okinawa, Japan), 2019.

[37] W. N. Everitt, G. I. Markovic, and A. N. Ilic, *Theory of linear operators in Hilbert space*. Pitman, 1981.

[38] J. Jung, S.-M. Lee, and H. Myung, "Indoor mobile robot localization and mapping based on ambient magnetic fields and aiding radio sources," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, pp. 1922–1934, Nov. 2015.

[39] F. Lindsten, *Rao-Blackwellised particle methods for inference and identification*. PhD thesis, Lindköping University, June 2011.

[40] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, "Simultaneous localization and mapping using ambient magnetic field," in *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, (Utah, USA), pp. 14–19, 2010.

[41] O. Tickoo, *Making sense of sensors: end-to-end algorithms and infrastructure design from wearable-devices to data center*. Apress., 2017.

[42] J. Hol, *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. PhD thesis, 2011. Linköping University.

[43] M. Kok and T. Schön, "Magnetometer calibration using inertial sensors," *IEEE Sensors Journal*, vol. 16, Jan. 2016.

[44] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[45] R. Rossi, "Chapter 3 - clothing for protection against heat and flames," in *Protective Clothing*, pp. 70 – 89, Woodhead Publishing, 2014.

[46] H. Takeuchi, Y. Yogo, T. Hattori, T. Tajima, and T. Ishikawa, "High-temperature magnetization characteristics of steels," *ISIJ International*, vol. 57, pp. 1883–1886, Oct. 2017.

[47] L. Niu, "A survey of wireless indoor positioning technology for fire emergency routing," in *IOP Conference Series: Earth and Environmental Science*, vol. 18, (Sarawak, Malaysia), Jan. 2014.

[48] A. Kemppainen, J. Haverinen, I. Vallivaara, and J. Röning, "Near-optimal SLAM exploration in Gaussian processes," in *Proceedings of IEEE Conference on Multisensor Fusion and Integration*, vol. 40, (Salt Lake City, Utah, USA), pp. 7–13, Sep. 2010.

[49] M. Kok, "Magnetic field SLAM." Guest Lecture, 2018. Delft University of Technology.

[50] K. B. Petersen and M. S. Pedersen, "The matrix cookbook." Technical University of Denmark, Nov. 2012.

[51] J. Candela and C. Rasmussen, "Analysis of some methods for reduced rank Gaussian process regression," in *Lecture Notes in Computer Science, Switching and Learning in Feedback Systems, European Summer School on Multi-Agent Control*, (Maynooth, Ireland), pp. 98–127, Sept. 2003.