Train Localization with Magnetic Field Distortions and Bayesian Methods

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Benjamin Siebler

Tag der mündlichen Prüfung:	03. Juli 2026
1. Referent:	Prof. Dr.-Ing. Uwe D. Hanebeck
2. Referent:	Prof. Dr. Gustaf Hendeby

Zusammenfassung

Die Verlagerung eines Anteils des Personen- und Güterverkehrs von der Straße auf die Schiene hat das Potential die vom Verkehrssektor ausgestoßenen Treibhausgase signifikant zu reduzieren. Eine Verlagerung auf die Schiene ist aber nur dann durchführbar, wenn die Transportleistung des Schienennetzes entsprechend erhöht wird. Diese Erhöhung kann mit zusätzlicher Schieneninfrastruktur oder durch eine Kapazitätssteigerung der existierenden Infrastruktur erreicht werden. Im direkten Vergleich erscheint die Steigerung der Kapazität existierender Strecken attraktiver, da das Bauen neuer Schieneninfrastruktur mit hohen Kosten, einem hohen Platzbedarf und langwierigen Planungs- und Bauphasen verbunden ist. Eine Steigerung der Kapazität ist dabei durch eine engere Taktung der Züge und eine Reduzierung der teils kilometerlangen Sicherheitsabstände realisierbar. Erreicht werden kann dies durch eine Erhöhung des Automatisierungsgrades des Schienenverkehres und die Entwicklung der dafür notwendigen Technologien.

Eine der Schlüsseltechnologien für die Automatisierung ist die absolute Lokalisierung aller Fahrzeuge im Schienennetz. Die absolute Position ist insbesondere für die Optimierung des Verkehrsflusses und die Manöversteuerung einzelner Züge essentiell. Im Vergleich zum Straßenverkehr spielt die Bestimmung der absoluten Position der Fahrzeuge im Schienenverkehr eine wichtigere Rolle, da durch die langen Bremswege eine reine Umfelderfassung mit Kameras, Laserscannern oder vergleichbaren Sensoren nicht für einen sicheren Betrieb ausreichend ist. Diese Arbeit befasst sich daher mit der Entwicklung und Evaluation eines neuartigen Lokalisierungssystems für Schienenfahrzeuge.

Das entwickelte Lokalisierungssystem basiert auf ortsabhängigen Verzerrungen des Magnetfeldes. Die Verzerrungen entstehen aus der Wechselwirkung des Erdmagnetfeldes mit magnetischen Materialien in der Eisenbahninfrastruktur oder in Gebäuden nahe der Strecke. Für die Lokalisierung muss zunächst eine Karte der Verzerrungen erstellt werden. Die Karte ordnet jeder Position im Schienennetz den dort zu erwartenden magnetischen Vektor zu. Die Position eines Zuges der sich durch das kartierte Schienennetz bewegt, kann dann anschließend durch einen Abgleich seiner Magnetometermessungen mit der vorab aufgezeichneten Karte bestimmt werden.

In dieser Arbeit werden die generellen Eigenschaften des Magnetfelds in Eisenbahnumgebungen untersucht und basierend darauf geeignete Lokalisierungsalgorithmen abgeleitet. Konkret werden Bayes'sche Schätzfilter entwickelt und untersucht, die durch eine rekursive Verarbeitung eine echtzeitfähige Magnetfeld-basierte Lokalisierung ermöglichen. Die entwickelten Filter berücksichtigen dabei explizit das Auftreten von Störgrößen in den Messungen und adressieren das Problem der Magnetometerkalibrierung. Die Magnetometerkalibrierung spielt eine zentrale Rolle für die praktische Anwendbarkeit der Magnetfeld-basierten Zuglokalisierung, da ohne Kalibrierung der Abgleich der Messdaten mit der Magnetfeldkarte erschwert oder verhindert wird. Die Kalibrierung ist dabei nicht nur von den Eigenschaften des Magnetometers abhängig, sondern auch vom Zug, in dem der Sensor montiert ist. Die Zugabhängigkeit entsteht dabei durch die Wechselwirkung von magnetischem

Material im Zug und dem externen Magnetfeld. Wird keine Kalibrierung durchgeführt, ist es unter Umständen nötig, ein separate Magnetfeldkarte für jeden zu lokalisierenden Zug zu erstellen. Um dies zu verhindern, wurde ein neuartiges Verfahren für die Kalibrierung entwickelt, das sich, im Gegensatz zu gängigen Verfahren, auch für Schienenfahrzeuge eignet. Gängige Verfahren benötigen eine Rotation des Magnetometers im Raum während es auf der zu lokalisierenden Plattform verbaut ist. Die Rotation muss dabei in einem homogenen Magnetfeld durchgeführt werden und hat das Ziel, die Beobachtbarkeit der Parameter sicherzustellen. Im konkreten Fall würde das bedeuten, dass der Sensor zusammen mit dem Zug im Raum rotiert werden muss. Aufgrund der großen Masse und Ausdehnung des Zuges, ist dies nicht ohne erheblichen Aufwand umsetzbar. Im Rahmen der Arbeit wurde daher ein Schätzfilter entwickelt, das die Kalibrierparameter simultan mit der Position bestimmt und keine Rotation erfordert. Das Filter nutzt dabei, dass die Beobachtbarkeit der Parameter auch durch die Bewegung des Zuges in einem inhomogenen Magnetfeld erwirkt werden kann.

Neben konkreten Schätzfiltern für die Magnetfeld-basierte Lokalisierung, wurde eine untere Schranke für die erreichbare Lokalisierungsgenauigkeit hergeleitet. Die Schranke basiert auf der Bayes'schen Version der Cramér-Rao Schranke und kann u. a. für die Bewertung von Filteralgorithmen genutzt werden. Die Herausforderung bei der Berechnung der Schranke besteht in der fehlenden analytischen Darstellung des Magnetfeldes. Das Magnetfeld ist nur aus Messungen bekannt und die analytische Berechnung des wahren Feldes ist praktisch unmöglich. Selbst bei einer perfekten Kenntnis der Umgebung und deren Materialeigenschaften wäre die Wechselwirkung zwischen Erdmagnetfeld und Umgebung zu komplex für eine geschlossene Berechnung. Zur Schließung der Lücke zwischen Theorie und Praxis wurde in dieser Arbeit ein Gaußprozess auf die Messdaten angewandt um ein probabilistisches Model des Magnetfeldes zu erstellen, das sich auf einfache Weise in die Schranke integrieren lässt.

Alle entwickelten Methoden werden mit Messdaten getestet und anschließend bewertet. Die dafür verwendeten Messdaten wurden mit drei verschiedenen Zügen und auf zwei unterschiedlichen Streckennetzen aufgezeichnet. Die Auswertung zeigt, dass die entwickelten Verfahren für die Magnetfeld-basierte Zuglokalisierung eine metergenaue Positionsbestimmung ermöglichen und die Verfahren das Potential haben, langfristig einen Beitrag zur Automatisierung des Schienenverkehrs zu leisten.

Abstract

Shifting transportation of persons and goods from the road to the rail has the potential to significantly reduce the greenhouse gas emissions of the transportation sector. For the shift from road to rail it is crucial that the capacity of the track networks is increased to serve the increased demand. In order to increase the capacity either new infrastructure has to be built or the existing infrastructure has to be used more efficiently. When comparing the two options it seems to be preferable to increase the capacity of the existing infrastructure since new tracks require a large amount of space, money, and time for the planning and building phase. In contrast, increasing the capacity of the existing tracks can be achieved by reducing the safety distances between consecutive trains from the absolute braking distance, that can be up to kilometers, to the relative braking distance. To enable the capacity increase, the degree of automation in railway traffic has to be increased and the required technologies have to be developed. One key technology for automation is absolute train localization that is essential to control the traffic in the network and the maneuvers of individual trains. The key role of the absolute position for automation of railway traffic is in contrast to its role in the automation of road transportation, in which the absolute position is not essential to maintain a safe operation of the vehicles. The importance of the absolute position for railways is a result of the long braking distances, which exceed the measurement range of sensors such as radars, laser scanners, and cameras. In contrast to road transportation, were the braking distances are much smaller, for railways it is thus not sufficient for safe operations to just rely on a supervision of the surrounding environment with the before mentioned sensors. In this thesis, the focus is therefore on the development and the evaluation of a novel train localization system to enable automated train operations.

The developed localization system is based on position dependent distortions of the magnetic field along a railway track. The distortions are a result of the interaction between the Earth magnetic field and the magnetic material in the railway infrastructure or buildings close to the track. For localization first, a map of the magnetic distortions is created that relates to each position on a track the corresponding magnetic vector at that position. Second, a train moving within a mapped track can estimate it absolute position by comparing multiple measurements of an on-board magnetometer to the magnetic vectors in the map.

In this thesis, general properties of the magnetic field in the railway domain are investigated and based on the findings suitable localization algorithm are derived. Concretely, recursive Bayesian filters are developed and evaluated that enable magnetic field-based train localization in real-time. The developed filters explicitly account for disturbances in the magnetometer measurements and address the problem of magnetometer calibration. Magnetometer calibration plays a central role when it comes to the applicability of magnetic field localization in a practical setting, because without calibration the comparison of the magnetometer measurements to a prior created magnetic field map becomes more difficult or even impossible. The calibration not only depends on properties of the sensor but also on the train in which the sensor is mounted. The dependency on the train is a result of the interaction of the magnetic material in the train and the external magnetic field. Without calibration it might be necessary to create an individual map for each train that we wish to localize. To avoid this, a novel calibration method was developed that, in contrast to commonly used methods, is also applicable to trains. Common methods require a rotation of the magnetometer in space while mounted on the target platform. For calibration, the rotation has to be performed in a homogeneous field and has the goal to ensure the observability of all parameters. In the concrete example of magnetic train localization, this means the sensor has to be rotated while mounted on the train. Due to the large extend and the high weight of trains this would require a considerable effort. In this thesis, therefore a Bayesian filter was developed that estimates the calibration parameters and the train position simultaneously without the need of rotating the train. The proposed filter uses that the observability of the parameters can be ensured also by moving through a inhomogeneous magnetic field rather than a rotation in a homogeneous field.

In addition to the development of concrete filter algorithms, a lower bound for the achievable localization accuracy was derived during the course of the thesis. The bound is based on the Bayesian version of the well-known Cramér-Rao lower bound and can be used, e.g., as a benchmark for the filter algorithms. The challenge in the calculation of the bound for magnetic localization lies in the missing analytical model of the magnetic field. The magnetic field is only known from measurements and the derivation of an analytical model is practically impossible. Even with perfect knowledge of the environment and its material properties, the interaction between the Earth magnetic field and the environment is too complex to find a closed form model. To close the gap between theory and the real-world, we propose in this thesis to fit a Gaussian process to the measurement data to obtain a probabilistic model of the magnetic field that fits nicely into the Bayesian framework of the lower bound.

All developed methods are tested and evaluated based on measurement data. The measurement data for the evaluation was collected with three different trains driving in two different track networks. The evaluation shows, that the methods developed for magnetic train localization enable meter-level accuracy and that magnetic train localization can potentially contribute towards a higher degree of automation in railway traffic.

Table of Contents

Abbreviations, Notation, and Symbols

Abbreviations

a.u.	Arbitrary units
AWGN	Additive white Gaussian noise
BCRLB	Bayesian Cramér-Rao lower bound
BIM	Bayesian information matrix
BRB	Bayerische Regiobahn
cdf	Cumulative distribution function
CRLB	Cramér-Rao lower bound
DLR	Deutsches Zentrum für Luft- und Raumfahrt (engl. German Aerospace Center)
ETCS	European train control system
EU	European union
FDE	Fault detection and exclusion
FIM	Fisher information matrix
GHG	Greenhouse gas
GNSS	Global navigation satellite system
GP	Gaussian process
HSB	Harzer Schmalspurbahnen
IMU	Inertial measurement unit
INS	Inertial navigation system
LS	Least squares
ML	Maximum likelihood
MMSEE	Minimum mean squared error estimate
MSE	Mean squared error
NMA	Noise model averaging
pdf	Probability density function

pmf	Probability mass function
psd	Power spectral density
RMSE	Root mean squared error
RTMS	Rail traffic management system
SLAC	Simultaneous localization and calibration
SLAM	Simultaneous localization and mapping
STFT	Short time Fourier transform

Notation

a, b, c	Real valued scalars
A, B, C	Discrete valued scalars
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	Vectors
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	Matrices
$\mathcal{A}, \mathcal{B}, \mathcal{C}$	Sets
$\lvert\mathbf{a}\rvert$	Euclidean norm of vector $\mathbf{a}$
$[\mathbf{a}]_i$	i-th element of vector $\mathbf{a}$
$[\mathbf{a}]_{i:j}$	i-th to j-th element of vector $\mathbf{a}$
$\dim(\mathbf{a})$	Dimension of vector $\mathbf{a}$
$[\mathbf{A}]_{ij}$	Element in i-th row and j-th column of matrix $\mathbf{A}$
$\mathbf{A}^{(ij)}$	Block in the i-th row and j-th column of block matrix $\mathbf{A}$
$[\mathbf{A}]_{i:}$	i-th row of matrix $\mathbf{A}$
$\mathbf{A}^\top$	Transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	Inverse of matrix $\mathbf{A}$
$\mathbf{I}_{n\times m}$	Identity matrix with n rows and m columns.
$\mathbf{0}_{n\times m}$	Zero matrix with n rows and m columns.
$\mathrm{f}(\cdot)$	Function with scalar image
$\mathbf{f}(\cdot)$	Function with vector image
$\mathbf{F}(\cdot)$	Function with matrix image
$\dot{x}$	Time derivative of x
$\mathrm{p}(\cdot)$	Probability density function

$\mathrm{pr}(\cdot)$	Probability mass function
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian density with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{N}(\mathbf{a}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian density with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at $\mathbf{a}$
$\mathcal{U}(0, l)$	Uniform distribution on $[0, l]$
$\hat{a}(b)$	Estimator of a as function of observation b
$\breve{a}$	MMSEE of a
[J1]	Own journal paper
[C1]	Own conference paper
[CA1]	Co-authored journal or conference paper
[ST1]	Supervised student theses
[1]	External reference

List of Symbols

k	Current time step
$\mathbf{x}_k$	Generic state space vector at time step k
$\mathbf{x}_{0:k}$	Sequence of $\mathbf{x}$ from time step 0 to k
$\mathbf{x}_k^{(i)}$	i-th sample/particle of generic state space vector at time step k
$w_k^{(i)}$	Weight of i-th particle at time step k
$\boldsymbol{\theta}$	Parameter vector
$\mathbf{z}_k$	Generic observation vector at time step k
$\mathbf{z}_{1:k}$	Sequence of $\mathbf{z}$ from time step 0 to k
$\boldsymbol{\mu}_k^{\mathbf{x},-}, \boldsymbol{\Sigma}_k^{\mathbf{x},-}$	Predicted Kalman filter mean and covariance of state vector $\mathbf{x}$ at time step k
$\boldsymbol{\mu}_k^{\mathbf{x},+}, \boldsymbol{\Sigma}_k^{\mathbf{x},+}$	Updated Kalman filter mean and covariance of state vector $\mathbf{x}$ at time step k
s	Along-track position
I	Track identifier
$\mathcal{T}$	Set of all track identifiers in the track network
$\mathcal{P}^{\text{topo}}$	Topological position $\mathcal{P}^{\text{topo}} = \{s, I\}$
O	Train orientation
$\mathbf{d}$	Dynamic train state $\mathbf{d} = \begin{bmatrix} s & \dot{s} & \ddot{s} \end{bmatrix}^\top$

$\mathbf{d}^{\mathrm{B}}$	Dynamic train state for BCRLB $\mathbf{d}^{\mathrm{B}} = \begin{bmatrix} s & \dot{s} \end{bmatrix}^\top$
$\mathcal{X}^{\mathrm{do}}$	Topological pose $\mathcal{X}^{\mathrm{do}} = \{\mathbf{d}, O\}$.
$\mathcal{X}$	Set of topological pose and track identifier $\mathcal{X} = \{\mathcal{X}^{\mathrm{do}}, I\}$
$\mathcal{X}_k^{(i)}$	i-th sample/particle of $\mathcal{X}$ at time step k
$\mathcal{X}_{0:k}$	Sequence of $\mathcal{X}$ from time step 0 to k
$\mathcal{Z}$	Set of observations from multiple sensors
$\mathcal{Z}_{0:k}$	Sequence of $\mathcal{Z}$ from time step 0 to k
$\mathcal{R}$	Set of train routes
$\mathcal{R}^{(i)}$	i-th route in the set of routes $\mathcal{R}$
$\mathcal{M}$	Set of models used in robust particle filter variants
$\mathcal{M}^{(i)}$	i-th model in the set of models $\mathcal{M}$
$\delta(\cdot)$	Dirac distribution
$\mathcal{D}$	Training data set for GP
$\mathbf{F}$	Fisher information matrix
$\mathbf{J}$	Bayesian information matrix
$\mathbf{M}$	MSE matrix

CHAPTER 1

Introduction

In the European union (EU), the transportation sector is responsible for almost one quarter of the overall greenhouse gas (GHG) emissions [1]. Therefore, modernizing transportation towards more energy efficiency can considerably contribute to the reduction of the overall emissions. The comparison of different modes of transport for freight and persons w.r.t. their GHG emissions clearly shows that railways are superior compared to road transportation in both cases [2]. From an environmental point of view, it is therefore desirable to increase the share of rail transportation. Although railways are superior in terms of GHG emissions, if we look on the modal split for person and freight transport in the EU only 6.3 % of the persons and 11.5 % of freight are transported with railways, trams or metros [1]. In comparison, the share of road transport is 89.5 % for persons and 53.3 % for freight. EU funded research initiatives and projects, like Shift2Rail and R2DATO, have the goal to increase the share of railway on the overall transportation by increasing the capacity of the European rail systems, improving the service quality, and by developing and introducing new technologies [3, 4]. For achieving these goals, automation of rail traffic plays an essential role. One concrete example where automation can considerably contribute to achieving the previously mentioned goals is virtual coupling. Virtual coupling is a new operational approach, in which multiple trains can form a platoon on-the-fly without the need of mechanical coupling between the individual trains. The coupling is done only virtually with a communication link that is used to coordinate the behavior of the platoon. Virtual coupling allows to reduce the distance between individual trains from the absolute braking distance to the relative braking distance, which increases the capacity of the already existing tracks [3, 5]. Furthermore, with virtual coupling train lengths can be adapted to the current volume of passengers.

Introducing automation and new operational concepts for railway traffic, such as virtual coupling, comes with several, yet unsolved, technical challenges. The braking distances of high-speed trains can be in the range of multiple kilometers and therefore is beyond the measurement range of most distance sensors, like cameras, radars and laser scanners. Thus, in contrast to cars, trains cannot solely rely on sensors that capture the surroundings to operate safely, but require absolute position information of trains even kilometers away. This makes a reliable train localization system and a system that communicates the train position to other trains in the surrounding crucial for automated railway traffic. Since the train position is safety critical it has to be reliable, accurate, and highly available. As of today, existing train localization systems rely on dedicated track-side infrastructure, which makes the systems expensive and slows down the deployment of new systems. Due to the aforementioned disadvantages of infrastructure-based systems, it is desirable to move the hardware for localization from the infrastructure into the train. Developing on-board localization systems that fulfill the needs of the railway sector w.r.t. safety, accuracy, and availability is an open challenge and is the subject of several EU initiatives and projects, such as Shift2Rail [3], R2DATO [6], and CLUG [7]. In these initiatives and projects, the focus is on the use of global navigation satellite systems (GNSS) such as GPS and Galileo for localization. While this is an important first step towards infrastructure-less train localization, satellite-based systems alone cannot provide an accurate position in the whole railway domain. In tunnels for example, GNSS signals are completely blocked and hence cannot be used for positioning, but even in environments in which GNSS signals are available the signal quality can be severely degraded.

In this thesis, therefore an alternative infrastructure-less railway localization system is developed that is independent from GNSS and that works also in tunnel environments. In particular, the thesis focuses on the use of the Earth magnetic field for railway localization. Magnetic field localization is based on distortions in the Earth magnetic field introduced by magnetic material in the environment. When the magnetic material is stationary also the distortions they introduce are stationary and characteristic for a certain part of the railway network. In the following the current state-of-the-art in railway and magnetic localization is discussed and the contributions of this thesis are presented.

1.1 State-of-the-Art

1.1.1 Railway Localization

Train localization systems differ from other localization systems in that they do not determine the position relative to a global coordinate system but a position relative to the track network. This does not necessarily mean that a train localization system does not know where the train is in a global coordinate system, but for automation of train traffic one is typically interested in where the train is in the network. The location within the track network will be called topological position and consists of two values, a discrete valued track identifier and a continuous valued along-track position that describes the position of the train on the track. In order to obtain the topological position, the localization systems need information of the track network. This information can be provided either by a map of the network or by track-side equipment located at known positions.

The advantage of the topological position is that it automatically accounts for the track geometry, i.e., the difference between two topological positions is the length of the track between these positions. Note, the difference between two topological positions is only unique when the sequence of tracks between them is defined. In practice this is typically fulfilled because train operators schedule the track sequences in advance.

In the following a short overview on different train localization systems is given. The overview is divided into two parts. In the first part, systems are described that require track-side equipment and in the second part, we describe systems that use only on-board equipment. Some more details and references regarding train localization can be found, e.g., in [8] and [9].

Localization with Track-side Equipment

Rail traffic management systems (RTMS) are divided into several layers [10]. The upper layers are concerned with coordinating the traffic flow in the railway network that means here the decision is made which train will run on which track. The decision is passed from the upper layer to the lower layers. In the lower layers the signals are set and it is ensured that the trains travel safely through the network without collisions. The layers related to signaling and safety are also called interlocking system.

Interlocking systems are typically divided into fixed blocks and the systems ensure that each block is only occupied by one train. In order to ensure this, the track vacancy is supervised continuously by dedicated track-side equipment. A prominent example for equipment used for track vacancy supervision are axle counters. Axle counters are placed on the start and at the end of each block. When a train enters a block, the number of its axles is counted. The same is done when the train leaves the block again. If the number of axles leaving the block is equal to the number that entered the block, the block is considered vacant. Between the start and the end of the block the axle counters provide no information about the train position. Thus, the traffic management system only has punctual information about the position of a train once it enters or leaves a block and the density of position information along the track is directly related to the length of the blocks.

Modern RTMS still rely on interlocking systems that use a block structure, but they supervise the positions of trains continuously. For example, the European train control system (ETCS) combines fixed reference points in the network and dead reckoning to continuously calculate the position of the trains in the network [11]. In ETCS, the reference points are called Eurobalise and are in essence a RFID tag that is placed between the rails. When a train passes over a balise, the balise is activated by a signal transmitted from the on-board ETCS equipment of the train. The balise then transmits a telegram to the on-board equipment making the balise the new reference point from which dead reckoning is performed. Since the train can only move along the track, the dead reckoning is done by a simple odometer such as a wheel encoder or Doppler radar. The accuracy of ETCS strongly depends on the quality of the odometer and the density of the balises. According to the European agency for railways the position accuracy should stay within $\pm(5\,\mathrm{m} + 5\,\% \cdot d_{\text{traveled}})$ with d_{traveled} being the traveled distance since the last balise [12]. From the accuracy requirement given by [12] it becomes obvious that for accurate localization the track network has to be covered densely with Eurobalises. If the error should stay below 10 m a balise has to be placed every 100 m of track. Furthermore, if from the balise also the driving direction should be identified at least a group of two balises has to be placed instead of a single balise. Equipping a large network with balises therefore quickly becomes expensive and time consuming to maintain. To overcome these drawbacks, as mentioned already above, several EU initiatives and projects are concerned with the development of on-board localization systems that only need a minimum of track-side equipment.

Infrastructure-less Localization

Current research and development activities for on-board train localization are strongly focused on GNSS and its peculiarities when used in the railway environment. The first and maybe most important peculiarity is that for trains we are not interested in their position in a global coordinate system but in their position in the track network. We call this also track-selective localization because we want to know the train position relative to the track the train is driving on. Therefore, the position obtained from GNSS, which is given in a global coordinate system such as the Earth-centered, Earth-fixed system, must be related to a position in the track network. This requires a map of the track network that describes the tracks in coordinates of a global coordinate system.

In the literature, different approaches are proposed for how the topological position of the train is derived from the GNSS position. The proposed approaches can be roughly divided into two categories. The first category of approaches calculate the topological position directly by combining GNSS observables, e.g., the pseudoranges, with the map [13] [C1]. Most of the proposed approaches fall into the second category and follow a more indirect two-step scheme. In this approaches first, a global position from the GNSS observables is calculated which is then combined in a second step with the information of the track map [14–20]. The advantage of the direct combination of map information and GNSS observables lies within the reduction of the required visible satellites to calculate the train position [13]. A problem that is present for both categories of approaches is that the GNSS observables and the position calculated from it is uncertain and it might not be obvious on which track the train is currently driving. This problem is particularly difficult in a parallel track scenario where the tracks are close together, e.g., in Germany the distance between tracks with a speed limit of $120\,\mathrm{km\,h^{-1}}$ can be as close as $3.8\,\mathrm{m}$ [21]. If the uncertainty of the GNSS position spreads out over multiple tracks ambiguity emerges. Some of the above mentioned papers therefore have a mechanism to cope with the ambiguity by handling multiple track hypotheses. In [13, 17] a particle filter is used, where the particles can follow different tracks. In contrast, in [16] multiple Kalman filters are used to track the individual hypothesis and to calculate their probability given the available measurements data.

GNSS is often combined with other sensor modalities such as inertial measurements units (IMUs) and odometers into a multi-sensor system. The combination of GNSS with other sensors improves the overall localization and can help to identify the correct track hypothesis. For example IMUs can help to detect the correct track after passing a switch based on the measured turn rates and accelerations [22, 23]. To bound the position errors in the absence of GNSS, the IMU can also be used to obtain absolute position information by comparing the inertial measurements with the known track curvature [24–26] or with a map of track irregularities [27]. The curvature-based localization only works on curved parts of the track network, on straight tracks or high-speed tracks with large curve radii the position accuracy most likely will be poor. The approach based on track irregularities relies on a high-resolution and up-to-date map of the track.

1.1.2 Magnetic Localization

In contrast to the methods for infrastructure-less localization from the previous section, this thesis proposes the use of the Earth magnetic field to bound the position error in GNSS-denied areas. The idea behind the proposed magnetic field-based localization system is derived from the observation that the Earth magnetic field along a railway track is distorted by magnetic material in its surrounding. In the case of the railway environment, magnetic material can be found basically everywhere. Mostly the found magnetic material is steel, e.g., in the rails and the masts holding up signals and the overhead

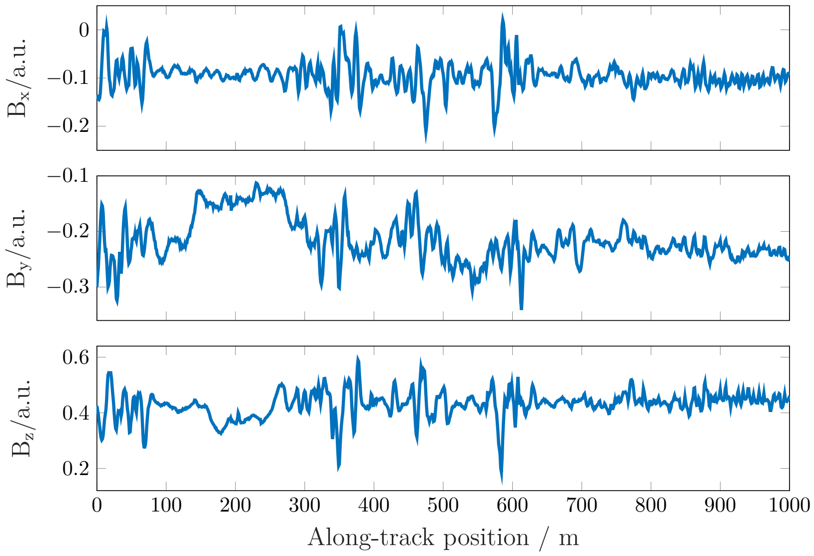

Figure 1.1: Example for the magnetic field in the railway domain. The shown magnetic field was recorded with a magnetometer on-board a regional train on a track near Augsburg. More details about the measurements can be found in Section 7.1.1.

power line. Furthermore, steel is also found in form of reinforcement in the concrete of buildings, platforms, bridges, and noise barriers. To get a first impression of the magnetic vector field in the railway environment, Figure 1.1 shows the magnetic field recorded on a 1 km long track segment. Details about the hardware setup used for this measurements can be found in Section 7.1.1. The shown magnetic field values are the measured flux density normalized internally by the sensor to the Earth magnetic field strength and hence are dimensionless. To emphasize this, we follow the notation of the sensor manufacturer and label the axes with arbitrary units (a.u.). Unfortunately, the manufacturer does not state the normalization value for the sensor used here but for newer sensor models a rough value of 40 µT is mentioned. The reason uncalibrated values are shown here is that no practical calibration procedure is available for magnetometers mounted on a train. The lack of appropriate calibration procedures is a problem that will be addressed in this thesis, but for the following general introduction to magnetic localization we can ignore this problem for now. From Figure 1.1 we can clearly see the distortions of the magnetic field and that the distortions are different at different positions on the track. When the distortions are caused by static, immobile, magnetic material also the resulting distortions are static. Thus, these distortions can be seen as a magnetic fingerprint of the track that, when stored in a map, can be used for localization. For position estimation the magnetic field measured with a train-mounted magnetometer is compared to the map to find the part of the map that fits best to the measurements. To find a unique position a single measurement is not sufficient and will cause ambiguities, in practice therefore a sequence of measurements is compared to the map.

The authors of [28] and [29] also propose the use of magnetic field measurements for train localization but in contrast to the work presented in this thesis, in which we passively sense the distorted Earth magnetic field, their idea relies on a custom build active sensor mounted close to the rail. The sensor induces an eddy current in the steel rail and then senses the resulting magnetic field. With the measured magnetic field then the train speed can be estimated [30] and certain infrastructure elements like switches can be detected and classified. The disadvantage of the active sensor is its mounting

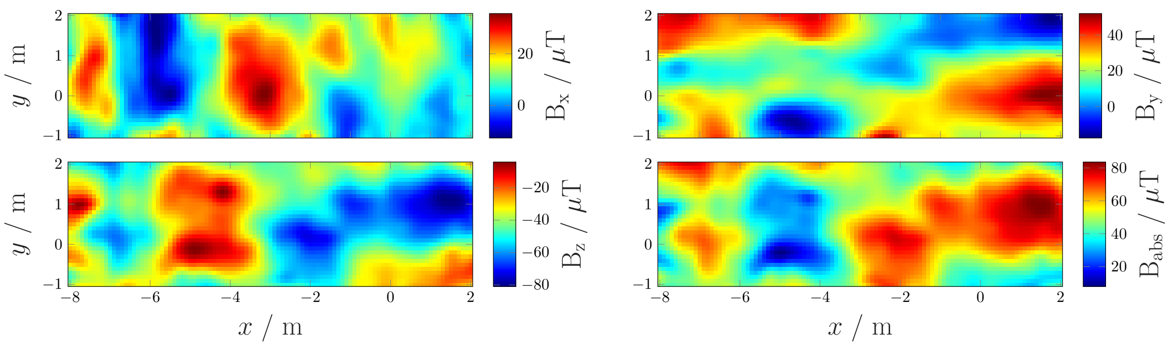

Figure 1.2: Flux density of the magnetic vector field measured in a laboratory of DLR. The four plots show the x-, y- and z-components of the magnetic field and its magnitude. The x- and y-components are aligned to the coordinate axes of the graph. The z-component is orthogonal to the other components such that they form a righted handed coordinate system.

position close to the rails, which complicates the retrofitting of old trains and since it is an active sensor it has to be shown that it does not interfere with the already existing train- and track-side equipment.

The idea of magnetic field-based localization, as it used in this thesis, is not new. It was and still is investigated also in other environments. Most notably is the use of the magnetic field for indoor localization. Intuitively it makes sense that especially in this environment the magnetic field attracted the attention of the research community. In indoor environments there is no ubiquitous localization system like GNSS. Therefore, the magnetic field offers a big benefit, it does not require dedicated infrastructure and it contains absolute position information that can be used to bound the long-term position error. In addition, from our experience the magnetic field in indoor environments shows rather strong variations, which results in a high position accuracy. To get an impression how the field looks in an indoor environment, Figure 1.2 shows the magnetic field recorded in an laboratory of the German Aerospace Center (DLR). In total an area of 10 m by 3 m was measured and the sensor was kept at an constant height. As we have seen already for the railway example, the indoor magnetic field shows strong position dependent distortions. For the indoor measurements we can also quantify the observed distortions because for the shown measurements the magnetometer was calibrated. If we compare the maximal and minimal values of the field's magnitude we see a difference of roughly 70 µT. This is more than the flux density of the Earth magnetic field in Germany which is around 50 µT. For comparison, the empirically determined noise standard deviation of the low-cost Kionix KMX62 magnetometer used in the measurements of Figure 1.2 was ≈ 0.23 µT. The magnetic field therefore not only varies considerably in indoor environments, but it also has a good signal-to-noise-ratio when measured with a low-cost magnetometer.

The distortions in indoor environments are caused by magnetic material in walls, floors, pillars, and in the ceiling. But also furniture can have an influence on the magnetic field. Examples for the use of the magnetic field for indoor localization can be found in [31–35]. These papers are based on particle filters that recursively estimate the position of a person or a robot. The particle filter has the advantage that it can cope with the nonlinearity that stems from the nonlinear relation between the position and the magnetic field. In addition, a recursive filter processes sequences of measurements which helps to resolve ambiguities that might exist after initialization. To improve the localization performance, the magnetometer can be combined with other sensor modalities, e.g., IMUs and wheel encoders. Besides

Bayesian filters, also pattern recognition methods are proposed for magnetic localization. Examples for such methods are convolutional neural networks [36] and dynamic time warping [37].

Since for indoor environments there is no ubiquitous localization system typically no ground truth position is available for creating the magnetic map. This results in a simultaneous localization and mapping (SLAM) problem that is well-known from the robotics domain (cf. [38]). In [39–45] algorithms are proposed that address the magnetic SLAM problem. In [39–44] Bayesian filters are used for the estimation while in [45] the authors propose a graph-based approach with a loop-closure detection based on the magnetic field.

In addition to the railway and indoor environment, the feasibility of magnetic localization using a particle filter has been demonstrated also for road environments [46]. Furthermore, even for airplanes is was demonstrated that magnetic localization is feasible by combining a magnetometer with an inertial navigation system (INS) [47, 48]. A particle filter-based INS aided SLAM approach for a magnetometer mounted in an airplane was proposed and evaluated in [49]. The magnetic field distortions in the airspace have less spatial variations compared to the other domains and are caused, according to [47, 48], mainly by the crustal field of the earth. The crustal field is the magnetic field caused by magnetic material in the Earth crust, and the observable spatial variability of it depends on the altitude of the airplane. Therefore, for different altitudes individual magnetic maps are required. Alternatively, methods to approximate the field at the current altitude can be used.

1.2 Contributions

The main goal of this thesis is to demonstrate the technical feasibility of magnetic field-based localization in the railway domain. In order to achieve this goal, algorithms are developed that account for the characteristics of the railway domain and that make the localization robust in the presence of disturbances. Furthermore, the theoretically achievable accuracy of magnetic localization is investigated. All the proposed algorithms are evaluated with measurement data collected with different train types. In the following, the contributions of this thesis are presented in more detail and references to the corresponding papers of the author are given.

I **Robust particle filter for magnetic localization:** To show the feasibility of magnetic localization in the railway domain we presented in [C2] an approach based on an error state Kalman filter that uses the magnetic field to bound the errors of an INS. Furthermore, in [C3] we showed that with a particle filter localization is possible using solely the magnetometer measurements. Besides these simple proofs of feasibility, in the scope of this thesis also filters that are robust in the presence of magnetic disturbances are investigated. This is of particular importance because magnetic localization is based on the comparison of the magnetic field measured by an on-board magnetometer with a prior recorded magnetic map. Disturbances in the magnetometer measurements therefore have a negative effect on the localization performance. Unfortunately, in the railway environment the measurements are strongly affected by disturbances. Frequently disturbances are caused by currents in the overhead power lines or moving magnetic material, e.g., cargo trains on a neighboring track. While the alternating field of the overhead power line can be filtered out relatively easy due to its known frequency, filtering out the disturbances caused by moving magnetic material is not that simple. In order to cope with disturbances that cannot be filtered out, a multiple model particle filter was developed that detects disturbances in the magnetometer measurements and excludes them from the filter update step. The filter was proposed and evaluated in [C4]. Furthermore, in [C5] we adapted the filter algorithm from [50] that uses multiple measurement models and softly switches between them to reduce the influence of disturbed measurements.

II **Joint train localization and track identification:** The task of a train localization system is to estimate the topological position of the train, i.e., the system has to determine the current track and the position of the train on that track. The track identifier is a discrete value and can only change when the train is passing a switch. A localization system has to account for this and has to track all possible track hypotheses until the correct track can be identified with certainty. In [C5] we proposed and evaluated a particle filter-based localization system that jointly estimates the position on the track and the track identifier. In the proposed particle filter, the different track hypotheses are handled by creating a copy of the particle filter as soon as the filter reaches a switch. Each of the filter copies continues tracking the train position for a different track. Based on the residuals of the different filters, a decision is made for one of the possible tracks. As an alternative for the copy filter, this thesis also investigates a straightforward particle filter that randomly distributes its particles across the possible tracks when reaching a switch.

III **Simultaneous localization and calibration:** When a magnetometer is mounted on a platform that contains magnetic material, the platform will interact with the external magnetic field and alter the magnetometer measurements. Thus, if two trains measure the magnetic field of a track with the same magnetometer the results may differ severely. This leads to difficulties in the practical implementation of magnetic train localization, e.g., a magnetic field map might be no longer universally valid for all trains but has to be created for each train individually. An approach to remove the influence of the platform from the measurements is to calibrate the magnetometer. Unfortunately, common calibration methods assume that the magnetometer and the platform on which it is mounted can be rotated in a homogeneous magnetic field. The rotation is required to render the calibration parameters observable. When the platform is a train this is not practical due to its sheer mass and volume. Therefore, this thesis presents a simultaneous localization and calibration (SLAC) approach that overcomes this problem by exploiting magnetic field distortions. The approach was first presented in [J1] in which it was evaluated based on measurements of a model train running in a laboratory of DLR. In [C6] and [J2] the approach was then evaluated with data from a real train experiment. In addition, in [C7] we showed the applicability of SLAC to a robotic platform driving in an indoor environment.

IV **Theoretical accuracy analysis:** During the development of a position estimator it is beneficial to have a benchmark for the estimator performance. For Bayesian estimators one possible benchmark is given by the Bayesian version of the Cramér-Rao lower bound (CRLB). The Bayesian Cramér-Rao lower bound (BCRLB) is a lower bound on the mean squared error (MSE) of any estimator. If an estimator attains the BCRLB it is called efficient and no estimator can achieve a lower MSE. The derivation of the CRLB and the closely related BCRLB requires a differentiable measurement model. For magnetic localization the measurement model is given by the magnetic map that is known only from measurements and no analytical model for it exists. Calculating the derivatives of the map therefore is not straightforward. In [J3] and [C8] we addressed this issue by adapting the idea proposed in [51] to magnetic localization. The key idea here is to approximate the magnetic map with a Gaussian process (GP). A GP is differentiable and it accounts for the uncertainty in the magnetic map and fits nicely into the Bayesian framework of the BCRLB. As a first step towards a BCRLB for magnetic train localization, we investigated in [J3] the applicability of the BCRLB for magnetic indoor localization on the basis of laboratory measurements. In [C8] we then transferred the BCRLB to the railway domain.

1.3 Structure of the Thesis

The thesis structure is shown in Figure 1.3 as a block diagram. The different blocks in the diagram are linked to the contributions from the previous section by the corresponding roman number. Chapter 2 and Chapter 3 are the foundation to the latter chapters. Chapter 2 lays down the fundamentals used in the development of the localization algorithms and the derivation of the BCRLB. Depending on the familiarity of the reader with Bayesian estimation, this chapter can be skipped. In Chapter 3 the magnetic field in the railway domain is analyzed and sources of disturbance are identified. The robust localization algorithms and the SLAC algorithm are presented in Chapter 4 and Chapter 5. The more theoretical part of the thesis is completed with Chapter 6, in which the BCRLB for magnetic localization is derived. An evaluation of all the developed methods and algorithms can be found in Chapter 7. Finally in Chapter 8 conclusions of the presented work are drawn and a short outlook on future work is given.

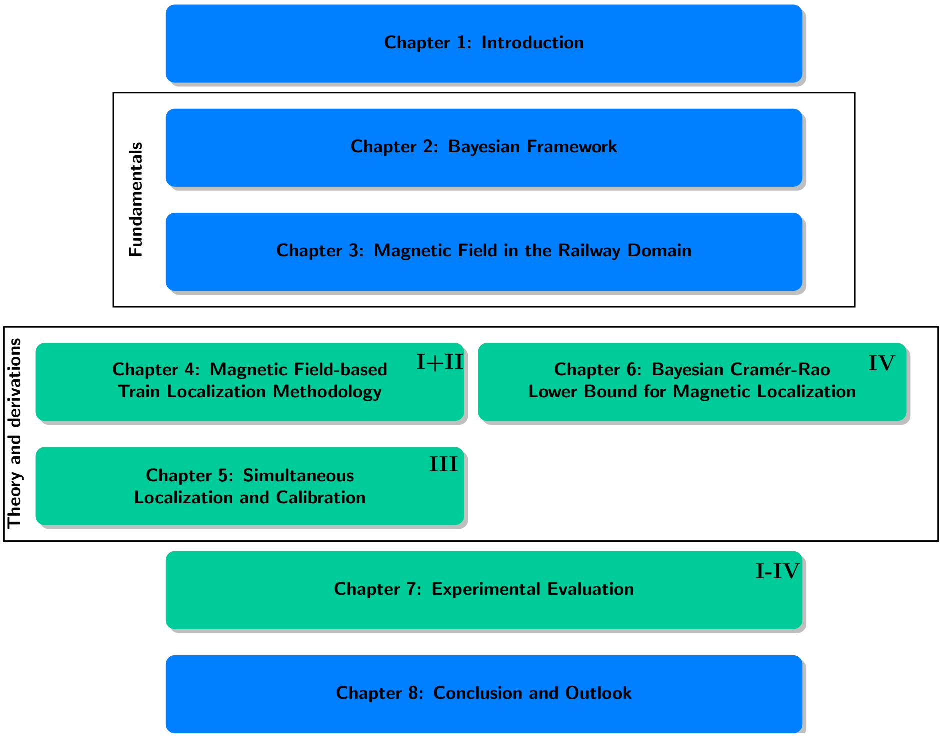

Figure 1.3: Block diagram of the thesis structure. The green blocks contain the main contributions of the thesis. The roman numbers in the blocks create the link to the contributions described in Section 1.2

CHAPTER 2

Bayesian Framework

The localization algorithms developed in Chapter 4 and Chapter 5, and the BCRLB in Chapter 6 are based on Bayesian methods to account for the inherent uncertainty of the used sensor data and to incorporate motion models into the estimation. Therefore, in this chapter a Bayesian framework is established that forms the foundation for the developed algorithms. The chapter starts with a general introduction to Bayesian estimation and filtering. Then, the linear Kalman filter and the nonlinear particle filter is discussed. This is followed by the theory of Gaussian processes. In the end of the chapter, the CRLB and its Bayesian version is presented.

2.1 Bayesian Estimation

Bayesian estimation is based on the posterior probability density function (pdf) of the quantity that we want to estimate. Let us assume for now that we want to estimate the value of parameter vector $\boldsymbol{\theta}$ given the observation $\mathbf{z}$. Then the posterior pdf of $\boldsymbol{\theta}$ is given by the conditional density

$$\mathrm{p}(\boldsymbol{\theta}|\mathbf{z}) = \frac{\mathrm{p}(\mathbf{z}|\boldsymbol{\theta})\,\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathbf{z})}. \tag{2.1}$$

The above equation is called Bayes' theorem and is easily obtained from the definition of conditional probability

$$\mathrm{p}(\mathbf{a}|\mathbf{b}) = \frac{\mathrm{p}(\mathbf{a},\mathbf{b})}{\mathrm{p}(\mathbf{b})}. \tag{2.2}$$

To avoid confusion, we would like to mention that in the following the terms pdf, density, and distribution are used synonymously. The posterior in (2.1) consists of three pdfs. The likelihood $\mathrm{p}(\mathbf{z}|\boldsymbol{\theta})$ that accounts for the information contained in the observation, the prior $\mathrm{p}(\boldsymbol{\theta})$ of the parameter vector that contains the information about $\boldsymbol{\theta}$ before any observations are available, and $\mathrm{p}(\mathbf{z})$ that depends only on the given observation and hence can be seen as a normalization constant.

From (2.1) the difference between Bayesian and maximum likelihood (ML) estimation becomes directly visible. An ML estimator treats $\boldsymbol{\theta}$ as a nonrandom constant for which no prior information is available. The ML estimator simply returns the estimate $\hat{\boldsymbol{\theta}}$ of the parameter vector that maximizes the likelihood for the given observation

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ \mathrm{p}(\mathbf{z}|\boldsymbol{\theta}). \tag{2.3}$$

In contrast, a Bayesian maximum a posteriori estimator returns the maximum of the posterior density

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ \mathrm{p}(\boldsymbol{\theta}|\mathbf{z}) = \arg\max_{\boldsymbol{\theta}} \ \mathrm{p}(\mathbf{z}|\boldsymbol{\theta})\,\mathrm{p}(\boldsymbol{\theta}) \ , \tag{2.4}$$

which accounts also for the prior pdf. The prior information has the function of a regularization during the maximization. The advantage of Bayesian over ML estimation is highly dependent on the available prior information and may not offer any advantage. But there are also estimation problems in which meaningful prior information is available. An example, that will be discussed in more detail in the next section, is dynamic state estimation in which the evolution of the state over time can be described by a mathematical model.

2.2 Recursive Bayesian Filtering

In practice, we often encounter the problem that the state of a dynamic system has to be estimated from a sequence of noisy observations. This problem can be addressed with a Bayesian estimator based on the posterior distribution of the complete state sequence

$$\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \frac{\mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k})}{\mathrm{p}(\mathbf{z}_{1:k})} \ , \tag{2.5}$$

where $\mathbf{x}_{0:k} = (\mathbf{x}_0, \ldots, \mathbf{x}_k)$ is the sequence of states $\mathbf{x} \in \mathbb{R}^{N_\mathbf{x}}$ from time step 0 to time step k and $\mathbf{z}_{1:k} = (\mathbf{z}_1, \ldots, \mathbf{z}_k)$ is a sequence of observations with $\mathbf{z} \in \mathbb{R}^{N_\mathbf{z}}$. The posterior distribution $\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ is also called full filter posterior since it is the pdf of the complete state history. In contrast to the full filter posterior, the filter posterior only accounts for the state at the time step k of the newest observation. The filter pdf is obtained from a marginalization of (2.5) w.r.t. the states up to time step $k-1$

$$\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{\int \mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k}) d\mathbf{x}_{0:k-1}}{\mathrm{p}(\mathbf{z}_{1:k})} \ , \tag{2.6}$$

where the integral is taken over the complete domain of $\mathbf{x}_{0:k-1}$. Here and in the following we will not explicitly state integral limits when the integral is taken over the whole domain of the corresponding variable. Furthermore, only a single integral sign is used in this case even when the integral runs over multiple dimensions. Note, the pdfs in (2.5) and (2.6) are called filter posterior only when observations exactly up to time step k are available. If instead the observation sequence $\mathbf{z}_{1:l}$ with $l > k$ is used the result is called smoothing distribution and for $l < k$ predictive distribution [52, 53].

In practical applications, the focus lies often on the calculation of the filter densities because they account for all the information available at the current time step and do not rely on future measurements, as it is the case for the smoothing distribution. For real-time applications it is also important that the calculation of the posterior is completed within a predefined time window after receiving a new observation. This becomes challenging when the posterior is calculated naively from (2.5) or (2.6) because the amount of data, and for (2.5) also the number of state vectors, that have to be processed grows over time. This issue is avoided through a recursive calculation of the posterior. The recursive calculation is based on the common assumption that the dynamic system under consideration is Markovian [52, 54]. For a Markovian system the transition density of the state at time step k depends only on the previous state and not the complete history

$$\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{0:k-1}) = \mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1}) \; . \tag{2.7}$$

This is still true when conditioned additionally on previous measurements

$$\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) = \mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1}) \; . \tag{2.8}$$

Furthermore, given the current state the likelihood of an observation in a Markovian system is independent from past observations and states

$$\mathrm{p}(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}) = \mathrm{p}(\mathbf{z}_k|\mathbf{x}_k). \tag{2.9}$$

The equations (2.7) - (2.9) show that in a Markovian system the state at a certain time step contains all the information of previous states and observations.

With the definition of conditional probability (2.2), and the above assumptions (2.8) - (2.9) the full posterior in (2.5) can be reformulated to

$$\begin{aligned}
\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) &= \frac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})\,\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k-1})\,\mathrm{p}(\mathbf{z}_{1:k-1})}{\mathrm{p}(\mathbf{z}_k|\mathbf{z}_{1:k-1})\,\mathrm{p}(\mathbf{z}_{1:k-1})} \\
&= \frac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)\,\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k-1})}{\mathrm{p}(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\
&= \frac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)\,\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1})\,\mathrm{p}(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})}{\mathrm{p}(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\
&= \frac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)\,\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})}{\mathrm{p}(\mathbf{z}_k|\mathbf{z}_{1:k-1})}\,\mathrm{p}(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \; .
\end{aligned} \tag{2.10}$$

The filter posterior density is obtained once more by marginalization of the state history

$$\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)}{\mathrm{p}(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \int \mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})\,\mathrm{p}(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{0:k-1} \; . \tag{2.11}$$

The posterior $\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ at time step k is now a function of the posterior $\mathrm{p}(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$ from the previous time step. To obtain the updated posterior for time step k, the posterior from the previous time step is multiplied with the transition density $\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})$ and the likelihood $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)$. The posterior update is usually divided into two steps. The first step is the prediction step in which the posterior from the previous time step is advanced in time

$$\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k-1}) = \mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})\,\mathrm{p}(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \; . \tag{2.12}$$

For the filtering density (2.11) the prediction is given by the Chapman-Kolmogorov equation

$$\begin{aligned} \mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) &= \int \mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})\, \mathrm{p}(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{0:k-1} \\ &= \int \mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})\, \mathrm{p}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}\ . \end{aligned} \tag{2.13}$$

The second step updates the predictive densities in (2.12) and (2.13) with the information of the current observation to obtain the new full posterior density

$$\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \frac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)\, \mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k-1})}{\mathrm{p}(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \tag{2.14}$$

and the filter density

$$\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)\, \mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\mathrm{p}(\mathbf{z}_k|\mathbf{z}_{1:k-1})}\ . \tag{2.15}$$

The practical implementation of (2.10) and (2.11) depends strongly on the estimation problem defined by the dynamic system whose state is to be estimated. This means it depends on the likelihood $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)$, the transition pdf $\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})$ of the state, and the prior distribution.

2.3 Linear Filtering - The Kalman Filter

If the dynamic system under consideration is linear and all involved distributions are Gaussian the filter posterior in (2.11) will be also Gaussian and can be calculated analytically, which yields the famous Kalman filter. The system under consideration here is given by the stochastic state space model

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \tag{2.16}$$

$$\mathbf{z}_{k+1} = \mathbf{H}_{k+1} \mathbf{x}_{k+1} + \mathbf{n}_{k+1}. \tag{2.17}$$

The dynamics of the state space vector $\mathbf{x}_k$ are defined by the system matrix $\mathbf{F}_k$ and the process noise vector $\mathbf{w}_k$. The observations available for state estimation are obtained from the linear model in (2.17), where $\mathbf{H}_{k+1}$ is the observation or measurement matrix and $\mathbf{n}_{k+1}$ is the measurement or observation noise. The noise vectors are both considered Gaussian and white

$$\mathbf{w}_k \sim \mathcal{N}\left(\mathbf{0}_{N_\mathbf{x} \times 1}, \mathbf{Q}_k\right) \tag{2.18}$$

$$\mathbf{n}_k \sim \mathcal{N}\left(\mathbf{0}_{N_\mathbf{z} \times 1}, \mathbf{R}_k\right)\ . \tag{2.19}$$

With the system model in (2.16) and the process noise density (2.18) we can calculate the transition density $\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})$ that is required for the prediction step (2.13)

$$\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1})\ , \tag{2.20}$$

where we used the properties of Gaussian distributions under linear transformations. If the posterior from the previous time step is Gaussian

$$\mathrm{p}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \boldsymbol{\mu}_{k-1}^{\mathbf{x},+}, \boldsymbol{\Sigma}_{k-1}^{\mathbf{x},+}) \,, \tag{2.21}$$

then the prediction equation becomes

$$\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1})\mathcal{N}(\mathbf{x}_{k-1}; \boldsymbol{\mu}_{k-1}^{\mathbf{x},+}, \boldsymbol{\Sigma}_{k-1}^{\mathbf{x},+})d\mathbf{x}_{k-1} \,. \tag{2.22}$$

For the linear Gaussian case the integral (2.22) has the closed form solution [52]

$$\begin{aligned} \mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) &= \mathcal{N}(\mathbf{x}_k; \mathbf{F}_{k-1}\boldsymbol{\mu}_{k-1}^{\mathbf{x},+}, \mathbf{F}_{k-1}\boldsymbol{\Sigma}_{k-1}^{\mathbf{x},+}\mathbf{F}_{k-1}^{\top} + \mathbf{Q}_{k-1}) \\ &= \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k^{\mathbf{x},-}, \boldsymbol{\Sigma}_k^{\mathbf{x},-}) \,. \end{aligned} \tag{2.23}$$

The result (2.23) uses that from (2.20) and (2.21) the joint Gaussian density $\mathrm{p}(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ can be calculated that then only has to be marginalized w.r.t. $\mathbf{x}_{k-1}$. Regarding the notation, the superscript minus $^-$ is used for the predicted mean and covariance and the superscript plus $^+$ for their updated posterior values.

With the predictive density (2.23) we now formulate the observation update by noting that the numerator in (2.15) is equal to the joint density of $\mathbf{z}_k$ and $\mathbf{x}_k$ conditioned on the measurement history

$$\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)\,\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathrm{p}(\mathbf{z}_k, \mathbf{x}_k|\mathbf{z}_{1:k-1}). \tag{2.24}$$

As for the prediction step, the joint density is again Gaussian and can be calculated directly from $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k)$ and $\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k-1})$. Once we have $\mathrm{p}(\mathbf{z}_k, \mathbf{x}_k|\mathbf{z}_{1:k-1})$, we can easily calculate the desired posterior density using the properties of the Gaussian distribution [52]

$$\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k^{\mathbf{x},+}, \boldsymbol{\Sigma}_k^{\mathbf{x},+}) \,, \tag{2.25}$$

where the mean $\boldsymbol{\mu}_k^{\mathbf{x},+}$ and covariance $\boldsymbol{\Sigma}_k^{\mathbf{x},+}$ are given by

$$\boldsymbol{\mu}_k^{\mathbf{x},+} = \boldsymbol{\mu}_k^{\mathbf{x},-} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\boldsymbol{\mu}_k^{\mathbf{x},-}) \tag{2.26}$$

$$\boldsymbol{\Sigma}_k^{\mathbf{x},+} = \boldsymbol{\Sigma}_k^{\mathbf{x},-} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\top} \,. \tag{2.27}$$

In (2.27) the innovation covariance matrix $\mathbf{S}_k$ and the Kalman gain $\mathbf{K}_k$ are given by

$$\begin{aligned} \mathbf{S}_k &= \mathbf{H}_k\boldsymbol{\Sigma}_k^{\mathbf{x},-}\mathbf{H}_k^{\top} + \mathbf{R}_k \\ \mathbf{K}_k &= \boldsymbol{\Sigma}_k^{\mathbf{x},-}\mathbf{H}_k^{\top}\mathbf{S}_k^{-1} \,. \end{aligned} \tag{2.28}$$

With (2.23) and (2.25)-(2.28) we have all the ingredients for the Kalman filter. One iteration of the filter algorithm, consisting of a prediction and a update step, is shown in Algorithm 2.1.

Algorithm 2.1 Generic Kalman filter

1: Input: Initial mean $\boldsymbol{\mu}_{k-1}^{\mathbf{x},+}$ and covariance $\boldsymbol{\Sigma}_{k-1}^{\mathbf{x},+}$ and the current observation $\mathbf{z}_k$
2: Output: Updated mean $\boldsymbol{\mu}_{k}^{\mathbf{x},+}$ and covariance $\boldsymbol{\Sigma}_{k}^{\mathbf{x},+}$

Prediction step
3: $\boldsymbol{\mu}_{k}^{\mathbf{x},-} = \mathbf{F}_{k-1}\boldsymbol{\mu}_{k-1}^{\mathbf{x},+}$
4: $\boldsymbol{\Sigma}_{k}^{\mathbf{x},-} = \mathbf{F}_{k-1}\boldsymbol{\Sigma}_{k-1}^{\mathbf{x},+}\mathbf{F}_{k-1}^{\top} + \mathbf{Q}_{k-1}$

Update step
5: $\mathbf{S}_k = \mathbf{H}_k\boldsymbol{\Sigma}_{k}^{\mathbf{x},-}\mathbf{H}_k^{\top} + \mathbf{R}_k$
6: $\mathbf{K}_k = \boldsymbol{\Sigma}_{k}^{\mathbf{x},-}\mathbf{H}_k^{\top}\mathbf{S}_k^{-1}$
7: $\boldsymbol{\mu}_{k}^{\mathbf{x},+} = \boldsymbol{\mu}_{k}^{\mathbf{x},-} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\boldsymbol{\mu}_{k}^{\mathbf{x},-})$
8: $\boldsymbol{\Sigma}_{k}^{\mathbf{x},+} = \boldsymbol{\Sigma}_{k}^{\mathbf{x},-} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\top}$

9: **return**

2.4 Sample-based Nonlinear Filtering

The Kalman filter is the analytical solution for the filter posterior when the dynamic system is linear and Gaussian. Unfortunately, for nonlinear systems, even when they are affected only by Gaussian noise, there are no general closed form solutions to the posterior. For nonlinear systems therefore numerical methods to approximate the posterior are widely used. One class of these numerical methods are Monte Carlo methods that rely on representing the density with random samples.

Assume we want to calculate the expected value w.r.t. the full posterior given by the integral

$$\mathbf{E}\left[\mathbf{f}(\mathbf{x}_{0:k})\right] = \int \mathbf{f}(\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})d\mathbf{x}_{0:k}\ , \tag{2.29}$$

with some function $\mathbf{f}:\mathbb{R}^{N_{\mathbf{x}}\cdot(k+1)} \to \mathbb{R}^M$. The solution of such integrals can be obtained from Monte Carlo integration. In the simplest form of Monte Carlo integration a set $\{\mathbf{x}_{0:k}^{(i)}\}_{i=1}^{N_\mathrm{p}}$ of N_p samples is drawn directly from the posterior $\mathbf{x}_{0:k}^{(i)} \sim \mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ and the integral is approximated by the mean over the values of $\mathbf{f}(\cdot)$ evaluated at the different samples [55]

$$\mathbf{E}\left[\mathbf{f}(\mathbf{x}_{0:k})\right] \approx \frac{1}{N_\mathrm{p}}\sum_{i=1}^{N_\mathrm{p}}\mathbf{f}(\mathbf{x}_{0:k}^{(i)}). \tag{2.30}$$

The approximation in (2.30) can be also obtained by approximating the posterior density with a Dirac mixture density

$$\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \frac{1}{N_\mathrm{p}}\sum_{i=1}^{N_\mathrm{p}}\delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)})\ , \tag{2.31}$$

where $\delta(\cdot)$ is the Dirac distribution. Replacing the full posterior in (2.29) with (2.31) reduces the integral to the sum (2.30) due to the sifting properties of the Dirac distribution.

2.4.1 Importance Sampling

Approximation (2.31) showed that it is possible to get a numerical approximation for the posterior based on samples. Unfortunately, in (2.31) these samples have to be drawn from the unknown posterior that we want to calculate. This problem is addressed by importance sampling, where the samples are no longer drawn directly from the posterior, but from another density $\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ called importance or proposal density in the literature [55]. In importance sampling the integrand in (2.29) is multiplied with $\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})/\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$

$$\mathbf{E}\left[\mathbf{f}(\mathbf{x}_{0:k})\right] = \int \mathbf{f}(\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})\frac{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})}d\mathbf{x}_{0:k}. \tag{2.32}$$

This is equal to the following expected value over the proposal density

$$\mathbf{E}\left[\frac{\mathbf{f}(\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})}\right] = \int \frac{\mathbf{f}(\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})}\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})d\mathbf{x}_{0:k}\ , \tag{2.33}$$

which can be yet again approximated with a set of samples but this time the samples are drawn from the proposal density. If the support of the proposal density contains the support of the posterior we may write

$$\mathbf{E}\left[\mathbf{f}(\mathbf{x}_{0:k})\right] \approx \frac{1}{N_\mathrm{p}}\sum_{i=1}^{N_\mathrm{p}} w_k^{(i)}\,\mathbf{f}(\mathbf{x}_{0:k}^{(i)})\ , \tag{2.34}$$

with the weights $w_k^{(i)} = \frac{\mathrm{p}(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k})}{\pi(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k})}$ and $\mathbf{x}_{0:k}^{(i)} \sim \pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. A comparison of (2.29) and (2.34) yields an alternative approximation of the posterior based on weighted samples drawn from the proposal density

$$\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \frac{1}{N_\mathrm{p}}\sum_{i=1}^{N_\mathrm{p}} w_k^{(i)}\delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)})\,. \tag{2.35}$$

The idea of importance sampling is therefore to approximate the integrals and the posterior with samples drawn from a density different to the posterior and account for that difference by assigning weights to the samples.

One critical point still to be addressed is the calculation of the weights. The weights require an evaluation of the posterior at the sampling points. As shown in the following, this is avoidable when the likelihood and the prior distribution in the numerator of the posterior in (2.5) can be evaluated at the sample points. To achieve this the trick is to also approximate the normalization constant in (2.5) with the samples from the proposal density

$$\begin{aligned}\mathrm{p}(\mathbf{z}_{1:k}) &= \int \mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k})d\mathbf{x}_{0:k} \\ &\approx \frac{1}{N_\mathrm{p}}\sum_{i=1}^{N_\mathrm{p}}\frac{\mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k}^{(i)})\,\mathrm{p}(\mathbf{x}_{0:k}^{(i)})}{\pi(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k})}\ .\end{aligned} \tag{2.36}$$

If the posterior in (2.32) is replaced with (2.5) and the approximated normalization constant (2.36) is inserted the expected value becomes

$$\mathbf{E}\left[\mathbf{f}(\mathbf{x}_{0:k})\right] \approx \frac{\int \frac{\mathrm{f}(\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})}\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})d\mathbf{x}_{0:k}}{\frac{1}{N_\mathrm{p}}\sum_{i=1}^{N_\mathrm{p}}\frac{\mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k}^{(i)})\,\mathrm{p}(\mathbf{x}_{0:k}^{(i)})}{\pi(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k})}} . \tag{2.37}$$

From the integral above we can once more find an approximation to the posterior

$$\mathrm{p}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)}\delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)}) , \tag{2.38}$$

where the weights are given by

$$w_k^{(i)} = \frac{\frac{\mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k}^{(i)})\,\mathrm{p}(\mathbf{x}_{0:k}^{(i)})}{\pi(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k})}}{\sum_{j=1}^{N_\mathrm{p}}\frac{\mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k}^{(j)})\,\mathrm{p}(\mathbf{x}_{0:k}^{(j)})}{\pi(\mathbf{x}_{0:k}^{(j)}|\mathbf{z}_{1:k})}} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^{N_\mathrm{p}}\tilde{w}_k^{(j)}} \tag{2.39}$$

and the samples are drawn from the proposal density. The weight are now normalized in the sense that $w_k^{(i)} \geq 0$ and $\sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} = 1$. The new approximation (2.38) enables the calculation of the posterior and its moments based solely on the evaluation of the likelihood and the prior at samples drawn from the proposal density.

2.4.2 Particle Filter

So far the importance sampling was done on the basis of the complete state history $\mathbf{x}_{0:k}$. This is avoided by applying sequential importance sampling [55] that only samples the current state and otherwise leaves the history unchanged. For Markovian systems sequential sampling is achieved through some simple assumptions about the proposal density. In particular, it is assumed that the state at time $k-1$ is independent from current observations at time step k. With this assumption the following factorization of the proposal density becomes possible [55]

$$\pi(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})\,\pi(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}). \tag{2.40}$$

Sequential importance sampling further requires that the system, whose posterior is estimated, is Markovian for which the following identity holds

$$\begin{aligned}\mathrm{p}(\mathbf{z}_{1:k}|\mathbf{x}_{0:k})\,\mathrm{p}(\mathbf{x}_{0:k}) &= \mathrm{p}(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) \\ &= \mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)\,\mathrm{p}(\mathbf{x}_k|\mathbf{x}_{k-1})\,\mathrm{p}(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) .\end{aligned} \tag{2.41}$$

With the factorized proposal density (2.40) and identity (2.41) the weight calculation in (2.39) is reformulated to

$$w_k^{(i)} = \frac{\dfrac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k^{(i)})\,\mathrm{p}(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})\,\mathrm{p}(\mathbf{x}_{0:k-1}^{(i)},\mathbf{z}_{1:k-1})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)},\mathbf{z}_{1:k})\,\pi(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{z}_{1:k-1})}}{\sum\limits_{j=1}^{N_\mathrm{p}} \dfrac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k^{(j)})\,\mathrm{p}(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)})\,\mathrm{p}(\mathbf{x}_{0:k-1}^{(j)},\mathbf{z}_{1:k-1})}{\pi(\mathbf{x}_k^{(j)}|\mathbf{x}_{0:k-1}^{(j)},\mathbf{z}_{1:k})\,\pi(\mathbf{x}_{0:k-1}^{(j)}|\mathbf{z}_{1:k-1})}}. \tag{2.42}$$

Taking into account the identity $\mathrm{p}(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) = \mathrm{p}(\mathbf{z}_{1:k-1}|\mathbf{x}_{0:k-1})\,\mathrm{p}(\mathbf{x}_{0:k-1})$, the quotient between $\mathrm{p}(\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k-1})$ and $\pi(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{z}_{1:k-1})$ in (2.42) is according to (2.39) the unnormalized weight $\tilde{w}_{k-1}^{(i)}$ for the i-th sample at time step $k-1$. Since the unnormalized weights show up in the nominator and denominator of (2.42) and their normalization constant is the same for all weights we can replace them also with the normalized weights, resulting in a recursive formulation for the weight calculation

$$w_k^{(i)} = \frac{\dfrac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k^{(i)})\,\mathrm{p}(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)},\mathbf{z}_{1:k})} w_{k-1}^{(i)}}{\sum\limits_{j=1}^{N_\mathrm{p}} \dfrac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k^{(j)})\,\mathrm{p}(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)})}{\pi(\mathbf{x}_k^{(j)}|\mathbf{x}_{0:k-1}^{(j)},\mathbf{z}_{1:k})} w_{k-1}^{(j)}} = \frac{\tilde{w}_k^{(i)}}{\sum\limits_{j=1}^{N_\mathrm{p}} \tilde{w}_k^{(j)}}\,. \tag{2.43}$$

Furthermore, due to the factorization (2.40) creating a sample of the state history at time step k only requires N_p samples of the current state $\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k})$. The new samples are then added to their corresponding state history to form a sample of the complete history $\mathbf{x}_{0:k}$ [56].

A direct implementation of sequential importance sampling to approximate the posterior will most likely give poor results. This is caused by the weight calculation, which weights the samples relative to each other. Over time when the samples start to explore the state space, the samples will only sparsely cover the part of the state space in which the posterior is concentrated. When this happens, the weights are not evenly distributed anymore and only a few samples will have nonzero weights. This effect is commonly referred to as degeneracy [52, 55, 57, 58]. Therefore, a high weight does not necessarily mean that a sample is at a point in the state space where the posterior is high, it only means the sample is at a point where the posterior is high compared to the other samples. If the number of samples cannot be increased to compensate this effect, which is often not possible due to the increase in computational complexity, a procedure is required that moves samples towards parts of the state space that are worth exploring. Such a procedure is resampling, which takes the samples and their corresponding weights and generates a new set of samples with equal weights. For resampling multiple algorithms exist [58, 59], but they all follow the same principle idea to reproduce samples with a high weight and remove samples with low weights from the set. Thus, resampling replaces the weighted set of samples with an unweighted set, where the “probability” of a sample is expressed by the number of its copies in the set rather than by its weight. One could say the new unweighted set has been drawn from the posterior, assuming the weighted set was an appropriate approximation.

The combination of sequential importance sampling and resampling results in the particle filter, that owes its name to the samples which may be considered as particles that move through the state space. In the literature a variety of particle filters exist [55], where one of the first versions of the particle filter was proposed in [60]. One iteration of a generic particle filter is shown in Algorithm 2.2. Like

Algorithm 2.2 Generic particle filter

1: Input: Weighted particle set $\{\mathbf{x}_{0:k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_\text{p}}$ and observations $\mathbf{z}_{1:k}$
2: Output: Updated set $\{\mathbf{x}_{0:k}^{(i)}, w_k^{(i)}\}_{i=1}^{N_\text{p}}$
3: **for all** $i \in \{1, \dots, N_\text{p}\}$ **do**
4: $\quad \mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k})$ //Sample new state (Prediction step)
5: $\quad \tilde{w}_k^{(i)} = \frac{\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k^{(i)})\,\mathrm{p}(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)},\mathbf{z}_{1:k})} w_{k-1}^{(i)}$ //Update weight (Update step)
6: **end for**
7: **for all** $i \in \{1, \dots, N_\text{p}\}$ **do**
8: $\quad w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^{N_\text{p}} \tilde{w}_k^{(j)}$ //Normalize weights
9: **end for**
10: **if** Resampling necessary **then**
11: $\quad$ Draw N_p particles from $\{\mathbf{x}_{0:k}^{(i)}, w_k^{(i)}\}_{i=1}^{N_\text{p}}$
12: $\quad$ Set all weights to $w_k^{(i)} = \frac{1}{N_\text{p}}$
13: **end if**
14: **return**

the Kalman filter, the particle filter consists of a prediction step and an update step. In the prediction step, the particle state is predicted by drawing from the proposal density and in the weight update the information from the current observation is considered. In line eleven and twelve of Algorithm 2.2 resampling is performed only when necessary. In order to decide when this is the case a common measure is the effective number of particles [56]

$$N_\text{eff} = \frac{1}{\sum_{i=1}^{N_\text{p}} \left(w_k^{(i)}\right)^2}. \tag{2.44}$$

The effective number of particles N_eff is equal to N_p when all weights are the same and it becomes $N_\text{eff} = 1$ when one particle has weight $w_k^{(i)} = 1$ and the rest have weight zero. Resampling is triggered when N_eff falls below a predefined threshold. The threshold can be considered as a tuning parameter of the filter.

To conclude this section, we want to give some remarks regarding the approximated posterior. The derived particle filter aims to approximate the full posterior by the Dirac mixture density in (2.38). If instead the filer density is desired a practical approach is to just neglect the history [52, 61, 62] and form a approximation of the filter density by

$$\mathrm{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_\text{p}} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \,. \tag{2.45}$$

Formally this approximation of the filter posterior is not correct since it ignores that in the filter density a marginalization over the state history has to be performed. A formally more correct approximation is given by the marginal particle filter [63]. Unfortunately, the computational complexity of the marginal particle filter is quadratic $\mathcal{O}({N_\text{p}}^2)$ in the number of particles when implemented naively, while the complexity of the particle filter in Algorithm 2.2 grows linearly $\mathcal{O}(N_\mathrm{p})$. Thus, in practice the simple approximation (2.45) is often preferred.

2.5 Gaussian Processes

A Gaussian process (GP) is a stochastic process described by a family of Gaussian distributed random variables $\{\mathrm{f}(\mathbf{y}) \in \mathbb{R} | \mathbf{y} \subseteq \mathbb{R}^{N_\mathrm{y}}\}$ [53]. The random variables $\mathrm{f}(\mathbf{y})$ in the GP are indexed by an input vector $\mathbf{y}$ and the process is fully defined by its covariance function $\mathrm{k}(\mathbf{y}, \mathbf{y}')$ between two input vectors and the mean function $\mathrm{m}(\mathbf{y})$ [64]. In a GP, the set of random variables $\{\mathrm{f}(\mathbf{y}^{(i)}) | \mathbf{y}^{(i)} \in \mathcal{Y}\}$ associated to the finite subset $\mathcal{Y} = \{\mathbf{y}^{(i)}\}_{i=1}^{N}$ of inputs is jointly Gaussian [65] and the density is given by [66]

$$\begin{bmatrix} \mathrm{f}(\mathbf{y}^{(1)}) \\ \vdots \\ \mathrm{f}(\mathbf{y}^{(N)}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathrm{m}(\mathbf{y}^{(1)}) \\ \vdots \\ \mathrm{m}(\mathbf{y}^{(N)}) \end{bmatrix}, \begin{bmatrix} \mathrm{k}(\mathbf{y}^{(1)}, \mathbf{y}^{(1)}) & \cdots & \mathrm{k}(\mathbf{y}^{(1)}, \mathbf{y}^{(N)}) \\ \vdots & \ddots & \vdots \\ \mathrm{k}(\mathbf{y}^{(N)}, \mathbf{y}^{(1)}) & \cdots & \mathrm{k}(\mathbf{y}^{(N)}, \mathbf{y}^{(N)}) \end{bmatrix} \right). \tag{2.46}$$

Since a stochastic process is characterized by the joint densities of all its finite sets of random variables [53] it can be seen from (2.46) that it sufficient to know $\mathrm{k}(\cdot, \cdot)$ and $\mathrm{m}(\cdot)$ to specify the properties of the process.

In principal the subset $\mathcal{Y}$ can be arbitrarily large and the input vectors in it can be arbitrarily close, thus a GP can be seen as the distribution of a random function $\mathrm{f}(\cdot)$. As for normal random variables, we can draw samples from the GP to obtain a realization of the scalar random function, which we denote by

$$\mathrm{f}(\mathbf{y}) \sim \mathcal{GP}(\mathrm{m}(\mathbf{y}), \mathrm{k}(\mathbf{y}, \mathbf{y}')). \tag{2.47}$$

This means a GP is a generalization of the multivariate Gaussian distribution to possibly infinite dimensions since a function can be seen as an infinitely long vector. The fact that for each subset of inputs the joint density of the function outputs is the Gaussian pdf in (2.46) enables closed form regression based on training data as long as the likelihood of the data is also Gaussian.

For GP regression the posterior density $\mathrm{p}(\mathbf{f}^* | \mathcal{D}, \mathcal{Y}^*)$ at some test inputs $\mathcal{Y}^* = \{\mathbf{y}^{*,(i)}\}_{i=1}^{N}$ conditioned on a training data set $\mathcal{D} = \{\mathcal{Y}^\mathcal{D}, \mathcal{Z}^\mathcal{D}\}$ is calculated, where the vector $\mathbf{f}^* = [\mathrm{f}(\mathbf{y}^{*,(1)}) \cdots \mathrm{f}(\mathbf{y}^{*,(N)})]^\top$ represents the values of the GP at the test inputs. The test data set is composed of a set of scalar observations $\mathcal{Z}^\mathcal{D} = \{z^{\mathcal{D},(i)}\}_{i=1}^{M}$ and a corresponding set of input vectors $\mathcal{Y}^\mathcal{D} = \{\mathbf{y}^{\mathcal{D},(i)}\}_{i=1}^{M}$. With the posterior we can then draw realizations of the random function or evaluate the posterior mean and covariance at different inputs.

To obtain the posterior, for now the observations in the data set are assumed to be noiseless observations $\mathcal{Z}^\mathcal{D} = \{f(\mathbf{y}), \mathbf{y} \in \mathcal{Y}^\mathcal{D}\}$. The joint density of the observations $\mathcal{Z}^\mathcal{D}$ and the GP values $\mathbf{f}^*$ conditioned on their associated inputs is then simply obtained from (2.46)

$$\mathrm{p}(\mathbf{f}^*, \mathcal{Z}^\mathcal{D} | \mathcal{Y}^*, \mathcal{Y}^\mathcal{D}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{m}^* \\ \mathbf{m}^\mathcal{D} \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathcal{Y}^*, \mathcal{Y}^*) & \mathbf{K}(\mathcal{Y}^*, \mathcal{Y}^\mathcal{D}) \\ \mathbf{K}(\mathcal{Y}^\mathcal{D}, \mathcal{Y}^*) & \mathbf{K}(\mathcal{Y}^\mathcal{D}, \mathcal{Y}^\mathcal{D}) \end{bmatrix} \right), \tag{2.48}$$

with $\mathbf{m}^* = [\mathrm{m}(\mathbf{y}^{*,(1)}) \cdots \mathrm{m}(\mathbf{y}^{*,(N)})]^\top$ and $\mathbf{m}^\mathcal{D} = [\mathrm{m}(\mathbf{y}^{\mathcal{D},(1)}) \cdots \mathrm{m}(\mathbf{y}^{\mathcal{D},(M)})]^\top$. The blocks of the covariance matrix in (2.48) are obtained from the covariance function evaluated for the different pairs of inputs. For example, the upper right block of the covariance matrix is given by

$$\mathbf{K}(\mathcal{Y}^*, \mathcal{Y}^\mathcal{D}) = \begin{bmatrix} \mathrm{k}(\mathbf{y}^{*,(1)}, \mathbf{y}^{\mathcal{D},(1)}) & \cdots & \mathrm{k}(\mathbf{y}^{*,(1)}, \mathbf{y}^{\mathcal{D},(M)}) \\ \vdots & \ddots & \vdots \\ \mathrm{k}(\mathbf{y}^{*,(N)}, \mathbf{y}^{\mathcal{D},(1)}) & \cdots & \mathrm{k}(\mathbf{y}^{*,(N)}, \mathbf{y}^{\mathcal{D},(M)}) \end{bmatrix}. \tag{2.49}$$

The remaining blocks are obtained in a similar way by replacing $\mathcal{Y}^*$ and $\mathcal{Y}^\mathcal{D}$ by the desired input sets. In the following, the shorthand notation $\mathbf{K}_{\mathcal{D}\mathcal{D}}$ for $\mathbf{K}(\mathcal{Y}^\mathcal{D}, \mathcal{Y}^\mathcal{D})$, $\mathbf{K}_{\mathcal{D}*}$ for $\mathbf{K}(\mathcal{Y}^\mathcal{D}, \mathcal{Y}^*)$, and $\mathbf{K}_{**}$ for $\mathbf{K}(\mathcal{Y}^*, \mathcal{Y}^*)$ will be used. Note, the matrices are the blocks of a block covariance matrix and therefore $\mathbf{K}_{\mathcal{D}*}^\top = \mathbf{K}_{*\mathcal{D}}$ holds. Since the joint distribution (2.48) is Gaussian the posterior can be directly calculated in closed form

$$\mathrm{p}(\mathbf{f}^* \,|\mathcal{D}, \mathcal{Y}^*) = \mathcal{N}\left(\mathbf{f}^*; \mathbf{m}^* + \mathbf{K}_{*\mathcal{D}}\mathbf{K}_{\mathcal{D}\mathcal{D}}^{-1}\left(\mathbf{z}_{1:M}^\mathcal{D} - \mathbf{m}^\mathcal{D}\right), \mathbf{K}_{**} - \mathbf{K}_{*\mathcal{D}}\mathbf{K}_{\mathcal{D}\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}*}\right), \tag{2.50}$$

where $\mathbf{z}_{1:M}^\mathcal{D}$ is a column vector containing all M observations from $\mathcal{Z}^\mathcal{D}$ stacked on top of each other. When the observations are affected by additive white Gaussian noise (AWGN) matrix $\mathbf{K}_{\mathcal{D}\mathcal{D}}$ is replaced with $\mathbf{K}_{\mathcal{D}\mathcal{D}} + (\sigma^z)^2\mathbf{I}_{M\times M}$, where $(\sigma^z)^2$ is the variance of the noise. This directly results from (2.48) when $(\sigma^z)^2\mathbf{I}_{M\times M}$ is added to the lower right block $\mathbf{K}_{\mathcal{D}\mathcal{D}}$ of the covariance matrix [64].

Before a GP can be used for regression, a covariance and mean function has to be selected that suits the data set. After selecting the functions their hyperparameters must be set. While choosing the covariance and mean function often is done manually, the hyperparameters are optimized numerically based on the given data. For optimization the marginal likelihood $\mathrm{p}(\mathcal{Z}^\mathcal{D}|\mathcal{Y}^\mathcal{D})$ obtained from the integral

$$\begin{aligned} \mathrm{p}(\mathcal{Z}^\mathcal{D}|\mathcal{Y}^\mathcal{D}) &= \int \mathrm{p}(\mathcal{Z}^\mathcal{D}, \mathbf{f}(\mathcal{Y}^\mathcal{D})|\mathcal{Y}^\mathcal{D})\,d\,\mathbf{f}(\mathcal{Y}^\mathcal{D}) \\ &= \int \underbrace{\mathrm{p}(\mathcal{Z}^\mathcal{D}|\,\mathbf{f}(\mathcal{Y}^\mathcal{D}), \mathcal{Y}^\mathcal{D})}_{\sim\mathcal{N}(\mathbf{f}(\mathcal{Y}^\mathcal{D}), (\sigma^z)^2\mathbf{I}_{M\times M})}\ \underbrace{\mathrm{p}(\mathbf{f}(\mathcal{Y}^\mathcal{D})|\mathcal{Y}^\mathcal{D})}_{\sim\mathcal{N}(\mathbf{m}^\mathcal{D}, \mathbf{K}_{\mathcal{D}\mathcal{D}})}\,d\,\mathbf{f}(\mathcal{Y}^\mathcal{D}), \end{aligned} \tag{2.51}$$

is used as cost function to find the parameters that best explain the available observations, where $\mathbf{f}(\mathcal{Y}^\mathcal{D})$ are the GP values at the inputs in the training data set.

With (2.51) the joint Gaussian of $\mathcal{Z}^\mathcal{D}$ and $\mathbf{f}(\mathcal{Y}^\mathcal{D})$ can be calculated from which we then can easily obtain the desired marginal likelihood

$$\mathrm{p}(\mathcal{Z}^\mathcal{D}|\mathcal{Y}^\mathcal{D}) = \mathcal{N}\left(\mathcal{Z}^\mathcal{D}; \mathbf{m}^\mathcal{D}, \mathbf{K}_{\mathcal{D}\mathcal{D}} + (\sigma^z)^2\mathbf{I}_{M\times M}\right). \tag{2.52}$$

2.6 Lower Bounds for Estimators

For linear systems with Gaussian noise the Kalman filter provides a closed form solution for the posterior density and hence captures all the information of the state given the measurements. For nonlinear estimation problems the calculation of the posterior density in closed form is impossible for most cases and hence approximations are used instead, e.g., a particle filter. Thus, in the nonlinear case it is often not clear which estimator and approximation should be chosen to achieve optimal results. It is therefore desired to have a lower limit on the achievable estimation accuracy to rate the quality of an estimator. A common and well-known approach to find such a limit is the derivation of lower bounds. Besides being a benchmark for estimators, a lower bound is also useful to gain insights into a particular estimation problem. In the following section the concept of lower bounds will be introduced on the example of the CRLB and its extension to Bayesian estimation.

2.6.1 Cramér-Rao Lower Bound

The classical, non-Bayesian, CRLB is a lower bound for the covariance matrix of unbiased estimators. The CRLB fulfills the matrix inequality

$$\mathbf{Cov}\left[\hat{\mathbf{x}}(\mathbf{z})\right] \geq \mathbf{F}^{-1}(\mathbf{x}) \; , \tag{2.53}$$

where $\hat{\mathbf{x}}(\mathbf{z})$ is an unbiased estimator of the true state vector $\mathbf{x}$ based on the observations $\mathbf{z}$. The bound is defined by the inverse Fisher information matrix (FIM) $\mathbf{F}^{-1}(\mathbf{x})$. The inequality sign states that the difference between the estimator covariance matrix and the bound is a semi positive definite matrix $\mathbf{Cov}\left[\hat{\mathbf{x}}(\mathbf{z})\right] - \mathbf{F}^{-1} \geq \mathbf{0}$. From this it follows that the diagonal elements fulfill the inequality

$$\left[\mathbf{Cov}\left[\hat{\mathbf{x}}(\mathbf{z})\right]\right]_{ii} \geq \left[\mathbf{F}^{-1}(\mathbf{x})\right]_{ii} . \tag{2.54}$$

The elements $[\mathbf{F}(\mathbf{x})]_{ij}$ of the FIM are given by the expected value of the derivatives of the log-likelihood

$$[\mathbf{F}(\mathbf{x})]_{ij} = -\,\mathbf{E}_{\mathbf{z}|\mathbf{x}}\left[\frac{\partial^2}{\partial[\mathbf{x}]_i \partial[\mathbf{x}]_j} \ln \mathrm{p}(\mathbf{z}|\mathbf{x})\right] \; , \tag{2.55}$$

where the index of the expectation operator $\mathbf{E}_{\mathbf{z}|\mathbf{x}}\left[\cdot\right]$ indicates that the expectation is taken w.r.t. the likelihood $p(\mathbf{z}|\mathbf{x})$. The bound is only valid when for the likelihood $p(\mathbf{z}|\mathbf{x})$ the derivatives

$$\frac{\partial \ln \mathrm{p}(\mathbf{z}|\mathbf{x})}{\partial[\mathbf{x}]_i} \quad \text{and} \quad \frac{\partial^2 \ln \mathrm{p}(\mathbf{z}|\mathbf{x})}{\partial[\mathbf{x}]_i [\mathbf{x}]_j} \; , \tag{2.56}$$

exist and are absolutely integrable for all elements of $\mathbf{x}$.

For many estimation problems Gaussian likelihoods are considered. It is therefore convenient to have an expression of the FIM for Gaussian likelihoods with mean $\boldsymbol{\mu}(\mathbf{x})$ and covariance $\boldsymbol{\Sigma}(\mathbf{x})$

$$[\mathbf{F}(\mathbf{x})]_{ij} = \left(\frac{\partial \boldsymbol{\mu}(\mathbf{x})}{\partial[\mathbf{x}]_i}\right)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}) \frac{\partial \boldsymbol{\mu}(\mathbf{x})}{\partial[\mathbf{x}]_j} + \frac{1}{2}\,\mathrm{trace}\left(\boldsymbol{\Sigma}^{-1}(\mathbf{x}) \frac{\partial \boldsymbol{\Sigma}(\mathbf{x})}{\partial[\mathbf{x}]_i} \boldsymbol{\Sigma}^{-1}(\mathbf{x}) \frac{\partial \boldsymbol{\Sigma}(\mathbf{x})}{\partial[\mathbf{x}]_j}\right) . \tag{2.57}$$

A detailed derivation of the CRLB and the FIM for Gaussians, as well as the required regularity conditions is given in [67, 68].

2.6.2 Bayesian Cramér-Rao Lower Bound

The CRLB only considers the likelihood and treats the state as a nonrandom parameter. Thus, the CRLB applies to ML estimators but not to Bayesian estimators. The extension of the CRLB to Bayesian estimators was first introduced by Van Trees [68]. Unlike the classical CRLB, the BCRLB is a bound on the MSE matrix and not the covariance. The BCRLB is defined by the matrix inequality

$$\mathbf{M} = \mathbf{E}_{\mathbf{x},\mathbf{z}}\left[(\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x})(\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x})^\top\right] \geq \mathbf{J}^{-1}\,, \tag{2.58}$$

where $\mathbf{M}$ is the MSE matrix of estimator $\hat{\mathbf{x}}(\mathbf{z})$, and $\mathbf{J}$ is the Bayesian information matrix (BIM). As for the classical case, the inequality sign means that $\mathbf{M} - \mathbf{J}^{-1} \geq \mathbf{0}$ is a semi positive definite matrix. The BIM is given by

$$\mathbf{J} = -\,\mathbf{E}_{\mathbf{x},\mathbf{z}}\left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{x},\mathbf{z})\right]. \tag{2.59}$$

Note, the expectation is now w.r.t. the joint density $\mathrm{p}(\mathbf{x},\mathbf{z})$. The operator $\Delta_{\mathbf{b}}^{\mathbf{a}} = \nabla_{\mathbf{b}}\nabla_{\mathbf{a}}^\top$ is the outer product of two nabla operators $\nabla_{\mathbf{a}} = [\frac{\partial}{\partial[\mathbf{a}]_1} \cdots \frac{\partial}{\partial[\mathbf{a}]_{\dim(\mathbf{a})}}]^\top$. The application of $\Delta_{\mathbf{b}}^{\mathbf{a}}$ to a function gives a Hessian like matrix. According to [68], the BIM's regularity conditions require that for the estimator bias

$$\mathbf{b}(\mathbf{x}) = \int [\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}]\,\mathrm{p}(\mathbf{z}|\mathbf{x})d\mathbf{z} \tag{2.60}$$

the following limits

$$\begin{aligned}\lim_{[\mathbf{x}]_i \to \infty} \mathbf{b}(\mathbf{x})\,\mathrm{p}(\mathbf{x}) &= \mathbf{0}\\ \lim_{[\mathbf{x}]_i \to -\infty} \mathbf{b}(\mathbf{x})\,\mathrm{p}(\mathbf{x}) &= \mathbf{0}\end{aligned} \tag{2.61}$$

hold for each element $[\mathbf{x}]_i$ of $\mathbf{x}$. Furthermore, the BCRLB requires that the first- and second-order partial derivatives of the joint density $\mathrm{p}(\mathbf{x},\mathbf{z})$ are absolutely integrable.

The BIM is closely related to the FIM. This is not directly visible from (2.59) but when we apply (2.2) on the joint density and substitute $\mathrm{p}(\mathbf{x},\mathbf{z}) = \mathrm{p}(\mathbf{z}|\mathbf{x})\,\mathrm{p}(\mathbf{x})$ into (2.59) we get

$$\begin{aligned}\mathbf{J} &= \mathbf{E}_{\mathbf{x}}\left[\mathbf{E}_{\mathbf{z}|\mathbf{x}}\left[-\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{z}|\mathbf{x})\right]\right] + \mathbf{E}_{\mathbf{x}}\left[-\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{x})\right]\\ &= \mathbf{E}_{\mathbf{x}}\left[\mathbf{F}(\mathbf{x})\right] + \mathbf{E}_{\mathbf{x}}\left[-\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{x})\right]\,,\end{aligned} \tag{2.62}$$

where $\mathbf{F}(\mathbf{x})$ is the FIM of $\mathrm{p}(\mathbf{z}|\mathbf{x})$ from (2.55). Equation (2.62) shows that the BIM consists of two parts. The first part is the expected value of the FIM w.r.t. the prior density of $\mathbf{x}$ and thus represents the information contained in the observations. The second part contains the information obtained from the prior distribution. In practice, the relation between BIM and FIM comes in handy because for many likelihood functions the FIM was already derived in the literature.

2.7 Summary

In this chapter, different theoretical concepts were briefly introduced that will form the basis for the algorithms developed later in this thesis. In particular, the concepts of Bayesian estimation and recursive filtering were introduced. For linear and Gaussian systems, it was shown that the posterior can be calculated exactly with the famous Kalman filter. For nonlinear systems a general exact solution for the posterior is not available, hence the idea of importance sampling was explained, that allows to approximate the posterior with a set of weighted samples. Then, a generic particle filter was derived on the basis of recursive importance sampling. In addition, GPs, CRLBs, and the BCRLB were shortly introduced. This will become important for the derivation of lower bounds for magnetic localization, where GPs will be integrated into the BCRLB.

Chapter 3

Magnetic Field in the Railway Domain

The magnetic field in the railway domain shows strong distortions caused by magnetic material along the tracks. If these distortions are fixed to a position and persistent over time, they can be used for localization. In addition to the desired position dependent distortions, temporal disturbances in the magnetic field are observed in practice that are caused, e.g., by electric currents or moving magnetic material in the vicinity of the magnetometer. The measured magnetic field is also influenced by the mounting position of the magnetometer inside the train. These disturbances and influences have to be considered in the development of a magnetic localization system. The purpose of this chapter is to give a brief overview on the spatial variability of the magnetic field and some sources of disturbance.

3.1 Spatial Variability

The spatial variability of the magnetic distortions plays an important role when it comes to the achievable localization accuracy. This seems intuitively plausible but will be also formalized in Chapter 6, in which the BCRLB for magnetic field-based localization is derived. To get a first impression of the spatial variability, here we will investigate how the signal power of the magnetic field distortions is distributed across the spatial frequency domain. In order to do this, the power spectral density (psd) for the magnitude of the magnetic field shown in Figure 1.1 is calculated.

For the calculation of the psd the recorded magnetometer time series data is transformed from the time domain to the spatial domain based on GNSS data and is then interpolated on a equidistant grid with a spacing of 0.1 m. The magnetic field in the spatial domain is now a function of the along-track position and therefore the unit of the frequency in its psd is m^{-1}. The spacing of 0.1 m and the sampling theorem limit the spatial frequencies in the psd to $5\,\mathrm{m}^{-1}$. Since the used magnetic field data is recorded with a real magnetometer, the data not only contains the magnetic distortions useful for localization, but also sensor noise and other disturbances. One example for a possible disturbance, that will be discussed also in the next section, is the magnetic field induced by currents in the overhead power line. Fortunately, the fixed frequency of the currents allows to filter them out from the raw magnetometer measurements. In contrast to the disturbances caused by the power line currents, the sensor noise cannot be removed from the magnetometer time series data due to its broad and flat frequency spectrum. As a consequence, the sensor noise is still part of the magnetometer time series data during the spatial transformation and the calculation of the psd. When we want to differentiate between the magnetic distortions useful for localization and the noise it is therefore necessary to relate the psd to the noise power density in the spatial frequency domain. In order to get an estimate of this

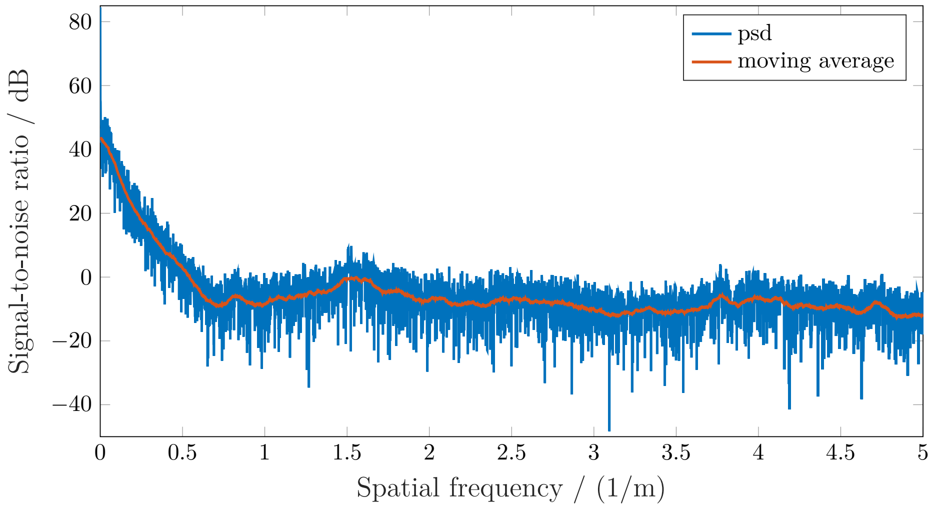

Figure 3.1: Signal-to-noise ratio of the magnetic field magnitude of the example shown in Figure 1.1. The signal-to-noise ratio is calculated from the psd of the magnetic field magnitude in the spatial domain and the noise power density of the magnetometer.

density we use a few seconds of magnetometer measurements during which the train is at standstill and remove the mean and the magnetic field induced by the overhead power line currents. From the resulting signal we then calculate the overall noise power and distribute this power equally in the whole spatial frequency domain to obtain the desired density. The assumption of an equally distributed noise power is a simplification and only a rough estimate, but gives a starting point to investigate the spatial variability of the magnetic field.

With the noise power density the signal-to-noise ratio at the different spatial frequencies can be calculated by dividing the psd of the data from Figure 1.1 by the estimated noise power density. The resulting signal-to-noise ratio in decibel is shown Figure 3.1. In addition to the strongly fluctuating signal-to-noise ratio also its moving average is shown to reveal the overall trend. From the moving average it is clearly visible that starting from around $0.54\,\mathrm{m}^{-1}$ the signal power is in the same order as the noise power or less. Taking into account that the amount of information one can extract from a signal strongly depends on the signal-to-noise ratio, frequencies $> 0.54\,\mathrm{m}^{-1}$ will only have a minor contribution w.r.t. the overall localization accuracy. We expect that frequencies around $0.22\,\mathrm{m}^{-1}$ and lower with a signal-to-noise ratio of 20 dB and higher will be much more important for localization. From a practical point of view this means that a train has to move in the order of $(0.22\,\mathrm{m}^{-1})^{-1} \approx 4.55\,\mathrm{m}$ to measure a significant change in the magnetic field and until accurate localization becomes possible.

3.2 Disturbance from Overhead Power Lines

On an electrified track the electric current in the overhead power line produces a magnetic field that alternates with the frequency of the current. The induced magnetic field is strong enough to have an considerable influence on the measurements of a train mounted magnetometer. This is illustrated in Figure 3.2 with a magnetometer measurement sequence, recorded once more with the setup described in Section 7.1.1. During the measurements the magnetometer was mounted inside the driver cab and

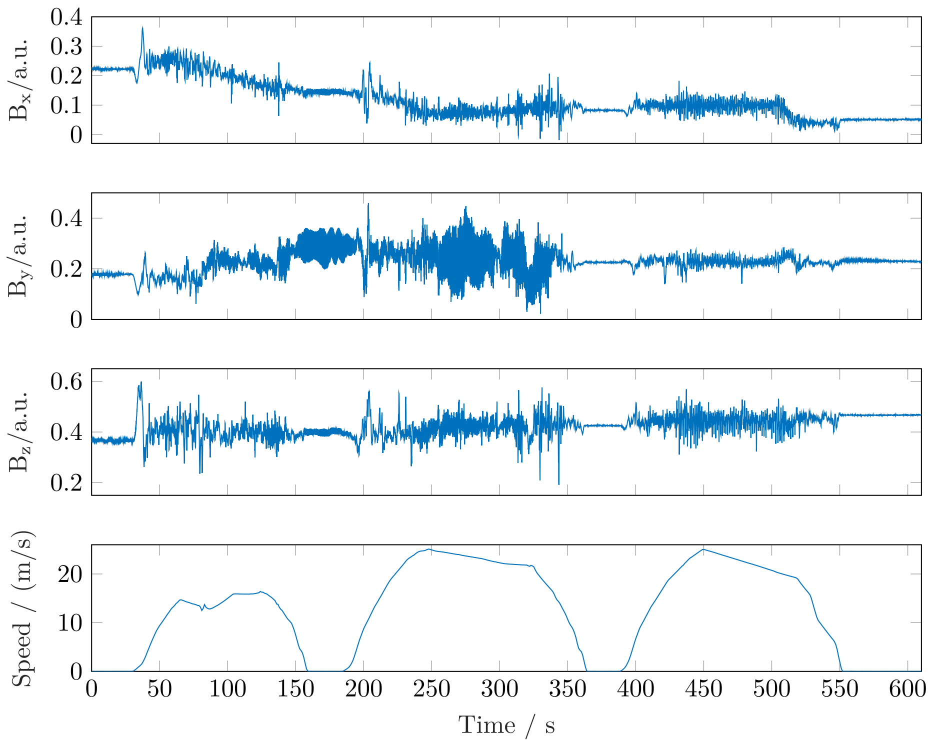

Figure 3.2: Measured magnetic flux density of the magnetic vector field and train speed during the measurements. The flux density B_y is strongly affected by the magnetic field of the overhead power line.

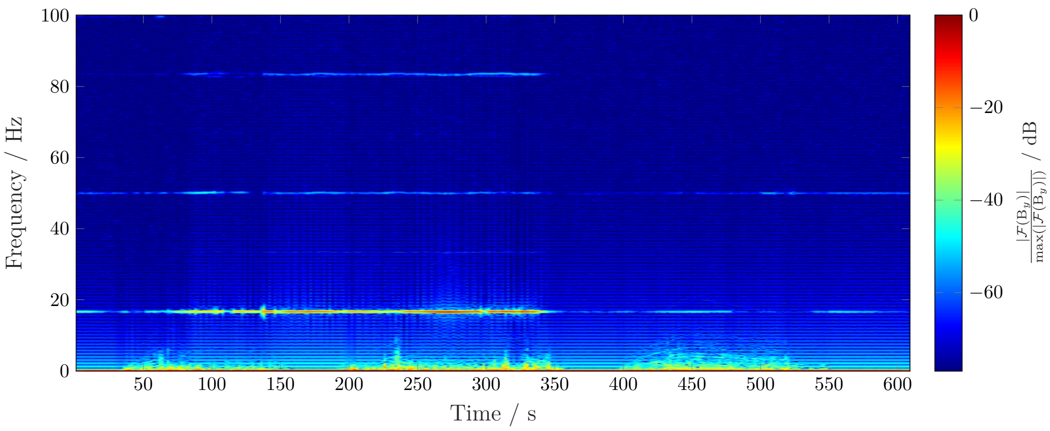

Figure 3.3: Short time Fourier transform of a roughly 600 s long magnetometer data set recorded with a diesel train driving on an electrified track. The magnetometer was mounted inside the driver cab during the measurements.

thus was not near the power line. From the data it seems that the magnetic field varies between 160 s and 180 s although the train is at standstill, as can be seen from the train speed in the lower part of Figure 3.2, and the measurements should be constant with some additive noise. These disturbances are especially present in the measured flux density B_y. However, also the other measurements are impacted to some degree. To further investigate these variations, the amplitude spectrum of the short time Fourier transform (STFT) of B_y is shown in Figure 3.3. The amplitude spectrum is given in dB, where the reference value is the maximum value of the STFT. In Figure 3.3 a strong line at the frequency 16.7 Hz is visible. This is the frequency of railway electrification in Germany, which confirms that the variations are caused by currents in the overhead power line. The STFT also shows horizontal lines at 50 Hz, the frequency of the European low-voltage network, and some harmonics of 16.7 Hz. The amplitude of the magnetic field at the different frequencies also varies with time, depending on the load on the power line.

Besides the signal components at fixed frequencies caused by electric currents, most of the signal power is allocated at frequencies below 16.7 Hz. These frequency components of the amplitude spectrum are the position dependent distortions of the Earth magnetic field important for position estimation. Since these distortions are persistent in time and at fixed positions, their frequencies in the STFT depend on the train speed during the measurements. In Figure 3.2 the train was traveling at speeds up to $25\,\mathrm{m\,s^{-1}}$. Considering the spatial variability of the magnetic field described in Section 3.1, a rough estimate for the speed at which the frequency of the position dependent distortions with a high signal-to-noise ratio start to overlap with the magnetic field of the overhead power line is obtained from the simple relation $16.7\,\mathrm{Hz}/0.22\,\mathrm{m^{-1}} \approx 76\,\mathrm{m\,s^{-1}}$. If the speed reaches this value, removing the power line induced disturbances becomes more difficult because a simple filter would also remove parts of the desired spatial variations.

It is important to mention that the disturbance from the overhead power line is not only encountered in electrical trains. In fact, the train with which the magnetometer data for Figure 3.2 and Figure 3.3 was recorded has a diesel engine. The currents that produce the magnetic field are caused by electrical trains in the vicinity that consume the power from the overhead power line. Thus, independent from the engine of the train, for magnetic localization we have to take care of the disturbances from the overhead power line. Further details of how the disturbances can be removed from the measurements are discussed in Chapter 7.

3.3 Influence of Mounting Position

The magnetic vector field does not only vary along the track, but it also varies in cross-track direction and with height. The magnetic field measured with a magnetometer therefore depends on its mounting position in the train. Another factor that influences the measurements are objects containing magnetic material in the vicinity of the magnetometer. In this section, some examples for the influence of the mounting position on the measurements are presented.

3.3.1 Nearby Magnetic Material

Magnetic material in the vicinity of a magnetometer influences its measurements. Since trains are partially made out of magnetic material this influence is present for every train mounted magnetometer. In the simplest case the magnetic material is not moving relative to the magnetometer. If this is the case, a calibration of the magnetometer would remove the influence of the material. Unfortunately, common calibration methods are not applicable for trains. Therefore, in the scope of this thesis a new calibration approach was developed. Magnetometer calibration and the proposed approach for trains is discussed in detail in Chapter 5, hence here the focus is on the influence of moving magnetic material that cannot be calibrated out.

One example for the influence of moving magnetic material on magnetometer measurements was investigated in [CA1]. In [CA1], we showed that when the magnetometer is mounted on the bogie of

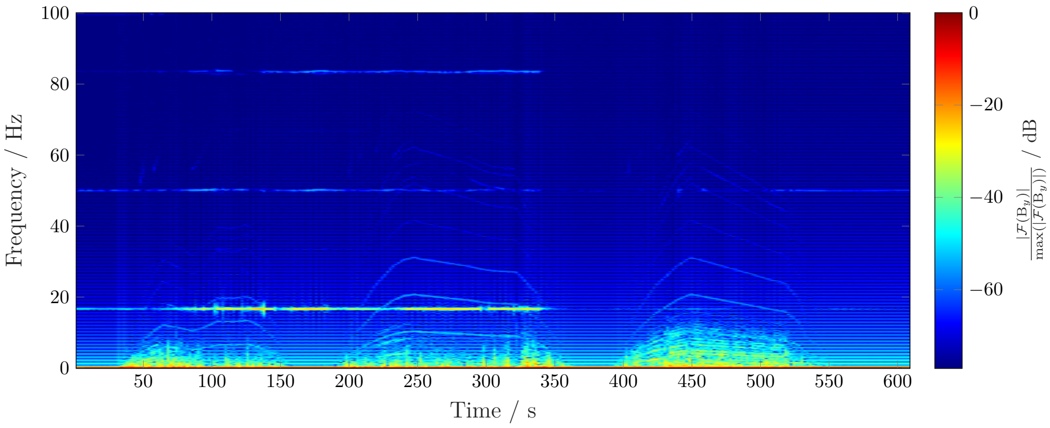

Figure 3.4: Short time Fourier transform of a data set recorded with a magnetometer mounted in the vicinity of a steel wheel. The data set was recorded simultaneously to the data set of Figure 3.2, in which the sensor was mounted inside the driver cab.

a train in the vicinity of a wheel, the wheel introduces speed dependent frequency components into the amplitude spectrum of the magnetometer measurements. This is also shown in Figure 3.4 based on a data set that was measured with the same setup as the data in [CA1]. Furthermore, the data of Figure 3.4 was measured simultaneously to the data set of Figure 3.2, so the train speed was the same for both data sets. The shape of the train's speed profile from Figure 3.2 is clearly visible in Figure 3.4. The speed dependent frequencies in the spectrum are directly related to the rotational frequency of the wheel and its harmonics. While mounting the sensor near the wheel has some benefits, e.g., in the supervised master thesis [ST1] these frequencies are used to estimate the train speed, this mounting position is not recommended for localization because everything in the measurements that is not related to the position has to be considered as a disturbance.

Another example for moving material are trains passing in the vicinity. These trains will interact with the Earth magnetic field and influence the magnetometer measurements. The influence of a passing trains depends on a couple of parameters, such as the distance in which it is passing and the type of train, e.g., in [CA2] we showed that a passing cargo train has a significantly stronger effect on the measurements when compared to a passenger train.

As was previously mentioned, disturbances due to moving material cannot be calibrated out. It is also rather difficult to model these disturbances in order to remove them from the data. In this thesis, we therefore propose in Chapter 4 particle filters that can detect disturbances and that avoid to incorporate measurements into the position estimate when disturbances are present.

3.3.2 Height and Lateral Position

The magnetic field not only changes when the train moves along the track it also changes when the magnetometer is mounted in a different height or lateral position. This implies that a magnetic map of the track network can be used for train localization only when the magnetometer of the train is mounted at the same height and lateral position as the magnetometer used to create the map. To illustrate the dependency of the magnetic field on the height and the lateral position, measurements were carried out with an eleven element magnetometer array mounted inside the advanced TrainLab of

Figure 3.5: Advanced TrainLab of the Deutsche Bahn and magnetometer array with eleven elements in vertical (right image) and lateral (lower left image) mounting position.

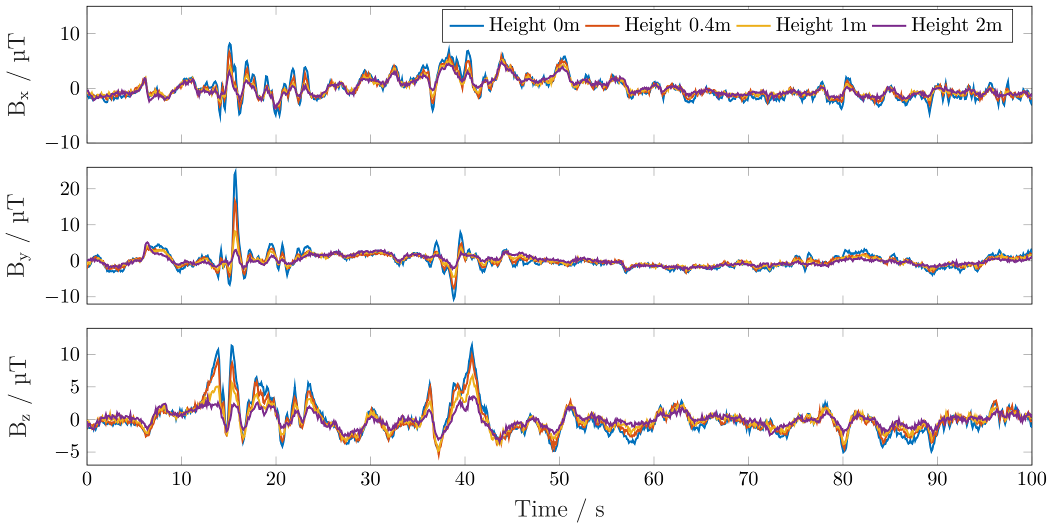

Figure 3.6: Measurements of four elements of the magnetometer array in vertical mounting position. The height is measured relative to the floor of the train.

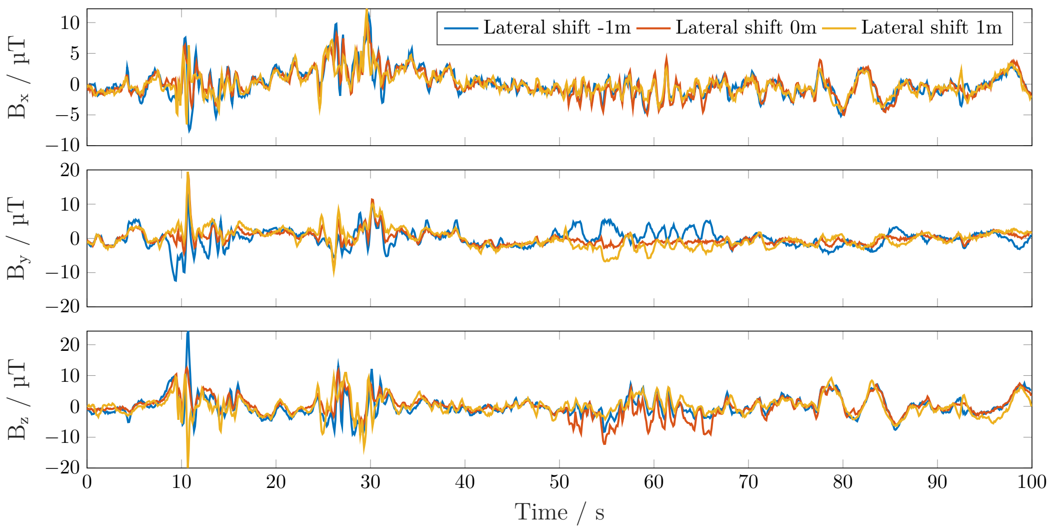

Figure 3.7: Measurements of three elements of the magnetometer array in lateral mounting position. The lateral shift is measured relative to the sensor in the middle of the train.

the Deutsche Bahn. The advanced TrainLab and the array in vertical and lateral orientation is shown in Figure 3.5. In contrast to the previous sections, KMX62 magnetometers from Kionix were used in the advanced TrainLab measurements that provide the magnetic field in µT. It should be mentioned that also this magnetometers were not calibrated due to the difficulties in the calibration of train mounted magnetometers described in Chapter 5. However, a comparison of the values measured by the different sensors should give a qualitative impression about the influence of the mounting position.

An example of how the height affects the magnetometer measurements is shown in Figure 3.6. The figure shows the measurements of four magnetometers from the vertically mounted array. One magnetometer was close to the floor of the train, and the other three are mounted at a height of ≈ 0.4 m, 1 m and 2 m above the floor. The mean was removed from each sensor axis to make the measurements better comparable and to not clutter the figure with the sensor biases and the mean magnetic field. In Figure 3.6 the general shape of the magnetic field is similar in all four data sets, but it is also noticeable that the measurements of the magnetometers get damped with an increasing mounting height. We believe this damping of the magnetic field is related to the distance of the magnetometer to the track bed and the rails. It is also interesting to note that a similar effect was observed in [J3] for an indoor environment, where the magnetic distortions became stronger the closer the sensor was to the floor or the ceiling. Based on the data in Figure 3.6, we would argue that the height dependency is something that should be accounted for by keeping the difference in the mounting height of the magnetometers during mapping and localization within a couple of centimeters.

Figure 3.7 shows data of three magnetometers from the array in lateral mounting position, shown in the lower left image of Figure 3.5. One sensor is mounted in the middle of the train and the other two are shifted ≈ 1 m to the left and to the right. As before, the mean of the data is removed to make the result better comparable. The comparison of the measured magnetic field values reveals that there are some parts with a similarity between the magnetometers but there are also parts, e.g., between 50 s and 70 s, during which strong differences are visible. In general, we would expect issues and a degraded accuracy when the lateral shift between the magnetometer during mapping and localization exceeds a couple of centimeters. Therefore, it is recommended to mount the magnetometer in the middle of the train. An alternative would be to have multiple maps that are recorded for different lateral positions.

3.4 Uniqueness

For localization it is important that the magnetic field can be uniquely distinguished for different positions in the track network. If this is not the case, ambiguities emerge in the position estimation that should be avoided. In the following we briefly analyze the magnetic field w.r.t. its capabilities to differentiate between tracks and positions on the same track.

3.4.1 Along-Track Position

A simple way to analyze the suitability of the magnetic field for along-track position estimation and its uniqueness is to look at the likelihood of the magnetometer measurements. If the magnetometer

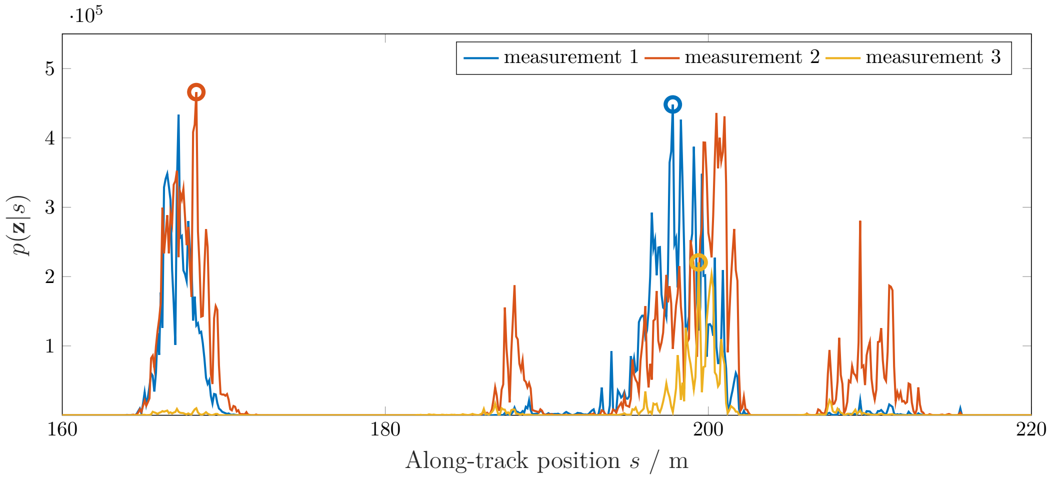

Figure 3.8: Likelihood as a function of the position for three noisy measurements simulated at along-track position 200 m. The circles mark the maxima of the likelihood with the same color.

measures the true magnetic vector field with some additive white Gaussian noise the likelihood of a measurement $\mathbf{z}$ is given by

$$\mathrm{p}(\mathbf{z}|s) = \mathcal{N}(\mathbf{z}; \mathbf{m}(s), (\sigma^{\mathrm{mag}})^2 \mathbf{I}_{3\times 3}) , \tag{3.1}$$

where s is the along-track position, σ^{mag} is the standard deviation of the magnetometer noise, and $\mathbf{m}(\cdot)$ is the magnetic map. The map is a function of the along-track position and returns for each position the magnetic field vector at that position. For a given magnetometer measurement $\mathbf{z}$, the likelihood (3.1) can be interpreted as a function of the position. For illustration, let the magnetic map be given by the magnetic vector field shown in Figure 1.1. If we now simulate three noisy measurements at the along-track position 200 m, the likelihood for each of these measurements can be calculated. The resulting likelihoods are shown in Figure 3.8 for a noise standard deviation of $\sigma^{\mathrm{mag}} = 0.005$, which is relatively small when compared to the amplitude of the distortions in Figure 1.1. Figure 3.8 shows the likelihood only between 160 m-220 m since this interval contains the maximum value for all three measurements.

All likelihoods have multiple local maxima that depend on the realization of the measurement noise. In Figure 3.8 the likelihood of the first and third measurement has its global maximum, indicated by the blue and yellow circles, close to the true position. The global maxima of the second measurement is more than 30 m away from the true position. Therefore, if the likelihood of a single measurement is used for ML estimation, the estimated position can jump considerably for different realizations of the measurement noise, resulting in a potentially low accuracy.

This simple example shows that in the presence of noise the magnetic field has ambiguities. Theoretically, a better sensor with a lower noise standard deviation could reduce the ambiguity. In practice, the improvement by reducing the magnetometer noise is limited because in addition to the sensor noise other disturbances are present in the measurements that at a certain point will dominate the estimation accuracy. There are also other approaches to tackle the ambiguities. For example, instead of processing a single measurement, sequences of measurements recorded while the train drives along the track can be processed. This effectively reduces the ambiguities in the likelihood, but has the disadvantage that the relative position, or distance, between the measurements has to be known or

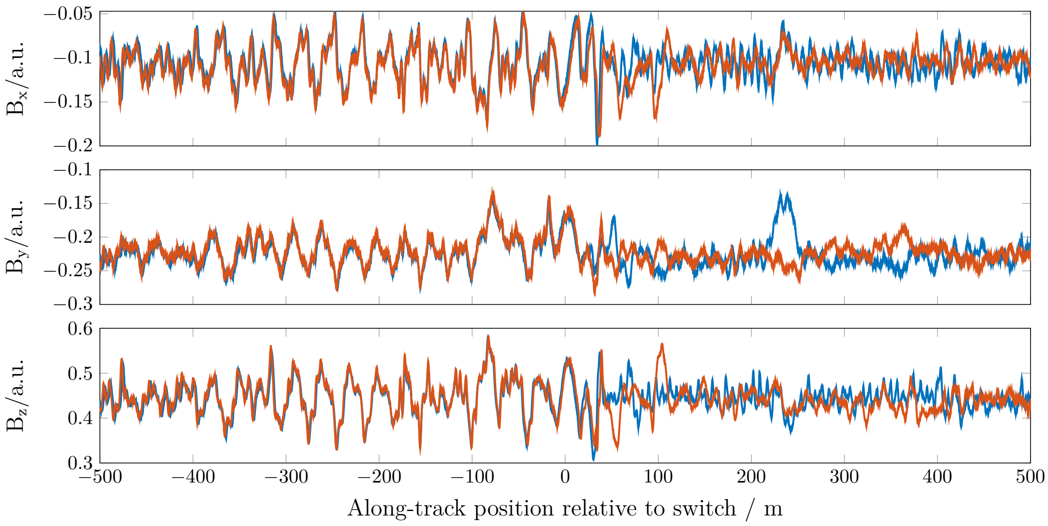

Figure 3.9: Magnetic field recorded on two parallel tracks. At the first 500 m both measurements are recorded on the same track and then the train passes a switch and the measurements are recorded on two parallel tracks.

multiple magnetometers at known distances are required. Another approach is to use Bayesian estimators that incorporate prior knowledge. A Bayesian estimator is based on the posterior density (2.1) that is proportional to the product of the likelihood and the prior. Hence, a prior that is localized around the true position effectively masks some of the local maxima in the likelihood. In this thesis, both approaches are combined in a recursive Bayesian filter that sequentially processes the magnetometer measurements and implicitly estimates the distance between consecutive measurements.

3.4.2 Track Identifier

In addition to an accurate estimation of the along-track position, it also important to be able to differentiate tracks in the track network based on their magnetic field. In this context an especially interesting case is the parallel track scenario, in which two tracks are only separated by a few meters. In Section 3.3.2 we saw the dependency of the magnetic field on the lateral mounting position of the magnetometer, this already gives an indication that the magnetic field might change significantly between parallel tracks. To confirm this presumption, Figure 3.9 depicts the magnetic field recorded during two 1 km long train runs over the relative position to a switch that splits a single track into two parallel tracks. The measurements are obtained once more from the measurement campaign described in Section 7.1.1. From −500 m to 0 m the train was driving on the same track for both measurement runs, then it passed a switch and the measurements were recorded on two parallel tracks. As expected, on the first 500 m the magnetic field of both runs shows a high similarity and repeatability. As soon the train passes the switch this is no longer the case and the magnetic field of the parallel tracks looks significantly different. In [CA2] we showed that the cross correlation between the magnetic field of parallel tracks is low and it is likely that the magnetic field can be used to differentiate between parallel tracks. The actual feasibility of track differentiation with the magnetic field was then demonstrated in [C5] on the basis of a likelihood ratio test that was used to detect the correct track after the train has passed a switch. Methods for track differentiation will be discussed in detail in Chapter 4 and concrete algorithms for it will be proposed.

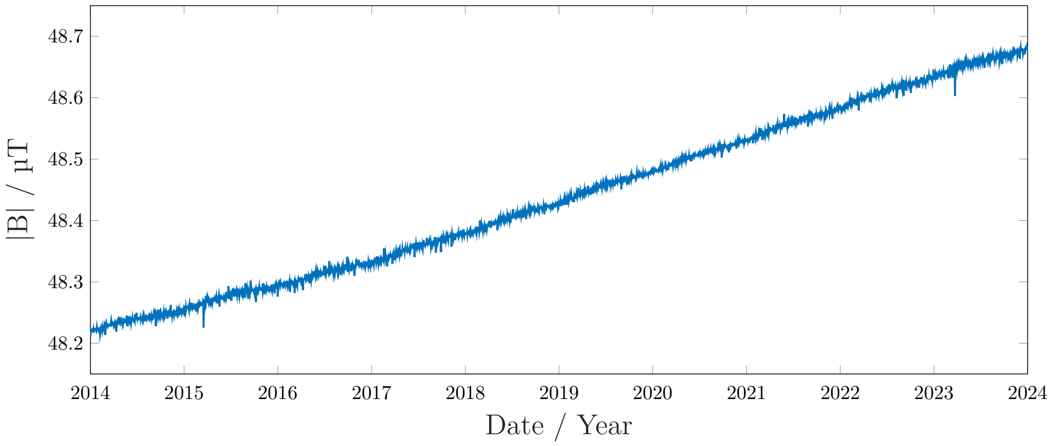

Figure 3.10: Magnitude of magnetic vector field over ten years measured in Fürstenfeldbruck, Germany.[1]

3.5 Stability of the Earth Magnetic Field

For the practical use of the magnetic field for localization its long-term stability is important. If the field would change rapidly, say at a day to day basis, also the maps have to be updated daily, which would require a lot of effort. The long-term stability depends on the stability of the Earth magnetic field and the magnetic material in the vicinity of the tracks. In Figure 3.10 the magnitude of the Earth magnetic field over a duration of ten years measured at the Erdmagnetisches Observatorium Fürstenfeldbruck in Bavaria Germany is shown. Over the ten years the overall change was in the range of 0.5 µT. If this is compared to the variability of the magnetic field, e.g., in Figure 3.6, this change is relatively small. Therefore, it is not expected that the change in the Earth magnetic field has a major impact on the long-term stability of magnetic maps in the railway domain. This is especially true when we compare the long-term stability of the Earth magnetic field with the impact changes in the railway environment have on the magnetic maps. An example for this was presented in [CA2], where the magnetic field along the same track was measured over multiple days. During nine measurement runs on the track the magnetic field was stable and showed a high similarity, but then a couple of days later the magnetic field changed noticeably. Further measurements then showed that after the change the magnetic field was stable again. In video recordings no visible changes were found hence we can only speculate what caused these changes. Possible reasons could be track maintenance or the activation of a magnetic track break [CA3].

Unfortunately, we are not yet able to make a final statement about the long-term stability, but if we neglect the rare occasions in which we encountered sudden changes, our observations indicate that the magnetic field is stable over multiple weeks. For example, we conducted two measurement campaigns on a small track network in northern Germany, as described in Section 7.1.2. The time between the two campaigns was roughly one month and within this time period the magnetic field stayed stable enough that the one month old map could be used for localization in combination with the algorithms proposed in Chapter 5. However, in the long run it should be considered to update the map regularly

[1] The results presented in this figure rely on the data collected at Erdmagnetisches Observatorium Fürstenfeldbruck (FUR). We thank FUR, for supporting its operation and INTERMAGNET for promoting high standards of magnetic observatory practice (www.intermagnet.org).

to account for sudden changes. One approach to do this without considerable manual labor could be the combination of magnetic localization with SLAM methods. A first idea how this could be done in the railway domain can be found in our paper [C9].

3.6 Summary

In this chapter an overview on the behavior of the magnetic field in the railway domain was given. The spatial variability of the magnetic field was discussed, which is an important factor for the achievable localization accuracy. Then it was shown that the magnetometer measurements are disturbed by the electric current in the overhead power lines. Fortunately, these disturbances are at known frequencies and hence can be filtered out. We also analyzed how the mounting position of the magnetometer affects the measurements. Here the conclusion was that the sensor should be mounted ideally in the middle of the train and away from moving magnetic material such as steel wheels. For magnetic localization it is also recommended to mount the magnetometer used for mapping roughly at the same position inside the train than the magnetometer used for localization. In addition, the uniqueness of the magnetic field was investigated on the basis of the likelihood function of the magnetometer measurements. The likelihood revealed ambiguities in the magnetic field, which can lead to poor performance when a ML estimator based on a single measurement is used for localization. Possible approaches to reduce the ambiguities is the use of Bayesian estimators and the processing of measurement sequences instead of a single measurement. To investigate the suitability of the magnetic field to differentiate between tracks, an example was shown based on measurements recorded during two runs. For both runs the train was first driving on the same track and after passing a switch continued on parallel tracks. The measurements revealed that even for close by parallel tracks the magnetic field shows strong differences.

The last section of this chapter was dedicated to the long-term stability of the magnetic field. The Earth magnetic field is rather stable over several years, so it is reasonable to assume that changes in the magnetic material in the railway environment are the dominant factor w.r.t. the long-term stability. In our measurements we experienced some abrupt changes in the magnetic field, but also that the magnetic field mostly stayed constant over multiple weeks. Nevertheless, in the long run it should be considered to update the magnetic maps regularly to account for changes.

Chapter 4

Magnetic Field-based Train Localization Methodology

In this chapter, filter algorithms for magnetic field-based train localization are developed. In this thesis, the focus is on particle filter-based approaches. The correlation-based approaches developed and presented in [C2] and [ST1] are not further discussed here because they dependent on reliable measurements of the train speed to convert the magnetometer measurements from the time domain into the spatial domain, which requires a train to have some sort of odometer. For older trains this is often not the case or it is not possible to get access to the data, which complicates the retrofitting of a train with a magnetic localization system. Thus, the focus is on particle filter-based approaches that only require sensors that can be mounted inside the vehicle body, such as magnetometers and inertial sensors. The developed particle filters are flexible when it comes to including additional sensors and hence, if available, an odometer can be integrated in a straightforward manner.

4.1 Preliminaries

4.1.1 Topological Position

The position of a train in a railway network is uniquely defined by the identifier I of the track on which the train is driving and the along-track position s within this track. Together the track identifier and along-track position define the topological position

$$\mathcal{P}^{\text{topo}} = \{s, I\} \,. \tag{4.1}$$

The along-track position is defined as a positive scalar in the interval $s \in [0, l_I]$, where l_I is the length of the track with identifier I. The identifier $I \in \mathcal{T}$ can take only the values from the set $\mathcal{T} \subset \mathbb{N}$ that contains one identifier for each track in the track network.

Thus, the along-track position describes the length of the track segment between the current train position and the start of the track. The start of the track is somewhat arbitrary and has to be fixed once relative to the network topology. Due to its definition, the along-track position jumps when the train passes a switch and the track identifier is changed. In the filter development, it is important to handle these jumps appropriately to avoid erroneous results. In the following, we always assume a known network topology, where the topology describes the connections between tracks and defines their start and end points.

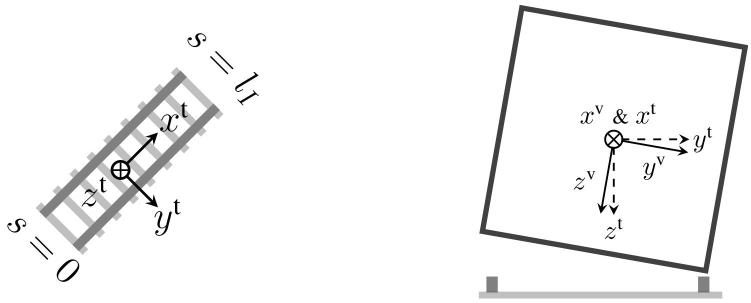

Figure 4.1: Definition of coordinate frames. (left) Definition of track frame seen from above. (right) Definition of vehicle frame in relation to the track frame when their x-axes are aligned.

4.1.2 Coordinate Frames

In this thesis different coordinate frames and their relations to each other are considered. In particular, the following frames will be used:

- **Local navigation frame:** The x- and y-axes of the local navigation frame lie in a plane tangent to the Earth surface. The x-axis points north and the y-axis east. The z-axis points downwards and is parallel to the gravity vector. The navigation frame depends on the current position since its x- and y-axes are always tangent to the Earth's surface [69].
- **Track frame:** The track frame depends on the track at the current train position. The plane spanned by its x- and y-axes is parallel to the track bed. The x-axis is parallel to the track heading and always points into the direction of an increasing along-track position. When looking into the direction of the x-axis the y-axis points to the right. To obtain a right-handed coordinate system, the z-axis points downwards into the track bed. The track frame is usually not aligned to the navigation frame. The attitude of the track relative to the navigation frame is defined by its slope, heading, and bank angles [70].
- **Vehicle frame:** The vehicle frame is fixed w.r.t. the train vehicle body. The x- and y-axes are parallel to the floor plane. The x-axis is parallel to the longitudinal axis of the train, the y-axis is parallel to the cross axis, and the z-axis is pointing downwards through the floor. For zero roll and pitch angles between the track and vehicle frame the x-axis of the vehicle frame points either in the direction of the x-axis of the track frame or in the opposite direction. The relation between the frames depends on how the train is placed on the track and at switches the relation can jump because the track frame x-axis always points into the direction of an increasing along-track position.
- **Body frame:** The body frame is attached to a sensor. For on-board sensors the body frame has an arbitrary but fixed rotation w.r.t. the vehicle frame. In the following, the body and vehicle frame are assumed to be aligned if not stated otherwise.

Some of the frames and their relation to each other are shown in Figure 4.1.

4.2 Bayesian Estimation of Topological Train Position

The goal of a train localization system is to estimate the topological position $\mathcal{P}^{\text{topo}}$ of a train based on some observations $\mathbf{z}$. In this thesis the localization follows a Bayesian approach, i.e., we want to determine the posterior density of the topological position

$$\mathrm{p}(\mathcal{P}^{\text{topo}}_{0:k}|\mathbf{z}_{1:k}) = \mathrm{p}(s_{0:k}, I_{0:k}|\mathbf{z}_{1:k}) \,. \tag{4.2}$$

Here the full posterior is estimated with particle filters. The filter density is then approximated by dropping the particle's history, for more details about this see Section 2.4. The topology of the track network and a map of the magnetic field along the tracks are assumed to be known at all times, even if they are not explicitly mentioned in the posterior.

In this chapter the observations $\mathbf{z}_{1:k}$ in (4.2) are the measurements of a vector magnetometer and the scalar measurements of an accelerometer. To indicate that the observations are obtained from multiple sensors $\mathbf{z}_{1:k}$ is replaced by the set

$$\mathcal{Z}_{1:k} = \{\mathbf{z}^{\text{mag}}_{1:k}, z^{\text{acc}}_{1:k}\} \,. \tag{4.3}$$

The vector magnetometer body frame is considered to be aligned with the vehicle frame. For a calibrated sensor, this can mean that either its physical axes are aligned with the vehicle frame or that its measurements are rotated based on the known relation between vehicle and body frame before they are handed to the filter. Similarly, the axis of the accelerometer is considered to be aligned with the x-axis of the vehicle frame. Since the along-track position and the magnetic map are defined in the track frame, the map has to be rotated into the vehicle frame to make it comparable to the measurements. Thus, the filter has to also estimate the vehicle orientation relative to the track. For orientation estimation it is assumed that the roll and pitch angles of the vehicle frame w.r.t. the track frame are zero. This is only an approximation but is roughly fulfilled in the railway domain. For the data used in [J2] the roll angle was for example below $\approx 2.5°$. As mentioned in the definition of the coordinate frames, with zero roll and pitch angles the x-axis of the vehicle frame is either aligned with the x-axis of the track frame or it points in the opposite direction. The orientation O therefore can only take two values

$$O \in \{-1, +1\} \,, \tag{4.4}$$

where $+1$ means the track and vehicle frames are aligned and -1 that their x-axes point in opposite directions. With this definition of the orientation, the rotation from the vehicle into the track frame is given by the rotation matrix

$$\mathbf{R}^{\text{t}}_{\text{v}}(O) = \begin{bmatrix} O & 0 & 0 \\ 0 & O & 0 \\ 0 & 0 & 1 \end{bmatrix} \,. \tag{4.5}$$

For this particular rotation matrix the inverse rotation $\mathbf{R}_\mathrm{t}^\mathrm{v} = (\mathbf{R}_\mathrm{v}^\mathrm{t})^\top$, from the track into the vehicle frame, is described with the same matrix. In the definition of the frames in Section 4.1.2, the body and vehicle frames are assumed to be aligned, therefore the rotation from the track frame to the body frame is given also by $\mathbf{R}_\mathrm{t}^\mathrm{b} = \mathbf{R}_\mathrm{t}^\mathrm{v}$.

For automation of railway traffic not only the train position is of interest, but also its speed $\dot{s}$ and acceleration $\ddot{s}$. Estimating the speed and acceleration has also the benefit that higher order motion models can be used that describe the dynamics of the train more accurately. Thus, the developed filters will estimate the following quantities

$$\mathcal{X}_k = \{s_k, \dot{s}_k, \ddot{s}_k, O_k, I_k\}. \tag{4.6}$$

Regarding the notation, the Bayesian filters in Chapter 2 are described w.r.t. a sequence of state space vectors $\mathbf{x}$. Since $\mathcal{X}$ contains a mixture of continuous-valued and discrete-valued random variables we combined them in a set rather than a vector, but the filter equations from Chapter 2 apply also to $\mathcal{X}$ when substituted for $\mathbf{x}$. For simplicity, we will call $\mathcal{X}$ also state. To keep the notation compact, the dynamic train state is introduced that comprises the continuous-valued elements of $\mathcal{X}$ related to the train motion

$$\mathbf{d}_k = \begin{bmatrix} s_k & \dot{s}_k & \ddot{s}_k \end{bmatrix}^\top . \tag{4.7}$$

In the following derivations of the filter algorithms for magnetic localization, two cases are differentiated. In the first case, it is assumed that the route of the train through the network is known in advance. The route is the sequence of track identifiers the train will pass over during its journey and will be denoted by

$$R_{1:N_\text{route}} = (R_1, \cdots, R_{N_\text{route}}) , \tag{4.8}$$

with $R_i \in \mathcal{T}$. This does not mean I_k is known because the route does not contain the time when the train is on which track. In the second more complicated case, the route is considered unknown and has to be retrieved from the magnetometer data jointly with the train position.

Furthermore, in both cases we assume for now that the used magnetometer is calibrated. The calibration of a train mounted magnetometer is not trivial in practice and will be discussed in detail in Chapter 5.

4.3 Along-Track Localization Filter for Known Routes

This section develops a particle filter for the easier case, where the route $R_{1:N_\text{route}}$ is considered to be known. The known route is accounted for in the posterior by conditioning the posterior on $R_{1:N_\text{route}}$

$$\mathrm{p}(\mathcal{X}_{0:k}|\mathcal{Z}_{1:k}, R_{1:N_\text{route}}) = \mathrm{p}(\mathbf{d}_{0:k}, O_{0:k}, I_{0:k}|\mathcal{Z}_{1:k}, R_{1:N_\text{route}}) . \tag{4.9}$$

The conditioning of the posterior on the route information simplifies the problem considerably. From the route information one long virtual track and magnetic map can be constructed and the filter does not have to cope with the nonlinearity when passing a switch. The track identifiers $I_{0:k}$ of the topological position are then obtained by mapping the estimated along-track position on the virtual track to the different tracks of the route. With the route information we thus can ignore the track identifier in the filter and handle it in a separate routine. Theoretically the route can contain loops, which would lead to discontinuities. For this case, an additional procedure is required to break the route into loop-free sections.

With the route information and the virtual track constructed from it, the estimation problem reduces to the estimation of the dynamic train state and the orientation, which, when combined, will be called topological pose

$$\mathcal{X}_k^{\text{do}} = \{\mathbf{d}_k, O_k\} \,. \tag{4.10}$$

With the freshly introduced topological pose $\mathcal{X}_k^{\text{do}}$ the posterior that has to be estimated is given by

$$\mathrm{p}(\mathcal{X}_{0:k}^{\text{do}}|\mathcal{Z}_{1:k}, R_{1:N_{\text{route}}}) = \frac{\mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\text{do}}, R_{1:N_{\text{route}}})\,\mathrm{p}(\mathcal{X}_k^{\text{do}}|\mathcal{X}_{k-1}^{\text{do}}, R_{1:N_{\text{route}}})\,\mathrm{p}(\mathcal{X}_{0:k-1}^{\text{do}}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})}{\mathrm{p}(\mathcal{Z}_k|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})} \,. \tag{4.11}$$

To estimate (4.11) with a particle filter, the transition density $\mathrm{p}(\mathcal{X}_k^{\text{do}}|\mathcal{X}_{k-1}^{\text{do}}, R_{1:N_{\text{route}}})$, the likelihood $\mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\text{do}}, R_{1:N_{\text{route}}})$, and a proposal density is required. The first two are given by the system model of the topological pose and the sensor measurement models. Formally, there is an optimal proposal density that minimizes the variance of the weights [71], but depending on the estimation problem the optimal density might not be tractable. However, approximations for the optimal proposal density exist, e.g., the Kalman filter-based approximations proposed in [71–73].

4.3.1 Measurement Models

The accelerometer and the magnetometer measurements are independent from each other, thus the overall measurement likelihood is given by the product of both likelihoods

$$\mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\text{do}}, R_{1:N_{\text{route}}}) = \mathrm{p}(\mathbf{z}_k^{\text{mag}}|\mathcal{X}_k^{\text{do}}, R_{1:N_{\text{route}}})\,\mathrm{p}(z_k^{\text{acc}}|\mathcal{X}_k^{\text{do}}) \,. \tag{4.12}$$

In the above equation we exploit that the accelerometer likelihood does not directly depend on the route when the orientation of the train is known. In order to fully define the overall likelihood from (4.12), the two individual measurement likelihoods will be described in the next paragraphs.

Magnetometer Model

The magnetometer measurements depend on the along-track position and the orientation of the train on the virtual track defined by the route information. To describe these dependencies, a magnetic map is required that relates the along-track position to the magnetic vector field at that position. For a known route, the magnetic map is constructed such that it contains the magnetic field in a single consistent track coordinate system. For the construction of the map therefore a coordinate system has to be defined for the virtual track and then all magnetic vectors have to be rotated accordingly. The coordinate system of the virtual track is arbitrary, a simple choice would be to use the track frame from the first track in the route and extend this until the end of the route. For a given route the map therefore can be seen as a vector valued function

$$\mathbf{m}^{\mathrm{t}}(s, R_{1:N_{\text{route}}}) : [0, l_{\text{route}}] \mapsto \mathbb{R}^3 , \tag{4.13}$$

where the route is considered to be a constant parameter and l_{route} is the length of the virtual track constructed from it. The superscript $^{\mathrm{t}}$ in (4.13) indicates that the magnetic vector returned from the map is measured in the track frame. With the map the magnetometer measurement model becomes

$$\mathbf{z}_k^{\text{mag}} = \mathbf{R}_{\mathrm{t}}^{\mathrm{b}}(O_k)\, \mathbf{m}^{\mathrm{t}}(s_k, R_{1:N_{\text{route}}}) + \mathbf{n}_k^{\text{mag}} , \tag{4.14}$$

with a white Gaussian measurement noise $\mathbf{n}_k^{\text{mag}} \sim \mathcal{N}(\mathbf{0}_{3\times1}, (\sigma^{\text{mag}})^2 \mathbf{I}_{3\times3})$. It is important to keep in mind that this simple model is only correct when the magnetometer is calibrated. For an uncalibrated magnetometer we cannot just rotate the data without introducing errors caused by the calibration parameters. Note, the measurements used in this thesis are considered to be given always in the body frame, hence for brevity the superscript ^{b} will be omitted for measurement vectors.

The likelihood of a magnetometer measurement required for the particle filter is obtained now simply from (4.14) and the sensor noise density

$$\mathrm{p}(\mathbf{z}_k^{\text{mag}} | \mathcal{X}_k^{\text{do}}, R_{1:N_{\text{route}}}) = \mathcal{N}(\mathbf{z}_k^{\text{mag}}; \mathbf{R}_{\mathrm{t}}^{\mathrm{b}}(O_k)\, \mathbf{m}^{\mathrm{t}}(s_k, R_{1:N_{\text{route}}}), (\sigma^{\text{mag}})^2 \mathbf{I}_{3\times3}) . \tag{4.15}$$

Accelerometer Model

In addition to the magnetometer measurements, the filter also processes the measurements of a scalar accelerometer. The accelerometer is considered to be aligned with the vehicle frame x-axis and thus is either parallel or antiparallel to the track frame x-axis. If the track frame is not tangent to the Earth surface and thus not aligned to the navigation frame, the accelerometer measures in addition to the train acceleration parts of the gravitational acceleration. The influence of the gravitational acceleration is ignored in the following model because for railway tracks in Germany the gradient is limited to 4 % for tracks used only for passenger trains and even less for freight trains [21]. Furthermore, the estimation of the track gradient with an accuracy noticeably higher than the actual value of the gradient is not an easy task that requires high quality gyroscopes, an accurate initial alignment, and regular information from additional aiding sensors to bound the error. Since we want to keep the hardware required for the magnetic localization system simple, the accelerometer model is reduced to

$$z_k^{\text{acc}} = O_k \ddot{s}_k + b^{\text{acc}} + n_k^{\text{acc}} , \tag{4.16}$$

where b^{acc} is the sensor bias and n_k^{acc} is white Gaussian noise with density $\mathcal{N}(0, (\sigma^{\text{acc}})^2)$. The accelerometer bias is considered to be a constant value that is set once during the standstill phase on the beginning of each train run. During the standstill phase the bias is obtained by averaging the accelerometer measurements. The severe simplifications in the model limit the usefulness of the acceleration measurements, but the model is good enough to provide some information on whether the train is accelerating or decelerating. This is particularly helpful when the magnetic field does not provide much information at certain parts of the track.

As for the magnetometer likelihood, the accelerometer likelihood is easily found from the measurement model (4.16) and the corresponding noise density

$$\mathrm{p}(z_k^{\mathrm{acc}}|\mathcal{X}_k^{\mathrm{do}}) = \mathcal{N}(z_k^{\mathrm{acc}}; O_k \ddot{s}_k + b^{\mathrm{acc}}, (\sigma^{\mathrm{acc}})^2) \; . \tag{4.17}$$

4.3.2 System Models

The prediction step of the particle filter requires the transition densities for the different parts of the state. The transition densities describe how the state evolves from one time step to the next. For the topological pose in the posterior (4.11) the transition is described separately for the dynamic train state and the orientation.

Dynamic Train State

The along-track position is limited to the interval $s \in [0, l_{\mathrm{route}}]$ where l_{route} is again the length of the virtual track constructed from the route information. When the nonlinearity at the interval, or route, endpoints is ignored, the dynamic train state $\mathbf{d}_k$ can be interpreted as the state vector of a linear state space model of the form

$$\mathbf{d}_k = \underbrace{\begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{F}^{\mathrm{d}}} \mathbf{d}_{k-1} + \mathbf{w}_{k-1}^{\mathrm{d}} \; , \tag{4.18}$$

where T is the sample period and the process noise $\mathbf{w}_{k-1}^{\mathrm{d}}$ is Gaussian with pdf $\mathcal{N}(\mathbf{0}_{3\times 1}, \mathbf{Q}^{\mathrm{d}})$ and covariance matrix

$$\mathbf{Q}^{\mathrm{d}} = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & \frac{T^2}{2} \\ \frac{T^3}{2} & T^2 & T \\ \frac{T^2}{2} & T & 1 \end{bmatrix} (\sigma^{\mathrm{d}})^2 \; . \tag{4.19}$$

The standard deviation σ^{d} is a tuning parameter that is roughly orientated on the acceleration capability of the train. In the literature this model is also referred to as discrete Wiener process acceleration model [74]. It should be noted that different approaches exist to model the covariance (4.19) depending on assumptions made about how the acceleration changes between two time steps. From a practical point view, we do not expect that these assumptions have a large impact on the overall performance particularly because we treat σ^{d} as a mere tuning parameter.

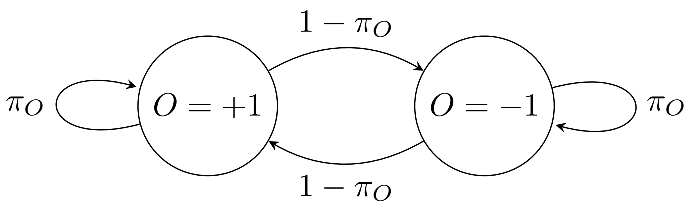

Figure 4.2: Markov chain that describes the transition between the two different values of the orientation.

If the nonlinearity at the interval endpoints is ignored, the transition density is now directly obtained from (4.18) and the noise density

$$\mathrm{p}(\mathbf{d}_k|\mathbf{d}_{k-1}, R_{1:N_{\text{route}}}) = \mathcal{N}(\mathbf{F}^{\mathrm{d}}\mathbf{d}_{k-1}, \mathbf{Q}^{\mathrm{d}}) \ . \tag{4.20}$$

If the nonlinearity at endpoints is not ignored, there is no closed form solution for the transition density of the dynamic train state. In the filter we cope with this issue by setting the proposal density to the transition density, which essentially removes the need to evaluate the transition density. This means only samples from the transition density are required which can be realized by performing a two-step prediction. First a sample is generated from the Gaussian (4.20) and second, if the sample is outside the interval it has to be restricted to the interval. There is no intuitive or correct way to do this restriction. Here we decided to just set a sample that is outside the interval to the closest interval endpoint. It should be mentioned, that the nonlinearity is only a problem when the train is close the interval borders because the Gaussian (4.20) is rather localized and drawing a sample far away from the predicted dynamic train state $\mathbf{F}^{\mathrm{d}}\mathbf{d}_{k-1}$ is unlikely.

Orientation

The orientation is a discrete valued random variable, thus the dynamic behavior can be described by a Markov chain [75]. Since the orientation has only two possible states $\{-1, +1\}$, the Markov chain is rather simple as shown in Figure 4.2. From Figure 4.2 the four different values of the probability mass function (pmf) $\mathrm{pr}(O_k|O_{k-1})$ can be read directly. As mentioned before, only routes without loops are considered in the filter. Without loops the orientation is constant and can never switch from one value to another. Nevertheless, it might be beneficial in practice to allow with a small probability a change between the values because otherwise it can happen that shortly after initialization only particles with the wrong orientation are left. The exact value for π_O is a tuning parameter that will be further described in the evaluation in Chapter 7.

4.3.3 Filter Algorithm

With Algorithm 2.2, the definition of the transition density and likelihoods, and by setting the proposal to the transition density

$$\pi(\mathcal{X}_k^{\mathrm{do}}|\mathcal{X}_{0:k-1}^{\mathrm{do}}, \mathcal{Z}_{1:k}) = \mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{X}_{k-1}^{\mathrm{do}}, R_{1:N_{\text{route}}}), \tag{4.21}$$

the particle filter algorithm for magnetic field-based along-track localization for known routes can be formalized. With Algorithm 2.2 and (4.21) the weight update equation is given by

$$\tilde{w}_k^{(i)} = \frac{\mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do},(i)}, R_{1:N_{\mathrm{route}}})\,\mathrm{p}(\mathcal{X}_k^{\mathrm{do},(i)}|\mathcal{X}_{k-1}^{\mathrm{do},(i)}, R_{1:N_{\mathrm{route}}})}{\pi(\mathcal{X}_k^{\mathrm{do},(i)}|\mathcal{X}_{0:k-1}^{\mathrm{do},(i)}, \mathcal{Z}_{1:k})} w_{k-1}^{(i)} = \mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do},(i)}, R_{1:N_{\mathrm{route}}}) w_{k-1}^{(i)} \,. \tag{4.22}$$

Setting the proposal density to the transition density is a frequent choice although it is not optimal because the proposal does not account for the information of the newest measurements. But it also has the advantage that no evaluation of the transition density is required. This is particular helpful when the transition density has no closed form expression as it the case here when a particle is close to the start or end of the virtual track.

For the proposed filter the new unnormalized weights are obtained from the multiplication of the old normalized weights with the product of the magnetometer and the accelerometer likelihoods

$$\tilde{w}_k^{(i)} = \mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{\mathrm{do},(i)}, R_{1:N_{\mathrm{route}}})\,\mathrm{p}(z_k^{\mathrm{acc}}|\mathcal{X}_k^{\mathrm{do},(i)}) w_{k-1}^{(i)} \,. \tag{4.23}$$

In the prediction step of the filter, the predicted dynamic train state of each particle is sampled from (4.20) and then constrained to the interval $[0, l_{\mathrm{route}}]$. The orientation of each particle is predicted by simulating the Markov chain from Figure 4.2, which means the orientation state is changed to the other value with probability $1-\pi_O$. The different computing steps of one iteration of the along-track particle filter for known routes is summarized in Algorithm 4.1. The filter presented in this section is mostly based on our paper [C3]. In contrast to Algorithm 4.1, the filter in [C3] did not use the accelerometer in the weight update and the orientation was not modeled with a Markov chain. Furthermore, the filter in [C3] used a simpler motion model that does not have an acceleration state. The mentioned modifications were done here to harmonize the system and measurement models across all filters presented in this thesis.

4.3.4 Calculation of Point Estimates

For practical applications often point estimates have to be calculated from the posterior. The calculation of point estimates for the along-track filter with a known route is rather straightforward since the along-track position is a continuous value in the interval $[0, l_{\mathrm{route}}]$.

A typical point estimate is the minimum mean squared error estimate (MMSEE), which minimizes the quadratic loss [52]. The MMSEE is the expected value of the desired quantity w.r.t. the posterior density. Hence, for the along-track position MMSEE we can write

$$\check{s}_k = \int s_k\,\mathrm{p}(s_k|\mathcal{Z}_{1:k}, R_{1:N_{\mathrm{route}}}) ds_k \approx \int s_k \sum_{i=1}^{N_{\mathrm{p}}} w_k^{(i)} \delta(s_k - s_k^{(i)}) ds_k = \sum_{i=1}^{N_{\mathrm{p}}} w_k^{(i)} s_k^{(i)} \,, \tag{4.24}$$

where $\check{}$ indicates the MMSEE of a quantity. The Dirac mixture approximation of the posterior reduces the MMSEE to the weighted mean over the along-track positions of the particles. To output the topological position, we can now check on which section of the virtual track $\check{s}_k$ is located and then

Algorithm 4.1 Particle filter for magnetic field-based along-track position estimation

Input: Particle set $\{\mathcal{X}_{k-1}^{\mathrm{do},(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_\mathrm{p}}$, observations $\mathcal{Z}_k$ and route information $R_{1:N_\mathrm{route}}$
Output: Updated set $\{\mathcal{X}_{k}^{\mathrm{do},(i)}, w_{k}^{(i)}\}_{i=1}^{N_\mathrm{p}}$
for all $i \in \{1, \dots, N_\mathrm{p}\}$ **do**
    $\mathbf{d}_k^{(i)} \sim \mathcal{N}(\mathbf{F}^\mathrm{d}\mathbf{d}_{k-1}^{(i)}, \mathbf{Q}^\mathrm{d})$
    if $s_k^{(i)} \notin [0, l_\mathrm{route}]$ **then**
        Set $s_k^{(i)}$ to closest endpoint of interval $[0, l_\mathrm{route}]$
    end if
    $O_k^{(i)} = O_{k-1}^{(i)}$
    if $u^{(i)} > \pi_O$ with $u^{(i)} \sim \mathcal{U}(0,1)$ **then**
        Set $O_k^{(i)}$ to opposite value of $O_{k-1}^{(i)}$
    end if
    $\mathcal{X}_k^{\mathrm{do},(i)} = \{\mathbf{d}_k^{(i)}, O_k^{(i)}\}$
    $\tilde{w}_k^{(i)} = \mathrm{p}(\mathbf{z}_k^\mathrm{mag}|\mathcal{X}_k^{\mathrm{do},(i)}, R_{1:N_\mathrm{route}})\, \mathrm{p}(z_k^\mathrm{acc}|\mathcal{X}_k^{\mathrm{do},(i)}) w_{k-1}^{(i)}$
end for
for all $i \in \{1, \dots, N_\mathrm{p}\}$ **do**
    $w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^{N_\mathrm{p}} \tilde{w}_k^{(j)}$
end for
if Resampling necessary **then**
    Draw N_p particles from $\{\mathcal{X}_{k}^{\mathrm{do},(i)}, w_{k}^{(i)}\}_{i=1}^{N_\mathrm{p}}$
    Set all weights to $w_k^{(i)} = \frac{1}{N_\mathrm{p}}$
end if
return

derive the corresponding identifier $\check{I}_k$ and along-track position on that track from it. The MMSEE of the speed and the acceleration is obtained by replacing the along-track position s in (4.24) with $\dot{s}$ or $\ddot{s}$. After calculating the speed and acceleration estimates, their sign has to be adopted depending on the track-frame of the track $\check{I}_k$.

In addition to the MMSEE, often a measure of uncertainty is desired. For that we can use the standard deviation of the posterior. The standard deviation is obtained by taking the square root of the along-track position variance obtained from the integral

$$\sigma_{s_k}^2 \approx \int (s_k - \check{s}_k)^2 \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \delta(s_k - s_k^{(i)}) ds_k = \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} (s_k^{(i)} - \check{s}_k)^2 \,. \tag{4.25}$$

For the train speed and acceleration, the standard deviation is obtained by replacing s_k and $\check{s}_k$ with the corresponding values. When replacing $\check{s}_k$ with the speed or acceleration MMSEE, it is important to first calculate the standard deviation before the sign is adapted to the track-frame of track $\check{I}_k$.

Table 4.1: Overview on the different model assumptions. For each model the undisturbed sensor axes are indicated by a "-"and disturbed axes by a "+".

	Model							
Sensor axis	$\mathcal{M}^{(1)}$	$\mathcal{M}^{(2)}$	$\mathcal{M}^{(3)}$	$\mathcal{M}^{(4)}$	$\mathcal{M}^{(5)}$	$\mathcal{M}^{(6)}$	$\mathcal{M}^{(7)}$	$\mathcal{M}^{(8)}$
x	-	-	-	+	+	-	+	+
y	-	-	+	-	+	+	-	+
z	-	+	-	-	-	+	+	+

4.4 Robust Along-Track Localization Filter

In Chapter 3 different types of disturbances are shown that can affect the magnetometer measurements. These disturbances are not accounted for in the along-track particle filter Algorithm 4.1 since the likelihood (4.15) only assumes AWGN. One can imagine that disturbances that introduce correlated errors into the magnetometer measurements will severely impair the weight update and the filter performance, and might even lead to a divergence of the particle set. An example of such disturbances is passing trains that can have a severe effect on the magnetometer measurements depending on the train type as shown in [CA2]. Therefore, in this thesis variants of the particle filter are investigated that are robust against such disturbances. In particular, two different filter algorithms are implemented and compared. The first filter was developed during this thesis and was published in [C4]. The second filter is based on [50] and was adapted for magnetic train localization in [C5].

4.4.1 Particle Filter with Fault Detection and Exclusion

The principle idea of the first robust filter is to tightly integrate a fault detection and exclusion (FDE) scheme into the filter algorithm. A fault here can mean that the magnetometer measurements are affected by magnetic disturbances or that there is an actual hardware fault in the sensor. The proposed fault detection scheme checks if the magnetometer measurements fit to the expected magnetic field stored in the map. If they deviate from the map, the measurements are excluded from the weight update.

The FDE scheme uses multiple magnetometer measurement models and calculates for each model the probability that it is correct given the magnetic map and the magnetometer measurements. Each of the models assumes that either none, one, two, or all three axes of a vector magnetometer are affected by a disturbance or a hardware fault. Based on the model probabilities, it is then decided which sensor axes contribute to the weight update. In total, the probabilities of eight different models are calculated. The models are collected in the set $\mathcal{M}$ and the individual models are addressed by $\mathcal{M}^{(i)}$ with $i \in \{1, \ldots, 8\}$. The different models and their assumptions about the magnetometer sensor axes are defined in Table 4.1, where the sensor axes that are assumed to be disturbed are marked with a "+" sign.

Calculation of Model Probabilities

For FDE the probability that a model is correct at a given time step must be calculated. For the j-th model and time step k this probability is given by

$$\begin{aligned}\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_{\text{route}}}) &= \frac{\mathrm{p}(\mathcal{Z}_k|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})\,\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})}{\mathrm{p}(\mathcal{Z}_k|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})} \\ &\propto \underbrace{\mathrm{p}(\mathcal{Z}_k|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})}_{=\lambda_k^{(j)} \text{ likelihood of model } j}\,\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})\,. \end{aligned} \tag{4.26}$$

As shown in Appendix A.1.1, equation (4.26) can easily be obtained from the rules for conditional probability distributions. The denominator $\mathrm{p}(\mathcal{Z}_k|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})$ is just a normalization constant. However, the normalization constant does not have to be explicitly calculated because normalization is also achieved by making sure the sum over all model probabilities is one. In (4.26) the factor $\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})$ is the predicted probability that the j-th model is correct at time step k. The predicted probability is obtained from the law of total probability with which the update of the model probabilities can be formulated in recursive form

$$\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}}) = \sum_{i=1}^{8} \mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{M}_{k-1}^{(i)})\,\mathrm{pr}(\mathcal{M}_{k-1}^{(i)}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})\,. \tag{4.27}$$

In (4.27) we assumed that the transition probabilities are conditionally independent from the observations and the route

$$\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{M}_{k-1}^{(i)}) = \mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{M}_{k-1}^{(i)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})\,. \tag{4.28}$$

The transition between different models is further assumed to be governed by a homogeneous Markov chain with the time independent transition probabilities $p^{(j,i)} = \mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{M}_{k-1}^{(i)})$. With (4.26) and (4.27) the normalized model probabilities become

$$\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_{\text{route}}}) = \frac{\lambda_k^{(j)} \sum_{i=1}^{8} p^{(j,i)}\,\mathrm{pr}(\mathcal{M}_{k-1}^{(i)}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})}{\sum_{j=1}^{8} \lambda_k^{(j)} \sum_{i=1}^{8} p^{(j,i)}\,\mathrm{pr}(\mathcal{M}_{k-1}^{(i)}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})}. \tag{4.29}$$

The crucial part of the fault detection is how the likelihoods $\lambda_k^{(j)}$ are chosen for each model. However, before we can choose the likelihoods we have to make sure they can be evaluated. In order to evaluate $\lambda_k^{(j)}$, the topological pose $\mathcal{X}_k^{\text{do}}$ has to be introduced into the likelihood by marginalization which yields

$$\begin{aligned}\lambda_k^{(j)} = \mathrm{p}(\mathcal{Z}_k|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}}) &= \int \mathrm{p}(\mathcal{Z}_k, \mathcal{X}_k^{\text{do}}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}}) d\mathcal{X}_k^{\text{do}} \\ &= \int \mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\text{do}}, \mathcal{M}_k^{(j)}, R_{1:N_{\text{route}}})\,\mathrm{p}(\mathcal{X}_k^{\text{do}}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}}) d\mathcal{X}_k^{\text{do}}\,. \end{aligned} \tag{4.30}$$

In the last line of (4.30) we assumed that the underlying process for the different models is Markovian and the likelihood $p(\mathcal{Z}_k|\mathcal{X}_k^{\text{do}}, \mathcal{M}_k^{(j)}, R_{1:N_{\text{route}}})$ is independent from previous measurements when conditioned on the state and the model. The second factor of the integrand in (4.30) is the predicted posterior conditioned on model j. Unfortunately, the calculation of the predicted posterior $\mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\mathrm{route}}})$ poses a problem in practice. The predicted posterior requires the calculation of the posterior for each possible sequence of models that ends up in the j-th model at time step k. The desired posterior is then obtained by combining the posterior densities for the different model sequences by weighting them with the probability that the corresponding sequence is correct, a more detailed discussion regarding this topic can be found in the appendix Section A.1.2. As a consequence, a bank of particle filters with one filter per sequence is required. Over time the number of filters in the bank grows exponentially, which renders the calculation unfeasible from a practical point of view due to memory and complexity limitations. In addition, it is likely that filters conditioned on the wrong model sequence perform rather poorly because they include disturbed measurements or exclude undisturbed measurements. In [C4] we thus proposed to only run one particle filter, which updates its weights based on the most probable model. For the filter update step this means that depending on the current model probabilities some of the measurements might not be considered in the new weights. For example, when from Table 4.1 model $\mathcal{M}^{(2)}$ has the highest probability the z-axis measurement is excluded and for model $\mathcal{M}^{(8)}$ no magnetometer update is performed. Since only one filter is used, we drop the model dependency in the predicted posterior and insert the Dirac mixture density of the filter into likelihood (4.30) to obtain the approximation

$$\begin{aligned}
\lambda_k^{(j)} &\approx \int \mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do}}, \mathcal{M}_k^{(j)}, R_{1:N_{\mathrm{route}}})\, \mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{Z}_{1:k-1}, R_{1:N_{\mathrm{route}}}) d\mathcal{X}_k^{\mathrm{do}} \\
&\approx \sum_{i=1}^{N_{\mathrm{p}}} w_{k-1}^{(i)}\, \mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do},(i)}, \mathcal{M}_k^{(j)}, R_{1:N_{\mathrm{route}}}) \;, \qquad (4.31)
\end{aligned}$$

where $w_{k-1}^{(i)}$ are the weights from the previous time step and $\mathcal{X}_k^{\mathrm{do},(i)}$ are the predicted particles at time step k. Thus, with the particle approximation the likelihood $\lambda_k^{(j)}$ becomes the weighted mean of the likelihoods $\mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do},(i)}, \mathcal{M}_k^{(j)}, R_{1:N_{\mathrm{route}}})$ evaluated at the predicted particles. Note, since we sample the predicted particles from the transition density the weights of the predicted posterior are simply the weights from the previous time step.

The likelihood $\mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do},(i)}, \mathcal{M}_k^{(j)}, R_{1:N_{\mathrm{route}}})$ for the i-th particle and model j is given by the product

$$\mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do},(i)}, \mathcal{M}_k^{(j)}, R_{1:N_{\mathrm{route}}}) = \mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{\mathrm{do},(i)}, \mathcal{M}_k^{(j)}, R_{1:N_{\mathrm{route}}})\, \mathrm{p}(z_k^{\mathrm{acc}}|\mathcal{X}_k^{\mathrm{do},(i)}) \;. \qquad (4.32)$$

In the likelihood the magnetometer part depends on the model and therefore differs from model to model. For the accelerometer likelihood this is not the case and the same pdf given by (4.17) is used for all models.

Model Likelihoods

The likelihood function $\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{\mathrm{do}}, \mathcal{M}_k^{(j)}, R_{1:N_{\mathrm{route}}})$ for the different models plays an essential role in the FDE. For the nominal model $\mathcal{M}^{(1)}$, in which all three sensor axes are not affected by disturbances, the likelihood is given by (4.15) from the along-track localization filter Algorithm 4.1. In (4.15) the three magnetometer axes are independent and the likelihood of the vector measurement is the product of the likelihoods of the individual axes

$$\begin{aligned}\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{\mathrm{do}},\mathcal{M}_k^{(1)},R_{1:N_{\mathrm{route}}}) &= \prod_{l=1}^{3}\mathcal{N}([\mathbf{z}_k^{\mathrm{mag}}]_l;[\mathbf{R}_{\mathrm{t}}^{\mathrm{b}}(O_k)\,\mathbf{m}^{\mathrm{t}}(s_k,R_{1:N_{\mathrm{route}}})]_l,(\sigma^{\mathrm{mag}})^2)\\ &= \prod_{l=1}^{3}\mathrm{p}^{\mathrm{nom}}([\mathbf{z}_k^{\mathrm{mag}}]_l|\mathcal{X}_k^{\mathrm{do}};l)\ ,\end{aligned} \tag{4.33}$$

where $[\cdot]_l$ stands for the l-th element of the enclosed vector and the semicolon in $\mathrm{p}^{\mathrm{nom}}(\cdot)$ means that l is considered a parameter of $\mathrm{p}^{\mathrm{nom}}(\cdot)$. In (4.33) it is assumed that for each magnetometer axis the measurement is a sample of a Gaussian density with the mean given by the rotated magnetic map. A measurement that is close to the rotated map thus has a high likelihood and a measurement that is far away from the map has a low likelihood. But how do we describe the likelihood of a measurement from a disturbed sensor axis? As discussed in Chapter 3, there are several sources of disturbance in the railway domain and it is neither clear nor trivial how to model these disturbances. It is therefore difficult to find an accurate likelihood for the disturbed magnetometer axes. However, in [C4] we proposed a simple likelihood suitable for fault detection. The proposed likelihood is the result of a simple thought: when the magnetometer axis is disturbed, the measured value is no longer localized around the mean given by the magnetic map. Based on this thought, the likelihood of a disturbed axis is designed to have an opposite behavior than the nominal likelihood, i.e., the likelihood of a disturbed axis increases the further away the measurement is from the map. In this thesis, the likelihood from [C4] that fulfills this property is used once more

$$\mathrm{p}^{\mathrm{dist}}([\mathbf{z}_k^{\mathrm{mag}}]_l|\mathcal{X}_k^{\mathrm{do}};l) = c^{\mathrm{dist}}\left[1-\exp\left(-\frac{1}{2(\sigma^{\mathrm{dist}})^2}([\mathbf{z}_k^{\mathrm{mag}}]_l-[\mathbf{R}_{\mathrm{t}}^{\mathrm{b}}(O_k)\,\mathbf{m}^{\mathrm{t}}(s_k,R_{1:N_{\mathrm{route}}})]_l)^2\right)\right]\ . \tag{4.34}$$

The function in (4.34) is not a proper pdf because it does not integrate to one. For the calculation of the model probabilities this does not pose a problem because the normalization in (4.29) ensures that the model probabilities always sum to one. When the support of $\mathrm{p}^{\mathrm{dist}}(\cdot)$ would be limited, a normalization constant could be calculated such that the function becomes a proper pdf, but we do not see an advantage in imposing this kind of restriction. In the end the goal is to have freedom in choosing the parameters of $\mathrm{p}^{\mathrm{dist}}(\cdot)$ to improve the performance of the fault detection. In Section 7.2.4 we will give some insight how the parameters can be selected and a concrete example for $\mathrm{p}^{\mathrm{dist}}(\cdot)$ is shown in Figure 7.5.

With (4.33) and (4.34) it is now straightforward to construct the likelihood of the magnetometer measurements for the different models

$$\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{\mathrm{do}},\mathcal{M}_k^{(j)},R_{1:N_{\mathrm{route}}}) = \prod_{l\in\mathcal{I}^{(j)}}\mathrm{p}^{\mathrm{nom}}([\mathbf{z}_k^{\mathrm{mag}}]_l|\mathcal{X}_k^{\mathrm{do}};l)\prod_{l\in\{1,2,3\}\setminus\mathcal{I}^{(j)}}\mathrm{p}^{\mathrm{dist}}([\mathbf{z}_k^{\mathrm{mag}}]_l|\mathcal{X}_k^{\mathrm{do}};l)\ , \tag{4.35}$$

where $\mathcal{I}^{(j)}$ is a set that contains the indices of the magnetometer axes that according to model j are not affected by disturbances. With (4.29), (4.31)–(4.32), and (4.35) we now have all ingredients to formulate the FDE particle filter algorithm.

Algorithm 4.2 Fault detection and exclusion particle filter for along-track localization

Input:$\{\mathcal{X}_{k-1}^{\text{do},(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_\text{p}}$, $\mathcal{Z}_k$, $R_{1:N_\text{route}}$ and $\{\text{pr}(\mathcal{M}_{k-1}^{(j)}|\mathcal{Z}_{1:k-1}, R_{1:N_\text{route}})\}_{j=1}^{8}$
Output: Updated sets $\{\mathcal{X}_{k}^{\text{do},(i)}, w_{k}^{(i)}\}_{i=1}^{N_\text{p}}$ and $\{\text{pr}(\mathcal{M}_{k}^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_\text{route}})\}_{j=1}^{8}$
Predict $\mathcal{X}_{k}^{\text{do},(i)}$ for all particles as in Algorithm 4.1 line 4–12
Calculate model likelihoods $\lambda_k^{(j)}$ with (4.31)
Updated model probabilities $\text{pr}(\mathcal{M}_{k}^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_\text{route}})$ with (4.29)
Update weights based on the most probable model according to (4.36). If all axes are disturbed update only with the accelerometer likelihood.
if Resampling necessary **then**
  Draw N_p particles from $\{\mathcal{X}_{k}^{\text{do},(i)}, w_{k}^{(i)}\}_{i=1}^{N_\text{p}}$
  Set all weights to $w_k^{(i)} = \frac{1}{N_\text{p}}$
end if
return

Filter Algorithm

In Algorithm 4.2 the FDE particle filter is described. The filter is in essence a normal particle filter with a measurement model that is adapted depending on the model probabilities. A filter iteration starts with the prediction of the topological pose $\mathcal{X}^\text{do}$ of all particles with the system model. With the predicted particles then the likelihoods $\lambda_k^{(j)}$ for all models are calculated and the model probabilities are updated according to (4.29). In the update step of the filter, the model probabilities are used to decide which magnetometer axes are included and excluded from the weight update. Thus, the weight update becomes

$$\tilde{w}_k^{(i)} = \text{p}(z_k^\text{acc}|\mathcal{X}_k^{\text{do},(i)}) \left(\prod_{l \in \mathcal{I}^{(j^\text{max})}} \text{p}^\text{nom}([\mathbf{z}_k^\text{mag}]_l | \mathcal{X}_k^{\text{do},(i)}; l) \right) w_{k-1}^{(i)} \tag{4.36}$$

with

$$j^\text{max} = \arg\max_j \text{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_\text{route}}) \,. \tag{4.37}$$

When $j = 8$ is the most probable model, all axes are considered disturbed. In this case, the weights are updated only with the accelerometer likelihood. Similar to the generic particle filter in Algorithm 2.2, after the weight update resampling is performed based on the effective number of samples and a predefined threshold.

4.4.2 Particle Filter with Multiple Measurement Noise Models

In this section a second variant of a robust particle filter algorithm adopted from [50] is presented. Similar to the FDE filter from the previous section, the filter from [50] uses multiple measurement models to cope with outliers and disturbed measurements, but instead of excluding measurements from the filter update, the filter softly reduces the influence of outliers on the updated weights. The filter achieves this by averaging over multiple measurement models, where each model represents a

Algorithm 4.3 Noise model averaging particle filter for along-track localization

Input:$\{\mathcal{X}_{k-1}^{\mathrm{do},(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_\mathrm{p}}$, $\mathcal{Z}_k$, $R_{1:N_\mathrm{route}}$ and $\{\mathrm{pr}(\mathcal{M}_{k-1}^{(j)}|\mathcal{Z}_{1:k-1}, R_{1:N_\mathrm{route}})\}_{j=1}^{N_\mathcal{M}}$
Output: Updated sets $\{\mathcal{X}_{k}^{\mathrm{do},(i)}, w_{k}^{(i)}\}_{i=1}^{N_\mathrm{p}}$ and $\{\mathrm{pr}(\mathcal{M}_{k}^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_\mathrm{route}})\}_{j=1}^{N_\mathcal{M}}$
Predict $\mathcal{X}_{k}^{\mathrm{do},(i)}$ for all particles as in Algorithm 4.1 line 4–12
Calculate model likelihoods $\lambda_k^{(j)}$ with (4.31)
Updated model probabilities $\mathrm{pr}(\mathcal{M}_{k}^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_\mathrm{route}})$ with (4.29)
Update and normalize weights $w_k^{(i,j)}$ for each model with (4.40)
Calculate the weights $w_k^{(i)}$ of the averaged posterior (4.42)
if Resampling necessary **then**
    Draw N_p particles from $\{\mathcal{X}_{k}^{\mathrm{do},(i)}, w_{k}^{(i)}\}_{i=1}^{N_\mathrm{p}}$
    Set all weights to $w_k^{(i)} = \frac{1}{N_\mathrm{p}}$
end if
return

different distribution of the measurement noise. Hence, in the following we will call the filter the noise model averaging (NMA) filter. In [C5] the NMA filter was adapted for magnetic field-based along-track localization. For magnetic along-track localization the posterior estimated by the NMA filter is given by

$$\begin{aligned}\mathrm{p}(\mathcal{X}_{0:k}^{\mathrm{do}}|\mathcal{Z}_{1:k}, R_{1:N_\mathrm{route}}) &= \sum_{j=1}^{N_\mathcal{M}} \mathrm{p}(\mathcal{X}_{0:k}^{\mathrm{do}}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k}, R_{1:N_\mathrm{route}})\,\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_\mathrm{route}}) \\ &\approx \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \delta(\mathcal{X}_{0:k}^{\mathrm{do}} - \mathcal{X}_{0:k}^{\mathrm{do},(i)}) \, . \end{aligned} \tag{4.38}$$

The posterior is the weighted average of $N_\mathcal{M}$ posteriors conditioned on the different models. The calculation of the model probabilities is identical to (4.29) and requires the likelihoods $\lambda_k^{(j)}$ for the different models.

The model specific likelihoods are calculated once more based on (4.31), which theoretically requires a predicted posterior for each model. The NMA filter therefore suffers from the same problem as the FDE filter and the correct calculation of the likelihoods requires an exponentially growing bank of filters that reflects all possible model sequences up to the current time step. As for the FDE filter, we make a simplification and use just one posterior. In the FDE filter, the predicted posterior was obtained from the posterior that is updated only with measurements that are considered undisturbed by the most probable model. For the NMA filter this is different and the predicted posterior $p(\mathcal{X}_k^{\mathrm{do}}|\mathcal{Z}_{1:k-1}, R_{1:N_\mathrm{route}})$ is obtained from the averaged posterior (4.38) of the previous time step. Assuming the same predicted posterior for all models, we can now find a simple way to calculate the weights of the NMA filter by approximating the posterior for the different models also with a Dirac mixture density

$$\mathrm{p}(\mathcal{X}_{0:k}^{\mathrm{do}}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k}, R_{1:N_\mathrm{route}}) = \sum_{i=1}^{N_\mathrm{p}} w_k^{(i,j)} \delta(\mathcal{X}_{0:k}^{\mathrm{do}} - \mathcal{X}_{0:k}^{\mathrm{do},(i)}) \, . \tag{4.39}$$

In the above equation the particles set $\{\mathcal{X}_{0:k}^{\mathrm{do},(i)}\}_{i=1}^{N_\mathrm{p}}$ is the same for all models and is obtained from predicting the particles from the Dirac mixture representation of (4.38) from the previous time step, and $w_k^{(i,j)}$ are the model specific weights for model j. The unnormalized model specific weights are obtained by updating the weights $w_{k-1}^{(i)}$ of the averaged posterior Dirac mixture approximation from the previous time step

$$\tilde{w}_k^{(i,j)} = \mathrm{p}(z_k^{\mathrm{acc}}|\mathcal{X}_k^{\mathrm{do},(i)})\,\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{\mathrm{do},(i)}, \mathcal{M}_k^{(j)}, R_{1:N_{\mathrm{route}}})w_{k-1}^{(i)}\,. \tag{4.40}$$

Inserting the particle approximations (4.39) into the averaged posterior of (4.38) yields

$$\begin{aligned}\mathrm{p}(\mathcal{X}_{0:k}^{\mathrm{do}}|\mathcal{Z}_{1:k}, R_{1:N_{\mathrm{route}}}) &\approx \sum_{j=1}^{N_\mathcal{M}} \mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_{\mathrm{route}}}) \sum_{i=1}^{N_\mathrm{p}} w_k^{(i,j)}\delta(\mathcal{X}_{0:k}^{\mathrm{do}} - \mathcal{X}_{0:k}^{\mathrm{do},(i)})\\ &= \sum_{i=1}^{N_\mathrm{p}} \delta(\mathcal{X}_{0:k}^{\mathrm{do}} - \mathcal{X}_{0:k}^{\mathrm{do},(i)}) \sum_{j=1}^{N_\mathcal{M}} w_k^{(i,j)}\,\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_{\mathrm{route}}})\,,\end{aligned} \tag{4.41}$$

where the weights of the mixed posterior can be identified by comparing (4.38) to (4.41)

$$w_k^{(i)} = \sum_{j=1}^{N_\mathcal{M}} w_k^{(i,j)}\,\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_{\mathrm{route}}})\,. \tag{4.42}$$

A summary of the algorithm is shown in Algorithm 4.3. Up to line five Algorithm 4.2 and Algorithm 4.3 are the same. In line six then some differences appear, instead of just updating one set of weights, the NMA filter first updates a set of weights for each noise model. The model specific weights are then combined in a second step into the weights of the NMA filter.

4.5 Track Identification

Until now the focus of the chapter was on the along-track localization on a known route. In this section the assumption of a known route is dropped and we focus on the case of an unknown route. More specifically, two different approaches to this problem will be proposed. In the first approach, the track identifier is estimated jointly with the along-track position. In the second approach, the complete route is estimated by running one filter for each possible route.

4.5.1 Filter for Joint Estimation of Identifier and Along-Track Position

We start with the approach in which the track identifier is part of the particle state. This requires the estimation of the joint posterior

$$\mathrm{p}(\mathcal{X}_{0:k}|\mathcal{Z}_{1:k}) = \mathrm{p}(\mathbf{d}_{0:k}, I_{0:k}, O_{0:k}|\mathcal{Z}_{1:k})\,. \tag{4.43}$$

To estimate the joint posterior, the along-track localization filter for known routes has to be modified. The key step in the new modified filter is the prediction of the particles. Unlike the along-track filter with a known route, we have to cope for the joint estimation with the case when a particle passes a switch during the prediction. While the prediction step is considerably modified for the joint estimation of the position and the track identifier, the update step only requires minor changes.

Prediction Step

The passing of a switch introduces a strong nonlinearity into the prediction of the filter state. For example, when the train is driving towards the end of a track and then passes a switch the along-track position jumps either to zero or the length of the track after the switch. In addition, at a switch the sign of the speed and acceleration can jump depending on the orientation of the new track frame. The same applies for the orientation state, when the new track frame is not aligned with the previous track frame.

In the filter algorithm for joint estimation of position and track identifier, the prediction step is divided into two parts. In the first part, the state is predicted almost exactly as in the along-track filter for known routes by applying the following steps to the particles

1. Predict the dynamic train state $\mathbf{d}_k^{(i)}$ of each particle with the linear model in (4.20) and do not limit the result to the length of the current track indicated by the track identifier $I_{k-1}^{(i)}$ of the particle.
2. Set the predicted track identifier to the value of the identifier from the previous time step $I_k^{(i)} = I_{k-1}^{(i)}$.
3. Predict the orientation state $O_k^{(i)}$ based on the Markov chain model in Figure 4.2.

After the first part of the prediction, it is checked if the predicted along-track position of a particle is outside the track boundaries. If this is the case, the following second part of the prediction is applied to the corresponding particles

1. Predict the new track identifier $I_k^{(i)}$ of the particle. This is done by checking which tracks are connected to the track the particle just left. If more than one track is connected to the track, the new track is chosen randomly such that the probability for each possible track is the same. Note, the connections between the tracks defined in the network topology are always supposed to be known, although it is not explicitly mentioned in the equations.
2. Now that the new track identifier is defined, the dynamic train state can be adapted accordingly. The along-track position is derived from the distance the particle has traveled since the switch. If the particle enters the new track over its starting point the new along-track position is equal to the traveled distance after passing the switch. If the train enters the track over its end point the new along-track position is equal to the length of the new track minus the distance traveled after passing the switch. It can happen, that the traveled distance after the switch exceeds the length of the new track. In this case one has to again predict randomly the new track identifier until the traveled distance is shorter than the new track. For the speed and the acceleration only the sign has to be changed depending on how the previous and new track are connected to each other. The sign has to be changed when a particle leaves its previous track over the track endpoint and

Algorithm 4.4 Particle filter for joint estimation of track identifier and along-track position

Input: Particle set $\{\mathcal{X}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_\mathrm{p}}$ and observations $\mathcal{Z}_k$
Output: Updated set $\{\mathcal{X}_{k}^{(i)}, w_{k}^{(i)}\}_{i=1}^{N_\mathrm{p}}$
for all $i \in \{1, \dots, N_\mathrm{p}\}$ **do**
    $\mathbf{d}_k^{(i)} \sim \mathcal{N}(\mathbf{F}^\mathrm{d}\mathbf{d}_{k-1}^{(i)}, \mathbf{Q}^\mathrm{d})$
    $I_k^{(i)} = I_{k-1}^{(i)}$
    $O_k^{(i)} = O_{k-1}^{(i)}$
    if $u^{(i)} > \pi_O$ with $u^{(i)} \sim \mathcal{U}(0,1)$ **then**
        Set $O_k^{(i)}$ to opposite value of $O_{k-1}^{(i)}$
    end if
    if $s_k^{(i)} \notin [0, l_{I_k^{(i)}}]$ **then**
        Predict new track identifier $I_k^{(i)}$ by randomly selecting one of the possible tracks.
        Calculate $s_k^{(i)}$ on the new track and adapt sign of $\dot{s}_k^{(i)}$ and $\ddot{s}_k^{(i)}$.
        Check if track frame has changed and adapt sign of $O_k^{(i)}$.
    end if
    $\mathcal{X}_k^{(i)} = \{\mathbf{d}_k^{(i)}, O_k^{(i)}, I_k^{(i)}\}$
    $\tilde{w}_k^{(i)} = \mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{(i)})\, \mathrm{p}(z_k^{\mathrm{acc}}|\mathcal{X}_k^{(i)}) w_{k-1}^{(i)}$
end for
for all $i \in \{1, \dots, N_\mathrm{p}\}$ **do**
    $w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^{N_\mathrm{p}} \tilde{w}_k^{(j)}$
end for
if Resampling necessary **then**
    Draw N_p particles from $\{\mathcal{X}_{k}^{(i)}, w_{k}^{(i)}\}_{i=1}^{N_\mathrm{p}}$
    Set all weights to $w_k^{(i)} = \frac{1}{N_\mathrm{p}}$
end if
return

enters the new track over the endpoint. The same is true, when a particle leaves its previous track over the track starting point and enters its new track over the starting point.

3. Finally, the trains orientation is adapted. The orientation only changes when the new track frame does not correspond to the previous one. The two cases in which the orientation has to be changed are identical to the cases in which the sign for the speed and the acceleration has to be changed.

Update Step

The update step only needs a minor modification compared to the along-track localization filter for a known route. The main modification concerns the map function. For the along-track filter with a known route the map (4.13) was a function of the complete route. From the route a virtual track was constructed and the localization was done w.r.t. the virtual track. Here the map is now a function of the along-track position and the track identifier

$$\mathbf{m}^{\mathrm{t}}(s_k, I_k) : [0, l_{I_k}] \times \mathcal{T} \mapsto \mathbb{R}^3 \ . \tag{4.44}$$

Consequentially, the likelihood of the magnetometer measurement conditioned on $\mathcal{X}_k$ given by the Gaussian

$$\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k) = \mathcal{N}(\mathbf{z}_k^{\mathrm{mag}}; \mathbf{R}_{\mathrm{t}}^{\mathrm{b}}(O_k)\, \mathbf{m}^{\mathrm{t}}(s_k, I_k), (\sigma^{\mathrm{mag}})^2 \mathbf{I}_{3\times 3}) \tag{4.45}$$

is now, in contrast to (4.15), also a function of the track identifier. With (4.45) the weight update for particle i becomes

$$\tilde{w}_k^{(i)} = \mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{(i)})\, \mathrm{p}(z_k^{\mathrm{acc}}|\mathcal{X}_k^{(i)}) w_{k-1}^{(i)} \ . \tag{4.46}$$

The weight update of the joint particle filter is in accordance with the weight update equation (4.23) of the along-track filter with a known route, we just exchanged the magnetometer likelihood (4.15) with (4.45).

Filter Algorithm

The prediction and update step for the joint track identifier and along-track position filter is summarized in Algorithm 4.4. In Algorithm 4.4 it is assumed that no disturbances occur in the magnetometer measurements and hence we used a prediction and update step that is similar to the one of the simple along-track filter from Section 4.3.

When the track identifier is estimated by the filter the calculation of point estimates differs from the calculation for the filter with a known route. For a known route the MMSEE was calculated by the weighted sum over all particles (4.24). Now all particles have their own track identifier and their along-track position is no longer defined on a common virtual track. Averaging the along-track position of all particles therefore does not yield a meaningful result. To avoid this issue, we propose here to provide a point estimate only when most of the posterior probability is allocated on a single track. We therefore first calculate the probability for the different track identifiers I_k that are currently part of the particle set

$$\mathrm{pr}(I_k|\mathcal{Z}_{1:k}) = \sum_{j \in \{i | I_k^{(i)} = I_k\}} w_k^{(j)} \ . \tag{4.47}$$

A proof of the above equation is given in the Appendix Section A.2. If the probability of one track exceeds a predefined threshold, the MMSEE of position, speed, and acceleration is calculated over all particles with that track identifier according to (4.24). The weights used in (4.24) do not sum up to one anymore when the particle cloud spreads over more than one track. For the MMSEE we therefore apply a simple heuristic and normalize the weights to one before the weighted sum for the MMSEE is calculated. The MMSEE of the topological position then comprises the most probable track and the MMSEE of the corresponding along-track position. If none of the track identifiers in the particle set reaches the probability threshold the position is considered to be not defined. This can be problematic for some applications, but without further information from track-side devices or forward-looking sensors, such as cameras and laser scanners, there will be always a period of uncertainty between the time the switch is passed and the time it is resolved.

4.5.2 Filter for Joint Estimation of Route and Along-Track Position

The second approach for localization on an unknown route estimates the complete route instead of the current track identifier. To estimate the route, we apply an idea similar to the one used already in the robust particle filters and calculate the probabilities of different hypotheses for the measurement model. That means we want to calculate for each possible route $\mathcal{R}^{(i)} \in \mathcal{R}$ through the network the probability $\mathrm{pr}(\mathcal{R}^{(i)}|\mathcal{Z}_{1:k})$ that the route is correct based on the available measurements, where $\mathcal{R}$ is the set of all $N_{\mathcal{R}}$ routes. Similar to the model probabilities in the robust filters, we can calculate the route probability by the equation

$$\mathrm{pr}(\mathcal{R}^{(i)}|\mathcal{Z}_{1:k}) = \frac{\mathrm{p}(\mathcal{Z}_k|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})\,\mathrm{pr}(\mathcal{R}^{(i)}|\mathcal{Z}_{1:k-1})}{\mathrm{p}(\mathcal{Z}_k|\mathcal{Z}_{1:k-1})}. \tag{4.48}$$

Thus, the probability $\mathrm{pr}(\mathcal{R}^{(i)}|\mathcal{Z}_{1:k})$ of a route conditioned on all observations until time step k is a function of the route probability from the previous time step and the likelihood of the current observation. In contrast to the model probabilities in Section 4.4, the route probability must not be predicted in time because the route is fixed from the beginning and cannot change over time, hence also the time index was omitted. In essence this means, the transition probability between different routes is zero. To update the route probability we therefore only need to find a feasible way to obtain the likelihood in (4.48). Similar to (4.30), the likelihood will be calculated by a marginalization over the topological train pose

$$\begin{aligned}
\mathrm{p}(\mathcal{Z}_k|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) &= \int \mathrm{p}(\mathcal{Z}_k, \mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})d\mathcal{X}_k^{\mathrm{do}} \\
&= \int \mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do}}, \mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})\,\mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})d\mathcal{X}_k^{\mathrm{do}} \\
&= \int \mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do}}, \mathcal{R}^{(i)})\,\mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})d\mathcal{X}_k^{\mathrm{do}}\,.
\end{aligned} \tag{4.49}$$

In the last line of the above integral we identify the integrand as the product of the likelihood of the current observation and the predicted posterior of the along-track particle filter for a known route. Thus, for the calculation of the different route probabilities we have to calculate a separate along-track particle filter for each route through the network. In a naive implementation, this would result in an unfeasible number of filters running in parallel. To keep the number of filters manageable, the set of considered routes is adapted dynamically as will be described in the next paragraph. Note, the

Algorithm 4.5 Particle filter for joint estimation of route and along-track position

Input: Set of particle filters $\mathcal{F}_{k-1}$ and observations $\mathcal{Z}_k$
Output: Updated particle filter set $\mathcal{F}_k$
for all $j \in \{1, \ldots, N_{\mathcal{F}}\}$ **do**
  Predict particle set of filter $\mathcal{F}^{(j)}_{k-1}$
  if Particle crosses a switch **then**
    Create copy of $\mathcal{F}^{(j)}_{k-1}$ and add to set $\mathcal{F}_{k-1}$
    Add track identifiers of the new tracks to the route of the corresponding filters
  end if
end for
Update the route probability for all filters with (4.49) and (4.50)
When the probability of a route is above a defined threshold, delete all other routes and their corresponding filters from the set of filters
for all $j \in \{1, \ldots, N_{\mathcal{F}}\}$ **do**
  Update the weights $w^{(i,j)}_{k-1}$ of the j-th filter
  Resample if necessary
end for
Add updated filters to $\mathcal{F}_k$
return

normalization constant $\mathrm{p}(\mathcal{Z}_k|\mathcal{Z}_{1:k-1})$ in (4.48) can be replaced with the sum over the unnormalized probabilities of all routes which yields the update equation for the route probabilities

$$\mathrm{pr}(\mathcal{R}^{(i)}|\mathcal{Z}_{1:k}) = \frac{\mathrm{p}(\mathcal{Z}_k|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})\,\mathrm{pr}(\mathcal{R}^{(i)}|\mathcal{Z}_{1:k-1})}{\sum\limits_{j=1}^{N_{\mathcal{R}}} \mathrm{p}(\mathcal{Z}_k|\mathcal{R}^{(j)}, \mathcal{Z}_{1:k-1})\,\mathrm{pr}(\mathcal{R}^{(j)}|\mathcal{Z}_{1:k-1})} \,. \tag{4.50}$$

Management of Routes

To limit the computational complexity and memory requirements, we have to keep the number of considered routes small. Let us assume the train starts on a known track, then all routes that start with that track have the same probability and the along-track filter will be the same. All routes that do not start on the correct track will have zero probability. As soon as the first particle of the along-track filter reaches a switch the situation becomes a bit more complicated. Now we have two possible routes, one includes the left track and one the right track. In this situation a copy of the along-track filter is created, which results in one copy for each possibly route. For each of the routes now the probability is calculated and updated with each new measurement. When a route probability is above a defined threshold, the switch is considered to be resolved and all other routes are no longer feasible and the corresponding filters will be deleted. Thus, at one point a hard decision is made for one of the routes. If one filter reaches a switch before a decision for a single route was made, it can happen that a third filter has to be created. As will be discussed in the evaluation in Chapter 7, the magnetic field enables a rather quick identification of the route, which means also the number of tracked routes should stay bounded.

Filter Algorithm

In Algorithm 4.5 one recursion of the filter algorithm is shown. The input to the algorithm is no longer a single set of weighted particles but rather a set of particle filters $\mathcal{F}_k = \{\mathcal{F}_k^{(j)}\}_{j=1}^{N_\mathcal{F}}$ that contains one filter for each tracked route. A filter $\mathcal{F}_k^{(j)}$ is itself a set that contains the route information, the weighted particle set, and the route probability. The prediction step for the individual filters is performed once more based on the motion model and the Markov chain from Section 4.3.2. If during the prediction a particle has left the known route over a branching switch, a copy of the filter is created such that one filter for each possible route exists. After the prediction step is completed, the route probabilities are calculated. In the case that a route probability exceeds a certain threshold all filters but one are deleted. For the weight update in Algorithm 4.5, we can either choose to use the update step from Algorithm 4.1 or from the two robust filters. If a robust filter is used, the individual filters $\mathcal{F}_k^{(j)}$ have to be adapted accordingly such that also the model probabilities are calculated. Point estimates are obtained only when the route is identified. When the route is identified, only one filter remains and the point estimates can be calculated as described for the particle filter on a known route.

Finally, we want to note that this section is based on the algorithm that we presented in [C5]. The main difference to [C5] lies in the motion model of the along-track filter that uses the acceleration as an input and in the way the decision for a route is made. In [C5] routes are removed based on a likelihood ratio test of the filter residuals. In this thesis we switch to route probabilities because this is a more intuitive and accessible way to judge the "correctness" of a certain route.

4.6 Summary

In this chapter different filter algorithms for magnetic field-based train localization were introduced. The first filter introduced is the simplest and estimates the along-track position and orientation of a train on a known route with magnetometer and accelerometer data. Based on this filter, two robust filter algorithms were presented that cope with unmodeled disturbances in the magnetometer data. The robust filters use multiple measurement models and calculate the probability of these models to either exclude disturbed measurements from the filter update step or to reduce their impact on the filter output. After the introduction of the robust filter algorithms, two possible approaches were proposed to cope with the case of an unknown route. In the first approach, a particle filter was designed that estimates the track identifier and the along-track position jointly by randomly assigning each particle a track after it passes a switch. In the second approach, the track identifier is not part of the filter state, but instead one filter per possible route is calculated. For each of those filters the probabilities that their corresponding route is correct are calculated and unlikely routes are removed in order to minimize the computing effort and memory requirements. The filters presented in this chapter will be evaluated and compared in Chapter 7.

CHAPTER 5

Simultaneous Localization and Calibration

In Chapter 4 different algorithms for the estimation of the topological train position were proposed. In all these algorithms it was assumed that, in the absence of disturbances, the magnetometer measurements are noisy samples of a known magnetic map. This assumption implies that the magnetometer used to create the map and the magnetometer used for localization are both calibrated. Unfortunately, the calibration of a train-mounted magnetometer is not a trivial task because common methods for magnetometer calibration cannot be used. In this chapter, we will discuss the underlying problem encountered in the railway domain and we will propose a filter that estimates the train position and the calibration parameters simultaneously.

5.1 Magnetometer Calibration

5.1.1 Sensor Model

For the calibration of a magnetometer we need to define its sensor model that creates a relation between the calibration parameters, the measured magnetic field, and the true magnetic field an ideal sensor would measure. In the literature, cf. [76–80], a common magnetometer model is given by

$$\mathbf{z}_k^{\mathrm{mag}} = \mathbf{C}\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}}\mathbf{m}^{\mathrm{n}} + \mathbf{b} + \mathbf{n}_k^{\mathrm{mag}} \,. \tag{5.1}$$

In the above model $\mathbf{m}^{\mathrm{n}}$ is the true physical magnetic field given in the navigation frame that a perfect sensor should measure when the sensor axes are aligned with the navigation frame and $\mathbf{n}_k^{\mathrm{mag}}$ is once more Gaussian sensor noise. To account for the rotation between body and navigation frame, the true magnetic field is rotated into the body frame by multiplying it with the time variant rotation matrix $\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}}$. For an uncalibrated sensor, the rotated true magnetic field has to be multiplied additionally with the real matrix $\mathbf{C} \in \mathbb{R}^{3\times 3}$ and a bias vector $\mathbf{b} \in \mathbb{R}^3$ has to be added. Matrix $\mathbf{C}$ accounts for soft iron effects and the bias $\mathbf{b}$ for hard iron effects. Both, the hard iron and soft iron effects, are caused by magnetic material in the platform on which the magnetometer is mounted. Hard iron effects are a consequence of permanently magnetized material in the vicinity of the sensor. Soft iron effects are a result of the interaction between magnetic material on the platform and the external magnetic field.

Model (5.1) is rather generic and cannot only account for hard and soft iron effects but also for imperfections of the sensor itself. Examples for such imperfections are non-orthogonality of the sensor axes, hardware induced biases, and scaling errors [76, 78]. When no particular structure is imposed on matrix $\mathbf{C}$ and vector $\mathbf{b}$ the model can also compensate for other linear magnetic effects and sensor imperfections [78]. Given the sensor model (5.1), the calibration of the sensor can be reduced to the task of estimating the elements of $\mathbf{C}$ and $\mathbf{b}$. In the following, we will take a closer look at how the calibration parameters can be estimated and under which conditions they are observable.

5.1.2 Observability of Calibration Parameters

For the observability analysis we start by defining the estimation problem. The estimation problem can be expressed in terms of a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{12}$ that contains all elements of $\mathbf{C}$ and $\mathbf{b}$

$$\boldsymbol{\theta} = \begin{bmatrix} [\mathbf{C}]_{1:} & [\mathbf{b}]_1 & [\mathbf{C}]_{2:} & [\mathbf{b}]_2 & [\mathbf{C}]_{3:} & [\mathbf{b}]_3 \end{bmatrix}^\top , \tag{5.2}$$

where $[\mathbf{C}]_{i:}$ is the i-th row of the calibration matrix $\mathbf{C}$ and $[\mathbf{b}]_i$ is the i-th element of the bias vector $\mathbf{b}$. With the parameter vector, the sensor model (5.1) can be reformulated to

$$\mathbf{z}_k^{\mathrm{mag}} = \mathbf{H}(\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}})\boldsymbol{\theta} + \mathbf{n}_k^{\mathrm{mag}} , \tag{5.3}$$

where $\mathbf{H}(\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}})$ is a matrix-valued function of the true magnetic field rotated into the body frame of the magnetometer

$$\mathbf{H}(\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}}) = \begin{bmatrix} \begin{bmatrix} (\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}})^\top & 1 \end{bmatrix} & \mathbf{0}_{1\times 4} & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{1\times 4} & \begin{bmatrix} (\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}})^\top & 1 \end{bmatrix} & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{1\times 4} & \mathbf{0}_{1\times 4} & \begin{bmatrix} (\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}})^\top & 1 \end{bmatrix} \end{bmatrix} . \tag{5.4}$$

In order to estimate the calibration parameters, a sequence of measurements $\mathbf{z}_{1:k}^{\mathrm{mag}}$ is required. Stacking these measurements into a long column vector and by using (5.3) we get a linear system of equations

$$\underbrace{\begin{bmatrix} \mathbf{z}_1^{\mathrm{mag}} \\ \vdots \\ \mathbf{z}_k^{\mathrm{mag}} \end{bmatrix}}_{\tilde{\mathbf{z}}} = \underbrace{\begin{bmatrix} \mathbf{H}(\mathbf{R}_{\mathrm{n},1}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}}) \\ \vdots \\ \mathbf{H}(\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}}) \end{bmatrix}}_{\mathbf{D}} \boldsymbol{\theta} + \underbrace{\begin{bmatrix} \mathbf{n}_1^{\mathrm{mag}} \\ \vdots \\ \mathbf{n}_k^{\mathrm{mag}} \end{bmatrix}}_{\tilde{\mathbf{n}}^{\mathrm{m}}} . \tag{5.5}$$

If the true magnetic field $\mathbf{m}^{\mathrm{n}}$ and the rotation between navigation and body frame $\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}}$ are known, a linear least squares (LS) estimator can be applied to obtain an ML estimate of the parameter vector

$$\hat{\boldsymbol{\theta}} = (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}^\top \tilde{\mathbf{z}} . \tag{5.6}$$

The LS estimate in (5.6) is only unique when matrix $\mathbf{D}$ has full column rank $\mathrm{rank}(\mathbf{D}) = 12$ [81]. The definition of $\mathbf{D}$ in (5.5) makes it clear that at least four measurements are required for this, as the number of rows and therefore the maximal rank is otherwise less than 12. In addition, the four measurements have to differ from each other because otherwise some of the rows are linearly dependent. Thus, to ensure a unique solution of the LS estimate and to render the parameter vector observable, the matrix $\mathbf{H}(\mathbf{R}_{\mathrm{n},k}^{\mathrm{b}} \mathbf{m}^{\mathrm{n}})$ has to be altered between measurements by changing the magnetic field $\mathbf{m}^{\mathrm{n}}$ or the sensor

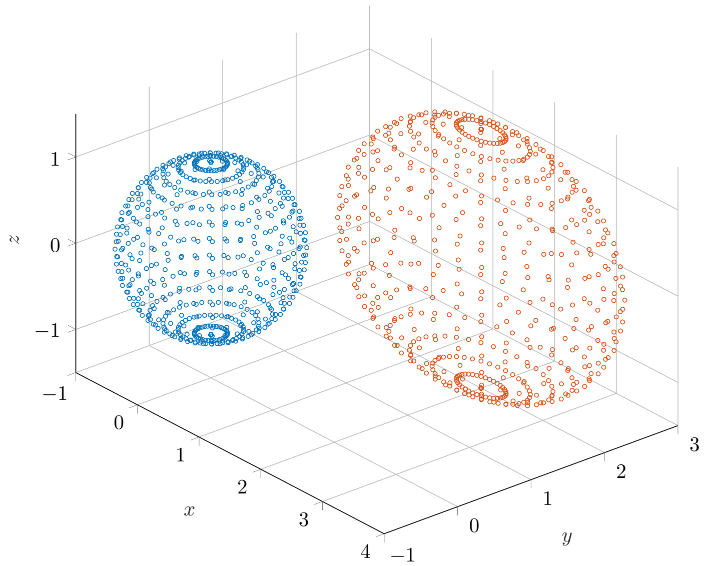

Figure 5.1: Unit sphere (blue) and ellipsoid (red) obtained by applying the linear transformation given by (5.1) to the sphere.

orientation $\mathbf{R}^{\mathrm{b}}_{\mathrm{n},k}$. As the measurements contain noise, more than four measurements should be used in the estimation to obtain good results. In this section only the LS estimator is considered, but the above considerations also apply in principle to Bayesian estimators: without a sufficient amount of different measurements the estimation performance will be poor.

In (5.6), the orientation and the magnetic field is considered to be known. As this assumption is rather unrealistic when the magnetometer and the platform on which it is mounted is not in a laboratory, the simple LS estimator cannot be employed in most applications. Therefore, more involved methods for calibration are required in practice. A popular method that does not require the sensor orientation is ellipsoid fitting [77, 80]. For ellipsoid fitting the platform with the magnetometer is placed in an area with a homogeneous magnetic field, e.g., an outside area without magnetic material in the immediate vicinity, and then the sensor is rotated around all axes. The magnetic field vectors a perfect magnetometer would measure during the rotation of the sensor is given by $\mathbf{R}^{\mathrm{b}}_{\mathrm{n},k}\mathbf{m}^{\mathrm{n}}$. Since a rotation of a vector does not change its length, the measurements lie on the surface of a centered sphere with a radius equal to the magnitude $|\mathbf{m}^{\mathrm{n}}|$ of the true magnetic field. For an uncalibrated magnetometer, the calibration matrix $\mathbf{C}$ transforms the sphere into an ellipsoid and the bias vector $\mathbf{b}$ translates the origin of the ellipsoid. Figure 5.1 shows an example for $\mathbf{C} = \mathrm{diag}([2 \quad 1 \quad 1.5])$ and $\mathbf{b} = [2 \quad 2 \quad 0]^\top$. Taking this into consideration, calibration is the task of finding values for $\mathbf{C}$ and $\mathbf{b}$ that map points from a sphere to points on an ellipsoid. In ellipsoid fitting this is achieved by finding parameters such that the magnitude of the calibrated magnetic field measurements is as close as possible to $|\mathbf{m}^{\mathrm{n}}|$.

For the calibration of train mounted magnetometers common calibration methods, such as ellipsoid fitting, are not applicable. This is due to the high train mass and size, which makes it almost impossible to rotate the magnetometer while it is rigidly mounted on the train. If the rotation of the magnetometer is not an option, the only possibility left to render the calibration parameters observable is to change the true external magnetic field $\mathbf{m}^{\mathrm{n}}$ instead. Fortunately, the magnetic field along a railway track shows strong distortions, which means instead of rotating the sensor to change the measured magnetic field we can simply move it along the railway track.

5.2 Calibration of a Train-Mounted Magnetometer

For a train-mounted uncalibrated magnetometer the sensor model is obtained by replacing the true external magnetic field in the body frame $\mathbf{R}^{\mathrm{b}}_{\mathrm{n},k}\mathbf{m}^{\mathrm{n}}$ from (5.1) with the rotated magnetic map $\mathbf{R}^{\mathrm{b}}_{\mathrm{t}}(O_k)\,\mathbf{m}^{\mathrm{t}}(s_k, R_{1:N_{\text{route}}})$ from (4.14)

$$\mathbf{z}^{\text{mag}}_k = \mathbf{C}\mathbf{R}^{\mathrm{b}}_{\mathrm{t}}(O_k)\,\mathbf{m}^{\mathrm{t}}(s_k, R_{1:N_{\text{route}}}) + \mathbf{b} + \mathbf{n}^{\text{mag}}_k\,. \tag{5.7}$$

Alternatively, the equation can be expressed in terms of the calibration vector

$$\mathbf{z}^{\text{mag}}_k = \mathbf{H}(\mathbf{R}^{\mathrm{b}}_{\mathrm{t}}(O_k)\,\mathbf{m}^{\mathrm{t}}(s_k, R_{1:N_{\text{route}}}))\boldsymbol{\theta} + \mathbf{n}^{\text{mag}}_k\,. \tag{5.8}$$

If in (5.8) the orientation O_k, the map $\mathbf{m}^{\mathrm{t}}(\cdot)$, the route $R_{1:N_{\text{route}}}$, and the along-track position s_k are known, the calibration parameters can be obtained once more with a linear estimator such as the one described by (5.6). To render the parameters observable, the train has to move to ensure that the external magnetic field changes. The travel distance necessary to get a good estimate depends on the strength and spatial variability of the magnetic field of the track the train is driving on.

In a practical setting, the main difficulty for performing the calibration is to obtain the true magnetic field $\mathbf{R}^{\mathrm{b}}_{\mathrm{t}}(O_k)\,\mathbf{m}^{\mathrm{t}}(s_k, R_{1:N_{\text{route}}})$ a calibrated magnetometer would measure. This requires the estimator to know the train orientation and position at all times and to have a map of the true magnetic field. Especially obtaining the map is a difficult endeavor, because the map should contain the true magnetic field and hence has to be recorded already with a calibrated sensor. To address this issue, in the remainder of this section we will show that under certain assumptions we can perform calibration also relative to a map that was recorded only with an uncalibrated magnetometer.

A map created from the measurements of an uncalibrated magnetometer is related to the true magnetic map by

$$\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\text{route}}}) = \tilde{\mathbf{C}}\mathbf{R}^{\tilde{\mathrm{b}}}_{\mathrm{t}}(s_k, R_{1:N_{\text{route}}})\,\mathbf{m}^{\mathrm{t}}(s_k, R_{1:N_{\text{route}}}) + \tilde{\mathbf{b}}\,. \tag{5.9}$$

In (5.9) $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{b}}$ are the calibration parameters of the magnetometer used to create the map. The rotation matrix $\mathbf{R}^{\tilde{\mathrm{b}}}_{\mathrm{t}}(s_k, R_{1:N_{\text{route}}})$ describes the orientation of the magnetometer body frame, indicated by the superscript $\tilde{\mathrm{b}}$, relative to the track frame during the mapping measurements and is a function of the train position. To get a relation between the uncalibrated map (5.9) and the measurements (5.7), first (5.9) is solved for the true map

$$\mathbf{m}^{\mathrm{t}}(s_k, R_{1:N_{\text{route}}}) = \mathbf{R}^{\mathrm{t}}_{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\text{route}}})\tilde{\mathbf{C}}^{-1}\left(\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\text{route}}}) - \tilde{\mathbf{b}}\right)\,, \tag{5.10}$$

where $\mathbf{R}^{\mathrm{t}}_{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}})$ describes the inverse rotation of $\mathbf{R}^{\tilde{\mathrm{b}}}_{\mathrm{t}}(s_k, R_{1:N_{\mathrm{route}}})$. Second, (5.10) is inserted into (5.7) which yields

$$\begin{aligned}
\mathbf{z}_k^{\mathrm{mag}} &= \mathbf{C}\underbrace{\mathbf{R}^{\mathrm{b}}_{\mathrm{t}}(O_k)\mathbf{R}^{\mathrm{t}}_{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}})}_{\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}}\tilde{\mathbf{C}}^{-1}\left(\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}}) - \tilde{\mathbf{b}}\right) + \mathbf{b} + \mathbf{n}_k^{\mathrm{mag}} \\
&= \mathbf{C}\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}\tilde{\mathbf{C}}^{-1}\left(\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}}) - \tilde{\mathbf{b}}\right) + \mathbf{b} + \mathbf{n}_k^{\mathrm{mag}} \\
&= \mathbf{C}\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}}) - \mathbf{C}\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{b}} + \mathbf{b} + \mathbf{n}_k^{\mathrm{mag}}\ .
\end{aligned} \tag{5.11}$$

This is again a linear model similar to (5.1) and can be written in short form as

$$\mathbf{z}_k^{\mathrm{mag}} = \bar{\mathbf{C}}_k\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}}) + \bar{\mathbf{b}}_k + \mathbf{n}_k^{\mathrm{mag}}\ , \tag{5.12}$$

where the calibration matrix and the bias vector are given by

$$\begin{aligned}
\bar{\mathbf{C}}_k &= \mathbf{C}\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}\tilde{\mathbf{C}}^{-1} \\
\bar{\mathbf{b}}_k &= -\mathbf{C}\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{b}} + \mathbf{b}\ .
\end{aligned} \tag{5.13}$$

Comparable to (5.8), (5.12) may also be formulated in terms of a parameter vector

$$\mathbf{z}_k^{\mathrm{mag}} = \mathbf{H}(\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}}))\bar{\boldsymbol{\theta}}_k + \mathbf{n}_k^{\mathrm{mag}}\ , \tag{5.14}$$

where $\bar{\boldsymbol{\theta}}_k$ contains the elements of $\bar{\mathbf{C}}_k$ and $\bar{\mathbf{b}}_k$. The parameters are now a mixture of the calibration parameters of both magnetometers, the magnetometer used in the map creation and the magnetometer that we want to calibrate. The parameters depend on the relative orientation $\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}$ of the two magnetometer body frames and hence are time variant as indicated by the time index k. The dependency on the time is not ideal and complicates the calibration considerably. Fortunately, in the railway environment the relative orientation $\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}$ behaves rather nicely. To show this, we decompose the rotation from the track frame to the body frame into two consecutive rotations

$$\mathbf{R}^{\tilde{\mathrm{b}}}_{\mathrm{t}}(s_k, R_{1:N_{\mathrm{route}}}) = \mathbf{R}^{\tilde{\mathrm{b}}}_{\tilde{\mathrm{v}}}\mathbf{R}^{\tilde{\mathrm{v}}}_{\mathrm{t}}(s_k, R_{1:N_{\mathrm{route}}})\ , \tag{5.15}$$

where the rotation between the body and vehicle frame is constant as long as the magnetometer is rigidly mounted to the vehicle. When this decomposition is applied also to $\mathbf{R}^{\mathrm{b}}_{\mathrm{t}}(O_k)$ the relative orientation between the body frames becomes

$$\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k} = \mathbf{R}^{\mathrm{b}}_{\mathrm{v}}\underbrace{\mathbf{R}^{\mathrm{v}}_{\mathrm{t},k}\mathbf{R}^{\mathrm{t}}_{\tilde{\mathrm{v}}}(s_k, R_{1:N_{\mathrm{route}}})}_{\mathbf{R}^{\mathrm{v}}_{\tilde{\mathrm{v}},k}}\mathbf{R}^{\tilde{\mathrm{v}}}_{\tilde{\mathrm{b}}}\ . \tag{5.16}$$

In (5.16) the outer left and right matrices are constant rotations, the only time varying part is the relative rotation between the different vehicles frames $\mathbf{R}^{\mathrm{v}}_{\tilde{\mathrm{v}},k}$. From our measurements, we have found that the differences in the roll and pitch angles of different vehicle frames at the same track position are rather small. Figure 7.10 in Chapter 7 shows an example, in which the difference in the two angles is in the range of 0.5°. Therefore, the difference between vehicles frames will be mainly due to the train orientation relative to the track. As a result, two cases must be distinguished. In the first case, both vehicles frames coincide which result in $\mathbf{R}_{\tilde{\mathrm{v}}}^{\mathrm{v}} \approx \mathbf{I}_{3\times3}$. In the second case, they are opposite and the rotation becomes

$$\mathbf{R}_{\tilde{\mathrm{v}}}^{\mathrm{v}} \approx \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{5.17}$$

The rotation matrix (5.17) is obtained from multiplying the matrix (4.5) for two different orientations. In both cases, matrix (5.16) becomes nearly a constant and it can change its value only when the train enters a new track since we assume a train does not change its orientation in the middle of a track. This assumption implies that for the mapping of a track measurements from a continuous train run are used or that the train orientation stayed constant throughout the measurements. From the above remarks one can conclude that the calibration parameters of (5.12) are approximately constant and can have only two different values. Furthermore, the values can only change when the train is switching the track. This means that two sets of parameters are required to cover both cases and the parameters have to be adapted accordingly when the track is changed. The actual estimation of the parameters is the same for both values of $\mathbf{R}_{\tilde{\mathrm{v}}}^{\mathrm{v}}$ and can be done with (5.14) and a LS estimator of the form (5.6) under the condition that the train position is known and the relative orientation of the vehicle frames does not change during calibration.

5.3 Joint Estimation of Calibration Parameters and Position

In the previous section we assumed that we know the train position and therefore also $\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}})$. With this assumption and (5.14) the calibration reduces to a linear estimation problem. Since the primary goal of this thesis is train localization with the magnetic field, this leads to a chicken-or-egg problem. To estimate the train position, we need the calibration parameters and to calibrate the magnetometer we need the position. This kind of problem is also encountered in other contexts. A prominent example of this is the SLAM problem, in which a map of the environment has to be created while simultaneously the position within the unknown map has to be estimated. The SLAM problem is discussed extensively for different environments and sensor modalities. SLAM is used, e.g., for magnetic field mapping [39, 40, 42, 45] and the mapping of wireless transmitters for localization with signals of opportunity [82, 83]. Furthermore, SLAM methods are used in combination with laser scanners [84, 85], cameras [86], or radars [87, 88] for localization in unknown environments. Even if at first glance the calibration problem and the SLAM problem do not appear to be comparable, they are mathematically similar. To acknowledge this similarity, we call the proposed procedure simultaneous localization and calibration (SLAC).

5.3.1 Rao-Blackwellized Particle Filter for SLAC

When the position and the calibration parameters are both unknown they have to be estimated jointly and we have to find the joint density

$$\mathrm{p}(\mathcal{X}^{\mathrm{do}}_{0:k}, \bar{\boldsymbol{\theta}}_k | \mathcal{Z}_{1:k}, R_{1:N_{\mathrm{route}}}) \,, \tag{5.18}$$

or at least some moments of it. In the above density, and for the rest of the chapter, the route of the train is considered to be known although in the following equations the conditioning on the route is dropped for brevity. The case for an unknown route is out of the scope of this thesis and will be considered in future work.

In contrast to SLAM, where the position and the map are estimated, here the position and the calibration parameters are estimated. Besides this difference, the underlying problem is the same. Hence, the idea of FastSLAM [89] is also applicable to the calibration problem. In FastSLAM, the joint density is decomposed into a map and a position part and then a Rao-Blackwellized particle filter is applied to find an approximation of the posterior. In this section we derive the concrete filter algorithm for SLAC. A more general introduction to Rao-Blackwellized particle filters can be found in [90]. The decomposition idea from FastSLAM applied to the SLAC problem factorizes the density (5.18) into a position part and a parameter part

$$\mathrm{p}(\mathcal{X}^{\mathrm{do}}_{0:k}, \bar{\boldsymbol{\theta}}_k | \mathcal{Z}_{1:k}) = \mathrm{p}(\bar{\boldsymbol{\theta}}_k | \mathcal{X}^{\mathrm{do}}_{0:k}, \mathcal{Z}_{1:k}) \mathrm{p}(\mathcal{X}^{\mathrm{do}}_{0:k} | \mathcal{Z}_{1:k}) \,. \tag{5.19}$$

The parameter-related part $\mathrm{p}(\bar{\boldsymbol{\theta}}_k | \mathcal{X}^{\mathrm{do}}_{0:k}, \mathcal{Z}_{1:k})$ is conditioned on the measurements and the position and thus the relation between the magnetometer measurements and the parameter vector is linear due to the magnetometer model (5.14). If we further assume a linear Gaussian model for the temporal behavior of the parameters, the parameter part of the posterior is analytically tractable and obtained with a linear Kalman filter. In the previous section the parameters were shown to be approximately constant, which will be expressed by the linear model

$$\bar{\boldsymbol{\theta}}_k = \bar{\boldsymbol{\theta}}_{k-1} + \mathbf{w}^{\bar{\boldsymbol{\theta}}}_{k-1} \,, \tag{5.20}$$

with the Gaussian process noise $\mathbf{w}^{\bar{\boldsymbol{\theta}}}_{k-1} \sim \mathcal{N}(\mathbf{0}_{12\times 1}, (\sigma^{\bar{\boldsymbol{\theta}}})^2 \mathbf{I}_{12\times 12})$. For truly constant parameters the standard deviation $\sigma^{\bar{\boldsymbol{\theta}}}$ of the process noise would be zero. Since the parameters are only approximately constant, the standard deviation $\sigma^{\bar{\boldsymbol{\theta}}}$ is treated here as a tuning parameter that defines how fast the parameters can change over time. With zero process noise the estimated uncertainty of the parameters would tend towards zero. When this happens, the parameters will no longer be adapted to the measurements. This is not desired in most applications, especially when the state space model of the filter is only an approximation of the true state behavior.

While the parameter part of (5.19) can be easily estimated with a Kalman filter, the calculation of the position part is more complicated. For the position part $\mathrm{p}(\mathcal{X}^{\mathrm{do}}_{0:k} | \mathcal{Z}_{1:k})$ the along-track particle filter from Section 4.3 is modified to account for the calibration parameters. The main modification to the filter is how the weight update for the particles is performed. The magnetometer update requires us to evaluate the likelihood $\mathrm{p}(\mathbf{z}^{\mathrm{mag}}_k | \mathcal{X}^{\mathrm{do}}_k)$ at the different particle positions. Due to the magnetometer model

(5.14) the likelihood cannot be evaluated directly because the calibration parameters are unknown. To resolve this problem, we will marginalize the magnetometer likelihood w.r.t. the parameter vector $\bar{\boldsymbol{\theta}}_k$. Before performing the marginalization, it is important to recall (2.10). In (2.10) the Markov property of the system allowed us to replace the likelihood $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})$ with $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)$. If we do not make this simplification, the new weight is obtained by multiplying the old weight with $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})$ instead of $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)$. This modification only affects the particle filter weight update, the rest of the filter algorithm remains unchanged. This can be seen easily from (2.41) and (2.42) when $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})$ is substituted for $\mathrm{p}(\mathbf{z}_k|\mathbf{x}_k)$. Thus, the magnetometer likelihood for the weight update of the i-th particle can be obtained from the integral

$$\begin{aligned} \mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^{\mathrm{mag}}) &= \int \mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}, \bar{\boldsymbol{\theta}}_k|\mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^{\mathrm{mag}}) d\bar{\boldsymbol{\theta}}_k \\ &= \int \mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\bar{\boldsymbol{\theta}}_k, \mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^{\mathrm{mag}})\, \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^{\mathrm{mag}}) d\bar{\boldsymbol{\theta}}_k \,. \end{aligned} \tag{5.21}$$

Note, in the above equation we only account for the dynamic train state $\mathbf{d}$ and not the orientation because the orientation is absorbed in the calibration parameters. The orientation is only required for the likelihood of the accelerometer. If no accelerometer is used, the orientation can be completely dropped from the estimated state. The first factor of the integrand in (5.21) is the likelihood of the current measurement $\mathbf{z}_k^{\mathrm{mag}}$ and is defined by the measurement model (5.14) and the respective measurement noise density

$$\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\bar{\boldsymbol{\theta}}_k, \mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^{\mathrm{mag}}) = \mathcal{N}(\mathbf{z}_k^{\mathrm{mag}}; \mathbf{H}(\tilde{\mathbf{m}}^{\mathrm{b}}(s_k^{(i)}))\bar{\boldsymbol{\theta}}_k, (\sigma^{\mathrm{mag}})^2\mathbf{I}_{3\times 3}) \,. \tag{5.22}$$

The second factor of the integrand is the predicted posterior of the calibration parameters. Because we use in the weight update the likelihood $\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^{\mathrm{mag}})$ instead of $\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathbf{d}_k^{(i)})$ the posterior of the parameters is here conditioned on all past measurements and the complete history of the dynamic train state. For this case, the relationship between the magnetometer data and the calibration parameters is linear and can be determined with a Kalman filter and is given by the Gaussian

$$\mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^{\mathrm{mag}}) = \mathcal{N}(\bar{\boldsymbol{\theta}}_k; \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),-}, \boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),-}) \,, \tag{5.23}$$

where $\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),-}$ is the predicted mean and $\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),-}$ the predicted covariance matrix of the Kalman filter at time step k for particle i. The mean and covariance have to be calculated for each particle history. Therefore, N_{p} Kalman filters have to be predicted and updated in each time step of the Rao-Blackwellized particle filter. Due to the system model (5.20) of the parameters, the predicted mean is the updated mean from the previous time step $\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),-} = \boldsymbol{\mu}_{k-1}^{\bar{\boldsymbol{\theta}},(i),+}$ and the predicted covariance is the updated covariance from the previous time step plus the process noise $\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),-} = \boldsymbol{\Sigma}_{k-1}^{\bar{\boldsymbol{\theta}},(i),+} + (\sigma^{\bar{\boldsymbol{\theta}}})^2\mathbf{I}_{12\times 12}$. For the Gaussian likelihood (5.22) and Gaussian predicted posterior (5.23) the integral in (5.21) has the closed form solution

$$\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^{\mathrm{mag}}) = \mathcal{N}(\mathbf{z}_k^{\mathrm{mag}}; \mathbf{H}_k^{(i)}\, \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),-}, \mathbf{H}_k^{(i)}\, \boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),-}(\mathbf{H}_k^{(i)})^\top + (\sigma^{\mathrm{mag}})^2\mathbf{I}_{3\times 3}) \,, \tag{5.24}$$

where $\mathbf{H}_k^{(i)} = \mathbf{H}(\tilde{\mathbf{m}}^{\mathrm{b}}(s_k^{(i)}))$. The covariance matrix of the likelihood is the sum of the covariance of the parameter vector projected into the measurement space and the measurement noise covariance.

Algorithm 5.1 Rao-Blackwellized particle filter for SLAC

Input: Particle set $\{\mathcal{X}_{k-1}^{\mathrm{do},(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_\mathrm{p}}$, Kalman filter estimates $\{\boldsymbol{\mu}_{k-1}^{\bar{\boldsymbol{\theta}},(i),+}, \boldsymbol{\Sigma}_{k-1}^{\bar{\boldsymbol{\theta}},(i),+}\}$, observations $\mathcal{Z}_k$ and route information $R_{1:N_\mathrm{route}}$

Output: Updated particle set $\{\mathcal{X}_k^{\mathrm{do},(i)}, w_k^{(i)}\}_{i=1}^{N_\mathrm{p}}$ and Kalman filter estimates $\{\boldsymbol{\mu}_k^{\tilde{\boldsymbol{\theta}},(i),+}, \boldsymbol{\Sigma}_k^{\tilde{\boldsymbol{\theta}},(i),+}\}$

for all $i \in \{1, \ldots, N_\mathrm{p}\}$ **do**

  $\mathbf{d}_k^{(i)} \sim \mathcal{N}(\mathbf{F}^\mathrm{d}\mathbf{d}_{k-1}^{(i)}, \mathbf{Q}^\mathrm{d})$

  if $s_k^{(i)} \notin [0, l_\mathrm{route}]$ **then**

    Set $s_k^{(i)}$ to closest endpoint of interval $[0, l_\mathrm{route}]$

  end if

  $O_k^{(i)} = O_{k-1}^{(i)}$

  if $u^{(i)} > \pi_O$ with $u^{(i)} \sim \mathcal{U}(0,1)$ **then**

    Set $O_k^{(i)}$ to opposite value of $O_{k-1}^{(i)}$

  end if

  $\mathcal{X}_k^{\mathrm{do},(i)} = \{\mathbf{d}_k^{(i)}, O_k^{(i)}\}$

  $\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),-} = \boldsymbol{\mu}_{k-1}^{\bar{\boldsymbol{\theta}},(i),+}$

  $\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),-} = \boldsymbol{\Sigma}_{k-1}^{\bar{\boldsymbol{\theta}},(i),+} + (\sigma^{\bar{\boldsymbol{\theta}}})^2\mathbf{I}_{12\times12}$

  $\mathbf{S}_k^{(i)} = \mathbf{H}_k^{(i)}\,\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),-}(\mathbf{H}_k^{(i)})^\top + (\sigma^\mathrm{mag})^2\mathbf{I}_{3\times3}$

  $\tilde{w}_k^{(i)} = \mathrm{p}(\mathbf{z}_k^\mathrm{mag}|\mathbf{d}_{0:k}^{(i)}, \mathbf{z}_{1:k-1}^\mathrm{mag})\,\mathrm{p}(z_k^\mathrm{acc}|\mathcal{X}_k^{\mathrm{do},(i)})w_{k-1}^{(i)}$

  $\mathbf{K}_k^{(i)} = \boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),-}(\mathbf{H}_k^{(i)})^\top(\mathbf{S}_k^{(i)})^{-1}$

  $\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} = \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),-} + \mathbf{K}_k^{(i)}(\mathbf{z}_k^\mathrm{mag} - \mathbf{H}_k^{(i)}\,\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),-})$

  $\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),+} = \boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),-} - \mathbf{K}_k^{(i)}\mathbf{S}_k^{(i)}(\mathbf{K}_k^{(i)})^\top$

end for

for all $i \in \{1, \ldots, N_\mathrm{p}\}$ **do**

  $w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^{N_\mathrm{p}} \tilde{w}_k^{(i)}$

end for

if Resampling necessary **then**

  Draw N_p particles and their associated Kalman filters

  Set all weights to $w_k^{(i)} = \frac{1}{N_\mathrm{p}}$

end if

return

The covariance of the likelihood is thus equal to the covariance of the Kalman filter innovation and will be denoted by the matrix $\mathbf{S}_k$ in the following. Intuitively this seems to be correct, when the parameters are uncertain the likelihood becomes "wider" and when the calibration parameters are known the width of the likelihood is only defined by the measurement noise, as it would be the case for a perfectly calibrated magnetometer.

With likelihood (5.24) we have all ingredients for the Rao-Blackwellized particle filter that calculates an approximation of the posterior (5.19) of the SLAC problem. The filter is described in detail in Algorithm 5.1. In the Rao-Blackwellized filter the prediction and update steps of the Kalman filters and particle filter are interleaved with each other. After the prediction of the particle and Kalman filter state, first, the particle weight is updated using the predicted density of the parameter vector and second, the Kalman filter is updated to obtain the posterior of the parameter vector. It should be pointed out that instead of estimating the parameter vector naively by a twelve-dimensional Kalman filter per particle, the parameter vector can be also decomposed into three four-dimensional parameter vectors that are estimated in separate Kalman filters. Each of the four-dimensional parameter vectors contains one row of matrix $\bar{\mathbf{C}}$ and the corresponding element of the bias vector $\bar{\mathbf{b}}$ and thus contains the calibration parameters for one of the three magnetometer axes. The decomposition is possible because $\mathbf{H}(\tilde{\mathbf{m}}^{\tilde{b}}(s_k, R_{1:N_{\text{route}}}))$ in (5.14) is a block diagonal matrix, the measurement noise for the magnetometer axes is assumed to be uncorrelated, and the prediction of the different parts of the parameter vector in (5.20) can be performed independently. The main benefit of decomposing the parameter vector into three smaller vectors is a reduction of the computational complexity due to the cubic complexity of the Kalman filter w.r.t. the number of estimated states [91].

5.3.2 Practical Considerations

Previously we discussed that the calibration parameters in the vector $\bar{\boldsymbol{\theta}}$ have different values for the two possible relative orientations between the vehicles frames during mapping and localization. In practice this has to be accounted for when SLAC is used. One possible approach to handle this is to store the estimated calibrations parameters and their covariance when the relative orientation is changed. The change of the relative orientation can be detected and handled by simply storing the train orientation during the mapping in the magnetic map. If the orientation changes for the first time, the Kalman filter state and covariance should be reset to their initial values to allow the parameters to be adapted quickly to the measurements. If the same orientation was already encountered before, the Kalman filter can use the stored estimates of the calibration parameters to continue the estimation. Depending on how well the assumption of constant parameters is fulfilled, one might have to inflate the covariance of the Kalman filters at that point.

The change in the parameters with the relative orientation of the vehicle frames is only encountered when the map is recorded with an uncalibrated magnetometer. If the true magnetic map is used, it is sufficient to estimate the train orientation relative to the track frame. Then, when the train orientation changes, it is sufficient to rotate the map accordingly while the calibration parameters stay the same. However, creating the true magnetic map is not easy, although it is not impossible. One approach would be to mount a calibrated magnetometer on a boom in front or in the back of a train and create the map with these measurements.

5.3.3 Calculation of Point Estimates

For the evaluation of the SLAC approach point estimates of the dynamic train state and the calibration parameters are required. For the dynamic train state the calculation of the MMSEE and the standard deviation is equal to Section 4.3.4. Here therefore only the calculation of the point estimates and standard deviation of the calibration parameters is discussed. For the parameter vector the MMSEE is defined by the integral

$$\begin{aligned}\check{\bar{\boldsymbol{\theta}}}_k &= \int \bar{\boldsymbol{\theta}}_k \, \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{Z}_{1:k})d\bar{\boldsymbol{\theta}}_k \\ &= \int \bar{\boldsymbol{\theta}}_k \left(\int \mathrm{p}(\mathcal{X}^{\mathrm{do}}_{0:k}, \bar{\boldsymbol{\theta}}_k|\mathcal{Z}_{1:k})d\mathcal{X}^{\mathrm{do}}_{0:k} \right) d\bar{\boldsymbol{\theta}}_k \ , \end{aligned} \tag{5.25}$$

where the inner integral is easily calculated by using the decomposition (5.19) and inserting the particle approximation for the position part

$$\begin{aligned}\mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{Z}_{1:k}) &= \int \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}^{\mathrm{do}}_{0:k}, \mathcal{Z}_{1:k})\mathrm{p}(\mathcal{X}^{\mathrm{do}}_{0:k}|\mathcal{Z}_{1:k})d\mathcal{X}^{\mathrm{do}}_{0:k} \\ &= \int \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}^{\mathrm{do}}_{0:k}, \mathcal{Z}_{1:k})\sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \delta(\mathcal{X}^{\mathrm{do}}_{0:k} - \mathcal{X}^{\mathrm{do},(i)}_{0:k})d\mathcal{X}^{\mathrm{do}}_{0:k} \\ &= \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}^{\mathrm{do},(i)}_{0:k}, \mathcal{Z}_{1:k}) \ . \end{aligned} \tag{5.26}$$

Substituting (5.26) into (5.25) gives the MMSEE of the parameter vector

$$\begin{aligned}\check{\bar{\boldsymbol{\theta}}}_k &= \int \bar{\boldsymbol{\theta}}_k \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}^{\mathrm{do},(i)}_{0:k}, \mathcal{Z}_{1:k})d\bar{\boldsymbol{\theta}}_k \\ &= \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \int \bar{\boldsymbol{\theta}}_k \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}^{\mathrm{do},(i)}_{0:k}, \mathcal{Z}_{1:k})d\bar{\boldsymbol{\theta}}_k = \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} \ . \end{aligned} \tag{5.27}$$

The integral can be identified as the posterior mean of the Kalman filter of particle i and the MMSEE estimate is the weighted sum over all Kalman filter means.

The covariance matrix of the parameter vector is calculated by replacing $\bar{\boldsymbol{\theta}}_k$ in (5.27) with the outer product of the vector difference $\bar{\boldsymbol{\theta}}_k - \check{\bar{\boldsymbol{\theta}}}_k$

$$\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}}} = \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \int (\bar{\boldsymbol{\theta}}_k - \check{\bar{\boldsymbol{\theta}}}_k)(\bar{\boldsymbol{\theta}}_k - \check{\bar{\boldsymbol{\theta}}}_k)^\top \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}^{\mathrm{do},(i)}_{0:k}, \mathcal{Z}_{1:k})d\bar{\boldsymbol{\theta}}_k = \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i)} \ . \tag{5.28}$$

For the covariance the result of the integral $\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i)}$ is not just the covariance of the i-th Kalman filter, recognizable by the lack of a $^+$ or $^-$ superscript, because $\check{\bar{\boldsymbol{\theta}}}_k$ is the same for all i and depends on the mean of all N_p Kalman filters. However, with a simple trick the integral in (5.28) can be calculated in a straightforward manner

$$\begin{aligned}
\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i)} =& \int (\bar{\boldsymbol{\theta}}_k - \check{\bar{\boldsymbol{\theta}}}_k + \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+})(\bar{\boldsymbol{\theta}}_k - \check{\bar{\boldsymbol{\theta}}}_k + \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+})^\top \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}_{0:k}^{\mathrm{do},(i)}, \mathcal{Z}_{1:k}) d\bar{\boldsymbol{\theta}}_k \\
=& \int (\bar{\boldsymbol{\theta}}_k - \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+})(\bar{\boldsymbol{\theta}}_k - \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+})^\top \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}_{0:k}^{\mathrm{do},(i)}, \mathcal{Z}_{1:k}) d\bar{\boldsymbol{\theta}}_k \\
&+ \int (\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k)(\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k)^\top \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}_{0:k}^{\mathrm{do},(i)}, \mathcal{Z}_{1:k}) d\bar{\boldsymbol{\theta}}_k \\
&+ \int (\bar{\boldsymbol{\theta}}_k - \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+})(\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k)^\top \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}_{0:k}^{\mathrm{do},(i)}, \mathcal{Z}_{1:k}) d\bar{\boldsymbol{\theta}}_k \\
&+ \int (\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k)(\bar{\boldsymbol{\theta}}_k - \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+})^\top \mathrm{p}(\bar{\boldsymbol{\theta}}_k|\mathcal{X}_{0:k}^{\mathrm{do},(i)}, \mathcal{Z}_{1:k}) d\bar{\boldsymbol{\theta}}_k \\
=& \boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),+} + (\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k)(\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k)^\top .
\end{aligned} \tag{5.29}$$

The integral in the second line is just the Kalman filter covariance. The argument of the expectation in the third line is a constant matrix and therefore the result is just the argument. The last two integrals are both zero because $\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k$ is a constant and the expectation of $\bar{\boldsymbol{\theta}}_k - \boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+}$ is the zero vector.

With (5.29) the overall covariance $\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}}}$ of the calibration parameters from (5.28) becomes

$$\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}}} = \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}},(i),+} + \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} (\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k)(\boldsymbol{\mu}_k^{\bar{\boldsymbol{\theta}},(i),+} - \check{\bar{\boldsymbol{\theta}}}_k)^\top . \tag{5.30}$$

The covariance in (5.30) is not only the weighted sum over the Kalman filter posterior covariance matrices but instead also accounts for the sample covariance of the parameter vector w.r.t. the MMSEE $\check{\bar{\boldsymbol{\theta}}}_k$ in its second term. This appears to be plausible, because without the sample covariance part the matrix $\boldsymbol{\Sigma}_k^{\bar{\boldsymbol{\theta}}}$ only depends on how certain the Kalman filters are about their estimates, but it would not take into account that the Kalman filter estimates depend on the particle trajectories that have also an inherent uncertainty.

5.3.4 Reduced SLAC

In the previous section the SLAC algorithm estimated the complete set of calibration parameters. This is not necessary when the soft iron effects are dominated by scale factors in the main diagonal of matrix $\mathbf{C}$. In such a case the estimation of the scale factors and biases are sufficient. When only the scale factors and the biases are considered, calibration matrix $\mathbf{C}$ is diagonal and the parameter vector becomes

$$\boldsymbol{\theta}^{\mathrm{red}} = \begin{bmatrix} [\mathbf{C}]_{11} & [\mathbf{b}]_1 & [\mathbf{C}]_{22} & [\mathbf{b}]_2 & [\mathbf{C}]_{33} & [\mathbf{b}]_3 \end{bmatrix}^\top . \tag{5.31}$$

With the reduced parameter vector, matrix (5.4) that relates the parameters to the measurements is reduced to

$$\mathbf{H}^{\text{red}}(\mathbf{R}^{\text{b}}_{\text{n},k}\mathbf{m}^{\text{n}}) = \begin{bmatrix} \begin{bmatrix} [\mathbf{R}^{\text{b}}_{\text{n},k}\mathbf{m}^{\text{n}}]_1 & 1 \end{bmatrix} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \begin{bmatrix} [\mathbf{R}^{\text{b}}_{\text{n},k}\mathbf{m}^{\text{n}}]_2 & 1 \end{bmatrix} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \begin{bmatrix} [\mathbf{R}^{\text{b}}_{\text{n},k}\mathbf{m}^{\text{n}}]_3 & 1 \end{bmatrix} \end{bmatrix}, \tag{5.32}$$

where $[\mathbf{R}^{\text{b}}_{\text{n},k}\mathbf{m}^{\text{n}}]_i$ is the i-th element of the true magnetic field vector in the body frame.

When SLAC is performed relative to an uncalibrated map, the structure of the calibration matrix $\bar{\mathbf{C}}$ depends on multiple factors that can make it a non-diagonal matrix. This can be seen from (5.13), where $\bar{\mathbf{C}}$ depends on the calibration parameters of the magnetometer used during mapping, the calibration parameters of the magnetometer that is to be localized, and the relative orientation of the body frames between the two magnetometers. A simple example in which $\bar{\mathbf{C}}$ is a diagonal matrix is when $\mathbf{C}$ and $\tilde{\mathbf{C}}$ are diagonal and both magnetometer body frames are aligned, this guarantees that $\mathbf{R}^{\text{b}}_{\tilde{\text{b}},k}$ in (5.16) is also diagonal.

Given $\bar{\mathbf{C}}$ is a diagonal matrix, the SLAC algorithm only has to be altered slightly. Mainly the measurement matrix $\mathbf{H}(\tilde{\mathbf{m}}^{\tilde{\text{b}}}(s_k, R_{1:N_{\text{route}}}))$ in (5.14) has to be replaced with a matrix of the form (5.32). Furthermore, the dimension of the Kalman filters for parameter estimation is reduced. Now three two-dimensional Kalman filters per particle can be used and each filter estimates the scale factor and bias of one magnetometer axis. The reduction in the Kalman filter state dimension reduces the computational complexity compared to the full SLAC filter. In the evaluation in Chapter 7 the full and the reduced SLAC algorithm are compared against each other to see if the reduced set of calibration parameters has a negative effect on the position accuracy.

5.4 Summary

In the beginning of this chapter, the magnetometer sensor model for calibration was presented and it was discussed under which conditions the calibration parameters are observable and how they can be estimated. Based on the sensor model, it was shown that in the railway domain, under certain assumptions, the magnetometer can be calibrated relative to a magnetic map that itself was recorded with an uncalibrated magnetometer. Performing this relative calibration requires knowledge about the position of the magnetometer. Since the primary goal of this thesis is train localization with the magnetic field, this led to a chicken-or-egg problem. To estimate the train position, we need the calibration parameters and to calibrate the magnetometer we need the position of the train on which it is mounted. To solve this problem, the SLAC algorithm was proposed that estimates the calibration parameters simultaneously with the position. The SLAC algorithm uses a Rao-Blackwellized particle filter that exploits the linear relationship of the calibration parameters and the magnetometer measurements when the estimation is conditioned on the trajectory of a particle. Finally, a reduced version of the SLAC algorithm was discussed that estimates only a scale factor and a bias for each magnetometer axis to further reduce the computational complexity. The proposed algorithms will be evaluated in Chapter 7 based on measurement data of two different trains.

CHAPTER 6

Bayesian Cramér-Rao Lower Bound for Magnetic Localization

In the previous chapters the focus was on the development of concrete filter algorithms for magnetic field-based localization. In this chapter, we now look at magnetic localization from a more theoretical point of view and analyze its fundamental properties by deriving a theoretical lower bound on the achievable position accuracy. In particular, the Bayesian version of the well-known Cramér-Rao lower bound will be derived. A lower bound has the benefit that it is universally valid and does not depend on a certain filter implementation. Such a bound therefore is a benchmark for any filter algorithm and gives insights about the achievable accuracy for a certain magnetic field. This chapter is largely based on our two papers [J3] and [C8], in which the BCRLB was derived for magnetic indoor localization of a robot and for magnetic train localization, respectively. As in the rest of this thesis, the focus of this chapter lies on train localization.

The derivation of a BCRLB requires an analytical description of the measurement model, the corresponding likelihood, and its derivatives. For magnetic train localization the measurement model contains the magnetic field map of the track network. In general, there exists no analytical expression for the map because the magnetic field along a railway track is the result of the interaction of the Earth magnetic field with the magnetic material in the environment. Therefore, to calculate the resulting magnetic field the environment and the properties of the material in it has to be known. However, even with perfect knowledge of the environment, it most likely will not be possible to get a closed form expression of the magnetic field due to the complex interaction of the Earth magnetic field with the surrounding material. This chapter therefore follows a different approach based on the idea presented in [51], where the measurement model is described with a GP that is trained on a given data set. For the case of magnetic train localization, the data set consists of magnetometer measurements and the along-track positions at which the measurements were performed. The advantage of the GP approach is that the trained GP is differentiable, it accounts for uncertainties, and nicely fits into the theoretical framework of the BCRLB due to the Bayesian treatment of the data. Furthermore, it has been shown in multiple papers [33,40,43,92,93] that GPs are a suitable model to represent magnetic field maps.

6.1 BCRLB for Measurement-based Likelihoods

Before the GP is integrated into the BCRLB, we will first introduce the equations of the BCRLB for likelihoods that depend on a measured data set. This requires only minor modifications to the bound but is necessary to properly account for the dependency of the bound on the data set. Furthermore, the equations derived in the following also account for motion models that depend on deterministic inputs.

Using the notation from Section 2.5, $\mathcal{D}$ is a data set that contains pairs of input locations and the corresponding measurements. For the example at hand, the input locations are along-track positions and the observations are measured magnetic field vectors. As mentioned in the previous paragraph, a sequence of deterministic inputs $\mathbf{u}_{0:k}$ is considered as well in the bound. The reason for including inputs will be motivated in Section 6.4.1. The input sequence is combined with the training data $\mathcal{D}$ in the set $\mathcal{D}_u = \{\mathcal{D}, \mathbf{u}_{0:k}\}$. The BCRLB conditioned on $\mathcal{D}_u$ is given by

$$\mathbf{M} = \mathbf{E}_{\mathbf{x},\mathbf{z}|\mathcal{D}_u} \left[(\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x})(\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x})^\top\right] \geq \mathbf{J}^{-1} , \tag{6.1}$$

where the BIM is defined as

$$\mathbf{J} = -\,\mathbf{E}_{\mathbf{x},\mathbf{z}|\mathcal{D}_u} \left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{x}, \mathbf{z}|\mathcal{D}_u)\right] . \tag{6.2}$$

The derivation of the BCRLB in (6.1) and (6.2) is not shown here because it is identical to the derivation of the standard BCRLB, that can be found in [68], but with the joint density $\mathrm{p}(\mathbf{x}, \mathbf{z})$ replaced with $\mathrm{p}(\mathbf{x}, \mathbf{z}|\mathcal{D}_u)$. With this substitution for the joint density, the estimator bias from Section 2.6.2 becomes

$$\mathbf{b}(\mathbf{x}) = \int_{-\infty}^{\infty} [\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}]\, \mathrm{p}(\mathbf{z}|\mathbf{x}, \mathcal{D}_u) d\mathbf{z} . \tag{6.3}$$

As regularity conditions now the following limits must hold for each element $[\mathbf{x}]_i$

$$\begin{aligned} \lim_{[\mathbf{x}]_i \to \infty} \mathbf{b}(\mathbf{x})\, \mathrm{p}(\mathbf{x}|\mathcal{D}_u) &= \mathbf{0} \\ \lim_{[\mathbf{x}]_i \to -\infty} \mathbf{b}(\mathbf{x})\, \mathrm{p}(\mathbf{x}|\mathcal{D}_u) &= \mathbf{0} . \end{aligned} \tag{6.4}$$

As for the standard BCRLB, the BIM (6.2) can be separated into two parts

$$\begin{aligned} \mathbf{J} &= \mathbf{E}_{\mathbf{x}|\mathcal{D}_u} \left[\mathbf{E}_{\mathbf{z}|\mathbf{x},\mathcal{D}_u} \left[-\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{z}|\mathbf{x}, \mathcal{D}_u)\right]\right] + \mathbf{E}_{\mathbf{x}|\mathcal{D}_u} \left[-\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{x}|\mathcal{D}_u)\right] \\ &= \mathbf{E}_{\mathbf{x}|\mathcal{D}_u} [\mathbf{F}(\mathbf{x})] + \mathbf{E}_{\mathbf{x}|\mathcal{D}_u} \left[-\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{x}|\mathcal{D}_u)\right] , \end{aligned} \tag{6.5}$$

where the first part accounts for the information in the observations and the second part accounts for prior knowledge. A comparison of the first and second line in (6.5) reveals that the FIM is now also a function of the combined data and input set $\mathcal{D}_u$

$$\mathbf{F}(\mathbf{x}) = \mathbf{E}_{\mathbf{z}|\mathbf{x},\mathcal{D}_u} \left[-\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \mathrm{p}(\mathbf{z}|\mathbf{x}, \mathcal{D}_u)\right] . \tag{6.6}$$

The FIM requires the density $\mathrm{p}(\mathbf{z}|\mathbf{x}, \mathcal{D}_u)$, which has the same form as the density (2.50) used for GP regression in Section 2.5. Therefore, GPs integrate seamlessly into the BCRLB.

6.2 BCRLB for Filtering

To find the BCRLB for magnetic train localization, it is helpful to first derive the generic bound for filtering problems. In a filtering problem the estimated state $\mathbf{x}$ is typically time variant. To obtain a bound for the state at time step $k+1$ we can naively implement the BCRLB from the previous section for the whole state sequence up to time step $k+1$ and then look at the corresponding part of the inverse BIM. With (6.2) the BIM for the state sequence $\mathbf{x}_{0:k+1}$ is given by

$$\mathbf{J}_{0:k+1} = -\,\mathbf{E}_{\mathbf{x}_{0:k+1},\mathbf{z}_{1:k+1}|\mathcal{D}_u}\left[\Delta^{\mathbf{x}_{0:k+1}}_{\mathbf{x}_{0:k+1}} \ln \mathrm{p}(\mathbf{x}_{0:k+1},\mathbf{z}_{1:k+1}|\mathcal{D}_u)\right] \;, \tag{6.7}$$

where the state sequence is interpreted as a long vector $\mathbf{x}_{0:k+1} = [\mathbf{x}_0^\top \cdots \mathbf{x}_{k+1}^\top]^\top$, in which the state vectors over all time steps are stacked on top of each other. In this naive formulation of the BCRLB for filtering problems, the dimension of the BIM is $N_\mathbf{x}(k+2) \times N_\mathbf{x}(k+2)$ and increases with the amount of considered time steps resulting in a high computational complexity for long state sequences. For the following derivations it is useful to introduce a compact notation, in which the joint density is denoted by

$$\mathrm{p}_{k+1} = \mathrm{p}(\mathbf{x}_{0:k+1},\mathbf{z}_{1:k+1}|\mathcal{D}_u) \;, \tag{6.8}$$

and the expected value w.r.t. p_{k+1} is represented by the operator $\mathbf{E}_{\mathrm{p}_{k+1}}[\cdot]$.

The authors in [94] address the complexity issue by deriving a recursive version of the bound, in which for each time step only the BIM of a single state vector with dimension $N_\mathbf{x} \times N_\mathbf{x}$ is calculated. The following derivation of the recursive bound follows the derivation given in [94]. The derivation starts by dividing the state vector into two parts $\mathbf{x}_{0:k+1} = [\mathbf{x}_{0:k}^\top \quad \mathbf{x}_{k+1}^\top]^\top$. With the divided state vector, the differential operator in (6.7) can be formulated as a block matrix

$$\Delta^{\mathbf{x}_{0:k+1}}_{\mathbf{x}_{0:k+1}} = \begin{bmatrix} \nabla_{\mathbf{x}_{0:k}} \\ \nabla_{\mathbf{x}_{k+1}} \end{bmatrix} \begin{bmatrix} \nabla^\top_{\mathbf{x}_{0:k}} & \nabla^\top_{\mathbf{x}_{k+1}} \end{bmatrix} = \begin{bmatrix} \Delta^{\mathbf{x}_{0:k}}_{\mathbf{x}_{0:k}} & \Delta^{\mathbf{x}_{k+1}}_{\mathbf{x}_{0:k}} \\ (\Delta^{\mathbf{x}_{k+1}}_{\mathbf{x}_{0:k}})^\top & \Delta^{\mathbf{x}_{k+1}}_{\mathbf{x}_{k+1}} \end{bmatrix} . \tag{6.9}$$

With (6.9) and (6.7) the BIM for $\mathbf{x}_{0:k+1}$ becomes a block matrix of the form

$$\mathbf{J}_{0:k+1} = -\,\mathbf{E}_{\mathrm{p}_{k+1}}\left[\begin{bmatrix} \Delta^{\mathbf{x}_{0:k}}_{\mathbf{x}_{0:k}} & \Delta^{\mathbf{x}_{k+1}}_{\mathbf{x}_{0:k}} \\ (\Delta^{\mathbf{x}_{k+1}}_{\mathbf{x}_{0:k}})^\top & \Delta^{\mathbf{x}_{k+1}}_{\mathbf{x}_{k+1}} \end{bmatrix} \ln \mathrm{p}_{k+1}\right] = \begin{bmatrix} \mathbf{A}_{k+1} & \mathbf{B}_{k+1} \\ \mathbf{B}^\top_{k+1} & \mathbf{C}_{k+1} \end{bmatrix} . \tag{6.10}$$

The BCRLB for the state at time step $k+1$ corresponds to the matrix with dimension $N_\mathbf{x} \times N_\mathbf{x}$ in the lower right corner of the inverse BIM $\mathbf{J}^{-1}_{0:k+1}$. This matrix is given by the inverse Schur complement of the upper left block $\mathbf{A}_{k+1}$ of matrix (6.10)

$$\mathbf{J}^{-1}_{k+1} = \left(\mathbf{C}_{k+1} - \mathbf{B}^\top_{k+1}\mathbf{A}^{-1}_{k+1}\mathbf{B}_{k+1}\right)^{-1} \;, \tag{6.11}$$

where $\mathbf{J}_{k+1}$ is the BIM for $\mathbf{x}_{k+1}$. So far (6.11) did not result in a considerable reduction of the computational complexity because in the Schur complement still the matrix $\mathbf{A}_{k+1}$ with dimension $(k+1)N_\mathbf{x} \times (k+1)N_\mathbf{x}$ is inverted. To understand how the dimension of matrices involved in the Schur complement can be reduced, we have to further break down the structure of the different matrices in (6.10).

Let us start with matrix $\mathbf{A}_{k+1}$, this matrix can be structured into four blocks by dividing the differential operator $\Delta_{\mathbf{x}_{0:k}}^{\mathbf{x}_{0:k}}$ analogously to (6.9)

$$\Delta_{\mathbf{x}_{0:k}}^{\mathbf{x}_{0:k}} = \begin{bmatrix} \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} & \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k}} \\ (\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k}})^\top & \Delta_{\mathbf{x}_{k}}^{\mathbf{x}_{k}} \end{bmatrix} . \tag{6.12}$$

With the differential operator in block form (6.12) matrix $\mathbf{A}_{k+1}$ can be written as 2×2 block matrix

$$\mathbf{A}_{k+1} = -\,\mathbf{E}_{\mathrm{p}_{k+1}} \left[\begin{bmatrix} \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} & \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k}} \\ (\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k}})^\top & \Delta_{\mathbf{x}_{k}}^{\mathbf{x}_{k}} \end{bmatrix} \ln \mathrm{p}_{k+1} \right] = \begin{bmatrix} \mathbf{A}_{k+1}^{(11)} & \mathbf{A}_{k+1}^{(12)} \\ (\mathbf{A}_{k+1}^{(12)})^\top & \mathbf{A}_{k+1}^{(22)} \end{bmatrix} . \tag{6.13}$$

Matrix $\mathbf{B}_{k+1}$ is divided into two separate blocks by splitting the differential operator according to $\Delta_{\mathbf{x}_{0:k}}^{\mathbf{x}_{k+1}} = [(\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k+1}})^\top \quad (\Delta_{\mathbf{x}_{k}}^{\mathbf{x}_{k+1}})^\top]^\top$ which results in

$$\mathbf{B}_{k+1} = -\,\mathbf{E}_{\mathrm{p}_{k+1}} \left[\begin{bmatrix} \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k+1}} \\ \Delta_{\mathbf{x}_{k}}^{\mathbf{x}_{k+1}} \end{bmatrix} \ln \mathrm{p}_{k+1} \right] = \begin{bmatrix} \mathbf{B}_{k+1}^{(1)} \\ \mathbf{B}_{k+1}^{(2)} \end{bmatrix} . \tag{6.14}$$

With the block matrices (6.13), (6.14), and the BCRLB in (6.11) the BIM for $\mathbf{x}_{k+1}$ is given by

$$\mathbf{J}_{k+1} = \mathbf{C}_{k+1} - \begin{bmatrix} (\mathbf{B}_{k+1}^{(1)})^\top & (\mathbf{B}_{k+1}^{(2)})^\top \end{bmatrix} \begin{bmatrix} \mathbf{A}_{k+1}^{(11)} & \mathbf{A}_{k+1}^{(12)} \\ (\mathbf{A}_{k+1}^{(12)})^\top & \mathbf{A}_{k+1}^{(22)} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{B}_{k+1}^{(1)} \\ \mathbf{B}_{k+1}^{(2)} \end{bmatrix} . \tag{6.15}$$

To obtain the recursive structure from [94], the upper block $\mathbf{B}_{k+1}^{(1)}$ of matrix $\mathbf{B}_{k+1}$ has to be a zero matrix with appropriate dimensions. This is always true when the dynamic system that describes the behavior of $\mathbf{x}$ is Markovian. This can be easily seen by noting that for a Markovian system the joint density $\mathrm{p}(\mathbf{x}_{0:k+1}, \mathbf{z}_{1:k+1}|\mathcal{D}_u)$ can be be decomposed according to

$$\begin{aligned} \mathrm{p}_{k+1} &= \mathrm{p}(\mathbf{x}_{0:k+1}, \mathbf{z}_{1:k+1}|\mathcal{D}_u) \\ &= \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{0:k+1}, \mathbf{z}_{1:k}, \mathcal{D}_u)\, \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k}, \mathcal{D}_u)\, \mathrm{p}(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}|\mathcal{D}_u) \\ &= \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u)\, \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_{k}, \mathcal{D}_u)\, \mathrm{p}(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}|\mathcal{D}_u) \\ &= \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u)\, \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_{k}, \mathcal{D}_u)\, \mathrm{p}_k \ . \end{aligned} \tag{6.16}$$

With (6.16) and (6.14) the matrix $\mathbf{B}_{k+1}^{(1)}$ can be written as

$$\begin{aligned} \mathbf{B}_{k+1}^{(1)} &= -\,\mathbf{E}_{\mathrm{p}_{k+1}} \left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k+1}} \ln \mathrm{p}_{k+1} \right] \\ &= -\,\mathbf{E}_{\mathrm{p}_{k+1}} \left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k+1}} \left(\ln \left[\mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u) \right] + \ln \left[\mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_{k}, \mathcal{D}_u) \right] + \ln \mathrm{p}_k \right) \right] . \end{aligned} \tag{6.17}$$

In (6.17) the three terms, on which the differential operator is applied, depend either on $\mathbf{x}_{0:k-1}$ or $\mathbf{x}_{k+1}$ but not both and therefore the result is always zero.

According to (6.15), when $\mathbf{B}_{k+1}^{(1)}$ is a zero matrix the equation for the BIM of $\mathbf{x}_{k+1}$ becomes

$$\mathbf{J}_{k+1} = \mathbf{C}_{k+1} - (\mathbf{B}_{k+1}^{(2)})^\top (\mathbf{A}_{k+1}^{-1})^{(22)} \mathbf{B}_{k+1}^{(2)} , \tag{6.18}$$

where $(\mathbf{A}_{k+1}^{-1})^{(22)}$ is the lower right block with dimension $N_\mathbf{x} \times N_\mathbf{x}$ from the inverse of $\mathbf{A}_{k+1}$, which can be calculated once more with a Schur complement

$$(\mathbf{A}_{k+1}^{-1})^{(22)} = \left(\mathbf{A}_{k+1}^{(22)} - (\mathbf{A}_{k+1}^{(12)})^\top (\mathbf{A}_{k+1}^{(11)})^{-1} \mathbf{A}_{k+1}^{(12)} \right)^{-1} . \tag{6.19}$$

Let us take a closer look at the matrix $\mathbf{A}_{k+1}$ and how the Schur complement (6.19) can be calculated efficiently, i.e., without explicitly inverting the $kN_\mathbf{x} \times kN_\mathbf{x}$ matrix $\mathbf{A}_{k+1}^{(11)}$. Due to the Markov property that gives rise to (6.16), matrix $\mathbf{A}_{k+1}$ is equal to the block matrix

$$\mathbf{A}_{k+1} = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^\top & \mathbf{D}_k^{(11)} + \mathbf{C}_k \end{bmatrix} , \tag{6.20}$$

where $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$ are the blocks from $\mathbf{J}_{0:k}$ of the state sequence $\mathbf{x}_{0:k}$ that are defined analogously to (6.10) and $\mathbf{D}_k^{(11)}$ is

$$\mathbf{D}_k^{(11)} = -\,\mathbf{E}_{\mathrm{p}_{k+1}} \left[\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u) \right] . \tag{6.21}$$

A detailed derivation of (6.20) and (6.21) is presented in Section B.1 of Appendix B. With (6.20) an alternative expression for the Schur complement in (6.19) is given by

$$(\mathbf{A}_{k+1}^{-1})^{(22)} = \left(\mathbf{D}_k^{(11)} + \mathbf{C}_k - \mathbf{B}_k^\top \mathbf{A}_k^{-1} \mathbf{B}_k \right)^{-1} . \tag{6.22}$$

A comparison of (6.22) to (6.11) shows that $\mathbf{C}_k - \mathbf{B}_k^\top \mathbf{A}_k^{-1} \mathbf{B}_k$ is just the BIM for the state at time step k and thus (6.22) becomes

$$(\mathbf{A}_{k+1}^{-1})^{(22)} = \left(\mathbf{D}_k^{(11)} + \mathbf{J}_k \right)^{-1} . \tag{6.23}$$

By inserting (6.23) into (6.18) we obtain

$$\mathbf{J}_{k+1} = \mathbf{C}_{k+1} - (\mathbf{B}_{k+1}^{(2)})^\top \left(\mathbf{D}_k^{(11)} + \mathbf{J}_k \right)^{-1} \mathbf{B}_{k+1}^{(2)} , \tag{6.24}$$

and we achieved our goal to find a recursive formulation for the calculation of the BIM and the associated BCRLB for filtering problems. In (6.24), all involved matrices have the dimension $N_\mathbf{x} \times N_\mathbf{x}$ and the computational complexity of the bound grows only linearly with the number of considered time steps. For better comparability to other publications about BCRLBs, in the remainder of the thesis the following, more common, notation from [94] will be used

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{(22)} - (\mathbf{D}_k^{(12)})^\top \left(\mathbf{D}_k^{(11)} + \mathbf{J}_k \right)^{-1} \mathbf{D}_k^{(12)} , \tag{6.25}$$

with the matrices $\mathbf{D}_k^{(22)} = \mathbf{C}_{k+1}$ and $\mathbf{D}_k^{(12)} = \mathbf{B}_{k+1}^{(2)}$.

The recursion in (6.25) must be initialized with the BIM $\mathbf{J}_0$ at time step zero. Since at time step zero no measurements are available, the BIM is retrieved from the prior distribution of the state vector. Once the initial BIM is set, the BIM for all time steps is obtained by applying the recursion (6.25), where the required matrices are the following expected values

$$\begin{aligned}
\mathbf{D}_k^{(11)} &= -\,\mathbf{E}_{\mathbf{x}_{k:k+1}|\mathcal{D}_u}\left[\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u)\right] \\
\mathbf{D}_k^{(12)} &= -\,\mathbf{E}_{\mathbf{x}_{k:k+1}|\mathcal{D}_u}\left[\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u)\right] \\
\mathbf{D}_k^{(22)} &= -\,\mathbf{E}_{\mathbf{x}_{k+1},\mathbf{z}_{k+1}|\mathcal{D}_u}\left[\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u)\right] \\
&\qquad - \mathbf{E}_{\mathbf{x}_{k:k+1}|\mathcal{D}_u}\left[\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u)\right] \ .
\end{aligned} \tag{6.26}$$

The matrices $\mathbf{D}_k^{(12)}$ and $\mathbf{D}_k^{(22)}$ are derived in a straightforward way by using the decomposition (6.16) and the definitions of $\mathbf{B}_{k+1}^{(2)}$ in (6.14) and $\mathbf{C}_{k+1}$ in (6.10). The matrix $\mathbf{D}_k^{(11)}$ is derived in Section B.1 of the appendix and the result is given by (B.4) and (6.21). In (6.26) the expected values are taken w.r.t. the joint density p_{k+1}, but because the arguments only dependent on states and observations at some time steps the expected value is calculated only over the relevant random variables. The remaining variables are simply marginalized from the joint pdf during the integration in the expected values.

For the implementation of the BCRLB it is useful to note that $\mathbf{D}_k^{(22)}$ is connected to the FIM. The connection is revealed by rewriting $\mathbf{D}_k^{(22)}$ to

$$\begin{aligned}
\mathbf{D}_k^{(22)} &= \mathbf{E}_{\mathbf{x}_{k+1}|\mathcal{D}_u}\left[-\,\mathbf{E}_{\mathbf{z}_{k+1}|\mathbf{x}_{k+1},\mathcal{D}_u}\left[\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u)\right]\right] \\
&\qquad - \mathbf{E}_{\mathbf{x}_{k:k+1}|\mathcal{D}_u}\left[\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u)\right] \\
&= \mathbf{E}_{\mathbf{x}_{k+1}|\mathcal{D}_u}\left[\mathbf{F}(\mathbf{x}_{k+1})\right] - \mathbf{E}_{\mathbf{x}_{k:k+1}|\mathcal{D}_u}\left[\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u)\right] \ ,
\end{aligned} \tag{6.27}$$

where

$$\mathbf{F}(\mathbf{x}_{k+1}) = -\,\mathbf{E}_{\mathbf{z}_{k+1}|\mathbf{x}_{k+1},\mathcal{D}_u}\left[\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u)\right] \tag{6.28}$$

is the FIM conditioned on the data set. Comparable to the non-recursive version of the BCRLB, the recursive version requires the calculation of the expected value of the FIM over all possible state values for the time step under consideration. Calculating the integral in the expected value is intractable for most problems. In practice, the expected value is therefore determined numerically by a Monte Carlo integration based on simulating the dynamic system that governs the temporal evolution of the state vector.

6.3 FIM for Gaussian Process Measurement Models

Until now, the recursive BCRLB has only assumed the Markov property and that the required gradients and expectations of the different pdfs exist. In this section, we will now introduce more restrictions on the measurement model. In the following, we will focus on measurement models of the form

$$z_k = \mathrm{f}(\mathbf{x}_k) + n_k \ , \tag{6.29}$$

where the measurement function $\mathrm{f}(\mathbf{x}_k)$ is a GP and n_k is white Gaussian noise with the density $n_k \sim \mathcal{N}(0, (\sigma^z)^2)$. Note, here only scalar measurement functions are considered that map the state vector $\mathbf{x} \in \mathbb{R}^{N_\mathbf{x}}$ to a scalar measurement $z \in \mathbb{R}$. To represent vector valued observations, a different GP for each component will be used and it is assumed that the components are independent from each other. The GP in (6.29) is trained, as described in Section 2.5, based on a data set $\mathcal{D}$. The data set $\mathcal{D} = \{\mathcal{X}^\mathcal{D}, \mathcal{Z}^\mathcal{D}\}$ contains M noisy observations $\mathcal{Z}^\mathcal{D}$ of the true but unknown measurement function and the input locations $\mathcal{X}^\mathcal{D}$ at which the observations were made. Here, the elements $\mathbf{x}^{\mathcal{D},(i)}$ with $i \in \{1, \ldots, M\}$ in the input training data set $\mathcal{X}^\mathcal{D}$ are state vectors $\mathbf{x}$ from the same state space for which the BCRLB is derived.

The task of the GP is to bridge the gap between the real-world and theory by giving a smooth representation of the unknown measurement function. There are many possible approximation schemes that could fulfill this task, but we follow the approach proposed in [51] and use a GP that not only approximates the function value but also assigns an uncertainty to each value. The uncertainty accounts for the fact that we will never have perfect knowledge about the magnetic field and thus this should be reflected also in the bound. In the following, we will further show that the GP posterior mean and variance are smooth and differentiable and therefore fit particularly well into the BCRLB framework.

For the recursive BCRLB given by (6.25) and (6.26) the measurement model is only relevant for the FIM (6.28). The calculation of the FIM for models of the form (6.29) is a straightforward process. Since $\mathrm{f}(\mathbf{x}_k)$ is a GP, noiseless observations of the function follow the Gaussian posterior density

$$\mathrm{p}(\mathrm{f}(\mathbf{x}_k)|\mathbf{x}_k, \mathcal{D}) = \mathcal{N}\left(\mathrm{f}(\mathbf{x}_k); \mu(\mathbf{x}_k, \mathcal{D}), \mathrm{r}(\mathbf{x}_k, \mathcal{D})\right) \;, \tag{6.30}$$

where the posterior mean $\mu(\mathbf{x}_k, \mathcal{D})$ and the variance $\mathrm{r}(\mathbf{x}_k, \mathcal{D})$ are scalar valued functions of the state vector and the training data set. Note, here we use $\mathcal{D}$ instead of $\mathcal{D}_u$ because the GP is not a function of the control input $\mathbf{u}$. As elaborated in Section 2.5, the posterior mean and variance of the GP are

$$\begin{aligned}
\mu(\mathbf{x}_k, \mathcal{D}) &= \mathrm{m}(\mathbf{x}_k) + \mathbf{K}_{\mathbf{x}_k\mathcal{D}} \left(\mathbf{K}_{\mathcal{D}\mathcal{D}} + (\sigma^z)^2 \mathbf{I}_{M\times M}\right)^{-1} \left(\mathbf{z}^\mathcal{D}_{1:M} - \mathbf{m}^\mathcal{D}\right) \\
r(\mathbf{x}_k, \mathcal{D}) &= \mathrm{k}(\mathbf{x}_k, \mathbf{x}_k) - \mathbf{K}_{\mathbf{x}_k\mathcal{D}} \left(\mathbf{K}_{\mathcal{D}\mathcal{D}} + (\sigma^z)^2 \mathbf{I}_{M\times M}\right)^{-1} \mathbf{K}_{\mathcal{D}\mathbf{x}_k} \;,
\end{aligned} \tag{6.31}$$

where $\mathbf{z}^\mathcal{D}_{1:M}$ is interpreted as a long vector of the staked observations in $\mathcal{Z}^\mathcal{D}$ and $\mathbf{m}^\mathcal{D}$ is a vector that contains the values of the mean function $\mathrm{m}(\cdot)$ evaluated at the corresponding state vectors in $\mathcal{X}^\mathcal{D}$. The covariance matrices $\mathbf{K}_{\mathcal{D}\mathcal{D}}$ and $\mathbf{K}_{\mathbf{x}_k\mathcal{D}} = \mathbf{K}^\top_{\mathcal{D}\mathbf{x}_k}$ use the short hand notation $\mathbf{K}_{\mathbf{ab}} = \mathbf{K}(\mathbf{a}, \mathbf{b})$ and are calculated according to (2.49).

The measurement noise in (6.29) is considered to be white and independent from the GP, thus the noisy measurements are the sum of two independent Gaussian random variables and we can write for their likelihood

$$\begin{aligned}
\mathrm{p}(z_k|\mathbf{x}_k, \mathcal{D}) &= \mathcal{N}\left(z_k; \mu(\mathbf{x}_k, \mathcal{D}), \mathrm{r}(\mathbf{x}_k, \mathcal{D}) + (\sigma^z)^2\right) \\
&= \mathcal{N}\left(z_k; \mu(\mathbf{x}_k, \mathcal{D}), \mathrm{r}^\mathrm{c}(\mathbf{x}_k, \mathcal{D})\right) \;.
\end{aligned} \tag{6.32}$$

Adding the noise on the GP effectively increases the uncertainty and the overall variance is the sum of the GP posterior variance at $\mathbf{x}_k$ and the measurement noise variance.

Because the likelihood (6.32) is Gaussian, the FIM can be obtained by inserting (6.32) into the FIM for general Gaussian densities (2.57), which yields

$$
\begin{aligned}
[\mathbf{F}(\mathbf{x}_k)]_{ij} =& \frac{\partial \mu(\mathbf{x}_k, \mathcal{D})}{\partial [\mathbf{x}_k]_i} (\mathrm{r}^{\mathrm{c}}(\mathbf{x}_k, \mathcal{D}))^{-1} \frac{\partial \mu(\mathbf{x}_k, \mathcal{D})}{\partial [\mathbf{x}_k]_j} \\
&+ \frac{1}{2} (\mathrm{r}^{\mathrm{c}}(\mathbf{x}_k, \mathcal{D}))^{-2} \frac{\partial \mathrm{r}^{\mathrm{c}}(\mathbf{x}_k, \mathcal{D})}{\partial [\mathbf{x}_k]_i} \frac{\partial \mathrm{r}^{\mathrm{c}}(\mathbf{x}_k, \mathcal{D})}{\partial [\mathbf{x}_k]_j} .
\end{aligned} \tag{6.33}
$$

Finding the FIM thus only requires the calculation of the partial derivatives of the posterior mean and covariance from (6.31) w.r.t. the components of the state vector. The partial derivatives can be written in a compact form

$$
\begin{aligned}
\frac{\partial \mu(\mathbf{x}_k, \mathcal{D})}{\partial [\mathbf{x}_k]_i} &= \frac{\partial \mathrm{m}(\mathbf{x}_k)}{\partial [\mathbf{x}_k]_i} + \sum_{l=1}^{M} \left[\mathbf{S}^{-1}(\mathbf{z}^{\mathcal{D}}_{1:M} - \mathbf{m}^{\mathcal{D}})\right]_l \frac{\partial \mathrm{k}(\mathbf{x}_k, \mathbf{x}^{\mathcal{D},(l)})}{\partial [\mathbf{x}_k]_i} \\
\frac{\partial \mathrm{r}^{\mathrm{c}}(\mathbf{x}_k, \mathcal{D})}{\partial [\mathbf{x}_k]_i} &= \frac{\partial \mathrm{k}(\mathbf{x}_k, \mathbf{x}_k)}{\partial [\mathbf{x}_k]_i} - 2 \left(\frac{\partial \mathbf{K}_{\mathbf{x}_k \mathcal{D}}}{\partial [\mathbf{x}_k]_i} \right) \mathbf{S}^{-1} \mathbf{K}_{\mathcal{D} \mathbf{x}_k} ,
\end{aligned} \tag{6.34}
$$

where $\mathbf{S} = \mathbf{K}_{\mathcal{D}\mathcal{D}} + (\sigma^z)^2 \mathbf{I}_{M \times M}$. A detailed derivation of (6.34) is given in Section B.2 of Appendix B. The equations for a concrete choice of the mean and covariance function is easily obtained by substitution of the corresponding derivatives into (6.34).

From (6.33) we can see that the FIM increases with the derivative of the GP mean. For magnetic localization this means that the achievable position accuracy depends on the spatial variability of the magnetic field. A higher spatial variability will lead to a high accuracy while a flat magnetic field will not provide any information.

Before we proceed and apply the BCRLB on magnetic train localization, we want to give a final note about the integration of the GP into the BCRLB. In the recursive BCRLB (6.25) we assumed that the measurement model has the Markov property. For measurement models with a GP that implies the GP is trained offline and the observations used for localization are not used to also update the GP. It seems possible to derive the BCRLB for an online updated GP, but then the recursive form of the bound most likely can no longer be used. Here we are mainly interested in the localization performance for a given map, thus we limited our investigation on the bound with an offline trained GP.

6.4 Concrete Equations for Magnetic Train Localization

With the BCRLB equations in their generic form we can begin to derive the concrete equations of the bound for a specific motion model and specific mean and covariance functions of the GP. In the bound, we will consider only calibrated magnetometers and the case when the train drives on a known route. Furthermore, nonlinear effects on the border of the virtual track constructed from the route are avoided by using a route that is longer than the distance traveled by the train, i.e., for the bound it looks like the train is running on an infinitely long track.

6.4.1 Motion Model

In the BCRLB the motion of the train is accounted for by a simple linear motion model. Similar to the description of the localization filters in Chapter 4, the state for which the bound is derived is called dynamic train state and will be denoted here by $\mathbf{d}_k^{\mathrm{B}}$. In contrast to (4.7), the dynamic state in the bound only contains the train position and speed but not the acceleration

$$\mathbf{d}_k^{\mathrm{B}} = \begin{bmatrix} s_k & \dot{s}_k \end{bmatrix}^\top . \tag{6.35}$$

In the bound the acceleration is instead considered as a deterministic input. This yields the motion model

$$\mathbf{d}_k^{\mathrm{B}} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{\mathbf{F}^{\mathrm{B}}} \mathbf{d}_{k-1}^{\mathrm{B}} + \underbrace{\begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix}}_{\mathbf{B}^{\mathrm{B}}} a_{k-1} + \mathbf{w}_{k-1}^{\mathrm{B}} , \tag{6.36}$$

with the Gaussian process noise $\mathbf{w}_{k-1}^{\mathrm{B}} \sim \mathcal{N}(\mathbf{0}_{2\times1}, \mathbf{Q}^{\mathrm{B}})$. The covariance matrix of the noise is set according to the continuous white noise acceleration model from [74]

$$\mathbf{Q}^{\mathrm{B}} = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix} q^{\mathrm{B}} , \tag{6.37}$$

where q^{B} is a tuning parameter representing the power spectral density of a continuous time noise that is used to model the uncertainty in the evolution of the train speed. Removing the acceleration from the state and using it as an input gives us more control over the train trajectory and therefore over the bound based upon it. With the input and a small enough process noise the train can be kept within the boundaries of the virtual track. This avoids that the train reaches parts of the track where the GP has no training data and where it performs only an extrapolation with a high variance.

With the motion model from (6.36) the matrices (6.26) of the BCRLB become

$$\begin{aligned} \mathbf{D}_k^{(11)} &= (\mathbf{F}^{\mathrm{B}})^\top (\mathbf{Q}^{\mathrm{B}})^{-1} \mathbf{F}^{\mathrm{B}} \\ \mathbf{D}_k^{(12)} &= -(\mathbf{F}^{\mathrm{B}})^\top (\mathbf{Q}^{\mathrm{B}})^{-1} = (\mathbf{D}_k^{(21)})^\top \\ \mathbf{D}_k^{(22)} &= \mathbf{E}_{\mathbf{d}_{k+1}^{\mathrm{B}} | \mathcal{D}_u} \left[\mathbf{F}(\mathbf{d}_{k+1}^{\mathrm{B}}) \right] + (\mathbf{Q}^{\mathrm{B}})^{-1} . \end{aligned} \tag{6.38}$$

The different matrices depend only on the system matrix $\mathbf{F}^{\mathrm{B}}$ of the motion model, the process noise covariance $\mathbf{Q}^{\mathrm{B}}$, and the expected FIM. A detailed derivation of these matrices is given in Appendix B.3.

6.4.2 Measurement Model

The last missing piece for the calculation of the BCRLB for magnetic-field based train localization is the FIM. For the FIM we use a simplified version of the measurements model (4.14) and only consider the case of a known route. In addition, the orientation of the train is assumed to be known.

The influence of an unknown orientation on the localization accuracy is limited to a short duration at the beginning of the train run because the orientation is typically estimated quickly and then stays constant. Furthermore, the orientation is a binary variable that does not fit particularly well into the concept of the BCRLB, which requires derivatives of the likelihood and motion model. Thus, we simplified (4.14) to

$$\mathbf{z}_k^{\mathrm{mag}} = \mathbf{m}^{\mathrm{t}}(s_k) + \mathbf{n}_k^{\mathrm{mag}} \,, \tag{6.39}$$

where the orientation is set to $\mathbf{R}_{\mathrm{t}}^{\mathrm{b}}(O_k) = \mathbf{I}_{3\times 3}$ and the measurement noise is AWGN with the pdf $\mathbf{n}_k^{\mathrm{mag}} \sim \mathcal{N}(\mathbf{0}_{3\times 1}, (\sigma^{\mathrm{mag}})^2 \mathbf{I}_{3\times 3})$. Since the bound is only calculated for known routes, the dependency on the route is dropped for brevity.

The magnetic map $\mathbf{m}^{\mathrm{t}}(s_k)$ relates the along-track position s_k to the magnetic vector field at that position. For the FIM calculation each component of the magnetic vector field is represented by a separate independent function

$$\mathbf{m}^{\mathrm{t}}(s_k) = \begin{bmatrix} \mathrm{m}^{\mathrm{t,x}}(s_k) & \mathrm{m}^{\mathrm{t,y}}(s_k) & \mathrm{m}^{\mathrm{t,z}}(s_k) \end{bmatrix}^\top \,. \tag{6.40}$$

Here, the functions $\mathrm{m}^{\mathrm{t},i}(s_k), i \in \{\mathrm{x}, \mathrm{y}, \mathrm{z}\}$, are GPs that are trained offline on a data set $\mathcal{D}$ recorded on the considered route. From (6.39) and (6.40) it follows that the likelihood of a measurement $\mathbf{z}_k^{\mathrm{mag}}$ is the multivariate Gaussian density

$$\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}} | \mathbf{d}_k^{\mathrm{B}}, \mathcal{D}) = \mathcal{N}\left(\mathbf{z}_k^{\mathrm{mag}}; \boldsymbol{\mu}^{\mathbf{m}^{\mathrm{t}}}(s_k, \mathcal{D}), \boldsymbol{\Sigma}^{\mathbf{m}^{\mathrm{t}}}(s_k, \mathcal{D}) + (\sigma^{\mathrm{mag}})^2 \mathbf{I}_{3\times 3}\right) \,, \tag{6.41}$$

where the mean vector contains the posterior means of the GPs

$$\boldsymbol{\mu}^{\mathbf{m}^{\mathrm{t}}}(s_k, \mathcal{D}) = \begin{bmatrix} \mu^{\mathrm{x}}(s_k, \mathcal{D}) & \mu^{\mathrm{y}}(s_k, \mathcal{D}) & \mu^{\mathrm{z}}(s_k, \mathcal{D}) \end{bmatrix}^\top \,. \tag{6.42}$$

The covariance matrix of the likelihood is the sum of the measurement noise covariance and the diagonal matrix

$$\boldsymbol{\Sigma}^{\mathbf{m}^{\mathrm{t}}}(s_k, \mathcal{D}) = \mathrm{diag}\left(\begin{bmatrix} \mathrm{r}^{\mathrm{x}}(s_k, \mathcal{D}) & \mathrm{r}^{\mathrm{y}}(s_k, \mathcal{D}) & \mathrm{r}^{\mathrm{z}}(s_k, \mathcal{D}) \end{bmatrix}\right) \,, \tag{6.43}$$

where $\mathrm{r}^i(s_k, \mathcal{D}), i \in \{\mathrm{x}, \mathrm{y}, \mathrm{z}\}$, are the posterior variances of the GPs for the different components of the magnetic map. The measurement noise covariance and the covariance matrix of the map are both diagonal, hence the three components of the measured magnetic field vector $\mathbf{z}_k^{\mathrm{mag}}$ are assumed to be independent of each other. Independence allows us to write the likelihood of the vector measurement as the product of the likelihoods of the three vector components

$$\mathrm{p}(\mathbf{z}_k^{\mathrm{mag}} | \mathbf{d}_k^{\mathrm{B}}, \mathcal{D}) = \prod_{i \in \{\mathrm{x,y,z}\}} \mathcal{N}([\mathbf{z}_k^{\mathrm{mag}}]_i; \mu^i(s_k, \mathcal{D}), \underbrace{\mathrm{r}^i(s_k, \mathcal{D}) + (\sigma^{\mathrm{mag}})^2}_{\mathrm{r}^{\mathrm{c},i}(s_k, \mathcal{D})}) \,. \tag{6.44}$$

Writing the likelihood in product form simplifies the calculation of the FIM. By inserting the product (6.44) into the FIM (B.18), we can see that because of the logarithm the FIM for each factor of the product can be calculated separately and then the results can be summed up instead of directly calculating the FIM for the vector measurement. For each factor the FIM is straightforwardly found from (6.33) and (6.34) in Section 6.3 by substituting the derivative of the covariance and mean function with the derivative of the chosen functions. In this thesis, the mean function is simply a constant function

$$\mathrm{m}(\mathbf{a}) = \bar{m} \, , \tag{6.45}$$

and the covariance function is the well-known squared exponential kernel

$$\mathrm{k}(\mathbf{a}, \mathbf{u}) = (\sigma^{\mathrm{k}})^2 \exp\left[-\frac{1}{2\lambda^2}(\mathbf{a}-\mathbf{u})^\top(\mathbf{a}-\mathbf{u})\right] \, , \tag{6.46}$$

where $\mathbf{a} \in \mathbb{R}^{N_\mathbf{a}}$ and $\mathbf{u} \in \mathbb{R}^{N_\mathbf{a}}$ are two input vectors. For the particular example of train localization, the input of the GP, and therefore also of the mean and covariance functions, is not a vector but the scalar along-track position of the train.

The derivative of the constant mean function is always zero. For the squared exponential kernel, the derivative w.r.t. the along-track position s_k is

$$\frac{\partial \mathrm{k}(s_k, s^{\mathcal{D},(l)})}{\partial s_k} = -\frac{1}{\lambda^2}\,\mathrm{k}(s_k, s^{\mathcal{D},(l)})(s_k - s^{\mathcal{D},(l)}) \, , \tag{6.47}$$

where $s^{\mathcal{D},(l)}$ is the along-track position of the l-th measurement in the data set $\mathcal{D}$ on which the GP is based. In (6.34) also the derivative of $\mathrm{k}(s_k, s_k)$ is required. This is always zero for the squared exponential kernel because when both its arguments have the same value it is a constant function. Note, since the likelihood only depends on the position s_k, its derivative and thus the FIM for the train speed $\dot{s}_k$ is always zero.

6.5 Summary

In this chapter the BCRLB for likelihoods that depend on a measured data set was introduced. This only required minor modifications of the BCRLB from Section 2.6.2, but was needed to properly account for the dependency of the bound on the data set. In addition, the modified bound considers a deterministic control input for the motion model. The control input allows to influence the trajectory of the train for which the bound is evaluated. For the modified bound then a recursive version was introduced that enables a computational efficient implementation of the bound for filtering problems. Furthermore, it was shown how measurement models that include a GP can be incorporated into the bound. Finally, the concrete recursive BCRLB equations for magnetic field-based train localization were derived.

Chapter 7

Experimental Evaluation

In this chapter, the algorithms introduced in the previous chapters are evaluated based on measurement data. The chapter starts in Section 7.1 with a description of the measurement campaigns that were carried out to collect the data for the evaluation. This is followed by an evaluation of the different localization filters from Chapter 4 in Section 7.2. In Section 7.3, it is shown that by using the SLAC algorithm from Chapter 5 localization is possible also with uncalibrated magnetometers. Section 7.4 concludes the chapter with the application of the BCRLB to measurement data to gain an impression of the theoretically achievable accuracy of magnetic field-based train localization.

7.1 Measurement Campaigns

For the evaluation the data sets of two separate measurement campaigns are used. In the first campaign, the measurement hardware was mounted on a single train during regular service. This allowed us to collect data during multiple runs in the same track network. During these runs the train passed the same switches multiple times, which will be used to evaluate the performance of the switch way identification of the localization filters. In the second campaign, the magnetic field was measured with two types of trains traveling on a different track network than during the first campaign. The data from the second campaign allows us to evaluate the SLAC approach based on real-world data.

7.1.1 Bayerische Regiobahn

The first measurement campaign was conducted by DLR with the support of the Bayerische Regiobahn (BRB). The BRB is a train operator that operates local trains in Bavaria, Germany. In this campaign the measurement hardware was mounted on an Alstom Coradia Lint41 train driving between the cities Augsburg and Ingolstadt. The Coradia Lint41 is a diesel powered train that drove up to $\approx 120\,\mathrm{km\,h^{-1}}$ during the measurements. The train and the sensor setup is shown in Figure 7.1. For the recording a sensor box was mounted inside a cabinet in the driver cab. The magnetic field and the train acceleration was measured with a Xsens MTi nine degree of freedom IMU at an rate of 200 Hz. The magnetometer data is internally normalized by the sensor and hence is dimensionless as described already in the introduction. To emphasize this, each plot showing the measured magnetometer data has the unit a.u. for arbitrary units. To relate this to a physical value, one a.u. can be interpreted as roughly $40\,\mu\mathrm{T}$. The body frame of the Xsens is not aligned with the vehicle frame. Thus, we rotated the sensor data into the vehicle frame to fulfill the assumption from Chapter 4 that considers these frames to be aligned.

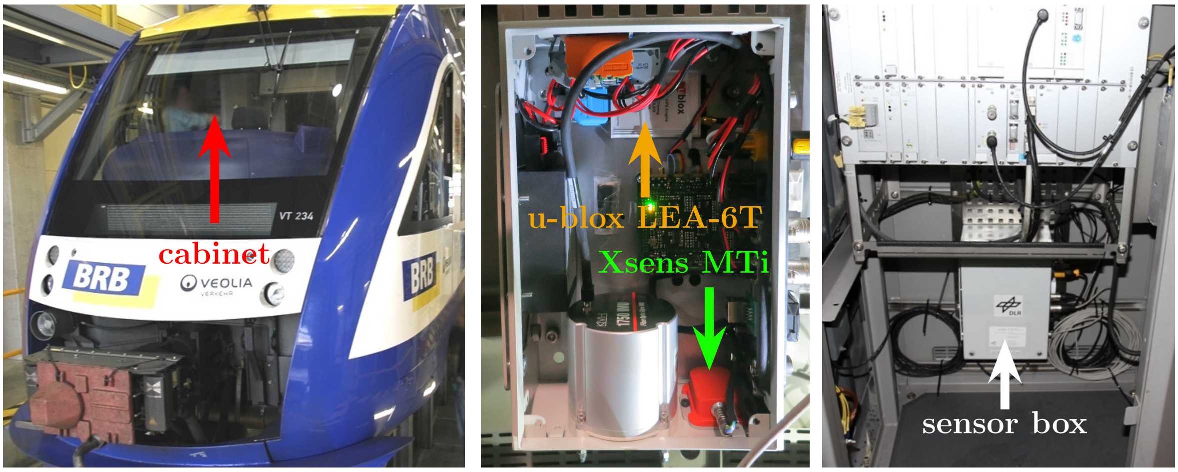

Figure 7.1: (left) Front view of regional train. (middle) Sensor box with u-blox LEA-6T GNSS receiver and Xsens MTi IMU. The IMU records the acceleration and the magnetic field. (right) Cabinet in the driver's cab with mounted sensor box.

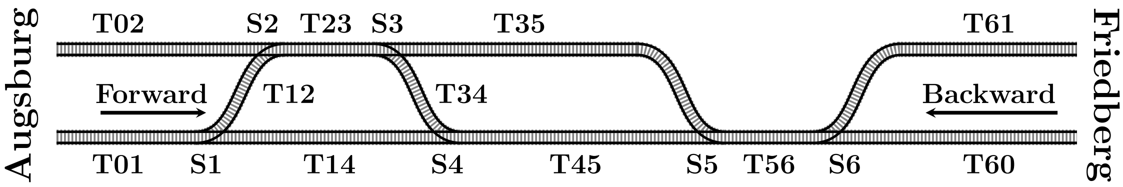

Figure 7.2: Topology of relevant track network between Augsburg main station and Friedberg station. In the topology only the six switches are considered that were traversed over both possible routes during the measurements.

Furthermore, the sensor box contains a low-cost single frequency u-blox LEA-6T GNSS receiver to record the ground truth position and speed. The GNSS observables measurement rate was set to 1 Hz.

For the evaluation of the filters from Chapter 4 we focus on measurements recorded on the tracks between Augsburg main station and Friedberg station. On its way between these two stations the train passes a multitude of switches, but many of them are traversed only over one of the two possible routes. As a result, the magnetic field on the second route was never measured and therefore cannot be considered in the evaluation. By ignoring switches that are traversed always over the same route, the track network used in the evaluation can be reduced to six switches between Augsburg and Friedberg. The topology of this simplified network is shown in Figure 7.2. For the evaluation six train runs through the network are used. The train is moving for 7.9 min to 10.3 min in these runs during which it travels the $\approx$ 7.5 km of tracks between Augsburg and Friedberg. In addition to the six runs used to evaluate the filters, the measurements of another four runs are used to create the magnetic map. In Table 7.1 the sequences of tracks and the switches passed in facing direction for the six evaluation runs are shown. Facing direction here means that the train has two possible routes to take. By looking at the table one can see that almost all switches are traversed from the facing direction at least once. The only exception is switch S4. S4 is traversed only over the trailing direction, which means the train always ends up on the same track and therefore the track does not have to be estimated by the filters. Table 7.1 additionally shows the driving direction of the train for each run. The direction is defined as shown in Figure 7.2. Throughout all runs, also the ones used for mapping, the train had the same orientation. The constant orientation plays an important role in the map creation because it allows us

Table 7.1: Routes of the different runs, the driving direction, and the switches passed in facing direction.

Run	Route	Direction	Switches passed facing
1	T02, T23, T34, T45, T56, T60	Forward	S3, S6
2	T02, T23, T34, T45, T56, T60	Forward	S3, S6
3	T60, T56, T35, T23, T12, T01	Backward	S5, S2
4	T01, T14, T45, T56, T61	Forward	S1, S6
5	T60, T56, T35, T23, T02	Backward	S5, S2
6	T60, T56, T35, T23, T12, T01	Backward	S5, S2

Figure 7.3: (top) Steam engine used during the measurements and steal cabinet on its back with the mounted sensor box. (bottom) HSB diesel railcar. In the railcar the sensor box was mounted directly on the floor inside the cabin.

to consider the magnetometer to be calibrated as will be explained in further details in Section 7.2.

More details about the measurements can be found also in the dissertation [70] that is also based upon these measurements.

7.1.2 Harzer Schmalspurbahnen

The second measurement campaign was performed 2015 with two different trains provided by the Harzer Schmalspurbahnen (HSB). During these measurements the trains were driving on a small regional track network in the north of Germany located in the Harz mountain range. For the evaluation of SLAC, the same track was recorded with two different train types, a steam locomotive and a diesel railcar. Concretely, the evaluation is based on a part of our data recorded on a track between

Quedlinburg and Gernrode. The data collection for the different trains was separated by roughly one month and was conducted with the same sensor setup.

The railcar and the steam locomotive are shown in Figure 7.3. In the campaign again the sensor box shown in Figure 7.1 is used but with updated sensors. The GNSS u-blox LEA-6T receiver was replaced by the newer u-blox LEA-M8T and the IMU was replaced by a Xsens MTi-G700. As before, the new magnetometer only outputs normalized values and we will use the unit a.u. in plots to highlight this fact, where one a.u. corresponds to $\approx 40\,\mu\mathrm{T}$. The measurement rate of the GNSS receiver is set to 1 Hz and the rate of the IMU to 200 Hz. It should be mentioned that the Xsens MTi-G700 outputs only the inertial data at the full rate, the magnetometer data is measured at a rate of 100 Hz. Compared to the older Xsens MTi this is a step back, but since the top speed of the used trains was only $\approx 50\,\mathrm{km\,h^{-1}}$ and the SLAC filters run well below 100 Hz this causes no issues. As shown in the top right part of Figure 7.3, on the steam engine the sensor box is mounted on the back in a steel cabinet. In the diesel railcar the box was mounted on the floor inside the cabin. The HSB measurement campaign was not planned for evaluating SLAC, i.e., neither the sensor hardware nor the placement of the sensors was optimized for SLAC. As will be shown in the result part for SLAC, this leads to some undesired effects. For example, the magnetometers in the two HSB trains are mounted approximately in the center of the track but not at the same height. In the steam locomotive the sensor was roughly at an height of ≈ 1.5 m above the rails and in the railcar the height above the rails was ≈ 1 m. Thus, even when the magnetometers are at exactly the same along-track position the linear calibration model is not necessarily fulfilled because of the height dependency of the magnetic distortions discussed in Section 3.3.2.

7.2 Evaluation of Localization with Calibrated Magnetometers

In this section, the different filter algorithms from Chapter 4 are evaluated with the measurement data from the BRB campaign. These filters assume a calibrated magnetometer during map creation and localization. As discussed in detail in Chapter 5, this constraint is almost impossible to achieve when the magnetometer is mounted inside a train. Fortunately, for the BRB campaign we can consider the magnetometer to be calibrated because the magnetic map of the track network is generated with data from the same measurement setup that was used to also record the data for the evaluation. Same setup here means that we use the same train and magnetometer mounted at the same position. Furthermore, it is necessary that the train does not change its orientation during the whole data recording because the biases of the magnetometer are fixed in the sensor frame and a rotation of the recorded data rotates also the bias vector, which would not be the case when the physical sensor is rotated. During the BRB campaign the train was always driving in the same orientation, therefore the assumption of calibrated magnetometers is fulfilled here. Note, in practice the use of an uncalibrated magnetometer requires a separate magnetic map for each train and for both possible orientations of the train on the track. It is obvious that this would strongly affect the practicality of magnetic train localization. This was also the reason why in this thesis the SLAC approach from Chapter 5 was developed that effectively allows to use one map for multiple trains. Nevertheless, testing and evaluating different filters on a pseudo-calibrated magnetic data set gives valuable insights about the performance and the limitations of magnetic localization.

In the following some details of the implementation are given and the different filter algorithms from Chapter 4 are evaluated.

7.2.1 Filtering the Overhead Power Line Magnetic Field

During the BRB measurement campaign the train was driving on a partially electrified track, which can be clearly seen in the magnetic field data in Figure 3.2 from Chapter 3. The current in the overhead AC power line has a known frequency of $f_{\text{ovr}} = 16.7\,\text{Hz}$. To remove the magnetic field induced by the current from the magnetometer data, the phase and amplitude of the alternating field can be estimated. The estimation is based on the the observation model (4.14) extended with the magnetic field generated by the power line currents

$$[\mathbf{z}_k^{\text{mag}}]_i = [\mathbf{R}_{\text{t}}^{\text{b}}(O_k)\,\mathbf{m}^{\text{t}}(s_k, R_{1:N_{\text{route}}})]_i + [\mathbf{a}_{\text{ovr}}]_i \sin(2\pi f_{\text{ovr}} kT + [\boldsymbol{\varphi}_{\text{ovr}}]_i) + [\mathbf{n}_k^{\text{mag}}]_i \;, \tag{7.1}$$

where $[\mathbf{z}_k^{\text{mag}}]_i$ is the observation of the i-th magnetometer axis with $i \in \{1, 2, 3\}$. The amplitudes and phases of the current induced magnetic field measured by the three sensor axes are collected in the vectors $\mathbf{a}_{\text{ovr}}$ and $\boldsymbol{\varphi}_{\text{ovr}}$.

On the first glance estimating the amplitude vector $\mathbf{a}_{\text{ovr}}$ and the phase vector $\boldsymbol{\varphi}_{\text{ovr}}$ appears to be a nonlinear problem, but applying the addition theorem [95] onto the second term in (7.1) considerably simplifies the estimation

$$\begin{aligned}
[\mathbf{a}_{\text{ovr}}]_i \sin(2\pi f_{\text{ovr}} kT + [\boldsymbol{\varphi}_{\text{ovr}}]_i) &= [\mathbf{a}_{\text{ovr}}]_i \cos([\boldsymbol{\varphi}_{\text{ovr}}]_i) \sin(2\pi f_{\text{ovr}} kT) \\
&\quad + [\mathbf{a}_{\text{ovr}}]_i \sin([\boldsymbol{\varphi}_{\text{ovr}}]_i) \cos(2\pi f_{\text{ovr}} kT) \\
&= [\mathbf{C}_{\text{ovr}}]_{i,1} \sin(2\pi f_{\text{ovr}} kT) + [\mathbf{C}_{\text{ovr}}]_{i,2} \cos(2\pi f_{\text{ovr}} kT) \;,
\end{aligned} \tag{7.2}$$

where the 3×2 matrix $\mathbf{C}_{\text{ovr}}$ contains the amplitude and phase dependent parameters of the current induced magnetic field. With (7.1), (7.2), and the assumption of a known train position s_k and orientation O_k the coefficients of the disturbance can be estimated with a linear LS estimator from the magnetometer measurements. To perform the LS estimation at least two vector observations are required to estimate the coefficient matrix. Depending on the signal-to-noise ratio of the measurements more measurements might be required for an accurate estimate. The estimated coefficients are considered constant for the duration in which the observations for the estimation are collected. Therefore, selecting the number of observations used in the estimation is a tradeoff between reducing the effect of noise and violating the constant parameter assumption.

However, the train position is unknown in practice. Thus, also the magnetic field that should be measured by the magnetometer in absence of any disturbance is unknown. This problem can be addressed by considering the magnetic field at the train position also as a unknown that has to be estimated jointly with the coefficient matrix $\mathbf{C}_{\text{ovr}}$ of the field induced by the power line current. To keep the complexity low, the unknown magnetic field is approximated by a simple linear function of time

$$[\mathbf{R}_{\text{t}}^{\text{b}}(O_k)\,\mathbf{m}^{\text{t}}(s_k, R_{1:N_{\text{route}}})]_i = [\mathbf{b}_{\text{mag}}]_i + [\mathbf{a}_{\text{mag}}]_i kT \;. \tag{7.3}$$

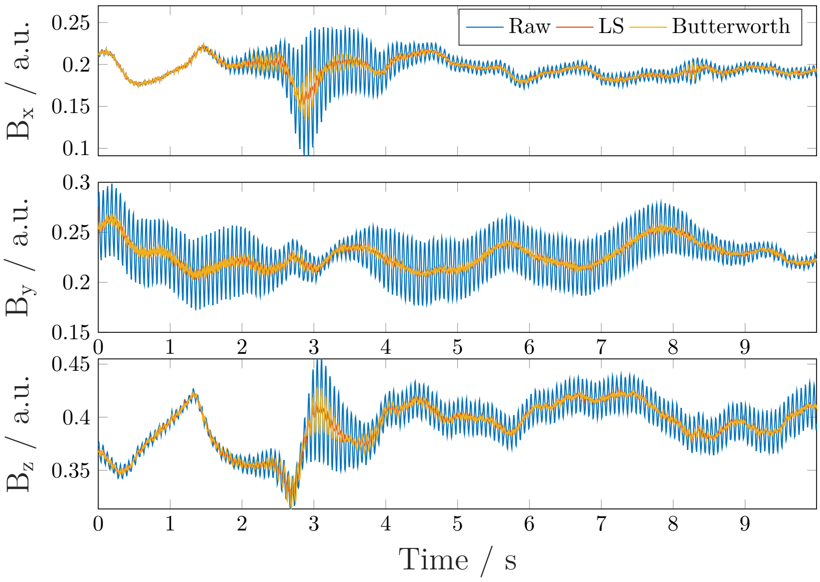

Figure 7.4: Raw 200 Hz magnetometer data (blue line) in comparison to the filtered data. The red line shows the magnetic field reconstructed from the LS estimate and the yellow line shows the result of a forward-backward Butterworth filter.

Inserting (7.2) and (7.3) into (7.1) yields the observation model

$$
\begin{aligned}
[\mathbf{z}_k^{\text{mag}}]_i &= [\mathbf{b}_{\text{mag}}]_i + [\mathbf{a}_{\text{mag}}]_i kT + [\mathbf{C}_{\text{ovr}}]_{i,1}\sin(2\pi f_{\text{ovr}} kT) + [\mathbf{C}_{\text{ovr}}]_{i,2}\cos(2\pi f_{\text{ovr}} kT) + [\mathbf{n}_k^{\text{mag}}]_i \\
&= \begin{bmatrix} 1 & kT & \sin(2\pi f_{\text{ovr}} kT) & \cos(2\pi f_{\text{ovr}} kT) \end{bmatrix} \begin{bmatrix} [\mathbf{b}_{\text{mag}}]_i \\ [\mathbf{a}_{\text{mag}}]_i \\ [\mathbf{C}_{\text{ovr}}]_{i,1} \\ [\mathbf{C}_{\text{ovr}}]_{i,2} \end{bmatrix} + [\mathbf{n}_k^{\text{mag}}]_i \\
&= \begin{bmatrix} 1 & kT & \sin(2\pi f_{\text{ovr}} kT) & \cos(2\pi f_{\text{ovr}} kT) \end{bmatrix} \boldsymbol{\theta}_i^{\text{mag+ovr}} + [\mathbf{n}_k^{\text{mag}}]_i \,.
\end{aligned} \tag{7.4}
$$

This is obviously a strong simplification and only makes sense when during the time the observations are collected for the estimation the train does not move too far. Theoretically, also higher order approximations could be used for the position dependent part, but in our evaluation the linear model seems to be appropriate as will be shown shortly.

With (7.4) the parameters of the position dependent magnetic field and the current induced field can be estimated with a simple LS estimator. Once the parameters are estimated, we reconstruct the position dependent magnetic field from (7.3) at the time of the current measurement and use this for the update of the localization filters. Therefore, the removal of the induced field must be performed in each update step of the filters based on a batch of the 200 Hz magnetometer data available at the time of the update.

Figure 7.4 shows the raw measurement data and the proposed LS-based reconstruction of the magnetic field alongside the raw data filtered with a forward-backward Butterworth band-stop filter. The forward-backward filter has zero-phase [96] and an order of 24. The edge frequencies are set to 16.7 ± 0.5 Hz. For the LS-based reconstruction ten samples of the 200 Hz data are used. The figure shows that in most parts of the example the two filtering methods remove the 16.7 Hz magnetic field equally well, but there are some parts, e.g., around 3 s when the amplitudes of the disturbances change, in which the proposed reconstruction method performs better. Thus, the performance of the LS method seems to be superior here, which made it the preferred choice to filter the magnetometer data before it is fed to the localization filters.

7.2.2 Map Generation

The magnetic map plays a crucial role in the filter algorithms since it is an integral part of the measurement model, in which it maps the topological position to the magnetic vector. The magnetic map has to be evaluated for each particle in the update step of the filters. The complexity of the map evaluation thus considerably contributes to the overall filter complexity. In the filter implementation the complexity is kept low by using a rather simplistic approach for the magnetic field map. In essence the map is a collection of arrays, in which each array describes the magnetic field of one track. The entries of each array are the magnetic vectors of the corresponding track at equally spaced along-track positions. When for a given along-track position the magnetic field is to be retrieved from the map, the equidistant grid enables the calculation of the index that corresponds to the closest grid point. Evaluating the map at the topological position of a particle therefore reduces to selecting the array assigned to its track identifier and performing a lookup of the magnetic vector at the index closest to the given along-track position.

The arrays that define the magnetic map are generated from magnetometer data recorded during four runs that are not used in the filter evaluation and that have been rotated into the track frame. During these four runs the train passes all tracks shown in Figure 7.2. To associate the magnetometer data to a position on the track network, the GNSS position of the u-blox receiver is interpolated onto the time grid of the 200 Hz magnetometer data and then the interpolated positions are matched to a geometric map of the track network. The map matching uses prior knowledge about the train route such that the GNSS position is always matched to the correct track. After matching the positions of the magnetometer data to the map, a simple linear interpolation is applied to get a magnetic vector for each along-track position in the desired equidistant grid. At a magnetometer sampling rate of 200 Hz and a top speed around $120\,\mathrm{km\,h^{-1}}$ the spacing between two measurements is below $0.2\,\mathrm{m}$. Within this distance the magnetic field does not change considerably, as discussed in Section 3.1, and a linear interpolation is sufficient. If the spacing between measurements is increased, it might be necessary to use higher order interpolation schemes. The chosen map spacing is somewhat arbitrary and was set to $\Delta s = 0.1\,\mathrm{m}$. Considering the spatial variability of the magnetic field, this value seems high enough to capture all features of the magnetic field while the memory requirements of the magnetic map is kept low. As will be shown in the evaluation, the accuracy of the position estimate is roughly one order of magnitude above the map resolution from which we conclude that the map resolution is not the limiting factor.

Before the map is created we also remove the magnetic field from the overhead power line with the approach from Section 7.2.1 to avoid having undesired, position independent, disturbances in the magnetic map. Note, the overhead power line magnetic field must be removed in a preprocessing step before the actual mapping, since the proposed approach has to be applied on the time series data provided by the magnetometer.

7.2.3 Implementation of Filter Algorithms

The BRB data will be used to evaluate four different filter algorithms. Concretely, we compare a filter that estimate the current track identifier as part of the filter state and filters that estimate the complete route of the train. The first filter is based on Algorithm 4.4, which essentially is an extension of the non-robust along-track localization filter Algorithm 4.1. For the other three filters Algorithm 4.1 and the two robust along-track filters, Algorithm 4.2 and Algorithm 4.3, are combined with Algorithm 4.5. The robust filters are not combined with the approach in which the track identifier is estimated as part of the state because this can lead to some problems when the particles are distributed on competing tracks. In this scenario, when enough particles are located on the wrong track, the robust filters potentially switch to the model that considers the measurements to be disturbed. If this happens, the particle weights are no longer updated with the magnetic data or the influence of the measurements on the weights is reduced to an extent that the filter starts to diverge or its accuracy is reduced.

The filter algorithm that estimates the current track identifier as part of the particle state and that distributes the particles randomly among the possible tracks when they pass a switch is a rather straightforward implementation of the pseudocode given in Algorithm 4.4. The same holds true for the filter that estimates the whole route based on the non-robust along-track filter, which combines the pseudocode from Algorithm 4.1 and Algorithm 4.5. The implementation of the filters that estimate the complete route based on robust filters is a bit more involved when it comes to the calculation of the route probabilities. In Appendix C therefore a detailed description of the calculation of the route probabilities for the two robust filters can be found.

The BRB data set contains periods of multiple seconds during which the train is at standstill. During these standstill phases it can happen that the filters lose track of the true train position when the filters are kept being updated with magnetometer data. This problem is caused by the magnetic field that can lead to multiple modes in the posterior. When the train is at standstill and the posterior has multiple modes, updating the filters with magnetic measurements from the same train position can lead to a loss of the correct mode. The loss of a mode in a multi-modal setting is not limited to magnetic localization, but is a general problem of particle filters [97, 98]. While driving this is not a big issue since the filters continuously “see” new magnetic field data and over time the magnetic field becomes unique. To make the filters more stable during standstill phases, we decided for a simple approach and implemented a standstill detector that uses only the magnetometer and accelerometer data available at the current time step. To detect a standstill phase, the variance of the magnetometer data and the amplitude of the acceleration is compared to a threshold and a flag is set when the magnetic field variance and the acceleration correspond to what we would expect during standstill. When a standstill is entered, the speed and the acceleration states of the particles are randomly reinitialized in an interval around zero. Once in standstill, the filters simply freeze their internal state and keep outputting the last estimated position. When the standstill detector indicates the train is moving again, the filters continue predicting and updating.

7.2.4 Filter Parametrization

The four different filters share a large part of their parameters to make their results comparable. In addition to the shared parameters, the robust filters require individual parameters related to their measurement models and model transition probabilities. In the following, the shared parameters are listed, and then the parameters specific for a single filter are introduced.

Shared Parameters

Each filter runs at a rate of 10 Hz with $N_\mathrm{p} = 4000$ particles and uses the motion model given by (4.18) and (4.19). The duration T between two time steps is the inverse of the filter update rate $T = 0.1\,\mathrm{s}$ and the standard deviation of the process noise if set to $\sigma^\mathrm{d} = 0.11\,\mathrm{m\,s^{-2}}$. The probability that in the Markov chain of the train orientation the orientation is flipped to the opposite value is set to $1 - \pi_O = 0.1$. It is therefore rather unlikely for the orientation to change. Theoretically the orientation cannot change at all since we assume the route of the train does not contain loops, but for the stability of the filters it is beneficial to leave a small possibility to change the orientation. The last two shared parameters are related to the accelerometer and magnetometer likelihoods. The standard deviation of the accelerometer likelihood is set to $\sigma^\mathrm{acc} = 0.4\,\mathrm{m\,s^{-2}}$. This value is rather high due to the simplifications made in (4.16) that require a downweighting of the measurements. For the magnetometer data all filters use a standard deviation $\sigma^\mathrm{mag} = 0.07$ as base value. For the non-robust filters this means they always use this standard deviation and likelihood (4.15) in the weight update. For the robust filters the base value and (4.15) is used only for undisturbed data. Note, σ^mag is dimensionless here because of the internal normalization of the Xsens magnetometer.

When the different filters pass a switch, they do not output a valid position estimate until one of the possible tracks or routes reach a probability of 0.9. When the probability of a track or route reaches this threshold a position estimate is outputted again. For the filters based on Algorithm 4.5 additionally all filters are deleted that are not on the route that reached the threshold.

The particle set of all filters is initialized in the same way. In the beginning all particles are on the correct track and are uniformly distributed in an 20 m wide interval around the true position obtained from the GNSS ground truth. Since the train is at standstill in the beginning of the runs, the speed and acceleration of the particles are uniformly distributed around zero. The speed interval is set to $1\,\mathrm{m\,s^{-1}}$ and the one for acceleration to $0.2\,\mathrm{m\,s^{-2}}$. The weights of all particles are set to $\frac{1}{N_\mathrm{p}}$. Resampling is only performed when the number of effective particles falls below $0.4 \cdot N_\mathrm{p}$.

Parameters of the FDE Filter

The FDE filter requires in addition to the shared parameters the transition probabilities $p^{(j,i)}$ used in the calculation of model probabilities in (4.29). In the evaluation, the transition probabilities are set such that the probability to stay in the same model is $p^{(i,i)} = 0.3$ with $i \in \{1, \dots, 8\}$. The remaining probability is equally distributed between the transition probabilities $p^{(j,i)} = 0.1$ with $i \neq j$. As a result, the probability to stay in the same model is considerably higher than the probability to switch

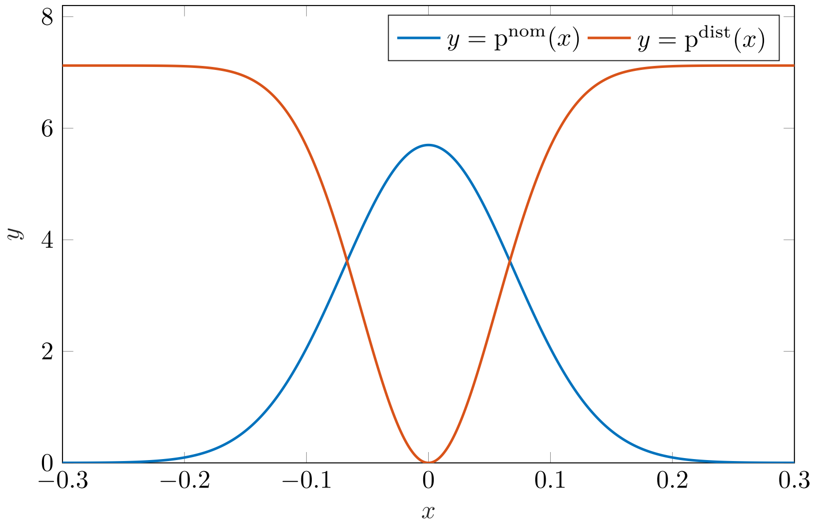

Figure 7.5: Likelihoods used in the FDE filter to calculate the model probabilities.

to another specific model. In the initialization the filter assumes that model one is correct and its probability is set to one.

Furthermore, the likelihood $\mathrm{p}^{\mathrm{dist}}(\cdot)$ from (4.34) for a disturbed magnetometer axis has to be parameterized. The parameters c^{dist} and σ^{dist} from (4.34) are tuning parameters and allow us to influence how quickly a measurement is rejected in the update step. For the evaluation the parameters are set to $\sigma^{\mathrm{dist}} = 0.8 \cdot \sigma^{\mathrm{mag}}$ and $c^{\mathrm{dist}} = (2\pi(\sigma^{\mathrm{dist}})^2)^{-\frac{1}{2}}$. The resulting likelihood is shown in Figure 7.5 in comparison to the Gaussian pdf for the undisturbed case that uses the base value of σ^{mag} as standard deviation. From Figure 7.5 it can be seen that for larger errors $\mathrm{p}^{\mathrm{dist}}$ exceeds the value of the nominal likelihood. That means that models that assume a disturbance in the measurements will get a higher probability in presence of disturbed measurements than models that use the nominal likelihood. As already mentioned in Section 4.4.1, the area enclosed by the function $\mathrm{p}^{\mathrm{dist}}(\cdot)$ is not normalized to one and therefore $\mathrm{p}^{\mathrm{dist}}(\cdot)$ is not a proper pdf. In the evaluation this does not cause any problems and enables us to choose a wide variety of parameters and functions. When the support of $\mathrm{p}^{\mathrm{dist}}(\cdot)$ would be limited to a fixed interval, one could also normalize it such that it becomes a proper pdf, but this might not yield the desired result. For example, in Figure 7.5 $\mathrm{p}^{\mathrm{dist}}(\cdot)$ was chosen such that its maximum value exceeds the maximum value of $\mathrm{p}^{\mathrm{nom}}(\cdot)$ to enable a fast switching between models when disturbed measurements are passed to the filter. If we would normalize $\mathrm{p}^{\mathrm{dist}}(\cdot)$, its maximum value would be limited by the width of the support and switching between models might be considerably slower.

Parameters of the NMA Filter

The NMA filter in the evaluation uses only two models for the magnetometer measurements. The first model assumes that all magnetometer measurements only have the nominal additive sensor noise. The first model thus uses likelihood (4.15) and the base value of the noise standard deviation as described in the shared parameter section. The second model assumes the measurements contain disturbances that are larger than the nominal sensor noise. The second model captures this with another Gaussian pdf that has a higher standard deviation of $1.5 \cdot \sigma^{\mathrm{mag}}$ that reduces the impact of a single measurement on the particle weights. The two different likelihoods are shown in Figure 7.6. As the FDE filter,

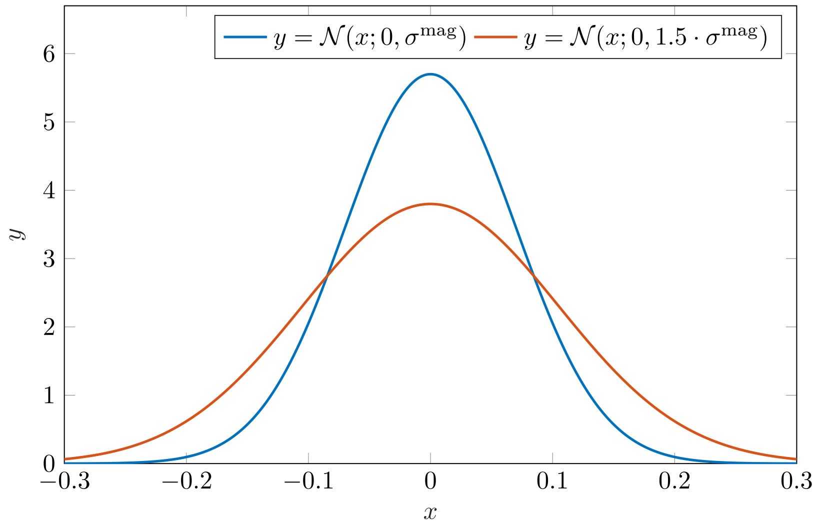

Figure 7.6: Likelihoods used in the NMA filter to calculate the model probabilities and to update the particle weights.

Table 7.2: Position error statistics for the different filters across all six runs. The shown statistics are based on the Monte Carlo Run with the highest RMSE.

	joint filter			copy filter			FDE filter			NMA filter		
Run	RMSE/m	r_{RMSE}/%	max/m	RMSE/m	r_{RMSE}/%	max/m	RMSE/m	r_{RMSE}/%	max/m	RMSE/m	r_{RMSE}/%	max/m
1	1.87	39	11.22	3.12	128	13.35	1.69	27	8.37	1.81	32	9.80
2	1.86	48	10.69	2.40	91	12.51	1.82	51	8.29	1.81	44	9.89
3	2.12	8	8.21	2.09	8	7.88	2.08	6	7.93	2.14	9	8.89
4	3.67	136	21.49	4.13	167	21.03	1.72	12	7.59	2.24	40	11.96
5	2.16	6	8.72	2.21	8	9.35	2.23	9	8.83	2.18	7	9.12
6	1.94	13	9.56	1.83	8	6.95	1.89	13	8.45	1.86	10	6.68

the NMA filter requires model transition probabilities. Here these values are set to $p^{(j,i)} = 0.5$ with $i, j \in \{1, 2\}$. At the first time step both models are considered to be equally probable.

7.2.5 Results

The evaluation result for the six runs from Table 7.1 will be presented in two parts. The first part is related to the position accuracy of the different filters. In the second part, the result of the track identification is presented and discussed. To make it easier to differentiate between the different filters from Chapter 4, in the following the filter for joint estimation of track identifier and along-track position, Algorithm 4.4, will be referred to as joint filter. The combination of the filter for the joint estimation of the route and the along-track position, Algorithm 4.5, with the non-robust Algorithm 4.1 will be called copy filter. The combinations of Algorithm 4.5 with the FDE filter, Algorithm 4.2, and the NMA filter, Algorithm 4.3, will be simply called FDE and NMA filter.

Position Accuracy

To evaluate the performance of the four different filters, each filter was evaluated 100 times in a Monte Carlo simulation. The Monte Carlo simulation has the goal to show the impact of the process noise realization on the filter result and to get an overall idea of the filter stability and the repeatability of the results. The results for the position accuracy are summarized in Table 7.2 divided into the six runs. The position error used in the statistics of Table 7.2 is calculated from the position estimate of the filters and the ground truth GNSS receiver position. The root mean squared error (RMSE) in the table is the highest RMSE among all Monte Carlo runs. To get an impression about how the RMSE differs between Monte Carlo runs, the table shows for each filter the increase from its lowest to its highest RMSE. The increase in percent is calculated according to

$$r_{\text{RMSE}} = \left(\frac{\text{RMSE}_{\text{max}}}{\text{RMSE}_{\text{min}}} - 1 \right) \cdot 100\% \, . \tag{7.5}$$

The shown values for r_{RMSE} are rounded to the next integer. In addition to the RMSE and r_{RMSE}, the table shows the maximum absolute position error for the Monte Carlo run that resulted in the highest position RMSE. It should be noted, that the statistics in Table 7.2 account only for phases in which the train was moving, but not the standstill phases. Cutting out the standstill phases avoids skewing the statistics in favor of filters that enter the standstill phases with a small error, because the filters do not update their estimate during standstill. If the statistics would account for the standstill phases, there might be a situation in which a filter seems to have a smaller RMSE than other filters although its error during driving is consistently higher.

From the error statistics it becomes apparent that all filters perform quite well and achieve a position RMSE below 5 m. Considering all six runs, the FDE and NMA filters have the lowest RMSE in most cases or are close to the filter with the lowest RMSE. The FDE filter also manages to keep the maximum error below 9 m across all runs. In contrast, the joint, copy, and NMA filter all have runs with a errors larger than 11 m. The increase r_{RMSE} from the lowest to the highest RMSE shows that the robust filters are more consistent over the different Monte Carlo runs. To get a better understanding where the differences between the filters are originating from, in the following we will have a closer look on the estimation error of run 4, for which larger differences in the filter performance were observed.

In Figure 7.7 the estimation error for run 4 in Table 7.1 is shown alongside the standstill phases, gray areas, and phases in which no position estimate is generated by the filters, green areas. The green areas correspond to the part of the data set in which the train is driving across a switch and the new track is not resolved yet. The gray areas are derived from the GNSS ground truth speed and thus do not perfectly align with the filter standstill phases that are derived purely from the standstill detector based on the magnetometer and accelerometer data, as discussed in Section 7.2.3. The green areas might look slightly different for the different filters, but for the sake of clearness only the green area of the copy filter is plotted. When looking at the position error, top plot in Figure 7.7, there is a sudden strong increase of the error around 110 s for all filters but the FDE filter. As can be seen from the plot and the maximum error in Table 7.2, the FDE filter here manages to keep the maximum error considerably smaller than the others filters. Compared to the non-robust filters, the NMA filter can

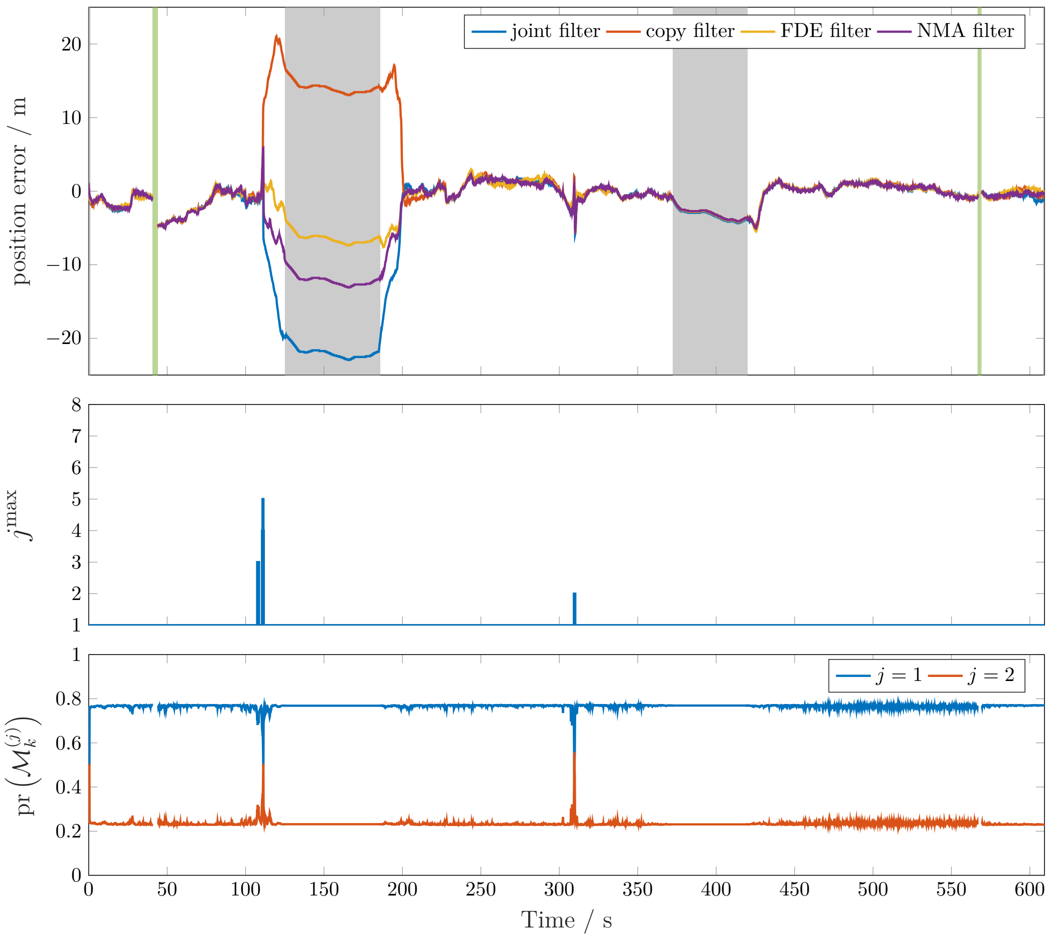

Figure 7.7: Position error of the four filters for run 4. The shown estimation error is from the Monte Carlo run with the highest RMSE. (top) Position error of the four filters. The gray areas highlight standstill phases. The green areas indicate the time when the copy filter reaches a switch and until the new track is resolved by the filter. (middle) Index of the model in the FDE filter with the highest probability. (bottom) Probability of the two models used in the NMA filter.

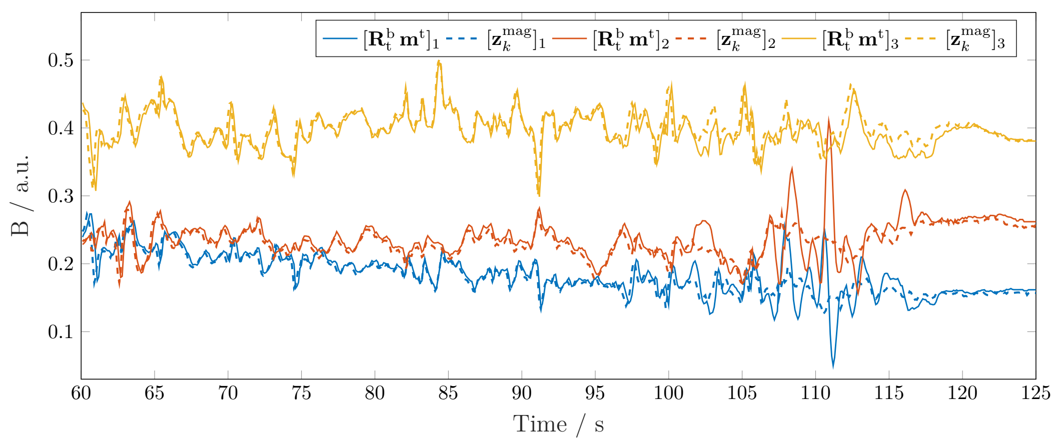

Figure 7.8: Cutout of the measured magnetic field and corresponding magnetic field in the map for run 4. The map magnetic field is obtained at the GNSS ground truth position and is rotated into the frame of the measurements with the true orientation.

also reduce the maximum error, although not as good as the FDE filter. To understand why the robust filters perform better here, we have to have a closer look on the model probabilities calculated by the filters. The plot in the middle of Figure 7.7 shows the index of the model selected in the update step of the FDE filter. During the time the estimation errors start to increase also the model index jumps from $j^{\max} = 1$, which corresponds to the model that uses all magnetometer measurements in the filter update, to another model index that rejects at least one of the sensor axes. A similar behavior is seen in the model probabilities of the NMA filter. Around the time the position error increases the probability of the first model is decreased and the probability of the second model is increased.

To check why the model probabilities start to change around 110 s and if this change is actually desired we have to have a look into the measurement data used as input for the filters. Therefore, Figure 7.8 shows the measured magnetometer data used in the filters together with the corresponding values in the magnetic map at the GNSS ground truth position rotated into the frame of the measurements. The measured values, dashed lines, show a good agreement with the magnetic field in the map until 100 s, then larger differences appear for a couple seconds. The changes in the model probabilities of the FDE and NMA filter are therefore a desired effect and lead to an exclusion of disturbed measurements, in the case of the FDE filter, and a downweighting in the case of the NMA filter. It should be mentioned that there is a shift in the along-track direction between the map and the measured data in Figure 7.8. This shift is caused by a position shift in the GNSS ground truth, which can have meter-level errors. These errors in the ground truth are also clearly visible in Figure 7.7 in the standstill phases. During these phases all filters output a constant position, but the error keeps changing due to the drift in the ground truth. Regarding the disturbances in the measurements, we found out that the same errors are present not only in run 4 but also in run 1 and run 2. A comparison of the train routes during these runs in Table 7.1, lead to the suspicion that the disturbance might be related to track T45. Further investigations of the data sets lead to the conclusion that the map of track T45 has parts where the map does not fit to the measurements. The cause for this must be some disturbance in the measurements used in the map creation, which could not be identified.

The example in Figure 7.7 and the statistics in Table 7.2 show that the evaluated filters can provide the train position also in the presence of disturbed measurements and erroneous maps. Especially the robust FDE and NMA filters can cope with these situations in the evaluation quite well and can keep the maximum error small compared to the other two filters. Furthermore, the robust filters achieve more consistent results w.r.t. the RMSE across the different Monte Carlo runs, as shown by the increase r_{RMSE} between the highest and the lowest RMSE.

Finally, we want to have a look on the cumulative distribution function (cdf) of the absolute position error across all six runs because the cdf gives a more complete impression about the overall error distribution than just the values in Table 7.2. The cdf in Figure 7.9 is again calculated without the standstill phases and uses for each of the six runs the Monte Carlo run with the highest RMSE value. The cdf once more confirms the superiority of the robust filters. Especially for the cdf values above 0.9 shown in Figure 7.9 the robust filters outperform the other filters. For lower values of the cdf the filter performance is close. This was somewhat expected, since in the absence of larger disturbances in the magnetometer data or in the map the overall filter performance is quite similar as can be seen also from Figure 7.7 after the first standstill phase. For higher values the cdf also shows a larger difference between the joint filter and the copy filter, although these filters are using always the same measurement

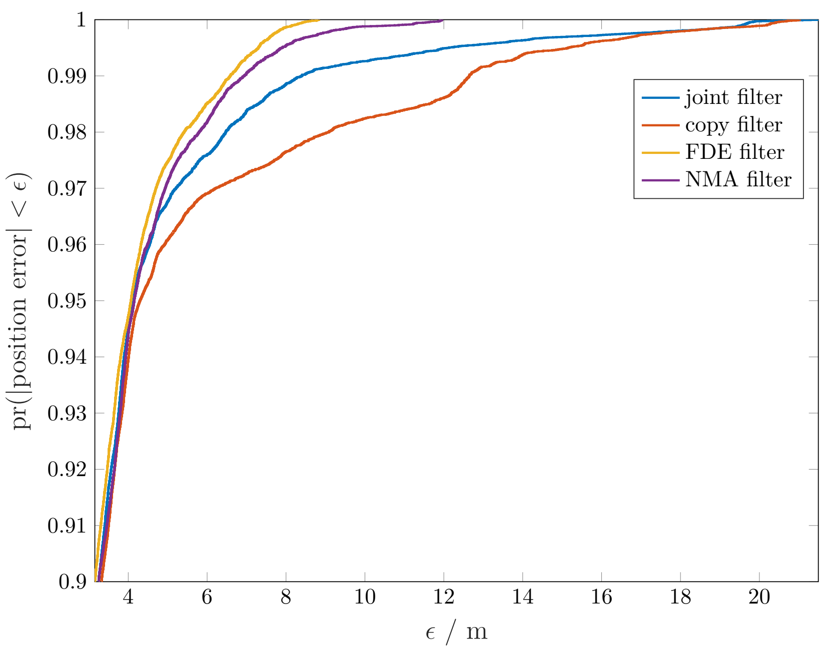

Figure 7.9: Cumulative distribution function of the absolute position error over all six runs.

model. We think this is mainly a random effect caused by the Monte Carlo simulation and the large differences between different realization of the process noise during run 1, 2, and 4. When there is a systematic difference between the accuracy of the joint and the copy filter we would expect a small benefit for the copy filter because the joint filter has to distribute its particles over multiple tracks, whereas the copy filter runs a separate filter for each possible route. This causes differences in how the point estimates are calculated in the two filters. The joint filter calculates the estimate only based on the particles that are on the most likely track, which means not always all particles are considered. In contrast, the copy filter always uses all particles. Nevertheless, the difference between the joint filter and the copy filter is to be expected small when there are no disturbances in the magnetometer data.

At this point we would like to point out that the presented results are based only on a small data set and therefore the comparison between the filters should be taken with a grain of salt. For a definite decision for one of the filters a more exhaustive evaluation should be performed in the future.

Track Identification

In addition to the position error of the filters, it is also important how accurate and fast the filters can determine on which track the train is driving after a switch was passed. In the evaluation, all filters were able to identify the correct track after passing a switch. The differences in the time and traveled distance required to identify the correct track were not dramatically different across all filters. Hence, only the track identification results for the copy filter are shown in Table 7.3. The times and distances are obtained once more from the Monte Carlo run with the highest RMSE.

Table 7.3 shows that the distance required to identify the correct track is between 15.5 m and 100.7 m. The distances therefore differ quite significantly. Even on the same switch one can observe larger differences, e.g., the required distance for track identification on switch S6 is more than doubled for

Table 7.3: Time and traveled distance until correct track is identified by the copy filter.

Switch		Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
S1	Time / s	-	-	-	3.2	-	-
	Dist. / m	-	-	-	47.7	-	-
S2	Time / s	-	-	6.1	-	6.1	2.8
	Dist. / m	-	-	34.7	-	33.2	15.5
S3	Time / s	2.1	2.0	-	-	-	-
	Dist. / m	28.9	29.1	-	-	-	-
S4	Time / s	-	-	-	-	-	-
	Dist. / m	-	-	-	-	-	-
S5	Time / s	-	-	2.9	-	3.0	2.6
	Dist. / m	-	-	42.40	-	44.6	38.2
S6	Time / s	5.6	2.4	-	1.9	-	-
	Dist. / m	100.7	39.6	-	27.7	-	-

run 1 in comparison to run 2 and 4. Especially the difference between run 1 and 2 is surprising, considering the train takes the exact same route during those runs, while during run 4 the train was taking the other track after passing switch S6. It is not entirely clear why this is the case and thus needs further investigation on a larger data set.

7.2.6 Summary

The evaluation showed the feasibility of magnetic localization with calibrated magnetometers in the railway domain based upon six different runs on a local track network in Germany. During the evaluation four different filters are compared to each other w.r.t. their position accuracy and the capability to identify the correct track after passing a switch. From the results it can be concluded that all filters were able to localize the train with a RMSE below 5 m and that all tracks have been be correctly identified. In absence of larger errors in the magnetic measurements, all filters performed on a similar level. However, in the presence of disturbed measurements the two proposed robust particle filter variants showed an improved accuracy and stability. The filters achieved this improvement by accounting for different measurement models that allowed them to either exclude disturbed measurements or to downweight them in the filter update step.

Finally, we would like to mention that for the given data set and the chosen number of particles all filters achieved real-time capability when running on a laptop CPU. Real-time capability here means that an average update rate beyond 10 Hz was achieved. Therefore, in average the processing of a single filter iteration was completed before a new measurement would be provided to the filter in an online processing scenario. The evaluation results are rather promising and motivate further investigations of the proposed filters with an extended data set.

7.3 Evaluation of Simultaneous Localization and Calibration

Different algorithms for magnetic field-based train localization were evaluated in the previous section, in which the magnetic map data and the data used for localization was recorded with the same hardware setup and the same orientation. Thus, there was no need to account for possible calibration effects and hence we just implemented the filters under the assumption of a calibrated data set. However, in order to use magnetic field-based train localization in a more practical setting, it cannot be assumed that a map created with the same hardware setup is available and neither is it possible to calibrate the magnetometer inside a train without considerable effort using common methods. These practical considerations lead to the development of the SLAC algorithms presented in Chapter 5. To show the feasibility of SLAC, the proposed algorithms are evaluated based upon data collected during the HSB measurement campaign with two different trains as described in Section 7.1.2. Concretely, the algorithms will be tested by creating a magnetic map with the steam engine data, which then will be used in the localization of the diesel railcar.

For the evaluation of SLAC we use one run during which the train travels $\approx$7 km. During the run the train started from standstill, accelerated, and then kept driving with an approximately constant speed. In average the speed during movement was around $\approx 39\,\mathrm{km\,h^{-1}}$. The evaluation presented in the following is based on our paper [J2]. As in [J2], track identification is not considered here. Nevertheless, the combination of SLAC and track identification is an interesting topic that should be considered in future work.

7.3.1 Map Generation

The map used in the evaluation is generated from the GNSS and magnetometer data of the steam engine. The map representation here follows an approach similar to Section 7.2.2. The map is therefore represented by an array of magnetic field vectors with associated along-track positions. In contrast to the map from the previous section, the magnetic vectors are here not rotated into the track frame because the SLAC algorithm already takes care of the rotation between the map and the body frame of the magnetometer used for localization.

The steam engine reaches only speeds around $50\,\mathrm{km\,h^{-1}}$. The low speeds combined with the magnetometer sampling rate of 100 Hz allows us to use once more a simple linear interpolation scheme without introducing larger deviations into the map. The resolution of the map is kept at $\Delta s = 0.1$ m. The track used in the evaluation is not electrified and the removal of the overhead power line induced magnetic field is not necessary.

7.3.2 Implementation of Evaluated Filter Algorithms

The filter used for SLAC is based on the implementation from our paper [J2] and follows Algorithm 5.1. Similarly to the filters evaluated in the previous section, the prediction step of the Rao-Blackwellized particle filter uses motion model (4.18) and (4.19). In contrast to the prediction step, the SLAC weight update step has two major differences when compared to the filters for calibrated data. The first change is that we forgo the accelerometer update. This was done to demonstrate the feasibility of SLAC using solely measurements from an uncalibrated magnetometer. The second difference is related to the magnetometer measurement model. The magnetometer likelihood (4.15) for calibrated magnetometers is replaced with (5.24). In (5.24), the train orientation is not explicitly needed since it is absorbed in the calibration parameters. The two mentioned changes in the update step render the train orientation redundant and it can be removed from the state of the Rao-Blackwellized particle filter used for SLAC. Another implementation detail that should be mentioned is that the Rao-Blackwellized particle filter uses three four-dimensional Kalman filters to estimate the calibration parameter vector instead of one filter that estimates all twelve parameters. As described in Section 5.3.1, the decomposition of the parameter vector is made possible by the particular structure of the estimation problem and reduces the complexity of the filter.

In addition to the SLAC algorithm that estimates the full twelve-dimensional parameter vector, also the reduced SLAC algorithm, from Section 5.3.4, is implemented and evaluated. For the reduced SLAC algorithm it was assumed that the calibration matrix $\bar{\mathbf{C}}$ in (5.13) is close to a diagonal matrix. This is the case when $\mathbf{C}$ and $\tilde{\mathbf{C}}$ are dominated by scaling effects in their main diagonal and when the relative orientation $\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k}$ is close to a diagonal matrix. For the HSB hardware this is not the case because the relative orientation is given by

$$\mathbf{R}^{\mathrm{b}}_{\tilde{\mathrm{b}},k} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} , \tag{7.6}$$

which leads, even when the matrices $\mathbf{C}$ and $\tilde{\mathbf{C}}$ are diagonal, to a matrix with the same structure as (7.6) with non-zero entries in the off-diagonal elements. Fortunately, for the rotation matrix (7.6) we can simply rotate the measured magnetic vector into the body frame of the map by a simple left multiplication of (5.12) with the inverse of (7.6)

$$\mathbf{R}^{\tilde{\mathrm{b}}}_{\mathrm{b},k}\mathbf{z}^{\mathrm{mag}}_k = \mathbf{R}^{\tilde{\mathrm{b}}}_{\mathrm{b},k}\bar{\mathbf{C}}_k\tilde{\mathbf{m}}^{\tilde{\mathrm{b}}}(s_k, R_{1:N_{\mathrm{route}}}) + \mathbf{R}^{\tilde{\mathrm{b}}}_{\mathrm{b},k}(\bar{\mathbf{b}}_k + \mathbf{n}^{\mathrm{mag}}_k) , \tag{7.7}$$

where $\mathbf{R}^{\tilde{\mathrm{b}}}_{\mathrm{b},k}\bar{\mathbf{C}}_k$ is now a diagonal matrix as desired. For the reduced SLAC implementation this means, we simply take the measurements of the railcar and rotate them into the map body frame before we pass the data to the update step of the particle and Kalman filter. The rest of the algorithm is basically the same as the full SLAC algorithm, just with a reduced set of parameters as described in Section 5.3.4. Alternatively to rotating the data, we could here also modify $\mathbf{H}^{\mathrm{red}}$ from (5.32) such that it maps the reduced parameter vector $\boldsymbol{\theta}^{\mathrm{red}}$ correctly to the measurements, the result is expected to be comparable.

7.3.3 Filter Parametrization

As in the evaluation of the filters with calibrated magnetometers, the SLAC filters are run at an update rate of 10 Hz and use the same number of particles $N_{\mathrm{p}} = 4000$. Compared to [J2], the number of particles was increased from $N_{\mathrm{p}} = 3000$ to have the same amount as in the previous section to make the results more comparable. With 4000 particles and an implementation in Matlab the overall processing time of the SLAC algorithms was well below the duration of the processed data set when the filters were executed on a laptop CPU.

The process and measurement noise parameters were tuned to the data set from the HSB campaign. The magnetometer noise standard deviation was set $\sigma^{\mathrm{mag}} = 0.025$ and the process noise standard deviation of the motion model to $\sigma^{\mathrm{d}} = 0.1\,\mathrm{m\,s^{-2}}$. As before, the standard deviation of the magnetometer noise has no unit because the sensor only provides normalized data. The process noise standard deviation of the state space model of the calibration parameters from (5.20) was experimentally selected and has an value of $\sigma^{\bar{\boldsymbol{\theta}}} = 0.005$. For constant parameters $\sigma^{\bar{\boldsymbol{\theta}}}$ should ideally be zero but allowing for some noise gives the filter more flexibility to change the parameters over time and therefore the possibility to account for deviations from the linear calibration model in (5.12).

The initial states of the particles are sampled from uniform distributions centered around the ground truth values of the train position, speed, and acceleration. The position ground truth is obtained from GNSS. The speed and acceleration are both set to zero in the initialization since the train is at standstill at the beginning of the data set. The uniform distribution of the particle positions has a width of 20 m. For the particle speed and acceleration states the width of the distributions is set to $1\,\mathrm{m\,s^{-1}}$ and $0.5\,\mathrm{m\,s^{-2}}$. The initial weights are all set to the value $\frac{1}{N_{\mathrm{p}}}$.

The initial parameter vector of the Kalman filters is set such that it corresponds to $\bar{\mathbf{C}} = \mathbf{I}_{3\times3}$ and $\bar{\mathbf{b}} = \mathbf{0}_{3\times1}$. The initial variance for all parameters is set to one. These values are the same for all particles and correspond to the assumption that the railcar magnetometer measures the same magnetic field that is also stored in the steam engine map.

7.3.4 Results

The results presented in the following include the results of the full and the reduced SLAC algorithms and the results of two particle filter variants without online calibration that are considered as benchmarks.

The first particle filter variant without online calibration uses a set of fixed calibration parameters. The fixed parameters are estimated from the railcar data recorded on the first couple of hundred meters of the track used in the evaluation. For the estimation the corresponding magnetic vector from the map is retrieved with the GNSS ground truth. The calibration parameters are then estimated with a LS estimator. The estimated parameters are passed to the filter that treats them as known and applies them to the magnetic map data before the weight update is performed.

The second filter variant uses uncalibrated data that has been only rotated into the body frame of the map according to (7.7). This variant therefore uses the same data as the reduced SLAC filter. Both particle filter variants are essentially a simplified version of Algorithm 4.1, in which the train orientation is considered known and thus is not estimated. Furthermore, the filters do not use the accelerometer information to make them comparable to the two SLAC filters.

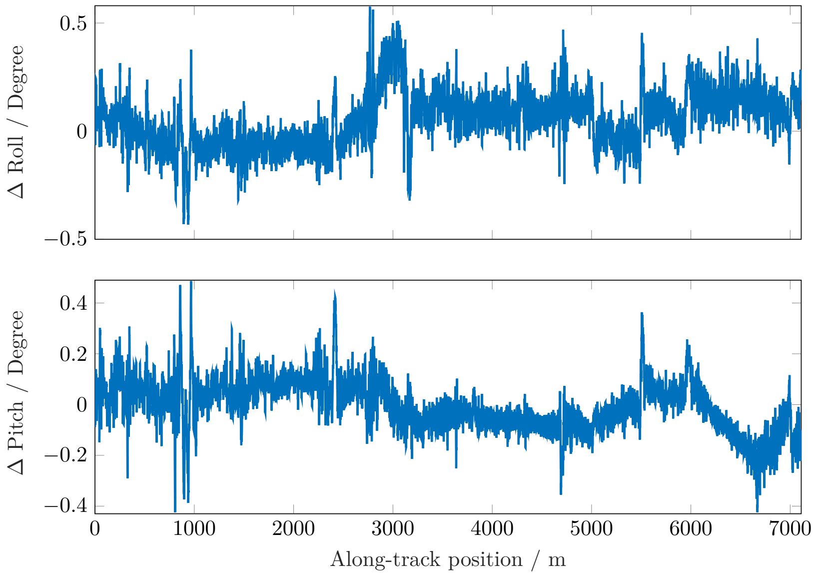

Figure 7.10: Roll and pitch angle difference between the steam engine and diesel railcar along the ≈ 7 km test track.

Attitude Difference between Railcar and Steam Engine

During the derivation of the SLAC algorithm in Chapter 5, some simplifications and assumptions were made that enabled us to perform a relative calibration to a magnetic map that was recorded with an uncalibrated magnetometer. The crucial assumption in the derivation was that the rotation matrix $\mathbf{R}^{\mathrm{v}}_{\tilde{\mathrm{v}},k}$ between two vehicle frames is nearly constant for two trains that drive on the same track.

To substantiate this assumption, the attitude of the railcar and the steam engine was calculated with a inertial navigation system that was aided by the data of the GNSS receiver. The sensor fusion is done by an error state Kalman filter, which loosely couples the two systems. The inertial navigation system uses the data of a KVH 1750 IMU that provides a high-quality fiber-optical gyroscope. The KVH IMU was placed in the same sensor box as the Xsens IMU, used to measure the magnetic field data, and can be recognized in Figure 7.1 as a silver metal cylinder.

Based on the estimated attitude of the two different trains, the difference in the roll and pitch angle was calculated based on the rotation matrix $\mathbf{R}^{\mathrm{v}}_{\tilde{\mathrm{v}},k}$. The resulting angle differences are shown in Figure 7.10 for the ≈ 7 km long track used for evaluating the SLAC algorithms. The figure shows, that the differences are small and below 0.5° for most of the time. Thus, the assumption of a constant $\mathbf{R}^{\mathrm{v}}_{\tilde{\mathrm{v}},k} \approx \mathbf{R}^{\mathrm{v}}_{\tilde{\mathrm{v}}}$ is fairly well fulfilled for the two considered trains. However, one should keep in mind the low speeds during the measurements. For higher speeds, or for trains with an active tilting system the difference might be larger. For larger deviations from the assumption of a constant relative rotation matrix, SLAC still could work since one of its advantages is the possibility to adapt the calibration parameters over time.

Position Error

For the evaluation of the position error a Monte Carlo simulation with 100 runs was performed. Each run uses a different realization of the process noise and initial particle distribution. The Monte Carlo

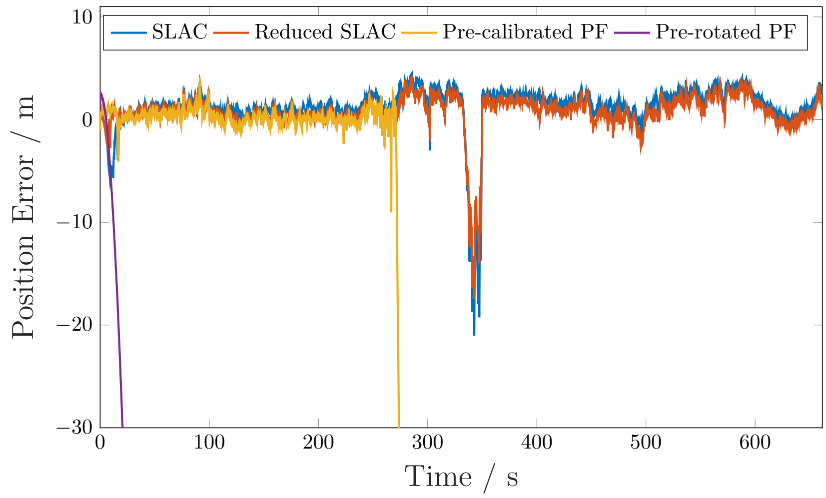

Figure 7.11: Position error over time. The error is calculated between the MMSEE of the filters and the GNSS ground truth.

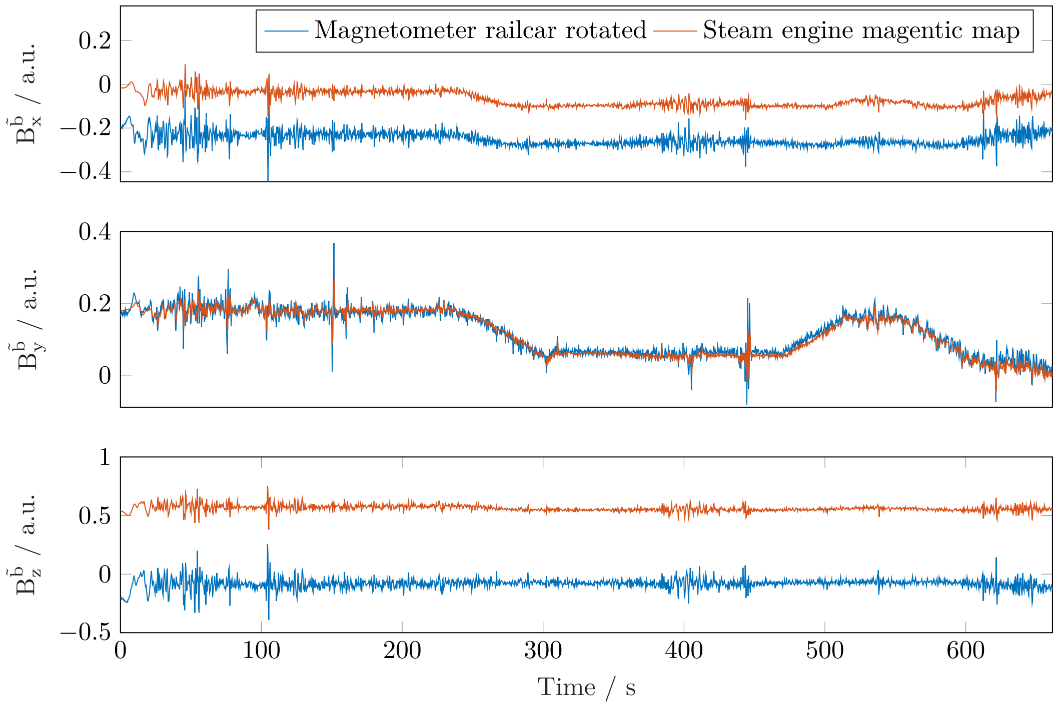

Figure 7.12: Data of the Diesel railcar rotated into the body frame of the steam engine and the corresponding values in the magnetic map from the steam engine.

simulation gives an insight in the repeatability of the results and the sensitivity of the filter to the process noise and the initial particle states.

The position errors over the whole train run are shown in Figure 7.11. The position errors are calculated between the GNSS ground truth and the along-track position MMSEE of the different filters. For the two SLAC filters the figure shows the Monte Carlo run with the highest RMSE. For the remaining two filters the runs with the lowest RMSE are shown. The errors of the pre-calibrated and pre-rotated filter grow up to hundreds of meters or the filters diverge and stop giving a valid output. Figure 7.11 thus only shows their errors until the point when they first surpass 30 m. From Figure 7.11 it can be seen that only the two SLAC filters are capable of accurately tracking the train position over the whole train run. The filter that uses only the rotated data, purple line, is unable to estimate the train position from the beginning on. This is not surprising when we look at Figure 7.12 that shows the rotated railcar data and the corresponding map data from the steam engine obtained with the GNSS ground truth. The rotated data has large offsets and is scaled relative to the map. What was somewhat surprising is the poor performance of the pre-calibrated filter. Until ≈ 270 s the error of the pre-calibrated filter is close to the errors of the SLAC filters. The good performance in the very beginning was expected since the pre-calibrated filter already starts with known calibrations parameters, whereas the SLAC filters first have to estimate them. We are not entirely sure why the error of the pre-calibrated filter suddenly starts to increases rapidly and the filter fails to estimate an accurate position. We suspect that this behavior has to do with some deviations from the assumption of a linear calibration model. More details on this topic are discussed further below when we have a look on the calibration parameters estimated by the SLAC algorithms.

The results in Figure 7.11 show that both SLAC algorithms achieve a similar position error with a small advantage for the reduced SLAC algorithm, e.g., in the beginning the reduced SLAC algorithm is able to achieve a smaller error and a faster convergence than the full SLAC algorithm. The slower convergence of the full SLAC algorithm could be explained by the additional degrees of freedom in the calibration parameters. The filter simply has to process more magnetometer data to get an accurate estimate of all twelve parameters. The advantage of the reduced SLAC filter is also reflected in the overall position RMSE. The full SLAC filter has an average RMSE, over all Monte Carlo runs, of 2.35 m and the reduced SLAC filter of 1.76 m. The highest observed RMSE is 2.70 m for full SLAC and 2.07 m for reduced SLAC. For both SLAC filters the RMSE over the different Monte Carlo runs is rather consistent and shows a spread, according to (7.5), of 31.49 % and 33.37 % for the full and reduced SLAC filter. In Figure 7.11 both SLAC variants show a larger error around 330 s-350 s. The larger error is caused by the absence of strong variations in the magnetic field data, as can be seen from Figure 7.12. This is a general phenomenon in magnetic localization and intuitively plausible, when there are no variations in the magnetic field or the variations are in the range of the sensor noise, the position cannot be estimated accurately. In a practical application this increase of the error can be mitigated by incorporating additional sensors in the estimation, e.g., in the railway domain a simple odometer sensor can help a magnetic localization system to bridge parts of the track with a “flat” and featureless magnetic field. Due to the generic nature of the particle filters used in this thesis, additional sensors can be easily integrated into the prediction or update step.

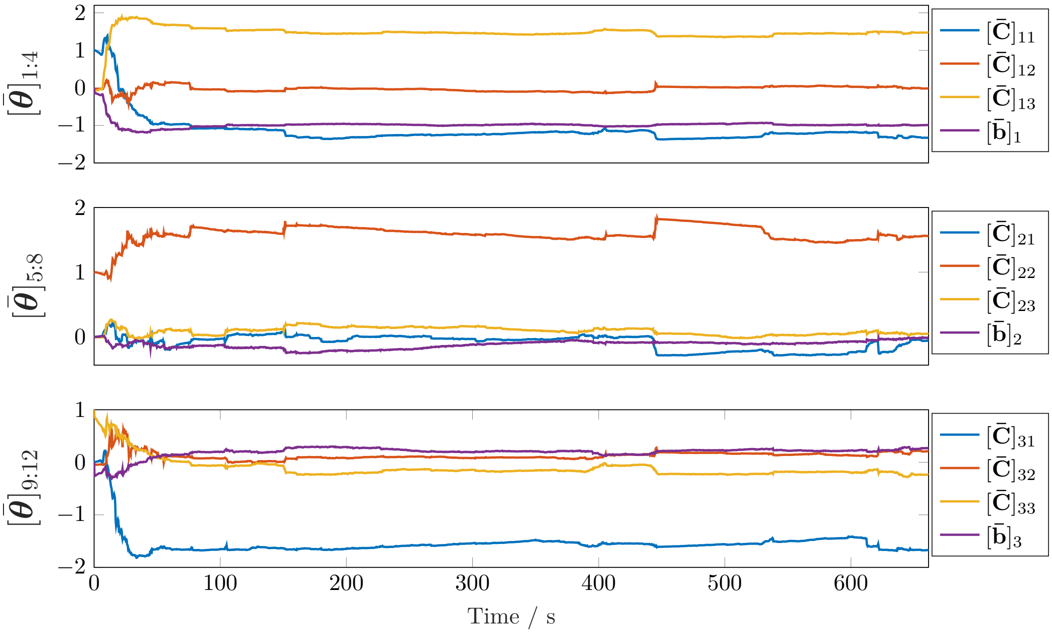

Figure 7.13: All twelve calibration parameters estimated by the full SLAC algorithm.

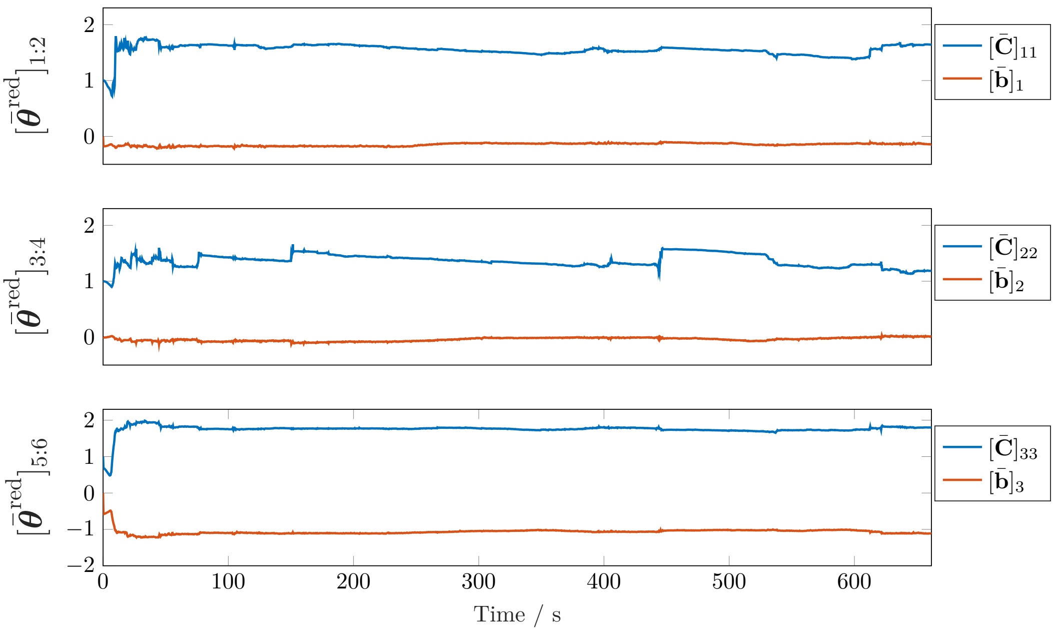

Figure 7.14: Scale factors and biases estimated by the reduced SLAC algorithm.

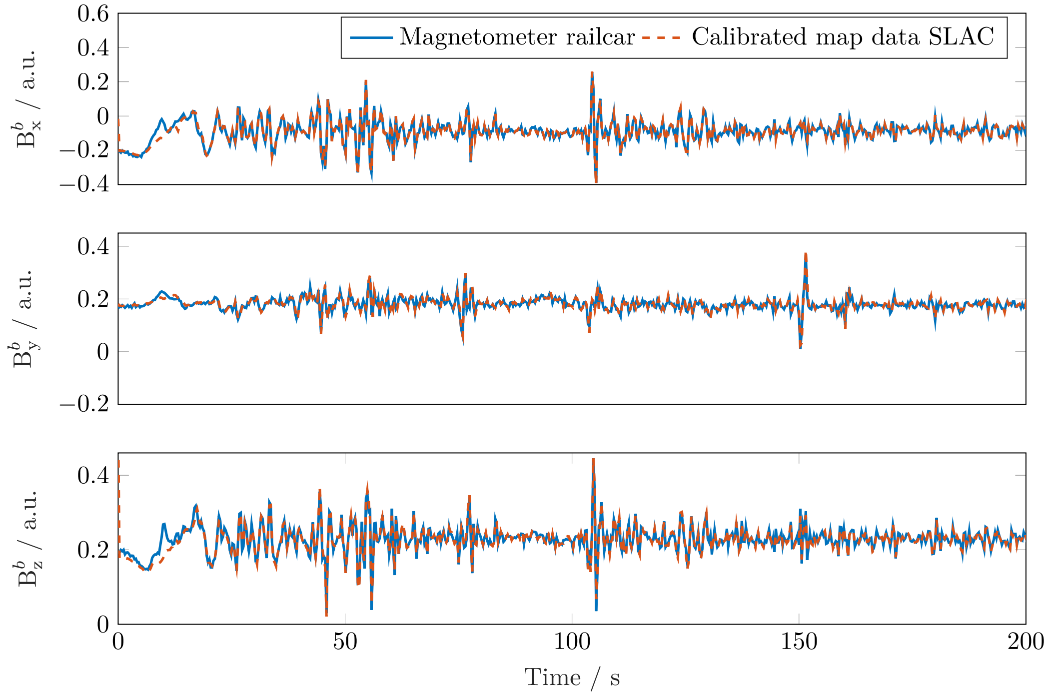

Figure 7.15: Diesel railcar data and the map data calibrated with the parameters estimated by the full SLAC algorithm. The shown calibrated map data is obtained with the MMSEEs of the along-track position and calibration parameters.

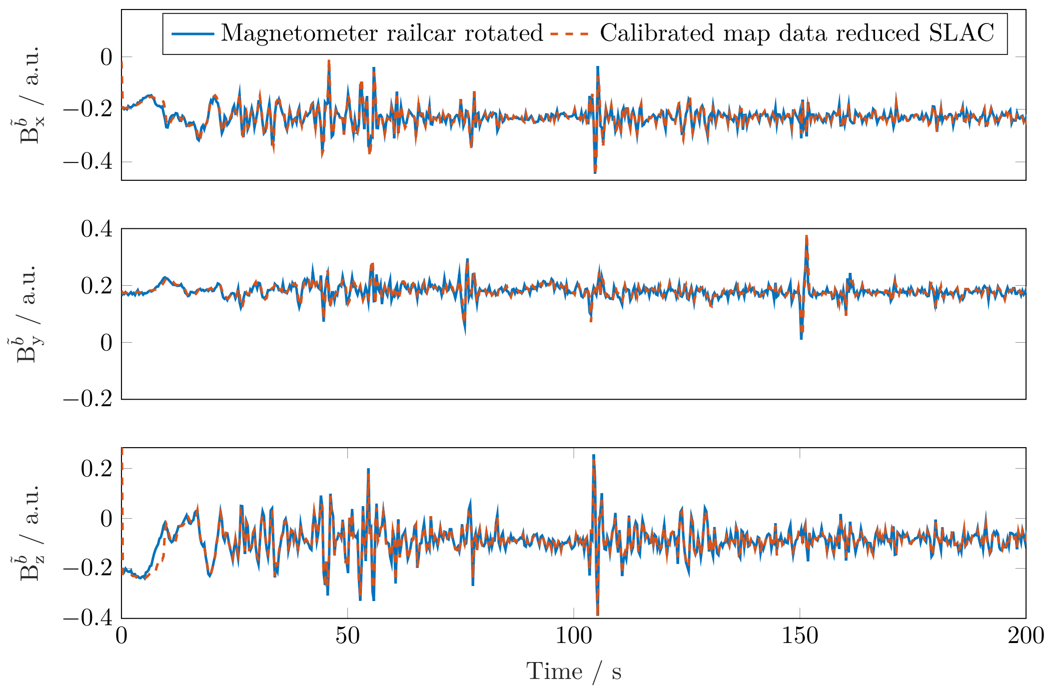

Figure 7.16: Diesel railcar data rotated into the body frame of the steam engine and the map data calibrated with the parameters estimated by the reduced SLAC algorithm. The shown calibrated map data is obtained with the MMSEEs of the along-track position and calibration parameters.

Calibration Parameters

The MMSEE of the calibration parameters obtained from the full and the reduced SLAC algorithm is shown in Figure 7.13 and Figure 7.14, respectively. In addition, Figure 7.15 and Figure 7.16 show the data handed to the filters and the map data calibrated with the estimated parameters and positions.

For both filters the estimated calibration parameters are not converging towards constant values but keep changing during the train run. This is in contrast to the results in our paper [J1], in which the full SLAC approach was evaluated with the data of a model train running in a laboratory environment and in which the estimated parameters did not show such strong variations. Since the same sensor type was used in [J1] and during the HSB measurement campaign, we believe that the observed changes in the estimated parameters are not caused by the sensor itself but by external effects and a mismatch of the model used for calibration, not present in the controlled laboratory environment of [J1].

The magnetometer model (5.14) assumes a perfect linear relation between the data in the map and the measured data. This is assumption is not exactly fulfilled during the HSB measurements, e.g., the magnetometers in the steam engine and in the railcar are not perfectly aligned in height, as explained in Section 7.1.2. The height difference alone introduces differences in the measured magnetic field that might not be easily compensated by a linear model. The same holds true for differences in the mounting position in the cross-track direction. Examples for the influence of the mounting position were also discussed Section 3.3.2. In addition, for the linear model to be correct, the environment in which the magnetometers are mounted has to be completely static and there should be no moving magnetic material in the vicinity. For sensors mounted in a train with rotating wheels, gearboxes, and engines this requirement cannot be fulfilled perfectly. Another possible complication for the estimation of the calibration parameters are imperfections in the map. These imperfections can be caused by erroneous GNSS positions and disturbed magnetometer measurements. An inaccurate GNSS measurement introduces shifts in the magnetic map, which can lead to the situation in which the filter expects a certain magnetic value at its current position, but gets a slightly different value. In order to match its measurements to the map, the filter then adapts its calibration parameters to compensate for the erroneous map. A comparison of the estimated parameters in Figure 7.13 and Figure 7.14 with the measurement data in Figure 7.12 reveals that the larger changes in the parameters occur mainly when there are larger distortions in the magnetic data. In the absence of strong distortions, the parameters are approximately constant or drift slowly. As soon as the SLAC filters get measurements with strong distortions, errors in the parameters and the position cause larger residuals in the Kalman and particle filter update step, which leads to a stronger adaption of the parameters in order to reduce the residuals.

Nevertheless, the SLAC filters are able to track the train position through the whole run, while the pre-calibrated filter could not cope with the above mentioned imperfections. This shows that the continuous estimation of the parameters in the SLAC algorithms not only enables the use of the same magnetic map for the localization of different trains, but it also enables magnetic localization in the presence of changing calibration parameters and imperfect conditions.

To check how well the calibrations parameters fit to the measurements and the map, Figure 7.15 and Figure 7.16 show the first 200 s of the railcar data handed to the filters and the corresponding calibrated data from the steam engine magnetic map. The shown map data was retrieved at the MMSEE of the along-track position and the calibration uses the MMSEE of the parameters. Note, the data for the full SLAC filter is the raw railcar magnetometer data, whereas the data shown for the reduced SLAC filter is the raw data rotated into the body frame of the map. In the first few seconds both figures show larger differences between the calibrated map data and the data handed to the filters. This is caused by the not yet fully estimated calibration parameters. After the initial larger differences, the calibrated map data nicely fits to the magnetometer data of the railcar, showing the feasibility of the SLAC algorithms to perform proper calibration.

7.3.5 Summary

In this section the full and the reduced SLAC algorithm was evaluated based on the data collected during the HSB measurement campaign. Both filters were able to estimate the train position over the complete train run with meter-level accuracy. In addition, it could be shown that the estimated parameters enable the calibration of the magnetic map of the steam engine such that it closely matches the magnetometer data recorded with the diesel railcar.

Due to imperfections in the measurements, the hardware setup, and the resulting violations of the linear model assumption, the estimated calibration parameters changed over the course of the train run. The changes in the estimated parameters are in contrast to the result in [J1], in which SLAC was evaluated under controlled conditions with measurements from a model train driving in a laboratory. However, the presented results for the HSB measurements show that the proposed SLAC algorithms are flexible enough to enable accurate localization under sub-optimal conditions.

7.4 Evaluation of BCRLB for Magnetic Train Localization

In this section, the GP-based BCRLB from Chapter 6 is applied to data from the BRB measurement campaign to investigate the theoretically achievable position accuracy. The calculated bound is compared to the MSE of a particle filter that uses the same GP as the bound in its measurement model as map. The comparison between the bound and the particle filter allows us to check if the filter is optimal in the MSE sense and can be also seen as sanity check if the bound gives reasonable values. The results in this chapter are mainly based on our paper [C8] with some minor changes and corrections that will be introduced in the remainder of this section.

7.4.1 GP Training

Before the bound can be calculated, the GP has to be trained on a given data set and the map for the particle filter has to be generated. For the following evaluation a 1 km long track segment from the BRB measurements campaign, described in 7.1.1, has been selected. This 1 km long segment is a part of track T45 from the track network shown in Figure 7.2. The chosen data set length is a tradeoff between the computational complexity to train and evaluate the GP and the informative value of the bound derived from it.

Table 7.4: GP hyperparameters obtained from optimization.

Axis	Sensor noise σ^{mag} / a.u.	Length Scale l / m	Kernel Variance σ^{k} / a.u.	Mean $\bar{m}$/a.u.
x	$2.64 \cdot 10^{-3}$	3.00	$1.06 \cdot 10^{-2}$	$1.75 \cdot 10^{-4}$
y	$4.56 \cdot 10^{-3}$	4.93	$1.69 \cdot 10^{-2}$	$3.75 \cdot 10^{-3}$
z	$3.50 \cdot 10^{-3}$	3.43	$2.00 \cdot 10^{-2}$	$1.62 \cdot 10^{-3}$

The data set used in the GP contains the three components of the magnetic vector field and the associated along-track position. For simplicity, the data set contains the magnetometer data in the vehicle frame, which is used also in Section 7.2, as the train orientation is assumed to be known and rotating the data into the track frame does not affect the bound. Similar to the evaluation of the different filter algorithms for magnetic localization in 7.2, the 16.7 Hz magnetic field induced by the overhead power line current is filtered out with the algorithm introduced in Section 7.2.1. The data set combines measurements of four runs over track T45. In particular, we use the data of run 1, 2, and 4 from Table 7.1. The fourth run was used before only for generating a part of the magnetic map. We combined here the data of four runs to introduce more uncertainty in the GP data and to avoid too optimistic covariance and sensor noise values in the training and regression phase. With the measurements from only one run on T45 the GP might be fitted to temporary disturbances that are not position dependent and that do not contain position information.

The position associated to each magnetometer measurement is obtained from GNSS measurements, which means the data across the different runs is not perfectly aligned. A miss alignment between the data sets leads to differences between magnetometer measurements associated to the same position that do not represent the true magnetic field or measurement noise. To get rid of these additional differences, we perform a fine alignment across the four runs. For the alignment we interpolate the magnetic data on an equidistant spatial grid with spacing 0.1 m and shift the interpolated data such that the cross correlation between the different runs is maximized.

A comparison of the magnetic field recorded during the four runs also revealed offsets between the different runs. Since these offsets can be compensated by proper calibration, e.g., with the proposed SLAC algorithms, and a constant offset does not contribute information to the bound, the mean of the magnetic vector field of each run was removed before its data was handed to the GP training.

Although the bound calculation is limited to a 1 km track segment, the computational complexity would be high and impractical if all available magnetometer data would be used in the GP due the high magnetometer sampling rate. Therefore, the amount of data considered in the GP is reduced in two steps. In the first step, spatial down-sampling is performed. In the spatial down-sampling step an equidistant grid with spacing 0.5 m is placed along the track and for each run only the data points closest to a grid point are stored and the rest is discarded. Given the spatial variability of the magnetic field, discussed in Section 3.1, and its overall smoothness a data point every 0.5 m is sufficient to accurately represent the magnetic field during GP training. In the second step, the number of data points is reduced once more by selecting for every grid point the data point of only one of the four

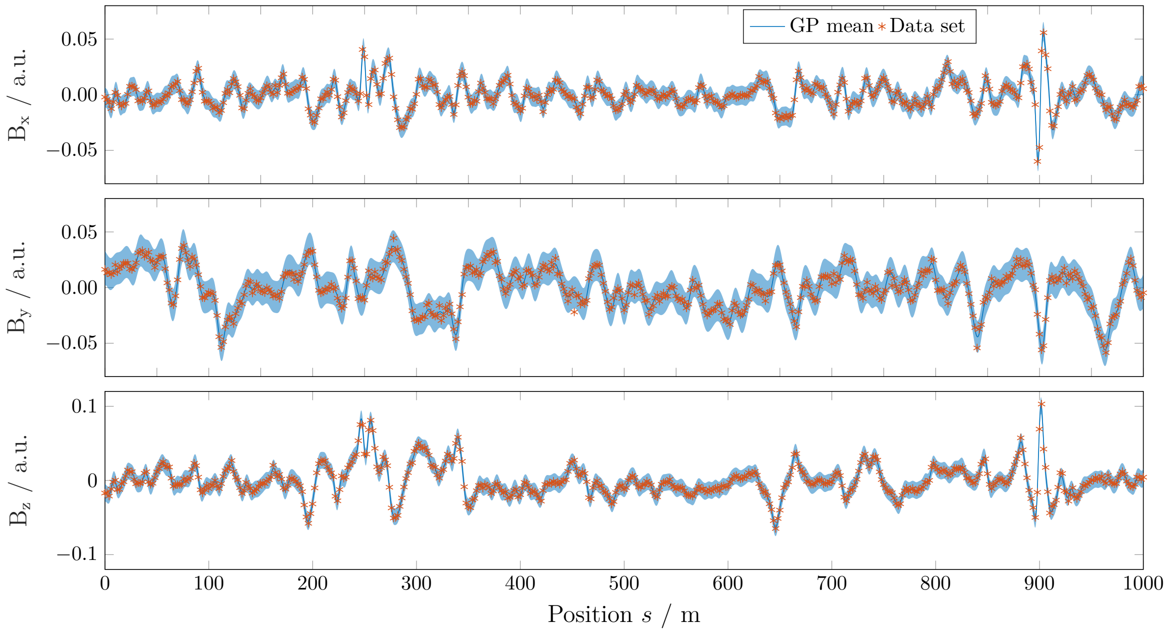

Figure 7.17: GPs representing the three components of the magnetic vector field of the 1 km long track segment. The solid line represents the posterior mean and the shaded area three time the standard deviation including sensor noise. The training data used by the GP regression is shown as red asterisks. The figure only shows every fourth data point to not clutter the plot. As before, the shown magnetic values are dimensionless since the used Xsens sensor performs internal normalization.

runs at random. With the second step three quarters of the remaining data is removed while all runs still contribute to the GP training and the differences between them are kept.

Based on the down-sampled data set, the GP training is performed. During the training the hyperparameters of the squared exponential kernel, the constant mean function, and the Gaussian sensor noise are optimized. For optimization we used the Matlab implementation of the GPML toolbox [99]. The optimization was performed individually for each component of the magnetic vector field and used only the data of the first 500 m to speed up the optimization. The hyperparameters obtained from the optimization are listed in Table 7.4 and the resulting GP is shown in Figure 7.17. The hyperparameters slightly changed w.r.t. the values presented in [C8] because here we changed the filter method that removes the 16.7 Hz magnetic field to the method presented in Section 7.2.1. Furthermore, the fine alignment is now done based on a finer grid spacing of 0.1 m. In [C8] the alignment was done after the spatial down-sampling with a coarser resolution.

7.4.2 Particle Filter Implementation

The particle filter that will be compared to the bound uses the same motion (6.36) and measurement (6.39) model as the bound. To compare the filter performance to the bound, we have to determine the MSE of the filter. The MSE matrix is given by the left hand side of (6.1) and is defined as the expectation of the estimation error w.r.t. the joint density of the state and the observations used in the estimation. Calculating the MSE therefore requires the evaluation of the filter over multiple realizations of the process and measurement noise. Here the MSE is calculated based on a Monte Carlo simulation. In the simulation different trajectories for the train position and speed are generated with

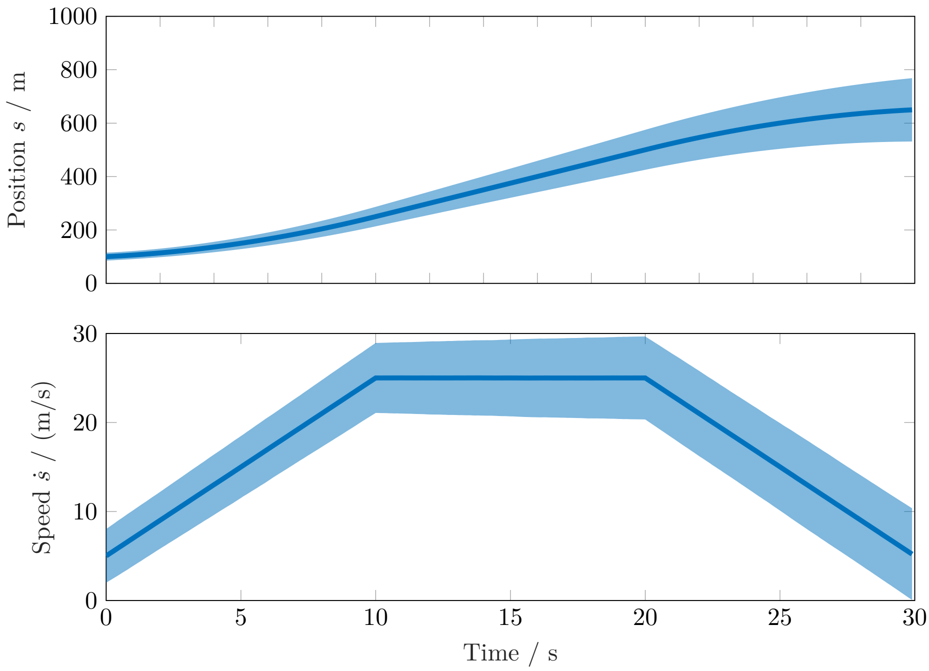

Figure 7.18: Distribution of state trajectories used in the calculation of the BCRLB and the filter MSE. The blue solid line is the mean and the shaded area three times the standard deviation.

(6.36). For each of these trajectories then multiple realization for the corresponding measurements are generated with (6.39).

In the particle filter implementation, we decided to not directly include the GP in the update step because this would mean that in each time step a GP regression has to be performed for all particle positions. In combination with the Monte Carlo simulation required for MSE calculation, this would lead to a high computational demand. To speed up the MSE calculation, the GP mean and covariance are evaluated once offline along an equidistant grid spanning the whole track segment. For each grid point the resulting mean and variance are stored in an array, similar to the maps used to evaluate the localization algorithms in the previous sections of this chapter. Thus, in the particle filter update step the GP regression for each particle is replaced with a simple look-up of the closest value in the array. The spacing between the grid points is set to 5 mm to minimize the impact of quantification effects on the particle filter estimation accuracy. Since here we are not interested in the real-time capability of the filter, but solely in its achievable accuracy, the number of used particles was increased, compared to the filters in the previous sections, to $N_{\mathrm{p}} = 10000$. Finally, we would like to mention that in the implementation of [C8] the variance in the weight update step was set to a too high value. This mistake led to a larger gap between the bound and the filter. This mistake has been corrected in the following results and the filter is now close to optimal.

7.4.3 Simulation Parameters

The results discussed in the next section are obtained from a Monte Carlo simulation based on the motion and measurement models in (6.36) and (6.39). Overall 2000 state trajectories are generated and for each of the trajectories 50 realizations of the corresponding measurements are generated.

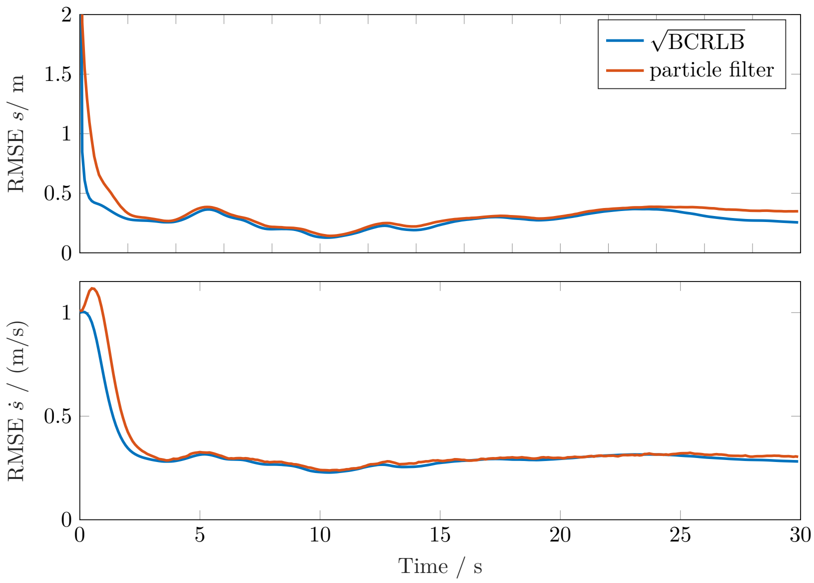

Figure 7.19: Square root of the position and speed BCRLB, and the RMSE of the particle filter. For the position RMSE the y-axis is zoomed into the area 0–2 m. The bound starts at 5 m as defined by the covariance matrix of the prior.

Therefore, the filter is evaluated one hundred thousand times and from the results then the MSE is calculated. The bound only requires a Monte Carlo simulation of the state to calculate the expected FIM in $\mathbf{D}_k^{(22)}$, as can be seen from (6.38). Taking the expectation over the measurements is not necessary since this is already taken care of by the FIM. The reduction of the Monte Carlo simulation to the simulation of the state trajectories is one of the advantages of the bound that makes its calculation faster than the brute force MSE calculation of the filter.

In the Monte Carlo simulation 300 time steps with a period of $T = 0.1\,\mathrm{s}$ are simulated, which results in a $30\,\mathrm{s} - T$ long train run. During the run the train is accelerating in the first 100 time steps with $a_k = 2\,\mathrm{m\,s^{-2}}$. In the next 100 time steps the train drives with a nearly constant speed and the acceleration input is set to $a_k = 0\,\mathrm{m\,s^{-2}}$. For the remainder of the run the train brakes and the input is set to $a_k = -2\,\mathrm{m\,s^{-2}}$. For the bound, as well as for the particle filter, the prior distribution of the state is given by a Gaussian with mean $\boldsymbol{\mu}_{\mathbf{d}_0^{\mathrm{B}}}^{\top} = [100\,\mathrm{m} \;\; 5\,\mathrm{m\,s^{-1}}]$. For the bound the initial covariance is set to $\boldsymbol{\Sigma}_{\mathbf{d}_0^{\mathrm{B}}} = \mathrm{diag}([25\,\mathrm{m}^2 \;\; 1\,\mathrm{m}^2\,\mathrm{s}^{-2}])$. For the filter the initial covariance is doubled in value to ensure that the particle cloud covers the true initial position and speed. The measurement noise variance depends on the sensor axis and corresponds to the value found by the GP hyperparameter optimization shown in Table 7.4. For the process noise standard deviation the value is set to $q^{\mathrm{B}} = 0.0625\,\mathrm{m}^2\,\mathrm{s}^{-3}$. From our point of view there is no exact rule how to choose the process noise. Here we selected a value small enough to avoid erratic trajectories while still having enough uncertainty in the movement model to show the gain in accuracy when the magnetic field information is incorporated into the estimation. To give an impression of the simulated scenario, Figure 7.18 shows the distribution of the generated state trajectories.

7.4.4 Results

Figure 7.19 shows the results of the bound and the particle filter for the Monte Carlo simulation described above. The shown values are the filter RMSE and the square root of the BCRLB for the train position and speed. The RMSE of the filter is calculated based on the error between its MMSEE and the true state of the simulated trajectories. The initial value of the bound is given by the covariance of the prior and hence the square root of the bound starts at 5 m for the position and at $1\,\mathrm{m\,s^{-1}}$ for the speed.

From Figure 7.19 we can see that the initial value of the bound is quickly reduced as soon as the information of the magnetic field is incorporated into the bound. Without the magnetic field the uncertainty in the motion model would lead to a monotonic increase of the bound as can be seen from the increasing variance of the simulated trajectories in Figure 7.18. A similar behavior can be observed for the filter, after the filter updates its particle weights with the magnetic data the uncertainty in the estimated position and speed is quickly reduced. We should note here that the speed is not directly observable from the magnetic field since the FIM w.r.t. the speed is zero. The observability is here only indirect and caused by the motion model that links the position over multiple time steps to the speed.

After the initial higher uncertainty is reduced, the bound is in the range of ≈ 0.13–$0.37\,\mathrm{m}$ for the position and ≈ 0.23–$0.32\,\mathrm{m\,s^{-1}}$ for the speed. The particle filter follows the bound closely and has in general a similar shape. For a few time steps the filter RMSE is below the bound, which theoretically is impossible. This behavior can be explained by the numerical and finite nature of the Monte Carlo simulation. The bound only holds when the MSE considers all possible realizations of the process and measurement noise. If only one realization is considered, the filter error can be considerably smaller than the bound. Since we are limited here to a finite amount of noise realizations, the bound is no longer strict, but we still can expect that an optimal filter has a MSE close to the bound as it is the case here.

The RMSE of the filter is in the range of ≈ 0.14–$0.39\,\mathrm{m}$ for the position and ≈ 0.24–$0.33\,\mathrm{m\,s^{-1}}$ for the speed. Formally the filter therefore is not optimal in terms of its MSE, although we simulated here an ideal scenario. However, considering the numerical uncertainty introduced by the finite Monte Carlo simulation the particle filter still performs close to optimal and it is questionable if better estimators exist. This argument is also supported by theory that does not guarantee that an estimator can attain the BCRLB due to some optimistic assumptions in the bound, e.g., the FIM does not account for ambiguities in the likelihood. The bound only looks at the derivatives of the likelihood, but does not consider that a certain magnetic value might not be unique. Thus, ambiguities in the magnetic field can lead to a disadvantage for concrete estimators compared to the bound because the estimators have to cope with the ambiguities.

7.4.5 Summary

In this section the BCRLB for magnetic field-based train localization from Chapter 6 was evaluated based on the BRB data set described in 7.1.1. With the BRB data set an individual GP for each component of the magnetic vector field was trained. The resulting GPs were then used to calculate the bound for a $\approx 30\,\mathrm{s}$ long train run. In addition to the bound, the RMSE of a particle filter was calculated based on data simulated from the GPs. The comparison of the filter and the bound revealed that the filter does not attain the bound in most cases and therefore is not optimal in the MSE sense. However, the filter followed the bound closely and by considering the optimistic assumptions made by the bound, and the numerical uncertainties introduced by the finite Monte Carlos simulation, the particle filter can be seen as an almost optimal choice for magnetic train localization.

Chapter 8

Conclusion and Outlook

8.1 Conclusion

Magnetic field-based train localization uses distortions of the Earth magnetic field introduced by magnetic material in the railway environment. The benefit of magnetic localization is its availability in GNSS-denied areas such as tunnels and train stations with closed roofs. But even in environments in which GNSS signals are available, magnetic localization can be beneficial since it provides an independent source of absolute position information. Thus, magnetic localization can help to increase the accuracy, availability, and integrity in the context of a multi-sensor localization system. This thesis had the goal to show the overall feasibility of magnetic field-based train localization and to identify and address problems encountered in its practical use. In order to achieve this goal, we introduced in Chapter 2 the theoretical foundation for the thesis. The general existence of magnetic distortions was shown in Chapter 3, in which we also discussed different properties of the distortions and investigated potential sources for disturbance.

Chapter 4 focused on the development of concrete Bayesian filter algorithms that can accurately estimate the train position and the correct track using only measurements of a train mounted magnetometer and a single axis accelerometer. The chapter started with a particle filter that assumes a known route for the train it wants to localize and that uses a straightforward weight update step based on a Gaussian likelihood. We then proposed two robust particle filter variants that explicitly account for disturbances in the magnetic map and the magnetometer measurements. The first filter variant tightly integrates an FDE scheme into its algorithm. The FDE scheme utilizes eight measurement models and calculates for each model its probability based on the magnetic map, the current particle set, and the newest magnetometer measurement. Based on the most probable model it is then decided which axes of the vector magnetometer are used to update the particle weights. This could mean that none, one, two, or all three of the available measurements are used to update the filter. In the case that all three axes of the magnetometer are disturbed, the filter does not update the weights with magnetic information and only updates them with the accelerometer measurement. The second robust filter variant does not make a hard decision and does not exclude measurements from the update step but rather makes a soft decision and reduces the influence of a measurement on the particle weights. To this end, the filter uses two different models for the sensor noise, one for the nominal case without disturbances, and one with a broader likelihood for the case of disturbed magnetometer measurements. For both models a set of intermediate particle weights are calculated that are then combined dependent on the model probabilities into the final weights of the particle filter.

The different particle filter variants were then extended to account for the case of unknown train routes. This means the filter has to also estimate and identify the track the train is currently driving on. For track identification two approaches were proposed. In the first approach, a particle reaching a switch is randomly allocated to the left or the right track. In the second approach, a copy of the particle filter is created as soon as one of its particles reaches a switch. Therefore, one filter per possible track and associated route is computed until the correct track is identified. To investigate the performance of the different filters and approaches for track identification, an evaluation was performed based on six runs on a local track network in Germany between Augsburg and Friedberg. During the six runs all filters achieved a position RMSE below 5 m and were able to identify all tracks correctly. However, the evaluation also showed an advantage of the robust filters that achieved a better accuracy and consistency when disturbances in the measurements were present.

In Chapter 4 the data for the magnetic map and the data used for localization was recorded with the same train and sensor hardware. This allowed us to pretend in the filter update step that the magnetometer was calibrated. In practice it is unrealistic to assume to have a magnetic map recorded with the setup used also for localization since this would require a dedicated map for each train. Therefore, magnetometer calibration is a key component for making magnetic field-based train localization feasible in a practical setting. Considering that for magnetometers a variety of calibration methods have been proposed over the years this seems to be not a problem. However, common calibration methods require a rotation of the magnetometer in a homogeneous field while mounted on the platform on which it will be used. For a train-mounted magnetometer this is not possible without larger efforts due to obvious reasons. To address this problem, in Chapter 5 we proposed an algorithm that estimates the calibration parameters and the train position simultaneously. This is possible because when the train moves along a track all axes of the magnetometer are sufficiently excited by the distortions in the Earth magnetic field to render the parameters observable. The rotation of the sensor in a homogeneous field is therefore replaced by a movement in a inhomogeneous field. In order to run SLAC in real-time, we proposed a Rao-Blackwellized particle filter that exploits the linear relation between the calibration parameters and the magnetometer measurements when conditioned on the train position. The filter therefore only has to represent the train state by particles while the parameters can be estimated with a Kalman filter. To show the feasibility of the proposed SLAC filter, an evaluation was conducted on the basis of train measurements with a diesel railcar and a steam engine driving on a rural track network in Germany. For the evaluation a magnetic map was created from the steam engine measurements that was then used to localize the diesel railcar. In the evaluation two SLAC filter versions were implemented. The first version estimated all calibration parameters, while the second version only estimated a scale factor and bias for each magnetometer axis. For both filters the results showed that they are capable of estimating the calibration parameters while maintaining a meter-level accuracy for the position.

Finally, in Chapter 6 a BCRLB for magnetic field-based train localization was derived. The BCRLB is a lower bound on the MSE and can be used as a benchmark for filter algorithms or to get an idea of the achievable position accuracy for a given magnetic field. The main problem encountered in the derivation of the BCRLB was the lack of a differentiable analytical model for the magnetic field map. Getting such a analytical model in practice is nearly impossible since this would require the knowledge of the properties of all magnetic material in the environment of the track network. However, even with perfect knowledge about the material properties the equations for the magnetic field might not be tractable in closed form. In this thesis, we therefore used an approximation of the magnetic field derived from a magnetic data set recorded along a railway track. Concretely, a GP was used to approximate the magnetic field since its Bayesian formulation fits well into the BCRLB framework and accounts for the noise in the magnetic data set. The bound was evaluated based on a GP that represents a 1 km long track segment. Alongside the bound also the MSE of a particle filter was obtained from a simulation based on the same GP. This served two purposes. First, the filter MSE is a sanity check if the bound is calculated correctly and second, it can be checked how well the filter performs. The results of the bound and the filter showed that from a theoretic point of view, magnetic localization enables an accuracy in the decimeter range. For the evaluated scenario the filter was not optimal in the MSE sense and it could not attain the bound. Nevertheless, the filter came close to the bound and was able to follow it closely during the whole simulation.

Overall, we showed in this thesis that the magnetic field along a railway track enables track-selective train localization with meter-level accuracy. For this purpose, different localization algorithms were developed and evaluated based on data recorded in two different track networks. In the algorithm development important issues, such as magnetometer calibration and robustness against disturbed measurements, were explicitly considered. Furthermore, a lower bound on the MSE was derived to provide a theoretical tool for filter evaluation and the investigation of the achievable position accuracy.

8.2 Outlook

This thesis showed the overall feasibility of magnetic field-based train localization and proposed algorithms to address some challenges encountered in its practical use. However, there are still open questions and issues that encourage further research.

One of the major topics of interest that should be explored in the future is the creation and maintenance of magnetic maps. In particular, the creation of maps in the absence of a reliable position reference system, such as GNSS, has to be addressed. Otherwise, the effort required to use magnetic localization in tunnels and train stations is severely increased. To address the mapping issue, SLAM methods seem to be an appropriate choice and should be applied and tested in the railway domain. In [C9] we made a first proposal for a graph-based SLAM method suitable for railways that could serve as a basis for further investigations in this direction.

Another important topic is magnetometer calibration since it is the key to use the same magnetic map across multiple trains. Here the proposed SLAC algorithms should be further evaluated and their limitations should be explored. Additionally, SLAC should be integrated into the algorithms that also estimate the track identifier when the train is passing a switch. Until now, SLAC was evaluated only for known routes. Looking beyond the railway use case, SLAC might be also useful in other domains since it can potentially enable magnetic localization for all platforms for which magnetometer calibration is not possible with common calibration methods. In [C7], we demonstrated the feasibility of SLAC for a small robot driving in a laboratory. Therefore, we could imagine that SLAC is also applicable to cars or even airplanes.

Finally, we see a large potential to further improve the proposed algorithms with respect to their accuracy and robustness by including additional sensor modalities into the estimation. These additional sensors can be something like laser scanners, radars, and odometers or just additional magnetometers. A magnetometer array for example opens new possibilities for position and speed estimation since it can measure the magnetic field on a short track segment instead of just a single point.

Appendix A

Proofs for Magnetic Field-based Train Localization

A.1 Model Probabilities of the Robust Particle Filter

A.1.1 Decomposition of Model Probability

$$
\begin{aligned}
\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k}, R_{1:N_{\text{route}}}) &= \frac{\mathrm{p}(\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k}, R_{1:N_{\text{route}}})}{\mathrm{p}(\mathcal{Z}_{1:k}, R_{1:N_{\text{route}}})} \\
&= \frac{\mathrm{p}(\mathcal{Z}_k|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})\,\mathrm{p}(\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})}{\mathrm{p}(\mathcal{Z}_k|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})\,\mathrm{p}(\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})} \\
&= \frac{\mathrm{p}(\mathcal{Z}_k|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})\,\mathrm{pr}(\mathcal{M}_k^{(j)}|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})}{\mathrm{p}(\mathcal{Z}_k|\mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}})}
\end{aligned} \tag{A.1}
$$

A.1.2 Practical Consideration on the Predicted Posterior

The predicted posterior is required to evaluate likelihood (4.30) in the model probability update (4.29). As discussed in Section 4.4.1, the calculation of the posterior

$$
\mathrm{p}(\mathcal{X}_k^{\text{do}}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\text{route}}}) \,, \tag{A.2}
$$

requires a bank of filters with an exponentially growing number of filters as time passes. Here we now elaborate on this topic and discuss how the predicted posterior can be obtained.

The posterior (A.2) is only conditioned on the current model at time step k but not the sequence of models that lead to it. The missing information about the model history implies that in the weight updates of the previous time steps the measurement model and thus the likelihood is not defined. It is therefore necessary to account for the model history when calculating (A.2). Once more we rely on marginalization for this which yields

$$\begin{aligned}\mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\mathrm{route}}}) &= \sum_{l=1}^{L} \mathrm{p}(\mathcal{X}_k^{\mathrm{do}}, \mathcal{M}_{0:k-1}^{(l)}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\mathrm{route}}}) \\ &= \sum_{l=1}^{L} \mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{M}_{0:k-1}^{(l)}, \mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\mathrm{route}}})\, \mathrm{pr}(\mathcal{M}_{0:k-1}^{(l)}|\mathcal{M}_k^{(j)}, \mathcal{Z}_{1:k-1}, R_{1:N_{\mathrm{route}}})\,, \end{aligned} \tag{A.3}$$

where the index l of the sum runs over all L possible model sequences $\mathcal{M}_{0:k-1}^{(l)}$. The second line shows that the posterior is in essence a weighted sum of L posteriors conditioned on model sequences that all end in $\mathcal{M}_k^{(j)}$. The weights of the sum are defined by the probability of the different sequences conditioned on the available measurements and the assumption that the j-th model is correct at time step k. Since the number of sequences L grows exponentially over time, also the number of required posteriors grows exponentially. Considering that each posterior has to be tracked by a separate filter, the need for a bank of filters becomes obvious.

A.2 Track Probability of the Joint Filter

The probabilities of the different tracks are directly obtained from the Dirac mixture approximation of the filter density by marginalization

$$\begin{aligned}\mathrm{pr}(I_k|\mathcal{Z}_{1:k}) &= \int \mathrm{p}(\mathbf{d}_k, I_k, O_k|\mathcal{Z}_{1:k}) d\mathbf{d}_k dO_k \\ &= \int \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \delta(\mathbf{d}_k - \mathbf{d}_k^{(i)}) \delta(O_k - O_k^{(i)}) \delta(I_k - {I_k}^{(i)}) d\mathbf{d}_k dO_k \\ &= \sum_{i=1}^{N_\mathrm{p}} w_k^{(i)} \delta(I_k - {I_k}^{(i)})\,. \end{aligned} \tag{A.4}$$

The last line in (A.4) is a pmf, in which the probability mass allocated to I_k is the sum of the weights of all particle for which $I_k^{(i)} = I_k$. Hence, we can alternatively write

$$\mathrm{pr}(I_k|\mathcal{Z}_{1:k}) = \sum_{j \in \{i | I_k^{(i)} = I_k\}} w_k^{(j)}\,. \tag{A.5}$$

Appendix B

Proofs for the Bayesian Cramér-Rao Lower Bound

This appendix contains some more detailed derivations related to the BCRLB for magnetic field-based localization that are not essential for the understanding of the idea presented in Chapter 6.

B.1 Recursive BCRLB

Due to the Markov property that gives rise to (6.16) the matrix $\mathbf{A}_{k+1}$ can be described as the block matrix (6.20) that creates a link between the BIM of different time steps. In the following, we will show the identity between the blocks in (6.13) and the blocks in (6.20). For the proof we first note that the BIM for the sequence $\mathbf{x}_{0:k}$ is, analogously to (6.10), given by the block matrix

$$\mathbf{J}_{0:k} = -\mathbf{E}_{\mathrm{p}_k}\left[\begin{bmatrix} \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} & \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_k} \\ (\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_k})^\top & \Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \end{bmatrix} \ln \mathrm{p}_k\right] = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^\top & \mathbf{C}_k \end{bmatrix} . \tag{B.1}$$

With the decomposition (6.16) of the joint pdf p_{k+1}, which is a result of the Markov property, it is straightforward to show that the upper left block $\mathbf{A}_{k+1}^{(11)}$ in (6.13) is identical to $\mathbf{A}_k$

$$\begin{aligned} \mathbf{A}_{k+1}^{(11)} &= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} \ln \mathrm{p}_{k+1}\right] \\ &= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} \left[\ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u) + \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u) + \ln \mathrm{p}_k\right]\right] \\ &= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} \ln \mathrm{p}_k\right] = -\mathbf{E}_{\mathrm{p}_k}\left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} \ln \mathrm{p}_k\right] = \mathbf{A}_k , \end{aligned} \tag{B.2}$$

where we have used in the last line that only p_k depends on the state sequence $\mathbf{x}_{0:k-1}$ and thus all other gradients are zero. Furthermore, the expected value in (B.2) can be calculated w.r.t. p_k rather than p_{k+1} because the argument does not depend on $\mathbf{x}_{k+1}$ and therefore $\mathbf{x}_{k+1}$ is marginalized out.

In a similar way it can be shown that the upper right block $\mathbf{A}_{k+1}^{(12)}$ in (6.13) is equal to $\mathbf{B}_k$

$$\begin{aligned} \mathbf{A}_{k+1}^{(12)} &= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_k} \ln \mathrm{p}_{k+1}\right] \\ &= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_k} \ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u) + \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u) + \ln \mathrm{p}_k\right] \\ &= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_k} \ln \mathrm{p}_k\right] = -\mathbf{E}_{\mathrm{p}_k}\left[\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_k} \ln \mathrm{p}_k\right] = \mathbf{B}_k , \end{aligned} \tag{B.3}$$

where in the last line again only the gradient of the term dependent on p_k is not vanishing. Finally, we use the Markov property once more to derive $\mathbf{A}_{k+1}^{(22)}$

$$\begin{aligned}
\mathbf{A}_{k+1}^{(22)} &= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \ln \mathrm{p}_{k+1}\right] \\
&= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathcal{D}_u) + \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u) + \ln \mathrm{p}_k\right] \\
&= -\mathbf{E}_{\mathrm{p}_{k+1}}\left[\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \ln \mathrm{p}(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathcal{D}_u)\right] - \mathbf{E}_{\mathrm{p}_k}\left[\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \ln \mathrm{p}_k\right] = \mathbf{D}_k^{(11)} + \mathbf{C}_k \,.
\end{aligned} \tag{B.4}$$

B.2 Derivatives of the GP Posterior Mean and Variance

In the following we present a proof for the derivatives of the GP posterior mean and covariance in (6.34). For the posterior mean the derivative w.r.t. the components of state vector $\mathbf{x}$ is directly obtained by substituting the definition of $\mathbf{K}_{\mathbf{x}_k\mathcal{D}}$ into (6.31)

$$\begin{aligned}
\frac{\partial}{\partial[\mathbf{x}]_i}\mu(\mathbf{x}_k, \mathcal{D}) &= \frac{\partial}{\partial[\mathbf{x}]_i}\left[\mathrm{m}(\mathbf{x}_k) + \mathbf{K}_{\mathbf{x}_k\mathcal{D}}\left(\mathbf{K}_{\mathcal{D}\mathcal{D}} + (\sigma^z)^2\mathbf{I}_{M\times M}\right)^{-1}\left(\mathbf{z}_{1:M}^{\mathcal{D}} - \mathbf{m}^{\mathcal{D}}\right)\right] \\
&= \frac{\partial}{\partial[\mathbf{x}]_i}\left[\mathrm{m}(\mathbf{x}_k) + \mathbf{K}_{\mathbf{x}_k\mathcal{D}}\underbrace{\mathbf{S}^{-1}\left(\mathbf{z}_{1:M}^{\mathcal{D}} - \mathbf{m}^{\mathcal{D}}\right)}_{=\tilde{\mathbf{z}}=\text{const.}}\right] \\
&= \frac{\partial}{\partial[\mathbf{x}]_i}\mathrm{m}(\mathbf{x}_k) + \frac{\partial}{\partial[\mathbf{x}]_i}\left[\mathrm{k}(\mathbf{x}_k, \mathbf{x}^{\mathcal{D},(1)}) \quad \cdots \quad \mathrm{k}(\mathbf{x}_k, \mathbf{x}^{\mathcal{D},(M)})\right]\tilde{\mathbf{z}} \\
&= \frac{\partial}{\partial[\mathbf{x}]_i}\mathrm{m}(\mathbf{x}_k) + \sum_{l=1}^{M}[\tilde{\mathbf{z}}]_l \frac{\partial}{\partial[\mathbf{x}]_i}\mathrm{k}(\mathbf{x}_k, \mathbf{x}^{\mathcal{D},(l)}) \,.
\end{aligned} \tag{B.5}$$

In the proof for the posterior variance derivative, we use the symmetry of the covariance function $\mathrm{k}(\cdot,\cdot)$. Symmetry here means that the arguments of the function can be interchanged without altering the value of the function, thus $\mathrm{k}(\mathbf{x}, \mathbf{x}^*) = \mathrm{k}(\mathbf{x}^*, \mathbf{x})$ holds for all covariance functions due to the required symmetry of the resulting covariance matrix. In the derivation of (6.34), we only have to consider the posterior variance of the GP from (6.31) because the noise variance $(\sigma^z)^2$ is independent from $\mathbf{x}$

$$\frac{\partial}{\partial[\mathbf{x}_k]_i}\mathrm{r}^{\mathrm{c}}(\mathbf{x}_k, \mathcal{D}) = \frac{\partial}{\partial[\mathbf{x}_k]_i}\mathrm{r}(\mathbf{x}_k, \mathcal{D}) \,. \tag{B.6}$$

For the derivative of the GP posterior variance $\frac{\partial}{\partial[\mathbf{x}_k]_i}\,\mathrm{r}(\mathbf{x}_k,\mathcal{D})$ we insert the definitions of the involved matrices into (6.31) and write the vector matrix products with sums, which yields

$$
\begin{aligned}
\frac{\partial}{\partial[\mathbf{x}_k]_i}\,\mathrm{r}(\mathbf{x}_k,\mathcal{D}) &= \frac{\partial}{\partial[\mathbf{x}_k]_i}\left[\mathrm{k}(\mathbf{x}_k,\mathbf{x}_k)-\mathbf{K}_{\mathbf{x}_k\mathcal{D}}\left(\mathbf{K}_{\mathcal{D}\mathcal{D}}+(\sigma^z)^2\mathbf{I}_{M\times M}\right)^{-1}\mathbf{K}_{\mathcal{D}\mathbf{x}_k}\right] \\
&= \frac{\partial}{\partial[\mathbf{x}_k]_i}\,\mathrm{k}(\mathbf{x}_k,\mathbf{x}_k)-\frac{\partial}{\partial[\mathbf{x}_k]_i}\left[\mathbf{K}_{\mathbf{x}_k\mathcal{D}}\mathbf{S}^{-1}\mathbf{K}_{\mathcal{D}\mathbf{x}_k}\right] \\
&= \frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}_k)}{\partial[\mathbf{x}_k]_i}-\frac{\partial}{\partial[\mathbf{x}_k]_i}\left[\begin{bmatrix}\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(1)}) & \cdots & \mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(M)})\end{bmatrix}\mathbf{S}^{-1}\begin{bmatrix}\mathrm{k}(\mathbf{x}^{\mathcal{D},(1)},\mathbf{x}_k)\\ \vdots \\ \mathrm{k}(\mathbf{x}^{\mathcal{D},(M)},\mathbf{x}_k)\end{bmatrix}\right] \\
&= \frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}_k)}{\partial[\mathbf{x}_k]_i}-\frac{\partial}{\partial[\mathbf{x}_k]_i}\left[\sum_{j=1}^{M}\mathrm{k}(\mathbf{x}^{\mathcal{D},(j)},\mathbf{x}_k)\sum_{l=1}^{M}\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(l)})[\mathbf{S}^{-1}]_{lj}\right] \\
&= \frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}_k)}{\partial[\mathbf{x}_k]_i}-\underbrace{\sum_{j=1}^{M}\sum_{l=1}^{M}\frac{\partial}{\partial[\mathbf{x}_k]_i}\left[\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(j)})\,\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(l)})[\mathbf{S}^{-1}]_{lj}\right]}_{\mathrm{s}(\mathbf{x}_k)}\,,
\end{aligned} \tag{B.7}
$$

where in the last line the symmetry of the covariance function is used. In order to obtain the derivative in (6.34), we have to show that the double sum is equal $2\left(\frac{\partial\mathbf{K}_{\mathbf{x}_k\mathcal{D}}}{\partial[\mathbf{x}_k]_i}\right)\mathbf{S}^{-1}\mathbf{K}_{\mathcal{D}\mathbf{x}_k}$. With the product rule we get

$$
\begin{aligned}
\mathrm{s}(\mathbf{x}_k) = &\sum_{j=1}^{M}\sum_{l=1}^{M}\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(j)})\frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(l)})}{\partial[\mathbf{x}_k]_i}[\mathbf{S}^{-1}]_{lj} \\
&+\sum_{j=1}^{M}\sum_{l=1}^{M}\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(l)})\frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(j)})}{\partial[\mathbf{x}_k]_i}[\mathbf{S}^{-1}]_{lj}\,.
\end{aligned} \tag{B.8}
$$

Due to the symmetry of the covariance matrix $\mathbf{S}$ the elements of the matrix and of its inverse fulfill the equation $[\mathbf{S}]_{lj}=[\mathbf{S}]_{jl}$. With the symmetry of $\mathbf{S}$ it can be seen that both double sums in (B.8) are equal by interchanging the indices l and j of one of the double sums. Thus, $\mathrm{s}(\mathbf{x}_k)$ becomes

$$
\begin{aligned}
\mathrm{s}(\mathbf{x}_k) &= 2\sum_{j=1}^{M}\sum_{l=1}^{M}\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(j)})\frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(l)})}{\partial[\mathbf{x}_k]_i}[\mathbf{S}^{-1}]_{lj} \\
&= 2\sum_{j=1}^{M}\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(j)})\left[\sum_{l=1}^{M}\frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(l)})}{\partial[\mathbf{x}_k]_i}[\mathbf{S}^{-1}]_{lj}\right] \\
&= 2\begin{bmatrix}\frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(1)})}{\partial[\mathbf{x}_k]_i} & \cdots & \frac{\partial\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(M)})}{\partial[\mathbf{x}_k]_i}\end{bmatrix}\mathbf{S}^{-1}\begin{bmatrix}\mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(1)})\\ \vdots \\ \mathrm{k}(\mathbf{x}_k,\mathbf{x}^{\mathcal{D},(M)})\end{bmatrix}\,,
\end{aligned} \tag{B.9}
$$

which is equivalent to the desired quadratic form in (6.34)

$$\mathrm{s}(\mathbf{x}_k) = 2\left(\frac{\partial \mathbf{K}_{\mathbf{x}_k\mathcal{D}}}{\partial [\mathbf{x}_k]_i}\right)\mathbf{S}^{-1}\mathbf{K}_{\mathcal{D}\mathbf{x}_k}\ . \tag{B.10}$$

B.3 BCRLB for the Train Motion Model with Deterministic Inputs

In this section the matrices (6.26) for the motion model given in (6.36) are derived. To keep the proofs short, the section starts with the introduction of some useful derivatives.

B.3.1 Useful Derivatives

In the following, we will calculate the gradients of log-likelihoods. When the likelihood is a multivariate Gaussian this will involve the gradient of quadratic forms. Therefore, it is useful to have an expression for gradients of the form $\nabla_{\mathbf{u}}^{\top}\mathbf{u}^{\top}\mathbf{A}\mathbf{u}$, where $\mathbf{u} \in \mathbb{R}^{N_{\mathbf{u}}}$ and $\mathbf{A} \in \mathbb{R}^{N_{\mathbf{u}}\times N_{\mathbf{u}}}$

$$\begin{aligned}
\nabla_{\mathbf{u}}^{\top}\mathbf{u}^{\top}\mathbf{A}\mathbf{u} &= \begin{bmatrix}\frac{\partial}{\partial[\mathbf{u}]_1} & \cdots & \frac{\partial}{\partial[\mathbf{u}]_{N_{\mathbf{u}}}}\end{bmatrix}\mathbf{u}^{\top}\mathbf{A}\mathbf{u} \\
&= \begin{bmatrix}\frac{\partial}{\partial[\mathbf{u}]_1} & \cdots & \frac{\partial}{\partial[\mathbf{u}]_{N_{\mathbf{u}}}}\end{bmatrix}\sum_{j=1}^{N_{\mathbf{u}}}\sum_{i=1}^{N_{\mathbf{u}}}[\mathbf{u}]_j[\mathbf{u}]_i[A]_{ij} \\
&= \begin{bmatrix}\sum_{j=1}^{N_{\mathbf{u}}}[\mathbf{u}]_j[A]_{1j} + \sum_{i=1}^{N_{\mathbf{u}}}[\mathbf{u}]_i[A]_{i1} & \cdots & \sum_{j=1}^{N_{\mathbf{u}}}[\mathbf{u}]_j[A]_{N_{\mathbf{u}}j} + \sum_{i=1}^{N_{\mathbf{u}}}[\mathbf{u}]_i[A]_{iN_{\mathbf{u}}}\end{bmatrix} \\
&= \begin{bmatrix}\sum_{j=1}^{N_{\mathbf{u}}}[\mathbf{u}]_j[A^{\top}]_{j1} + \sum_{i=1}^{N_{\mathbf{u}}}[\mathbf{u}]_i[A]_{i1} & \cdots & \sum_{j=1}^{N_{\mathbf{u}}}[\mathbf{u}]_j[A^{\top}]_{jN_{\mathbf{u}}} + \sum_{i=1}^{N_{\mathbf{u}}}[\mathbf{u}]_i[A]_{iN_{\mathbf{u}}}\end{bmatrix} \\
&= \mathbf{u}^{\top}(\mathbf{A}^{\top} + \mathbf{A})\ .
\end{aligned} \tag{B.11}$$

We will also often encounter derivatives of the form $\nabla_{\mathbf{u}}^{\top}\mathbf{u}^{\top}\mathbf{v}$ and $\nabla_{\mathbf{u}}^{\top}\mathbf{v}^{\top}\mathbf{u}$, which are given by

$$\nabla_{\mathbf{u}}^{\top}\mathbf{u}^{\top}\mathbf{v} = \nabla_{\mathbf{u}}^{\top}\mathbf{v}^{\top}\mathbf{u} = \begin{bmatrix}\frac{\partial}{\partial[\mathbf{u}]_1} & \cdots & \frac{\partial}{\partial[\mathbf{u}]_{N_{\mathbf{u}}}}\end{bmatrix}\sum_{i=1}^{N_{\mathbf{u}}}[\mathbf{u}]_i[\mathbf{v}]_i = \mathbf{v}^{\top}\ . \tag{B.12}$$

Finally, the derivative of the linear map $\mathbf{u}^{\top}\mathbf{A}$ with $\mathbf{A} \in \mathbb{R}^{N_{\mathbf{u}}\times N_{\mathbf{A}}}$ w.r.t. $\mathbf{u}$ is required for the next section

$$\begin{aligned}
\nabla_{\mathbf{u}}\mathbf{u}^{\top}\mathbf{A} &= \begin{bmatrix}\frac{\partial}{\partial[\mathbf{u}]_1} & \cdots & \frac{\partial}{\partial[\mathbf{u}]_{N_{\mathbf{u}}}}\end{bmatrix}^{\top}\mathbf{u}^{\top}\mathbf{A} \\
&= \begin{bmatrix}\frac{\partial}{\partial[\mathbf{u}]_1} & \cdots & \frac{\partial}{\partial[\mathbf{u}]_{N_{\mathbf{u}}}}\end{bmatrix}^{\top}\begin{bmatrix}\sum_{i=1}^{N_{\mathbf{u}}}[\mathbf{u}]_i[A]_{i1} & \cdots & \sum_{i=1}^{N_{\mathbf{u}}}[\mathbf{u}]_i[A]_{iN_{\mathbf{A}}}\end{bmatrix} \\
&= \mathbf{A}\ .
\end{aligned} \tag{B.13}$$

B.3.2 Proof of the BCRLB for Linear Motion Models with AWGN

For the matrices $\mathbf{D}_k^{(11)}$, $\mathbf{D}_k^{(12)}$, and $\mathbf{D}_k^{(22)}$ in (6.26) derivatives of $\ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u)$ are required. The density in the logarithm is found directly from the motion model (6.36) and the process noise pdf

$$\mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u) = \mathcal{N}(\mathbf{d}_{k+1}^{\mathrm{B}}; \mathbf{F}^{\mathrm{B}}\mathbf{d}_k^{\mathrm{B}} + \mathbf{B}^{\mathrm{B}}a_k, \mathbf{Q}^{\mathrm{B}}) \,. \tag{B.14}$$

The logarithm of (B.14) is the sum of a constant c and a quadratic form

$$\ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u) = c - \frac{1}{2}\left[(\mathbf{d}_{k+1}^{\mathrm{B}} - \mathbf{F}^{\mathrm{B}}\mathbf{d}_k^{\mathrm{B}} - \mathbf{B}^{\mathrm{B}}a_k)^\top(\mathbf{Q}^{\mathrm{B}})^{-1}(\mathbf{d}_{k+1}^{\mathrm{B}} - \mathbf{F}^{\mathrm{B}}\mathbf{d}_k^{\mathrm{B}} - \mathbf{B}^{\mathrm{B}}a_k)\right] \,. \tag{B.15}$$

Depending on the matrix from (6.26), we first have to differentiate the quadratic form w.r.t. $\mathbf{d}_k^{\mathrm{B}}$ or $\mathbf{d}_{k+1}^{\mathrm{B}}$. These derivatives are obtained by applying the identities (B.11) and (B.12) to the expanded quadratic form in (B.15) and using the symmetry of the noise covariance matrix $\mathbf{Q}^{\mathrm{B}}$

$$\begin{aligned}
\nabla_{\mathbf{d}_k^{\mathrm{B}}}^\top \ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u) &= -(\mathbf{d}_k^{\mathrm{B}})^\top(\mathbf{F}^{\mathrm{B}})^\top(\mathbf{Q}^{\mathrm{B}})^{-1}\mathbf{F}^{\mathrm{B}} - (\mathbf{B}^{\mathrm{B}}a_k - \mathbf{d}_{k+1}^{\mathrm{B}})^\top(\mathbf{Q}^{\mathrm{B}})^{-1}\mathbf{F}^{\mathrm{B}} \\
\nabla_{\mathbf{d}_{k+1}^{\mathrm{B}}}^\top \ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u) &= -(\mathbf{d}_{k+1}^{\mathrm{B}})^\top(\mathbf{Q}^{\mathrm{B}})^{-1} + (\mathbf{F}^{\mathrm{B}}\mathbf{d}_k^{\mathrm{B}} + \mathbf{B}^{\mathrm{B}}a_k)^\top(\mathbf{Q}^{\mathrm{B}})^{-1} \,.
\end{aligned} \tag{B.16}$$

These derivatives now have to be differentiated a second time w.r.t. the state vector at time step k or $k+1$

$$\begin{aligned}
\nabla_{\mathbf{d}_k^{\mathrm{B}}}\nabla_{\mathbf{d}_k^{\mathrm{B}}}^\top \ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u) &= -(\mathbf{F}^{\mathrm{B}})^\top(\mathbf{Q}^{\mathrm{B}})^{-1}\mathbf{F}^{\mathrm{B}} \\
\nabla_{\mathbf{d}_k^{\mathrm{B}}}\nabla_{\mathbf{d}_{k+1}^{\mathrm{B}}}^\top \ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u) &= (\mathbf{F}^{\mathrm{B}})^\top(\mathbf{Q}^{\mathrm{B}})^{-1} \\
\nabla_{\mathbf{d}_{k+1}^{\mathrm{B}}}\nabla_{\mathbf{d}_{k+1}^{\mathrm{B}}}^\top \ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u) &= -(\mathbf{Q}^{\mathrm{B}})^{-1} \,,
\end{aligned} \tag{B.17}$$

where we used (B.13) and (B.16). The matrices (B.17) are all deterministic quantities and their expected values are therefore simply the matrices themselves. Thus, the matrices $\mathbf{D}_k^{(11)}$, $\mathbf{D}_k^{(12)}$, and $\mathbf{D}_k^{(22)}$ are found by comparing their definition in (6.26) with (B.17)

$$\begin{aligned}
\mathbf{D}_k^{(11)} &= -\mathbf{E}_{\mathbf{d}_{k:k+1}^{\mathrm{B}}|\mathcal{D}_u}\left[\Delta_{\mathbf{d}_k^{\mathrm{B}}}^{\mathbf{d}_k^{\mathrm{B}}} \ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u)\right] = (\mathbf{F}^{\mathrm{B}})^\top(\mathbf{Q}^{\mathrm{B}})^{-1}\mathbf{F}^{\mathrm{B}} \\
\mathbf{D}_k^{(12)} &= -\mathbf{E}_{\mathbf{d}_{k:k+1}^{\mathrm{B}}|\mathcal{D}_u}\left[\Delta_{\mathbf{d}_k^{\mathrm{B}}}^{\mathbf{d}_{k+1}^{\mathrm{B}}} \ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u)\right] = -(\mathbf{F}^{\mathrm{B}})^\top(\mathbf{Q}^{\mathrm{B}})^{-1} \\
\mathbf{D}_k^{(22)} &= -\mathbf{E}_{\mathbf{d}_{k+1}^{\mathrm{B}},\mathbf{z}_{k+1}|\mathcal{D}_u}\left[\Delta_{\mathbf{d}_{k+1}^{\mathrm{B}}}^{\mathbf{d}_{k+1}^{\mathrm{B}}} \ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{d}_{k+1}^{\mathrm{B}}, \mathcal{D}_u)\right] \\
&\quad - \mathbf{E}_{\mathbf{d}_{k:k+1}^{\mathrm{B}}|\mathcal{D}_u}\left[\Delta_{\mathbf{d}_{k+1}^{\mathrm{B}}}^{\mathbf{d}_{k+1}^{\mathrm{B}}} \ln \mathrm{p}(\mathbf{d}_{k+1}^{\mathrm{B}}|\mathbf{d}_k^{\mathrm{B}}, \mathcal{D}_u)\right] \\
&= -\mathbf{E}_{\mathbf{d}_{k+1}^{\mathrm{B}},\mathbf{z}_{k+1}|\mathcal{D}_u}\left[\Delta_{\mathbf{d}_{k+1}^{\mathrm{B}}}^{\mathbf{d}_{k+1}^{\mathrm{B}}} \ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{d}_{k+1}^{\mathrm{B}}, \mathcal{D}_u)\right] + (\mathbf{Q}^{\mathrm{B}})^{-1} \\
&= \mathbf{E}_{\mathbf{d}_{k+1}^{\mathrm{B}}|\mathcal{D}_u}\left[\mathbf{E}_{\mathbf{z}_{k+1}|\mathbf{d}_{k+1}^{\mathrm{B}},\mathcal{D}_u}\left[-\Delta_{\mathbf{d}_{k+1}^{\mathrm{B}}}^{\mathbf{d}_{k+1}^{\mathrm{B}}} \ln \mathrm{p}(\mathbf{z}_{k+1}|\mathbf{d}_{k+1}^{\mathrm{B}}, \mathcal{D}_u)\right]\right] + (\mathbf{Q}^{\mathrm{B}})^{-1} \,,
\end{aligned} \tag{B.18}$$

where the first term in the last line is the expected value of the FIM as shown in (6.27). The matrices in (B.18) are in correspondence with the equations shown in [94, 100].

APPENDIX C

Calculation of Route Probabilities

This section describes how the route probabilities are calculated for the FDE and the NMA filter. The route probabilities are updated based on (4.50). The update requires the evaluation of the likelihood given by (4.49). For the non-robust along-track localization filter for known routes from Algorithm 4.1 this does not impose any problem since we only have one, supposedly correct, likelihood for the magnetometer. This is not the case for the FDE and the NMA filter, where we have to cope with different measurement models for the magnetometer and therefore different likelihoods.

C.1 FDE Particle Filter

The FDE particle filter updates the particle weights always based on the most likely model. When the most likely model corresponds to the case where all three magnetometer axes are disturbed, the weights are not updated with any magnetic information. Unfortunately, the case where the complete magnetic measurement vector is discarded does not integrate very well into the calculation of the route probabilities in (4.50). The issue that emerges here is that on a switch one of the FDE filters is always on the wrong track and it is very likely that this filter discards all magnetometer measurements and as a result its likelihood (4.49) only considers the accelerometer, which does not provide any information about the route. This is not a problem if we run a single filter on a known route, but for multiple filters on different routes we have to update the route probability based on (4.50), which requires that all likelihoods are well defined and can be weighted relative to each other in a meaningful way. To address this issue, we decided for the simplest approach that worked well in the evaluation. As for a single filter, the weights of the individual filters are updated based on the most likely model and if the complete magnetometer measurement vector is discarded only the acceleration is used. In contrast to the likelihood in the weight update, the likelihood (4.49) for the route probability update is constantly set to the likelihood of the first model in Table 4.1. Therefore, the route probability always accounts for all magnetometer measurements and the likelihood (4.49) becomes

$$\begin{aligned}
\mathrm{p}(\mathcal{Z}_k|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) &= \int \mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do}}, \mathcal{R}^{(i)})\, \mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) d\mathcal{X}_k^{\mathrm{do}} \\
&\approx \sum_{j=1}^{N_\mathrm{p}} w_{k-1}^{(j,i)}\, \mathrm{p}(\mathcal{Z}_k|\mathcal{X}_k^{\mathrm{do},(j,i)}, \mathcal{R}^{(i)}) \\
&= \sum_{j=1}^{N_\mathrm{p}} w_{k-1}^{(j,i)}\, \mathrm{p}(\mathbf{z}_k^{\mathrm{mag}}|\mathcal{X}_k^{\mathrm{do},(j,i)}, \mathcal{R}^{(i)})\, \mathrm{p}(z_k^{\mathrm{acc}}|\mathcal{X}_k^{\mathrm{do},(j,i)}) \,, \qquad \text{(C.1)}
\end{aligned}$$

where the predicted posterior was replaced by its Dirac mixture approximation and the superscript (j, i) stands for the j-th particle in the particle filter for route $\mathcal{R}^{(i)}$. In terms of route identification, the FDE filter and the non-robust filter use the same marginal likelihood and thus have here the same disadvantage that leaves them vulnerable to disturbances during route identification, which is not ideal. For the FDE filter thus better approaches have to be investigated in future work.

C.2 NMA Particle Filter

In the noise model averaging filter, the likelihood for the route probability is always well defined because measurements are never discarded in the weight update but only downweighted depending on the current model probabilities. The likelihood required for the route probability calculation can be obtained once more by marginalization

$$\begin{aligned}
\mathrm{p}(\mathcal{Z}_k|\mathcal{R}^{(i)},\mathcal{Z}_{1:k-1}) &= \sum_{l=1}^{N_\mathcal{M}} \int \mathrm{p}(\mathcal{Z}_k, \mathcal{M}_k^{(l)}, \mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) d\mathcal{X}_k^{\mathrm{do}} \\
&= \sum_{l=1}^{N_\mathcal{M}} \int \mathrm{p}(\mathcal{Z}_k|\mathcal{M}_k^{(l)}, \mathcal{X}_k^{\mathrm{do}}, \mathcal{R}^{(i)})\, \mathrm{p}(\mathcal{M}_k^{(l)}, \mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) d\mathcal{X}_k^{\mathrm{do}} \,, \qquad \text{(C.2)}
\end{aligned}$$

where $\mathrm{p}(\mathcal{M}_k^{(l)}, \mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})$ can be written as the product of the predicted posterior and the predicted model probability

$$\begin{aligned}
\mathrm{p}(\mathcal{M}_k^{(l)}, \mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) &= \mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{M}_k^{(l)}, \mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})\, \mathrm{pr}(\mathcal{M}_k^{(l)}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) \\
&= \mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})\, \mathrm{pr}(\mathcal{M}_k^{(l)}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) \,. \qquad \text{(C.3)}
\end{aligned}$$

In the last line of (C.3) we replaced the model specific predicted posterior again with a model independent one, as it is done in the NMA filter. Otherwise, a growing number of filters has to be run to obtain the required posteriors. If we insert (C.3) into (C.2) and replace $\mathrm{p}(\mathcal{X}_k^{\mathrm{do}}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1})$ with its Dirac mixture, we obtain an approximation of the desired likelihood

$$\mathrm{p}(\mathcal{Z}_k|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) = \sum_{l=1}^{N_\mathcal{M}} \mathrm{pr}(\mathcal{M}_k^{(l)}|\mathcal{R}^{(i)}, \mathcal{Z}_{1:k-1}) \sum_{j=1}^{N_\mathrm{p}} \mathrm{p}(\mathcal{Z}_k|\mathcal{M}_k^{(l)}, \mathcal{X}_k^{\mathrm{do},(j,i)}, \mathcal{R}^{(i)}) w_{k-1}^{(j,i)} \,, \qquad \text{(C.4)}$$

where the superscript (j, i) stands again for the j-th particle in the particle filter associated to route $\mathcal{R}^{(i)}$. The inner sum of (C.4) can be seen as expected likelihood of model l w.r.t. the predicted posterior density. The outer sum then calculates a weighted mean of the expected likelihoods, where the weights are given by the current model probabilities.

List of Figures

List of Tables

List of Algorithms

Own Publications

Journal Papers

[J1] B. Siebler, S. Sand, and U. D. Hanebeck, “Localization With Magnetic Field Distortions and Simultaneous Magnetometer Calibration,” *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3388–3397, Feb. 2021.

[J2] B. Siebler, A. Lehner, S. Sand, and U. D. Hanebeck, “Simultaneous Localization and Calibration (SLAC) Methods for a Train-Mounted Magnetometer,” *NAVIGATION: Journal of the Institute of Navigation*, vol. 70, no. 1, Mar. 2023.

[J3] B. Siebler, S. Sand, and U. D. Hanebeck, “Bayesian Cramér-Rao Lower Bound for Magnetic Field-Based Localization,” *IEEE Access*, vol. 10, pp. 123 080–123 093, 2022.

Conference Papers

[C1] B. Siebler, F. de Ponte Müller, O. Heirich, and S. Sand, “Algorithms for Relative Train Localization with GNSS and Track Map: Evaluation and Comparison,” in *2017 International Conference on Localization and GNSS (ICL-GNSS)*, Jun. 2017, pp. 1–7.

[C2] B. Siebler, O. Heirich, and S. Sand, “Bounding INS Positioning Errors with Magnetic-Field-Signatures in Railway Environments,” in *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, Sep. 2017, pp. 3224–3230.

[C3] ——, “Train Localization with Particle Filter and Magnetic Field Measurements,” in *2018 21st International Conference on Information Fusion (FUSION)*, Jul. 2018, pp. 1–5.

[C4] B. Siebler, O. Heirich, A. Lehner, S. Sand, and U. D. Hanebeck, “Robust Particle Filter for Magnetic field-based Train Localization,” in *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, Sep. 2022, pp. 1849–1858.

[C5] B. Siebler, O. Heirich, S. Sand, and U. D. Hanebeck, “Joint Train Localization and Track Identification based on Earth Magnetic Field Distortions,” in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Apr. 2020, pp. 941–948.

[C6] B. Siebler, A. Lehner, S. Sand, and U. D. Hanebeck, “Evaluation of Simultaneous Localization and Calibration of a Train Mounted Magnetometer,” in *Proceedings of the 34th International*

Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), Sep. 2021, pp. 2285–2293.

[C7] B. Siebler, T. Gerstewitz, S. Sand, and U. D. Hanebeck, "Magnetic Field-based Indoor Localization of a Tracked Robot with Simultaneous Calibration," in *2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2023, pp. 1–6.

[C8] B. Siebler, S. Sand, and U. D. Hanebeck, "Bayesian Cramér-Rao Lower Bounds for Magnetic Field-based Train Localization," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Apr. 2023, pp. 814–820.

[C9] B. Siebler, A. Lehner, S. Sand, and U. D. Hanebeck, "Magnetic Field Mapping of Railway Lines with Graph SLAM," in *2024 27th International Conference on Information Fusion (FUSION)*, Jul. 2024, pp. 1–8.

Co-Authored Papers

[CA1] O. Heirich and B. Siebler, "Train-side Passive Magnetic Measurements," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, May 2015, pp. 687–692.

[CA2] O. Heirich, B. Siebler, and E. Hedberg, "Study of Train-Side Passive Magnetic Measurements with Applications to Train Localization," *Hindawi Journal of Sensors*, vol. 2017, 2017.

[CA3] A. Lehner, T. Strang, O. Heirich, B. Siebler, S. Sand, P. Unterhuber, D. Bousdar Ahmed, C. Gentner, R. Karasek, and S. Kaiser, "Impact of Track Brakes on Magnetic Signatures for Localization of Trains," in *Fifth International Conference on Railway Technology: Research, Development and Maintenance*, 2022.

Supervised Student Theses

[ST1] E. Hedberg and M. Hammar, "Train localization and speed estimation using on-board inertial and magnetic sensors," Master's thesis, Linköping University, 2015.

List of References

[1] European Commission, Directorate-General for Mobility and Transport , *EU transport in figures: Statistical pocketbook 2022*. Publications Office of the European Union, 2022. [Online]. Available: https://data.europa.eu/doi/10.2832/216553

[2] European Environment Agency, *Rail and waterborne - Best for low-carbon motorised transport*. Publications Office of the European Union, 2021. [Online]. Available: https://data.europa.eu/doi/10.2800/85117

[3] Shift2Rail, "Shift2Rail Multi-Annual Action Plan (Amended version finally adopted on 14 November 2019)." [Online]. Available: https://rail-research.europa.eu/wp-content/uploads/2020/09/MAAP-Part-A-and-B.pdf (Accessed: 7. Jul. 2025).

[4] "R2DATO Project Objectives." [Online]. Available: https://projects.rail-research.europa.eu/eurail-fp2/objectives/ (Accessed: 7. Jul. 2025).

[5] J. Felez and M. A. Vaquero-Serrano, "Virtual Coupling in Railways: A Comprehensive Review," *Machines*, vol. 11, May 2023.

[6] "R2DATO Technical Enablers." [Online]. Available: https://projects.rail-research.europa.eu/eurail-fp2/technical-enablers/ (Accessed: 7. Jul. 2025).

[7] "CLUG High-Level Mission Requirements Definition." [Online]. Available: https://clugproject.eu/sites/default/files/2021-02/d2.1_hl_mission_requirements_v2.14_pu_final_0.pdf (Accessed: 7. Jul. 2025).

[8] J. Marais, J. Beugin, and M. Berbineau, "A Survey of GNSS-Based Research and Developments for the European Railway Signaling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2602–2618, Oct. 2017.

[9] J. Otegui, A. Bahillo, I. Lopetegi, and L. E. Díez, "A Survey of Train Positioning Solutions," *IEEE Sensors Journal*, vol. 17, no. 20, pp. 6788–6797, Oct. 2017.

[10] P. Winter, *Compendium on ERTMS*. Hamburg: Eurailpress, 2009.

[11] P. Stanley, *ETCS for Engineers*. Hamburg: Eurailpress, 2011.

[12] European Union Agency for Railways, "CCS TSI Appendix A – Mandatory specifications: SUBSET-041." [Online]. Available: https://www.era.europa.eu/era-folder/1-ccs-tsi-appendix-mandatory-specifications-etcs-b4-r1-rmr-gsm-r-b1-mr1-frmcs-b0-ato-b1 (Accessed: 7. Jul. 2025).

[13] O. G. Crespillo, O. Heirich, and A. Lehner, “Bayesian GNSS/IMU Tight Integration for Precise Railway Navigation on Track Map,” in *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, May 2014, pp. 999–1007.

[14] K. Lüddecke and C. Rahmig, “Evaluating Multiple GNSS Data in a Multi-Hypothesis Based Map-Matching Algorithm for Train Positioning,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2011, pp. 1037–1042.

[15] F. Böhringer, “Gleisselektive Ortung von Schienenfahrzeugen mit bordautonomer Sensorik,” Ph.D. dissertation, Universität Karlsruhe (TH), 2008.

[16] M. Lauer and D. Stein, “A Train Localization Algorithm for Train Protection Systems of the Future,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 970–979, 2015.

[17] O. Heirich, “Bayesian Train Localization with Particle Filter, Loosely Coupled GNSS, IMU, and a Track Map,” *Hindawi Journal of Sensors*, vol. 2016, May 2016.

[18] K. Kim, S. Seol, and S.-H. Kong, “High-speed Train Navigation System based on Multi-sensor Data Fusion and Map Matching Algorithm,” *International Journal of Control, Automation and Systems*, vol. 13, no. 3, pp. 503–512, Jun. 2015.

[19] T. Albrecht, K. Lüddecke, and J. Zimmermann, “A Precise and Reliable Train Positioning System and its Use for Automation of Train Operation,” in *2013 IEEE International Conference on Intelligent Rail Transportation Proceedings*, Aug. 2013.

[20] W. Jiang, S. Chen, B. Cai, J. Wang, W. ShangGuan, and C. Rizos, “A Multi-Sensor Positioning Method-Based Train Localization System for Low Density Line,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 425–10 437, Nov. 2018.

[21] H. Jochim and F. Lademann, *Planung von Bahnanlagen*. München: Carl Hanser Verlag, 2009.

[22] O. Heirich, P. Robertson, A. C. García, and T. Strang, “Bayesian Train Localization Method Extended by 3D Geometric Railway Track Observations From Inertial Sensors,” in *2012 15th International Conference on Information Fusion*, Jul. 2012, pp. 416–423.

[23] A. Broquetas, A. Comerón, A. Gelonch, J. M. Fuertes, J. A. Castro, D. Felip, M. A. López, and J. A. Pulido, “Track Detection in Railway Sidings Based on MEMS Gyroscope Sensors,” *Sensors*, vol. 12, no. 12, pp. 16 228–16 249, Nov. 2012.

[24] W. Löffler and M. Bengtsson, “Train Localization During GNSS Outages: A Minimalist Approach Using Track Geometry And IMU Sensor Data,” in *2024 27th International Conference on Information Fusion (FUSION)*, 2024, pp. 1–8.

[25] S. S. Saab, “A Map Matching Approach for Train Positioning Part I: Development and Analysis,” *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 467–475, Mar. 2000.

[26] ——, “A Map Matching Approach for Train Positioning Part II: Application and Experimentation,” *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 476–484, Mar. 2000.

[27] Y. Zhou, Q. Chen, R. Wang, G. Jia, and X. Niu, "Onboard Train Localization Based on Railway Track Irregularity Matching," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[28] S. Hensel, C. Hasberg, and C. Stiller, "Probabilistic Rail Vehicle Localization With Eddy Current Sensors in Topological Maps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1525–1536, 2011.

[29] S. Hensel and C. Hasberg, "Bayesian techniques for onboard train localization," in *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, Aug. 2009, pp. 361–364.

[30] T. Engelberg and F. Mesch, "Eddy current sensor system for non-contact speed and distance measurement of rail vehicles," *Computers in Railways VII*, pp. 1261–1270, 2000.

[31] 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 *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 913–920.

[32] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1028–1035, Oct. 2009.

[33] N. Akai and K. Ozaki, "Gaussian Processes for Magnetic Map-Based Localization in Large-Scale Indoor Environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 4459–4464.

[34] E. Le Grand and S. Thrun, "3-Axis Magnetic Field Mapping and Fusion for Indoor Localization," in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2012, pp. 358–364.

[35] A. Solin, S. Särkkä, J. Kannala, and E. Rahtu, "Terrain Navigation in the Magnetic Landscape: Particle Filtering for Indoor Positioning," in *2016 European Navigation Conference (ENC)*, May 2016, pp. 1–9.

[36] I. Ashraf, M. Kang, S. Hur, and Y. Park, "MINLOC:Magnetic Field Patterns-Based Indoor Localization Using Convolutional Neural Networks," *IEEE Access*, vol. 8, pp. 66 213–66 227, 2020.

[37] K. P. Subbu, B. Gozick, and R. Dantu, "LocateMe: Magnetic-Fields-Based Indoor Localization Using Smartphones," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, pp. 1–27, Oct. 2013.

[38] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[39] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, "Simultaneous Localization and Mapping Using Ambient Magnetic Field," in *2010 IEEE Conference on Multisensor Fusion and Integration*, Sep. 2010, pp. 14–19.

[40] M. Kok and A. Solin, "Scalable Magnetic Field SLAM in 3D Using Gaussian Process Maps," in *2018 21st International Conference on Information Fusion (FUSION)*, Jul. 2018, pp. 1353–1360.

[41] ——, "Online One-Dimensional Magnetic Field SLAM with Loop-Closure Detection," in *2024 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2024, pp. 1–7.

[42] P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. J. Julian, M. Garcia 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 *International Conference on Indoor Positioning and Indoor Navigation*, Oct. 2013, pp. 1–10.

[43] F. Viset, J. T. Gravdahl, and M. Kok, "Magnetic field norm SLAM using Gaussian process regression in foot-mounted sensors," in *2021 European Control Conference (ECC)*, Jun. 2021, pp. 392–398.

[44] F. Viset, R. Helmons, and M. Kok, "An Extended Kalman Filter for Magnetic Field SLAM Using Gaussian Process Regression," *Sensors*, vol. 22, no. 8, Apr. 2022.

[45] J. Jung, J. Choi, T. Oh, and H. Myung, "Indoor Magnetic Pose Graph SLAM with Robust Back-End," in *Robot Intelligence Technology and Applications 5*, 2018, pp. 153–163.

[46] J. A. Shockley and J. F. Raquet, "Navigation of Ground Vehicles Using Magnetic Field Variations," *NAVIGATION*, vol. 61, no. 4, pp. 237–252, 2014.

[47] A. Canciani and J. Raquet, "Airborne Magnetic Anomaly Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 67–80, Feb. 2017.

[48] ——, "Absolute Positioning Using the Earth's Magnetic Anomaly Field," *NAVIGATION*, vol. 63, no. 2, pp. 111–126, 2016.

[49] T. N. Lee and A. J. Canciani, "MagSLAM: Aerial simultaneous localization and mapping using Earth's magnetic anomaly field," *NAVIGATION*, vol. 67, pp. 95–107, 2020.

[50] B. Liu, "Robust particle filter by dynamic averaging of multiple noise models," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 4034–4038.

[51] Y. Zhao, C. Fritsche, G. Hendeby, F. Yin, T. Chen, and F. Gunnarsson, "Cramér–Rao Bounds for Filtering Based on Gaussian Process State-Space Models," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 5936–5951, Dec. 2019.

[52] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge: Cambridge University Press, 2014.

[53] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Mineola: Dover Publications, 2007.

[54] F. Gustafsson, *Statistical Sensor Fusion*. Lund: Studentlitteratur, 2012.

[55] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.

[56] 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, Feb. 2002.

[57] S. C. Surace, A. Kutschireiter, and J.-P. Pfister, “How to Avoid the Curse of Dimensionality: Scalability of Particle Filters with and without Importance Weights,” *SIAM Review*, vol. 61, no. 1, pp. 79–91, 2019.

[58] B. Sileshi, C. Ferrer, and J. Oliver, “Particle Filters and Resampling Techniques: Importance in Computational Complexity Analysis,” in *2013 Conference on Design and Architectures for Signal and Image Processing*, Oct. 2013, pp. 319–325.

[59] J. D. Hol, T. B. Schön, and F. Gustafsson, “On Resampling Algorithms for Particle Filters,” in *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, Sep. 2006, pp. 79–82.

[60] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings-F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[61] F. Gustafsson, “Particle Filter Theory and Practice with Positioning Applications,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, Jul. 2010.

[62] S. Godsill, A. Doucet, and M. West, “Maximum a Posteriori Sequence Estimation Using Monte Carlo Particle Filters,” *Annals of the Institute of Statistical Mathematics*, vol. 53, no. 1, pp. 82–96, Mar. 2001.

[63] M. Klaas, N. de Freitas, and A. Doucet, “Toward Practical N2 Monte Carlo: The Marginal Particle Filter,” in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, 2005, pp. 308–315.

[64] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: The MIT Press, 2006.

[65] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2016.

[66] J. Wang, “An Intuitive Tutorial to Gaussian Process Regression,” *Computing in Science & Engineering*, vol. 25, no. 4, pp. 4–11, 2023.

[67] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River: Prentice Hall, 1993.

[68] H. L. Van Trees, K. L. Bell, and Z. Tian, *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Filtering Theory*. Hoboken: John Wiley & Sons, 2013.

[69] J. Wendel, *Integrierte Navigationssysteme*. München: Oldenbourg Wissenschaftsverlag, 2011.

[70] O. Heirich, “Localization of Trains and Mapping of Railway Tracks,” Ph.D. dissertation, Technische Universität München, 2020.

[71] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[72] J. F. G. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet, “Sequential Monte Carlo Methods to Train Neural Network Models,” *Neural Computation*, vol. 12, no. 4, pp. 955–993, Apr. 2000.

[73] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, “The Unscented Particle Filter,” in *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[74] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation: Theory, Algorithms and Software*. New York: John Wiley & Sons, 2001.

[75] L. E. Sucar, *Probabilistic Graphical Models: Principles and Applications*. Springer, 2021.

[76] M. Kok and T. B. Schön, “Magnetometer Calibration Using Inertial Sensors,” *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5679–5689, Jul. 2016.

[77] V. Renaudin, M. H. Afzal, and G. Lachapelle, “Complete Triaxis Magnetometer Calibration in the Magnetic Domain,” *Hindawi Journal of Sensors*, vol. 2010, 2010.

[78] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, “Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1293–1306, Apr. 2011.

[79] Y. Wu and W. Shi, “On Calibration of Three-Axis Magnetometer,” *IEEE Sensors Journal*, vol. 15, no. 11, pp. 6424–6431, Nov. 2015.

[80] J. Fang, H. Sun, J. Cao, X. Zhang, and Y. Tao, “A Novel Calibration Method of Magnetic Compass Based on Ellipsoid Fitting,” *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 6, pp. 2053–2061, Jun. 2011.

[81] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River: Prentice-Hall, 2002.

[82] C. Gentner, T. Jost, W. Wang, S. Zhang, A. Dammann, and U.-C. Fiebig, “Multipath Assisted Positioning with Simultaneous Localization and Mapping,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6104–6117, 2016.

[83] M. Ulmschneider, S. Zhang, C. Gentner, and A. Dammann, “Multipath Assisted Positioning With Transmitter Visibility Information,” *IEEE Access*, vol. 8, pp. 155 210–155 223, 2020.

[84] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, “An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, Oct. 2003, pp. 206–211.

[85] K. Li, M. Li, and U. D. Hanebeck, “Towards High-Performance Solid-State-LiDAR-Inertial Odometry and Mapping,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5167–5174, Jul. 2021.

[86] T. Taketomi, H. Uchiyama, and S. Ikeda, “Visual SLAM algorithms: a survey from 2010 to 2016,” *IPSJ Transactions on Computer Vision and Applications*, vol. 9, Jun. 2017.

[87] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio, “Landmark based Radar SLAM Using Graph Optimization,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 2559–2564.

[88] Z. Hong, Y. Petillot, and S. Wang, “RadarSLAM: Radar based Large-Scale SLAM in All Weathers,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 5164–5170.

[89] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2002.

[90] T. Schön, F. Gustafsson, and P.-J. Nordlund, “Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[91] F. Daum, “Nonlinear filters: beyond the Kalman filter,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, Aug. 2005.

[92] 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, no. 4, pp. 1112–1127, Aug. 2018.

[93] T. Edridge and M. Kok, “Mapping the magnetic field using a magnetometer array with noisy input Gaussian process regression,” in *2023 26th International Conference on Information Fusion (FUSION)*, Jun. 2023, pp. 1–7.

[94] P. Tichavsky, C. Muravchik, and A. Nehorai, “Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering,” *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, 1998.

[95] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig, *Taschenbuch der Mathematik*, 7th ed. Frankfurt am Main: Wissenschaftlicher Verlag Harri Deutsch, 2008.

[96] The MathWorks, “Zero-phase digital filtering - Documentation.” [Online]. Available: https://www.mathworks.com/help/releases/R2023b/signal/ref/filtfilt.html (Accessed: 11. Jul. 2025).

[97] J. Vermaak, A. Doucet, and P. Perez, “Maintaining Multi-Modality through Mixture Tracking,” in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Oct. 2003, pp. 1110–1116.

[98] S. Maskell and S. Julier, “Optimised Proposals for Improved Propagation of Multi-Modal Distributions in Particle Filters,” in *Proceedings of the 16th International Conference on Information Fusion*, Jul. 2013, pp. 296–303.

[99] C. E. Rasmussen and H. Nickisch, “Gaussian Processes for Machine Learning (GPML) Toolbox,” *Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, 2010.

[100] H. L. Van Trees and K. L. Bell, *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*. John Wiley & Sons, 2007.