Lehrstuhl für Kommunikation und Navigation
Technische Universität München

Localization of Trains and Mapping of Railway Tracks

Dipl.-Ing. Oliver Heirich

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzende:	Prof. Dr.-Ing. Sandra Hirche
Prüfer der Dissertation:	1. Prof. Dr. sc. nat. Christoph Günther
	2. Prof. Fredrik Gustafsson, Ph.D.

Die Dissertation wurde am 10.07.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 10.06.2020 angenommen.

Bild Umschlag: iStock.com/no_limit_pictures

Abstract

Future railway systems and applications address a railway traffic that is both safe and efficient. Prominent examples are: train-side collision avoidance systems, virtual coupling, and autonomous train driving. These systems and applications rely on a train localization that provides an exact location of the train within the track network. This train localization is safety-critical and therefore the approach requires a continuous availability and a track-selective accuracy. Current train localization is based on infrastructure with costly way-side components. This thesis focuses on the onboard train localization that uses exclusively sensors that are mounted on the train. Onboard train localization needs appropriate sensors in combination with a track map and a method that computes a track ID and a location on that track. This thesis presents research on train-mounted sensors, several localization methods and a mapping approach for the required track map. The localization and mapping methods are based on a Bayesian estimation framework and estimate the probabilistic posterior distribution.

The train mounted sensors GNSS, IMU, and magnetometer are analyzed as well as novel approaches of track feature measurements are presented with inertial measurements of train kinematics caused by the track geometry, the measurements of passive magnetic track features, and the measurements of vibratory track features.

The special track map may initially not be available, suffer from incompleteness, insufficient accuracy or outdated information. Therefore, a simultaneous localization and mapping approach, called RailSLAM, is presented. This probabilistic mapping method addresses the creation and maintenance of the feature rich track map.

Three train localization methods are presented and implemented: a simple map-match method which requires very accurate positioning, a probabilistic particle filter method and a probabilistic multiple hypothesis tracker method. The train localization methods are evaluated for track-selectivity and compared. The evaluation is based on recorded measurement data of a regional train at regular passenger service. The particle filter and the multiple hypothesis filters showed results with 100 % track-selectivity even with real measurements and a prior recorded RailSLAM track map over 230 km of train data. The RailSLAM track map is created from multiple train runs and comprises nine geometric values: a 3-D position, 3-D attitude and 3-D curvatures.

Contents

Chapter 1

Introduction

1.1 Motivation

Since the beginning of railway transportation in the 19'th century, it is a concern to prevent train accidents. Therefore, train control systems with signaling have been established for a safe railway traffic. These systems manage the exclusive access of trains to certain tracks and sections for a collision free operation. Over time, the technical systems became more and more elaborated, but severe train accidents still happened.

Currently, a global trend in railways is a modernization of the train control systems. This modernization comprises improved train control and automatic train protection (ATP) systems for higher safety and efficiency. In Europe, the European train control system (ETCS) is currently deployed in many countries [1]. In the United States, the positive train control (PTC) system is currently implemented in many trains [2, 3]. Beside the modernization of train control systems, new concepts for a safe and efficient railway traffic have been proposed and developed in recent years. The research of this thesis is motivated with the development of the railway collision avoidance system (RCAS) at the German Aerospace Center DLR (*Deutsches Zentrum für Luft- und Raumfahrt*). RCAS is an infrastructure-less and decentralized system, which is installed on trains [4]. Usually, railway collision avoidance is realized by signaling with ATP. RCAS is an independent system that is not connected to the signaling system. The main system components of RCAS are a train localization unit with onboard sensors, a radio communication system, and an interface for the train driver. Each train localizes itself within the railway track network and communicates its location and its current velocity to all other trains or track workers. Each train receives messages from trains in the vicinity and all train trajectories are checked for potential collision courses. Finally, an interface for the train driver visualizes the current traffic state and issues an alert if necessary. In addition to the collision avoidance functionally, RCAS can provide a complementary and independent speed monitoring to prevent derailments. As an independent system to any ATP, RCAS is suitable to increase safety as an overlay system, as a fallback system as well as a safety system on side-tracks where train protection systems are not installed.

The development of RCAS generated a need for a track-selective and continuous train localization based on measurements of onboard mounted sensors. This localization requires also a special map of the railway environment. In this thesis, a map-based train localization concept with onboard sensors is designed and analyzed. Furthermore an automated mapping concept, called RailSLAM, is developed. This thesis encompasses a theoretical assessment of the novel concepts, implementations as well as evaluations with measurements recorded on a regional train.

The safety performance report of the European Railway Agency [5] counts yearly 86 significant accidents by collisions and 94 derailments as average performance figures per year between 2010-2012 of the European Union. This report concludes, that accidents by train collision or derailment happens every second day on average in the EU. The figures on train collisions include train-to-object and train-to-train collisions. About 12 % of the collisions in 2012 were train-to-train collisions [5]. Reported indicators to dangerous incidents are called *precursors to accident* and the ratio of reported precursors to significant accidents is 12:1. There are 468 reported precursors to accident on *wrong-side signaling failures* and 2318 precursors to accident of *signals passed at danger* [5]. These figures relate to 7.6 dangerous incidents per day on average in the EU. A collision avoidance system, such as RCAS, can warn the train driver in advance and is likely to be effective at preventing these incidents.

Additionally to safety, the efficiency of railway traffic can be optimized in terms of capacity, which refers to an increased number of trains per hour and track. The moving block method considers an increased capacity with a reduced safety distance between trains down to the braking distance and is considered in ETCS level 3 [1]. In contrast, a state-of-the-art interlocking uses fixed blocks of a track section. The moving block is dynamically defined around the train at a train control computer with a safety distance from the rear of the train to the train front plus the braking distance [1]. This safety distance is smaller than the fixed block. The technical elements are a train localization unit, wireless communication links from each train to the central train control and a cab signaling. A further reduction of the safety distance below the braking distance is possible with a new application called virtual coupling [6, 7, 8]. In virtual coupling, a train follows another train at a very close distance of a few meters. For safe operations, the train driver is excluded from the distance control. Virtual coupling requires a direct, robust wireless communication link between the trains as well as a train localization unit in combination with a reliable distance estimation.

Finally, automatic train operation (ATO) systems can operate a train without a driver or attendant autonomously in the highest automation grade, called unattended train operation (UTO) [9]. There are more than 25 subway systems with UTO by 2011 [10]. These systems require special infrastructure at wayside and platforms for the replacement of the train driver. UTO systems are often in enclosed railway environments with no level-crossings and are monitored by a centralized train supervision [11]. Furthermore, the railway traffic of a subway is a homogenous traffic with similar travel speeds and usually one operator, the same type of trains and one schedule. In contrast to homogenous traffic, a mixed traffic comprises cargo and passenger trains on the same track network with widely varying travel speeds. Fully autonomous trains for mixed traffic with no special infrastructure is currently a challenge. Rüdiger Grube, the former director of Germany's largest railway company (Deutsche Bahn AG), has announced autonomous trains in mixed traffic until 2023 [12].

All mentioned railway applications collision avoidance, virtual coupling, and autonomous train driving require an accurate and reliable train localization as well as a spacial map of the railway tracks.

1.2 Problem Statement

Future safety critical railway applications and systems require a train localization unit with sensors mounted on the train. Train localization determines a location called topological location in the following. This topological location consists of a track ID and a one-dimensional location on that track. The topological location cannot be measured directly with onboard mounted sensors, because the track ID and the origin for the 1-D location are arbitrary definitions and not physical quantities. This circumstance is designated as the *train localization problem.* Furthermore, the major requirements for safety critical train localization are a continuous availability and a track-selective accuracy. The train localization challenge is to design a train localization unit, that is based on onboard sensors, with a continuous availability and a track-selective accuracy. Moreover, appropriate sensors must be identified for their suitability with respect to train localization. The solution must cope with hidden variables such as the topological location, imprecise measurements from multiple sensor sources, and measurement outages.

Onboard train localization does not rely on cost-intensive track-side equipment. Instead, a train localization based on onboard mounted sensors requires a special map with track characteristics. However, this special map may be initially not available, has insufficient accuracy, and may be incomplete or outdated. These map issues are referred to as *map problem*. The special track map has to be created or updated with a mapping approach. Therefore, sensor measurements of the track geometry need to be associated with topological locations. The challenge for the mapping approach is to design a concept for an automated map generation and map maintenance based on measurements of onboard sensors.

Finally, the problems of train localization and track mapping are closely linked with each other: On the one hand, the train localization with onboard sensors requires a map and the track-selective accuracy depends on the map quality. On the other hand, the mapping requires a location for the correct data association and a track-selective location accuracy is in turn vital for the map quality. A missing map for the localization and a required location for the mapping is known as the *chicken-and-egg problem* [13] about what comes first: the map or the location.

1.3 Approach and Contribution

In this thesis, the train localization problem is addressed with a probabilistic train localization estimation, a track map with geometric track features and the following onboard sensors: global navigation satellite system (GNSS) receivers, inertial measurement units (IMUs), and magnetic sensors. The focus of this thesis is on low-cost and commercial-off-the-shelf (COTS) sensors.

GNSS alone does not fulfill the requirements of track selectivity and continuity. The accuracy of GNSS is not sufficient in some scenarios in the railway environment and the continuity is not always given, especially in tunnels. Beside the well-known combination of IMU and GNSS in an inertial navigation system (INS), this thesis focuses also on a different and non-integrative method with the IMU measurements: A special property of railways is the strong physical constraint of train kinematics by the rails of the track. An IMU can measure the train kinematics caused by the track geometry. The train location can be estimated in combination with a special map that contains the track geometry linked to topological locations. This special map is called track map in the following. This thesis provides detailed analysis on inertial measurements and limits the scope of GNSS measurements to the level of the solved position solution. Additionally, novel approaches for train localization are presented using magnetic measurements or vibration measurements.

The map problem and the chicken-and-egg problem are solved with an algorithmic approach named simultaneous localization and mapping (SLAM). The proposed method in this thesis is called RailSLAM. This method is able to estimate a new track map without prior information and also to improve an existing track map. The train localization method is either a part of the SLAM method or used as a stand-alone method once the track map has been created. In this work, three train localization methods and one SLAM method are presented and evaluated. The methods are based on a Bayesian estimation framework and estimate the probabilistic posterior. The three localization approaches are:

- a simple and straightforward map-match method,
- a probabilistic particle filter method,
- a probabilistic multiple hypothesis tracking method.

The mapping process uses a probabilistic SLAM approach and is able to start with or without a prior known track map.

The evaluation and test uses real train data from a regional passenger train at regular service. The sensor data set was recorded on multiple train runs between Augsburg and Ingolstadt in Germany and the data set contains synchronized measurements of multiple IMUs with magnetometers, a GNSS receiver, and a camera. The recorded videos from the camera were used as reference for the proposed track-selective evaluation method. The three localization methods are evaluated for track selectivity with different track maps.

The contributions of this thesis in the field of onboard train localization and track mapping are summarized in the following:

- Analysis of onboard measurements with GNSS, IMU and magnetic for train localization.
- Identification, analysis and description of novel measurement approaches for train localization based on train kinematic measurements with inertial sensors, magnetic measurements, and vibration measurements.

- Definition of the estimation problem for train localization and SLAM for railways based on Bayesian theory.
- Design, description and implementation of three train localization methods with GNSS and IMU. Evaluation for track-selective accuracy and performance comparison with train measurements over 230 km.
- Design, description, implementation and evaluation of the RailSLAM method for an automated localization and mapping of railway tracks with GNSS and IMU. The accuracy of the track map is evaluated with a reference map.

Furthermore, the discussions in Chapter 8 contain suggestions for improvements and indications for future research.

1.4 State of the Art and Related Work

Modern Train Control Systems

In Europe, the ETCS is currently deployed in many countries [1]. One focus in ETCS is a harmonization of technical systems between countries for a simplified cross-border traffic with a common signaling system. The technical modernization encompasses a balise based signaling with inherent train positioning, speed monitoring, cab signaling, train integrity and wireless communications for train control between train and control center [14]. Cab signaling is an onboard interface for the train driver to visualize the current signaling state without the need for light signals or signs in the infrastructure [14]. Train integrity monitors the wagon set if the train is still complete or if some wagons got decoupled.

In the United States, the PTC system is currently implemented in many trains [2, 3]. PTC uses a global positioning system (GPS) based train localization with the main purpose to ensure collisions. At railway switches, the predefined switch way is verified by the traveled switch way, so the system can detect a wrong switch stand. Furthermore, PTC monitors unauthorized train access to tracks in work zones and detects over-speed, which is a cause for derailments [3]. The PTC system and ETCS are both digital, centralized, and communication based train control system including train positioning.

Train Localization with Track-side Sensors

Collision free railway operations is primarily achieved by train protection systems with interlocking. Interlocking assures, that trains can only access a track section or a block if it is recognized to be vacant. State-of-the-art train protection systems rely on track-side infrastructure with beacons and signaling devices. Examples for track-side infrastructure are magnetic coils, cable loops, contacts, track-circuits, axle counters, transponders and balises [1, 15]. For continuous location data, a train odometry measures the speed and computes the traveled distance from wheel turns, for instance.

The train location within a certain track section is of interest for the train control center. In contrast, a train driver is interested in the distance between the train and the next signaling unit. In the case that there is a stop signal, a train needs to brake and stop safely before the signal.

There are two main classes of ATP systems, the intermittent train control systems and the continuous train control systems for higher train speeds. Intermittent train control systems, such as ETCS Level 1 or the German PZB (*Punktförmige Zugbeeinflussung*), provide information about the upcoming railway signal only at the discrete locations of the beacons. The focus is on the distance between train and railway signal and not on an absolute location. Continuous train control systems, such as ECTS Level 2 or the German LZB (*Linienzugbeeinflussung*) uses Balises or cable loops to localize the train with a higher resolution of the train location. The track selectivity is achieved with designated balise IDs or with a near-field communication over cable loops that is locally limited and only available at the particular track and track section. A location along the track can be determined at the balise positions or at loop crossings, e.g. every 50 m for LZB. Between balises or loop crossings, the distance is measured via wheel turns for a continuous localization [1, 15].

Train Localization with Onboard Sensors

There are multiple options of train localization with onboard sensors and a map. The different approaches vary in sensor types or combinations, processing methods and evaluation scope:

For railway localization, inertial sensors are often used in combination with GNSS receivers in an integrated navigation system [16, 17]. The yaw turn rate measurements of an IMU can also be used for the switch-way identification [18, 19] and has been analyzed in [20]. Approaches with GNSS and IMU are found in [18, 17, 19] and extensions with eddy current sensor in [16, 21, 22]. Sensors such as cameras [23, 24] or LIDAR

[25, 26] can directly identify the different switch ways and contribute to the track-selective result. The eddy current sensor, a metal detector for characteristic railway features, can be used for a switch-way detection, or as speed and displacement sensor [16, 21, 22]. A study of a tightly coupled localization with GNSS satellite ranges, Doppler measurements and a track map is presented in [27]. Marais et al. [28] show a survey of GNSS based railway signaling and train localization approaches. The processing methods of each study are different and dependent on the sensors, the filtering method, and the algorithmic integration of the map. Saab proposed a train localization using a map matching technique with a correlation of the curvature signature [29]. There are two approaches for the track-selective localization estimation: the multiple hypothesis filter and the particle filter. A multiple hypothesis filter handles and maintains multiple estimates on several tracks in the vicinity and is commonly used [16, 22, 19, 26]. The hypotheses are the different tracks and a new hypotheses is generated and maintained after passing a switch. The particle filter instead handles the different track hypotheses with a large set of particles. Localization with a particle filter, onboard sensors and a map was proposed in [30]. Fouque et al. [27] defined a marginalized particle filter with raw GNSS data for the identification of the carriageway and the along location of a road vehicle. Hensel et al. [21] showed a particle filter approach for railways based on an eddy current sensor and a map.

Only a few approaches evaluate larger data sets with a track-selectivity statistics: Lauer and Stein [22] used GNSS and a velocity sensor and showed a gain in track-selective accuracy and confidence between a simple map-match and a proposed estimation algorithm. Böhringer [16] evaluated an integrated navigation system (GNSS, IMU) in combination with an eddy current sensor for switch-way identification. Even with a moderate switch detection rate of 70 % of the eddy current sensor, the combined results with GNSS and IMU received 99.78 % of track-selective accuracy. These results are based on real train runs of 120 km with 113 switches and considered in a comparison with results of this thesis on Chapter 8. Hensel et al. [21] showed no direct figures on the traveled switch-way estimation, but improved switch detection (98.23 %) and classification (99.64 %) with an eddy current sensor of 831 switches. This study focuses on switches as position input and the classification discriminates between merging and splitting switch runs.

The contributions of this thesis address the following research gaps: Analysis of train measurements with focus on low-cost and COTS sensors for train localization, especially inertial sensors and magnetometers; a

Bayesian motivation of the train localization problem; evaluation and comparison of a particle filter, multi-hypothesis filter and a simple map-match approach for train localization all compared with an identical measurement data set.

Railway Track Mapping

Railway tracks are planned by railway line design offices and railway infrastructure companies [15]. The railway track information comprise plans for constructions, railway line overview maps, but also speed limitation, altitude, curvature and superelevation profiles. For train localization, this information may be too detailed as in construction plans, or too coarse as in generalized overview maps. As an example for a map, the DBOpenData [31] route atlas contains position data of major railway lines but generalizes parallel tracks of the same route by just one line. The data from plans may not be existing in the required generalization or format (e.g. paper maps). Further issues are an outdated map, incomplete content or the availability of maps by rights of property.

Design plans may consider only a theoretical railway track. The map with track positions and profiles can be achieved by three state-of-the-art methods:

- A classical survey considers position measurements from total station theodolites. These total stations determine points relative to a base station, usually from an electronic theodolite (azimuth, elevation) and electronic distance measurements, such as laser ranging [32]. As a limitation, measurements and post-processing are carried out by a time consuming manual process with a low grade of automation.

- A train-borne survey with dynamic measurements by special trains called track geometry car. Track geometry cars are equipped with expensive, high-end survey instrumentation, real time kinematic (RTK) GNSS and used to check the track condition [33]. The intended purpose is a high precision measurement of the track mostly by a single run.

- A photogrammetry method with data recorded by planes, drones [34] or satellites. A further processing step extracts positions of railway tracks from remote sensing data, such as images, LIDAR or radar. All methods require an additional processing step to link the positions to the abstract track definitions.

Alternatively to the mentioned methods, the mapping can also be realized with standard trains in regular service. In contrast to the train survey by a track geometric car, a standard train will run several times over the same tracks and may be equipped with sensors of lower cost and quality. In robotics, the localization and mapping with the same onboard sensors and has been realized by the SLAM algorithm. SLAM estimates a pose (position and heading) and a map at the same time and has been described by [13, 35] and in many others. The SLAM algorithm considers a probabilistic representation of the map and the location and depends on a revisit of previous locations. This revisit, also denoted as loop closure, enables a correction and reduction of uncertainty of the map and the location. The EKF-SLAM [35] considers a Kalman filter with positions of the landmarks stored in the state vector and the covariance matrix. As a limitation, only a relative small number of features are possible due to the quadratic growth of the covariance matrix. A further development for larger environments is the use of particle filters with independent maps in each particle by the FastSLAM approach of Montemerlo et al. [36]. In an advancement, FastSLAM 2.0 [37] uses a Rao-Blackwellized scheme for an optimized efficiency by a reduced number of particles. An example for pedestrian indoor navigation with inertial sensors was presented with FootSLAM [38]. As major differences to robotic SLAM, FootSLAM uses just an intrinsic sensor (IMU) and no extrinsic environment scanners.

The main difference of railway and robotics SLAM is the map, that is mainly the trajectory of the train run instead of distant features and free space for an arbitrary motion. Additionally, the railway SLAM uses absolute positions from GNSS sensors. The SLAM method is used in railways to improve the accuracy of the track geometry and the data association to the correct track and location with every run on the same tracks. Hasberg [17] proposed the SLAM method for railways and is considered as the most related work to this thesis in terms of track mapping. In that work, a probabilistic spline representation was used for the map with measurements from an integrated navigation system. The cubic splines allow a sparse data set for 2-D positions of the track representation.

1.5 Thesis Outline

This outline describes the content of the chapters: The fundamentals are found in Chapter 2 with an introduction to Bayesian theory, state estimation algorithms, and probabilistic methods for localization and SLAM. Chapter 3 describes the fundamentals for a navigation within the railway domain and introduces the railway track map. Chapter 4 presents an analysis of train measurements for GNSS, IMU and magnetic field sensors in typical railway environments. The IMU is further analyzed as kinematic sensor for track geometry and as a vibration sensor for characteristic track signatures. Chapter 5 presents the probabilistic theory of Bayesian train localization and RailSLAM. The algorithmic implementations are found in Chapter 6 and with three different methods of the map based train localization and one SLAM method. Chapter 7 contains a description of the experimental data set and evaluation methods. The evaluation results of the three localization methods and RailSLAM with measurement data are presented in Chapter 8. There, the methods are discussed, compared and suggestions of improvements are given. Finally, a summary and conclusion is presented in Chapter 9.

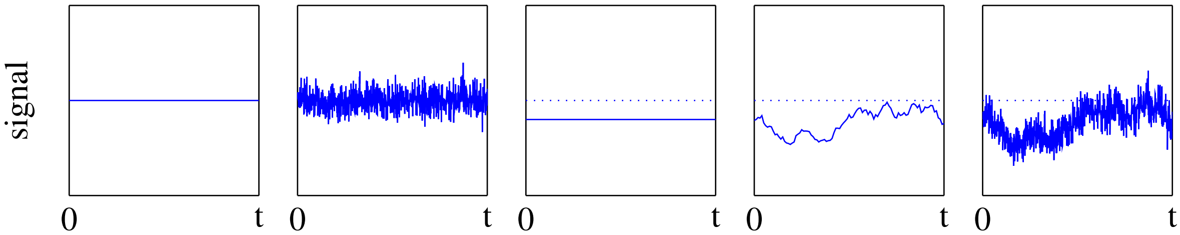
Chapter 2

Bayesian Framework

2.1 Probabilistic Theory and Bayesian Estimation

The general motivation for a probabilistic approach is uncertainty, which arises from imperfect and unknown information. Examples for uncertainty are an unpredictable environment, limited and noisy measurements, unknown and hidden states and partially observable variables as well as approximations of the used models and algorithms (see [13]). In probabilistic theory, the uncertainty is represented by probabilities and probability distributions. Estimation is the process to find an estimate from uncertain information. Bayesian estimation originates on concepts in an essay from Thomas Bayes and Richard Price [39].

As a motivation, Figure 2.1 shows typical random measurement distortions of an ideal measurement signal over time. Random errors can not be observed and adjusted in advance with a calibration. A calibration can only adjust for deterministic errors such as scale errors, offset, or non-linearity. The ideal signal is shown in Figure 2.1a) and the signal and additional



a) signal(t)

b) signal(t) + white noise

c) signal(t) + bias

d) signal(t) + bias drift(t)

e) signal(t) + bias drift(t) + white noise

Figure 2.1: Signals with distortions.

white Gaussian noise is shown in Figure 2.1b). Figure 2.1c) shows a constant but unknown bias or a nearly constant bias with very slow change. Examples are a varying bias at power-up, a uncompensated temperature effects or aging of the sensor. A faster changing bias over time is shown in Figure 2.1d). Figure 2.1e) represents a typical measurement with the random errors of white noise and a time varying bias. The unknown ideal signal is the desired information and the goal of an estimation is to find this information from the randomly distorted signals.

Probability Theory

The following defines the notation, basic probabilistic properties and basic concepts of a probabilistic framework. These definitions and concepts are prerequisite for the derivations of probabilistic train localization and railway SLAM. The following probabilistic properties can be found in many textbooks, such as [13, 40, 41].

The random variable x may be defined as discrete or continuous stochastic value and can take specific values with a particular probability. Furthermore, x can be also a vector and group several different random variables, also with continuous and discrete variables together. In this thesis, the estimation methods for localization and mapping comprises continuous and discrete variables. Furthermore, the random variables in this thesis are discretized in time with time steps and the current time step is indexed with k.

The probability distribution of a continuous random variable x is defined with probability density function (PDF) and denoted with $\mathrm{p}(x)$. The integral of the PDF is unity by definition:

$$\int \mathrm{p}(x)dx = 1. \tag{2.1}$$

The function for a discrete random variable is the probability mass function $\mu(x)$. There, each discrete outcome x^i, indexed with i, is associated with a point mass $\mu(x^i)$. This point mass is also the probability of the realization of one outcome. The probabilities of all n possible outcomes sum up to 1:

$$\sum_i^n \mu(x^i) = 1. \tag{2.2}$$

Discrete values can be described with the Dirac (impulse) delta function $\delta(\cdot)$ in the continuous domain. This Dirac delta function is defined by [41]:

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0. \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

and

$$\int_{-\infty}^{\infty} \delta(x) dx = 1. \tag{2.4}$$

A particular discrete value or sample x^i can be defined in the continuous domain with $\delta(x - x^i)$. The corresponding PDF of a discrete random variable for the continuously ranged variable x is [41]:

$$\mathrm{p}(x) = \sum_{i}^{n} \mu(x^i)\delta(x - x^i). \tag{2.5}$$

The joint probability distribution contains the combination of the random variables x and y:

$$\mathrm{p}(x, y). \tag{2.6}$$

A conditional probability is defined as [39]:

$$\mathrm{p}(x|y) = \frac{\mathrm{p}(x, y)}{\mathrm{p}(y)}. \tag{2.7}$$

The product rule is given by rearranging this equation and the joint distribution function $p(x, y)$ can be decomposed into two factored distributions:

$$\mathrm{p}(x, y) = \mathrm{p}(x|y) \cdot \mathrm{p}(y), \tag{2.8}$$

or alternatively into:

$$\mathrm{p}(x, y) = \mathrm{p}(y|x) \cdot \mathrm{p}(x). \tag{2.9}$$

In the case that there are multiple joint variables, the product rule can be applied multiple times, which refers to the chain rule in the Appendix A. The well-known Bayes' rule is derived from Eq. (2.8) and Eq. (2.9):

$$\mathrm{p}(x|y) = \frac{\mathrm{p}(y|x) \cdot \mathrm{p}(x)}{\mathrm{p}(y)}. \tag{2.10}$$

The posterior probability distribution is $\mathrm{p}(x|y)$, and called posterior in the following. The prior probability distribution is $\mathrm{p}(x)$ in this case and called prior. The Bayes' rule is used in a special way for estimation: The posterior is computed from the prior and the information update with the likelihood $\mathrm{p}(y|x)$. The denominator $\mathrm{p}(y)$ is a normalization constant, so that the calculated posterior is a proper PDF which integrates to unity (see Eq. (2.1)).

Conditional independency is described in [39] with: "Events are independent when the happening of any one of them does neither increase nor abate the probability of the rest". In the case that x is independent of y, the conditional probability $\mathrm{p}(x|y)$ simplifies to:

$$\mathrm{p}(x|y) = \mathrm{p}(x). \tag{2.11}$$

In combination with the product rule, a joint probability of $\mathrm{p}(x, y)$ can be split into independent parts, provided that x is independent of y:

$$\mathrm{p}(x, y) = \mathrm{p}(x|y) \cdot \mathrm{p}(y) = \mathrm{p}(x) \cdot \mathrm{p}(y). \tag{2.12}$$

Marginalization or the theorem of total probability is defined with [13]:

$$\begin{aligned}\mathrm{p}(x) &= \int \mathrm{p}(x, y)dy, \\ &= \int \mathrm{p}(x|y)\mathrm{p}(y)dy.\end{aligned} \tag{2.13}$$

The marginalization of the continuous variable y considers all possible values by integration. For a discrete variable y, the integral is replaced by a sum over all possible outcomes. The marginalization is used in the following ways: With marginalization, it is possible to remove y from the joint distribution function $\mathrm{p}(x, y)$ or from $\mathrm{p}(x|y)\mathrm{p}(y)$, even if x is dependent on y. In the following, this marginalization is used to eliminate a past state estimate. Another use case is the computation of the unknown distribution of $\mathrm{p}(x)$. There, $\mathrm{p}(x)$ is expanded according to Eq. (2.13) and solved with $\mathrm{p}(x|y)$ and $\mathrm{p}(y)$ that are either known or computable [13]. This type of marginalization is used when the joint distribution function is split with the product rule in $\mathrm{p}(y|x)\mathrm{p}(x)$ and the estimation of a next time step of $\mathrm{p}(x)$ requires knowledge from y. The unknown distribution $\mathrm{p}(x)$ is then expanded to Eq. (2.13) and computed. One example in the following is SLAM, in which the estimation of location and map is split in two estimations, but the location estimation still requires a map. Another example is Rao-Blackwellization, in which the state is split in linear and non-linear states that depend on each-other and different filters are used to compute these states.

Appendix A summarizes the probability calculus rules with more complex examples with additional joint variables and additional conditional variables. These rules will be used in the following to decompose the posterior probability distribution of the estimation problems into factored probability distributions. This process will be called factorization. In the context of this thesis, estimation problems are dynamic and involve probability distributions at different time steps or over several time steps. The current time step will be denoted with the index k, the previous or past with $k-1$ and the next time step in the future is indexed with $k+1$. A sequence of the random variable or estimation state x with all time steps from the start time to the current time is indexed with $x_{0:k}$. A sequence of all past time steps of x is denoted with $x_{0:k-1}$. A posterior PDF $p(x_k)$ is indexed with k and a prior PDF $p(x_{k\text{-}1})$ with $k-1$. Different notations in the literature use also $k+1$ for the posterior and k for the prior (see [40]). In terms of estimation, a hidden state x is estimated from measurements, where x is a vector of random variables and the elements are either discrete or continuous. The state is called hidden, because the true values are not known and also direct measurements contain uncertainty.

If $\mathrm{p}(x)$ is not further specified, it can be any arbitrary probability distribution. One particular PDF will be used in the context of this thesis: the Gaussian normal PDF. This PDF is commonly used, because many estimation methods are based on analytical derivations with the Gaussian PDF. The Gaussian PDF, denoted with $\mathcal{N}\left(\mu, \sigma^2\right)$, is defined with x as argument, μ as mean and σ^2 as variance:

$$\mathrm{p}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{2.14}$$

For a n-dimensional vector $\boldsymbol{x}$, the Gaussian distribution $\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ is multivariate with $\boldsymbol{\mu}$ as mean vector and $\boldsymbol{\Sigma}$ as covariance matrix:

$$\mathrm{p}(x) = \frac{1}{\sqrt{(2\pi)^n \cdot \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{2.15}$$

A particle filter uses sampled distributions. As an advantage, a sampled probability distribution can represent arbitrary distributions. A continuous probability distribution $\mathrm{p}(x)$ can be approximated with a limited sample number N of discrete samples x^i and weights w^i:

$$\mathrm{p}(x) \approx \sum_{i=1}^{N} w^i \delta(x - x^i). \tag{2.16}$$

The weights are similar to the point masses of Eq. (2.2) with the continuous function being approximated by samples. In contrast, Eq. (2.5) is exact because the discrete distribution with all n possible outcomes is included with the corresponding point masses. A distribution can be sampled by equidistant sampling points, e.g. same distances on a grid, uniform random distributed or arbitrary distributed.

Assumptions and Approximations

In the following estimation filters, several assumptions and approximations are made in order to realize a computational feasible estimation (see [13]). The estimation methods in the following comprises assumptions and approximations regarding to:

- conditional independence, also over time,
- completeness of state variables, models or functions,
- linearity of models or functions,
- probability distributions.

The Markov property assumes a conditional independence over time: a future state is conditionally independent to a past state if an interim state is known [13]. As a reminder, a state is here a set of random variables that are represented with probability distributions. For example, if the probability distribution of the current state with time step k is known, the future hidden state at time step $k+1$ is only conditionally dependent on the current state at k and not on the past hidden state at $k-1$. Further, the state representation may be incomplete due to computational complexity or because the additional states are not observable. Measurement models and transition models are an approximation with assumptions on the model function, linearity and the probability distribution. A commonly used assumption for measurements is additive white Gaussian noise and a linear measurement function of the state vector. The assumption that measurements are independent over time is often not correct: Unmodeled systematic errors or pre-filtered measurements cause correlations and thus dependency over time. The sampled distribution of Eq. (2.16) is an approximation because a limited number of samples, or particles respectively, are used to represent the distribution of a probability function.

Graphical Model and Posterior

A graphical model can be used for the design of a probabilistic model and further for a refinement or a comparison and differentiation between two approaches. Graphical models represent a graphical layout of a probabilistic model by visualizing observations and dependencies between random variables and also over time. A Bayesian network is a graphical model for random variables with directed and acyclic dependencies. The dynamic Bayesian network (DBN) shows additionally the causal dependencies over time. The hidden Markov model (HMM) can be seen as a special case of a dynamic Bayesian network [42]. A HMM is specified by a hidden and unknown random variable with a dependency in time to the previous hidden variable. Measurements have no dependency over time.

A simple positioning problem is formulated as an example: The hidden state of the position is X, as its true position is unknown and a sensor measures a dependent variable Z with an uncertainty. The DBN or HMM in Fig. 2.2 shows three sequential time steps.

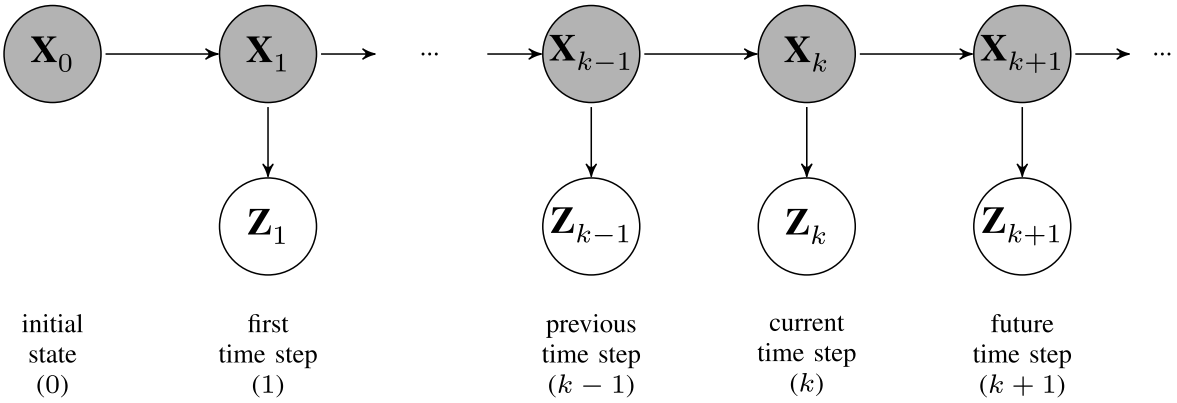


Figure 2.2: Dynamic Bayesian Network (DBN) with hidden state variables (X) and measurements (Z) over three time steps.

The current hidden state at time k depends on the last state $k-1$, as defined in general in an HMM and can be seen also in the model in Fig. 2.2. The given example contains the series of hidden variables over time $X_{0:k} = \{X_0, ..., X_k\}$ including the initial state X_0. The observations $Z_{1:k} = \{Z_1, ..., Z_k\}$ are conditionally independent between each other and conditional dependent on the true position X. This means, the outcome of the measurement Z is caused by the position X. The full posterior probability density contains the full history of the hidden state conditioned on the measurements, and is defined by:

$$\mathrm{p}(X_{0:k}|Z_{1:k}). \tag{2.17}$$

The filter posterior considers only the current state x_k and is defined by:

$$\mathrm{p}(X_k|Z_{1:k}). \tag{2.18}$$

Dynamic Bayesian Filter

The full posterior and the filter posterior are factorized in the following into different probability densities for further processing. The resulting probability densities are called likelihood, transition and prior. The filter computes the posterior density function from the measurement update, state transition and the prior density function. This filter is called a recursive filter because the output, i.e. posterior, for one time step becomes the input, i.e. prior, in the next time step.

The transition density contains the evolution of the estimate over time. The likelihood is a measure how likely the measurement fits to the estimate.

The full posterior of Eq. (2.17) is factorized in the Appendix B with Eq. (B.1) and splits the posterior in several factors:

$$\underbrace{\mathrm{p}(X_{0:k}|Z_{1:k})}_{\text{full posterior}} = \underbrace{\eta}_{\substack{\text{norm.}\\\text{factor}}} \cdot \underbrace{\mathrm{p}(Z_k|X_k)}_{\text{likelihood}} \cdot \underbrace{\mathrm{p}(X_k|X_{k\text{-}1})}_{\text{transition}} \cdot \underbrace{\mathrm{p}(X_{0:k\text{-}1}|Z_{1:k\text{-}1})}_{\text{prior}} . \qquad (2.19)$$

The posterior can be computed sequentially from the prior distribution, transition distribution and the likelihood of the measurements. A realization for the estimation of the full posterior is the particle filter.

The filter posterior of Eq. (2.18) is factorized in the Appendix B with Eq. (B.2), Eq. (B.3) and results in the general Bayesian filter definition. The filter posterior is computed from the prediction distribution and the likelihood by:

$$\underbrace{\mathrm{p}(X_k|Z_{1:k})}_{\text{filter posterior}} = \underbrace{\eta}_{\substack{\text{norm.}\\\text{factor}}} \cdot \underbrace{\mathrm{p}(Z_k|X_k)}_{\substack{\text{likelihood}\\\text{update, correction}}} \cdot \underbrace{\int \underbrace{\mathrm{p}(X_k|X_{k\text{-}1})}_{\text{transition}} \cdot \underbrace{\mathrm{p}(X_{k\text{-}1}|Z_{1:k\text{-}1})}_{\text{prior}} dX_{k\text{-}1}}_{\text{prediction}} . \qquad (2.20)$$

The prediction is computed with a marginalization of the previous state X_{k-1} [13]. Therefore, the prediction distribution $\mathrm{p}(X_k|Z_{1:k-1})$ is expanded by Eq. (2.13), because the current state depends on the previous state, as seen in Fig. 2.2. This type of equation is also known as the Chapman-Kolmogorov equation and refers to the marginalization of the past steps [40]. This means, that the information of the previous step is included while the dependency of X_{k-1} is excluded from the prediction distribution. This Bayesian filter is computed sequentially in two steps: first, the prediction step of Eq. (2.20) is processed, followed by the update step. An example for the realization of this Bayesian filter is the Kalman filter.

2.2 Probabilistic Data Filters

2.2.1 Kalman Filters

The Kalman filter implements the estimation of the filter posterior for linear state-space models with additive white Gaussian noise [43]. Every posterior, prior and initial PDF are represented with a Gaussian normal PDF with the state vector $\boldsymbol{x}$ as mean, and the covariance matrix $\boldsymbol{P}$:

$$\mathrm{p}(X_k|Z_{1:k}) \approx \mathcal{N}\left(\boldsymbol{x}_k, \boldsymbol{P}_k\right). \tag{2.21}$$

The generic Kalman filter algorithm implements the Bayesian filter of prediction and update Eq. (2.20). The Kalman filter estimates the state vector $\boldsymbol{x}$ and covariance matrix $\boldsymbol{P}$ with linear system model $\boldsymbol{F}$ and a process noise $\boldsymbol{Q}$. The prediction step or time update with control input $\boldsymbol{u}$ and input matrix $\boldsymbol{B}$ is [44]:

$$\boldsymbol{x}_{k|k\text{-}1} = \boldsymbol{F}_k \boldsymbol{x}_{k\text{-}1} + \boldsymbol{B}_k \boldsymbol{u}_k, \tag{2.22a}$$

$$\boldsymbol{P}_{k|k\text{-}1} = \boldsymbol{F}_k \boldsymbol{P}_{k\text{-}1} \boldsymbol{F}_k^T + \boldsymbol{Q}_k. \tag{2.22b}$$

The innovation is the difference between the predicted measurement and the actual measurement. Innovation $\boldsymbol{y}_k$ and innovation covariance $\boldsymbol{S}_k$ are defined with measurement matrix $\boldsymbol{H}$ and the sensor noise $\boldsymbol{R}$ [44]:

$$\boldsymbol{y}_k = \boldsymbol{z}_k - \boldsymbol{H}_k \boldsymbol{x}_{k|k\text{-}1} \tag{2.23}$$

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k\text{-}1} \boldsymbol{H}_k^T + \boldsymbol{R}_k. \tag{2.24}$$

The Kalman gain $\boldsymbol{K}$ is defined with:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k\text{-}1} \boldsymbol{H}_k^T \boldsymbol{S}_k^{-1}. \tag{2.25}$$

Finally, the measurement update with gain and innovation is:

$$\boldsymbol{x}_k = \boldsymbol{x}_{k|k\text{-}1} + \boldsymbol{K}_k \boldsymbol{y}_k, \tag{2.26a}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k\text{-}1} (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T, \tag{2.26b}$$

where $\boldsymbol{I}$ is an identity matrix and the covariance update is in the Joseph form [44].

For non-linear measurement or system models, there is the extended Kalman filter (EKF) with a linearization with a Taylor series approximation [44]. Another variant of the Kalman filter for non-linearities is the unscented Kalman filter (UKF) [45, 46]. The UKF uses the unscented transform and computes samples of a Gaussian distribution.

Error State Kalman Filter

The error-state Kalman filter (ESKF) is also known as linearized Kalman filter or indirect Kalman filter [47]. The use-case is the estimation with a non-linear system model and non-linear relations between state and measurement. For example, the update of angular changes of an attitude requires non-linear computations.

The ESKF estimates an error $\Delta\boldsymbol{x}$ in the update step, and the estimation state $\boldsymbol{x}$ is corrected with this error afterward. The vectors of ESKF state $\Delta\boldsymbol{x}$ and estimation state $\boldsymbol{x}$ are of same size. The goal is to estimate $\boldsymbol{x}$ while the error $\Delta\boldsymbol{x}$ is an auxiliary estimate that corrects the estimation state $\boldsymbol{x}$.

At first, this estimation state $\boldsymbol{x}$ is propagated with the non-linear system model:

$$\boldsymbol{x}_{k|k\text{-}1} = \boldsymbol{f}_k(\boldsymbol{x}_{k\text{-}1}, \boldsymbol{u}_k). \tag{2.27}$$

The process noise is defined for the prediction of the ESKF covariance. The error state system matrix $\boldsymbol{F}$ is either designed to propagate the errors, or $\boldsymbol{F}$ is the Jacobian of the system matrix [47]:

$$\boldsymbol{F}_{x,k} = \tfrac{d}{dx}\boldsymbol{f}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_k}, \tag{2.28}$$

In contrast to the standard Kalman filter, the covariance matrix $\boldsymbol{\Sigma}^{\Delta}$ is now computed for the ESKF:

$$\boldsymbol{\Sigma}^{\Delta}_{k|k\text{-}1} = \boldsymbol{F}_k\boldsymbol{\Sigma}^{\Delta}_{k\text{-}1}\boldsymbol{F}_k^T + \boldsymbol{Q}_k. \tag{2.29}$$

The ESKF state $\Delta\boldsymbol{x}$ is not predicted and can be omitted or set to zero: $\Delta\boldsymbol{x}_{k|k\text{-}1} = 0$. The error $\Delta\boldsymbol{x}_k$ is estimated in the update step. The measurement matrix is here a non-linear function of the *estimation state* $\mathrm{h}(\boldsymbol{x})$. The innovation is the difference between the predicted measurement of the estimation state $\boldsymbol{h}_k(\boldsymbol{x}_{k|k\text{-}1})$ and the actual measurement $\boldsymbol{z}_k$, see Eq. (2.23). The error $\Delta\boldsymbol{x}_k$ is now computed from the innovation, the Kalman gain $\boldsymbol{K}_k$, and without a predicted error state:

$$\Delta\boldsymbol{x}_k = \boldsymbol{K}_k \cdot (\boldsymbol{h}_k(\boldsymbol{x}_{k|k\text{-}1}) - \boldsymbol{z}_k). \tag{2.30}$$

The Kalman gain and the covariance update are the same as in the standard Kalman filter, see Eq. (2.25) and Eq. (2.26):

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}^{\Delta}_{k|k\text{-}1}\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{\Sigma}^{\Delta}_{k|k\text{-}1}\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{2.31a}$$

$$\boldsymbol{\Sigma}^{\Delta}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{\Sigma}^{\Delta}_{k|k\text{-}1}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T. \tag{2.31b}$$

Finally, the correction step corrects the errors from the estimation state:

$$\boldsymbol{x}_k = \boldsymbol{x}_{k|k\text{-}1} - \Delta\boldsymbol{x}_k. \tag{2.32}$$

2.2.2 Multiple Hypotheses Tracking

The main drawback of Kalman filters is the need for a Gaussian representation with a single mode. A Multiple Hypotheses Tracking (MHT) filter estimates a Gaussian mixture with different and discrete hypotheses. The filter tracks all hypothesis, but may output only the most probable hypothesis at each time step for further processing. A hypothesis is either a decision of a path in the motion model or a data association of measurements called correspondence (see [13]). This hypothesis is an auxiliary variable and time invariant because the decision or association does not change over time. Each hypothesis will be denoted with $\mathcal{H}$ and indexed with j. The posterior is now extended with an additional joint variable from the filter implementation. This can be done, as the filter can estimate any additional variable, even an own auxiliary variable of the filter. The product rule splits the extended joint posterior into the initial posterior, which is conditioned on a discrete decision or association, and all hypotheses conditioned on the measurements:

$$\mathrm{p}(X_k, \mathcal{H}|Z_{1:k}) = \mathrm{p}(X_k|Z_{1:k}, \mathcal{H}) \cdot \mathrm{p}(\mathcal{H}|Z_{1:k}). \tag{2.33}$$

The desired posterior is computed with a discrete marginalization over all hypothesis:

$$\mathrm{p}(X_k|Z_{1:k}) = \sum_j \underbrace{\mathrm{p}(X_k|Z_{1:k}, \mathcal{H}^j)}_{\text{posterior with decision}} \cdot \underbrace{\mathrm{p}(\mathcal{H}^j|Z_{1:k})}_{\text{weight } w^j}. \tag{2.34}$$

As the hypotheses are discrete, the marginalization is a sum over all realizations of $\mathcal{H}$. The second factor is the weight w^j of j'th hypothesis. The factorization of the weight estimation involves Bayes' rule and a normalization factor η:

$$\underbrace{\mathrm{p}(\mathcal{H}^j|Z_{1:k})}_{w_k^j} = \eta \cdot \underbrace{\mathrm{p}(Z_k|Z_{1:k\text{-}1}, \mathcal{H}^j)}_{\text{likelihood}} \cdot \underbrace{\mathrm{p}(\mathcal{H}^j|Z_{1:k\text{-}1})}_{w_{k\text{-}1}^j}. \tag{2.35}$$

The weight estimate consists of a measurement likelihood and a prior weight. The likelihood is defined with the j-th innovation $\boldsymbol{y}^j$ (see Eq. (2.23)) and the innovation covariance $\boldsymbol{S}^j$ (see Eq. (2.24)) [41]:

$$\mathrm{p}(Z_k|Z_{1:k\text{-}1}, \mathcal{H}^j) \propto \exp\left(-\frac{1}{2}(\boldsymbol{y}^j)^T(\boldsymbol{S}^j)^{-1}\boldsymbol{y}^j\right). \tag{2.36}$$

The normalization factor η is calculated from the condition, that all weights sum up to one:

$$\eta = \frac{1}{\sum_j \tilde{w}_k^j}. \tag{2.37}$$

The different hypothesis are represented by a Gaussian PDFs and estimated by a Kalman filter, EKF or UKF. The resulting distribution is a mixture of weighted Gaussians:

$$\mathrm{p}(X_k|Z_{1:k}) \approx \sum_j w_k^j \cdot \mathcal{N}\left(\boldsymbol{x}_k^j, \boldsymbol{P}_k^j\right). \tag{2.38}$$

New hypothesis may be added if new decisions for paths or correspondences are necessary. The amount of hypotheses are maintained by pruning, which eliminates unlikely hypotheses with a low weight. An output function may select and output the most likely hypothesis with the highest weight, while the filter continues to track all hypotheses.

2.2.3 Particle Filter

A particle filter represents a probability distribution with discrete samples of the estimation state, called particles. The probability distribution with particles is represented by the sample density and the particle weight. A particle filter estimates the full posterior of Eq. (2.19) and can handle non-linear models and arbitrary probability distributions of posterior, state-transitions and measurements [48]. The posterior of Eq. (2.19) is represented by the particles set $\{\boldsymbol{x}_{0:k}^i, w_k^i\}_{i=1}^N$ and each particle consists of a state vector hypothesis $\boldsymbol{x}^i$ and the weight w^i. The particle probability density is also called particle cloud and is mathematically represented by a sum of Dirac-delta functions $\delta(x)$, as defined in Eq. (2.16). The full posterior is now approximated with a finite number of particles N:

$$\mathrm{p}(X_{0:k}|Z_{1:k}) \approx \sum_{i=0}^{N} \omega_k^i \cdot \delta(\boldsymbol{x}_{0:k} - \boldsymbol{x}_{0:k}^i). \tag{2.39}$$

The i-th particle $\boldsymbol{x}_{0:k}^i$ and the weight w^i of N particles represent one sample of the posterior of all time steps until k. In other words, a particle $\boldsymbol{x}_{0:k}^i$ contains the i-th realization of the estimation state including the state history.

A probability density function can be represented by particles or samples in three ways: The first case uses constant sample spacings, where the weights contain the information of the probability function with $w^i = p(x^i)$. The second case considers constant weights $w^i = \frac{1}{N}$, where the sample density contains the probability information. The third combines different weights with arbitrary sample densities.

Importance Sampling

The posterior density function cannot be sampled directly. Therefore, particles are generated from a proposal function, also known as importance density which is feasible to compute [48]:

$$x^i_{0:k} \sim \mathrm{q}(X_{0:k}|Z_{1:k}). \tag{2.40}$$

Afterward, these particles are weighted [48], while the weights are proportional to the fraction of posterior over proposal function:

$$w_k = \eta \frac{\mathrm{p}(X_{0:k}|Z_{1:k})}{\mathrm{q}(X_{0:k}|Z_{1:k})}. \tag{2.41}$$

The importance density (proposal function) covers the domain of the posterior and is computationally feasible, but different to the posterior. This difference is compensated with the weights, that correct the importance density to the posterior. The general proposal function is defined in a recursive form:

$$\mathrm{q}(X_{0:k}|Z_{1:k}) = \mathrm{q}(X_k|X_{0:k\text{-}1}, Z_{1:k}) \cdot \underbrace{\mathrm{q}(X_{0:k\text{-}1}|Z_{1:k\text{-}1})}_{\text{recursive part}}. \tag{2.42}$$

The general weight function is computed from 2.41 with 2.42 and the full posterior factorization 2.19 in the nominator:

$$w_k = \eta \frac{\mathrm{p}(Z_k|X_k) \cdot \mathrm{p}(X_k|X_{k\text{-}1})}{\mathrm{q}(X_k|X_{0:k\text{-}1}, Z_{1:k})} \cdot \underbrace{\frac{\mathrm{p}(X_{0:k\text{-}1}|Z_{1:k\text{-}1})}{\mathrm{q}(X_{0:k\text{-}1}|Z_{1:k\text{-}1})}}_{w_{k-1}}. \tag{2.43}$$

A popular choice of the proposal function is with the transition function, while the weight function contains the likelihood [48, 40]:

$$\mathrm{q}'(X_{0:k}|Z_{1:k}) = \underbrace{\mathrm{q}'(X_k|X_{k\text{-}1}, Z_k)}_{\text{transition}} \cdot \underbrace{\mathrm{q}'(X_{0:k\text{-}1}|Z_{1:k\text{-}1})}_{\text{recursive part}}, \tag{2.44}$$

The weight function is then:

$$w_k = \eta \cdot \mathrm{p}(Z_k|X_k) \cdot w_{k-1}. \tag{2.45}$$

Nevertheless, the choice and definition of the proposal function is a crucial part of the particle filter design process. The weights of the particle filter sum up to one, similar to Eq. (2.1), where the sum over all probabilities is one. The normalization factor η is computed from this property in a deterministic way:

$$\eta = \frac{1}{\sum_{i=1}^{N} \tilde{w}_k^i}. \tag{2.46}$$

For many applications, the particle distribution is unfeasible to process. A characteristic value of a distribution are the minimum mean square estimate (MMSE) or minimum variance estimate and a corresponding covariance [40]. The MMSE with a mean and covariance is computed as output estimate:

$$\hat{\boldsymbol{x}}_k = \sum_{i=1}^{N} \omega_k^i \cdot \boldsymbol{x}_k^i, \tag{2.47a}$$

$$\hat{\boldsymbol{P}}_k = \sum_{i=1}^{N} \omega_k^i (\boldsymbol{x}_k^i - \hat{\boldsymbol{x}}_k)(\boldsymbol{x}^i - \hat{\boldsymbol{x}}_k)^T. \tag{2.47b}$$

The generic particle filter algorithm in Fig. 2.3 contains the main steps with importance sampling, resampling and output estimate.

```
function UPDATE PARTICLE FILTER($x_{k\text{-}1}^{1:N}, \omega_{k\text{-}1}^{1:N}, u_k, \nu_k, z_k, \epsilon_k$)
    for all $N$ particles do
        $x_k^i$ = sampleProposal($x_{k\text{-}1}^i, u_k, \nu_k$)
        $\tilde{\omega}_k^i$ = weight($x_{k\text{-}1}^i, \omega_{k\text{-}1}^i z_k, \epsilon_k$)
    end for
    $\omega_k^{1:N}$ = normalize($\tilde{\omega}_k^{1:N}$) with Eq. (2.46)
    $[\hat{x}_k, \hat{P}_k]$ = outputEstimate($x_k^{1:N}, \omega_k^{1:N}$) with Eq. (2.47)
    if isTimeForResample($\omega_k^{1:N}$) then
        $[x_k^{1:N}, \omega_k^{1:N}]$ = resample($x_k^{1:N}, \omega_k^{1:N}$)
    end if
    return $x_k^{1:N}, \omega_k^{1:N}, \hat{x}_k, \hat{P}_k$
end function
```

Figure 2.3: Generic particle filter update function for one time step and measurement update.

Likelihood

The sensor likelihood is the conditional density $p(Z_k|X_k)$. This function calculates a probability how likely the measurement z fits to the predicted measurement $\mathrm{h}(x^i)$ of one particle. The non-linear measurement model is denoted with $\mathrm{h}(\cdot)$. The likelihood evaluates the innovation $y^i = z - \mathrm{h}(x^i)$

of each particle (see Eq. (2.23)). A popular choice is a measurement model with additive white noise n:

$$z = \mathrm{h}(x) + n. \tag{2.48}$$

The likelihood function is then defined with the innovation vector $\boldsymbol{y}_k^i$ of the i-th particle and the covariance of the sensor noise $\boldsymbol{\Sigma}_z$:

$$\mathrm{p}(Z_k|X_k) \hat{=} \sum_{i=1}^{N} \mathcal{N}\left(\boldsymbol{y}_k^i, \boldsymbol{\Sigma}_z\right). \tag{2.49}$$

The resulting probability is further used as a weighting of the different particles. In practice, the likelihoods can be used without constants, such as the constant of the Gaussian normal function, because the normalization factor η compensates any constant afterward. Likelihood functions can be also used in the proposal function [13]. There, particles are sampled from a measurement PDF.

Resampling

After several time steps, some particles may carry a very high weight while the rest has a very low weight. These low weighted particles are inefficient and this process is called degeneration. In order to avoid this, a systematic resampling [48] of the particle distribution can solve this problem.

A metric for particle depletion is the effective number of particles $\hat{N}_{\text{eff}}$ [40]. The particle distribution is resampled if $\hat{N}_{\text{eff}}$ is below a threshold N_{th}:

$$N_{\text{th}} > \underbrace{\frac{1}{\sum_i (w_k^i)^2}}_{\hat{N}_{\text{eff}}}. \tag{2.50}$$

The threshold N_{th} depends on the total number of particles N_p and is practically defined within $1 < N_{\text{th}} < N_p$.

Particle Filter Challenges

The drawbacks and challenges for a particle filter application are:

- *Divergence* happens, if the particles and the estimate are apart from the true values and cannot recover. The result is an unstable filter, and can be compensated with a continued resampling, the design of the proposal function and the insertion of extra sampling noise (see

[40]). In the case that a divergence happens, a filter monitoring can detect a severe divergence by appropriate measurements and restart the filter [40].

- *Overconfidence* happens, if the measurement noise is too small and correlations are disregarded in the sensor model. The filter converges very quickly to the measurements and results in a too small particle distribution after resampling. This can lead to the described divergence, if the true values are not covered with near particles anymore.
- *Degeneracy* of the particle distribution is the effect where nearly all weight is accumulated on one or a few particles. The state-of-the-art approach is resampling the particle distribution [48].
- The *dimensionality* problem arises from the fact, that every additional dimension of the state vectors requires a much higher number of particles. This leads to a high computational complexity [49]. A low dimensional state vector for the particle filter is desirable and this can be achieved with the use of different filters and a state dimension reduction of the particle filter.

State Dimension Reduction

The dimension of the estimation state can be reduced with a combination of different filters. Kalman filters are optimal for linear and Gaussian cases while the particle filter can handle non-linear distributions. The state dimension of the particle filter can be reduced to the dimension of the non-linear states. Therefore the state vector is split into linear states X^l and non-linear states X^n, and the joint posterior is now:

$$\mathrm{p}(X^l, X^n|Z). \tag{2.51}$$

A separation of linear and non-linear states with dependencies on each other result in one Kalman filter per particle. This approach is called marginalized particle filter [40] or Rao-Blackwellized particle filter [50].

In the case that the estimation of the linear states are not dependent from the non-linear states, the product rule splits the joint state vector:

$$\mathrm{p}(X^l, X^n|Z) = \mathrm{p}(X^n|X^l, Z) \cdot \mathrm{p}(X^l|Z), \tag{2.52}$$

and the particle filter uses the linear estimate as input. The linear states are estimated with a Kalman filter in a first step. Afterward, the linear distribution is sampled by particles and the non-linear states are estimated with a particle filter.

2.3 Localization and Mapping

In the context of this thesis, the terms position, location and navigation will be used in a defined meaning: A position is expressed in a coordinate frame, such as a geographic GPS position measurement. A location will be used in combination with a map, especially for the train location on the railway tracks. This location contains a track identifier (ID) and a 1-D location on that track. The processes for the estimation of a position or a location are called positioning or localization, respectively. Navigation will be used in this context as the comprehensive estimation of positions, locations and velocities. The common navigation tasks by path planning or guidance are not considered here.

2.3.1 Localization

The location and the map are defined in a probabilistic manner. That means, that the location values and the map content can be ambiguous and represented by a probability distribution. Figure 2.4 shows the block diagram of a generic sequential Bayesian localization with a map. The location is computed from sensor inputs, the map and the prior location of the last time step.

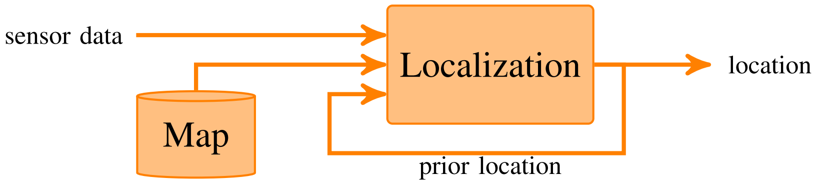


Figure 2.4: Block diagram of a generic Bayesian localization.

The posterior of a generic localization problem contains the hidden state X, conditioned on all measurements $Z_{1:k}$, all control inputs $U_{1:k}$ and the map M. The process of localization is divided into smaller steps by a factorization. The main components are the transition distribution and an update of sensor information. The transition is computed from previous location, map and motion. The sensor updates are dependent on the location. If the sensors measure directly a property of the map, the measurement update is conditionally dependent on the map. The factorized version for the *full posterior* of the localization is defined as:

$$\begin{aligned}&\mathrm{p}(X_{0:k}|Z_{1:k},U_{1:k},M) = \\ &\qquad \eta\cdot\underbrace{\mathrm{p}(Z_k|X_k,M)}_{\substack{\text{meas.}\\ \text{likelihood}}}\cdot\underbrace{\mathrm{p}(X_k|X_{k\text{-}1},U_k,M)}_{\text{transition}}\cdot\underbrace{\mathrm{p}(X_{0:k\text{-}1}|Z_{1:k\text{-}1},U_{1:k\text{-}1},M)}_{\text{prior}}.\end{aligned} \tag{2.53}$$

The factorized *filter posterior* of the generic localization contains:

$$\begin{aligned}&\mathrm{p}(X_k|Z_{1:k},U_{1:k},M) = \\ &\quad \eta\cdot\underbrace{\mathrm{p}(Z_k|X_k,M)}_{\substack{\text{meas.}\\ \text{likelihood}}}\cdot\int\underbrace{\mathrm{p}(X_k|X_{k\text{-}1},U_k,M)}_{\text{transition}}\cdot\underbrace{\mathrm{p}(X_{k\text{-}1}|Z_{1:k\text{-}1},U_{1:k\text{-}1},M)}_{\text{prior}}dX_{k\text{-}1}.\end{aligned} \tag{2.54}$$

2.3.2 Mapping

Mapping is the process to create or change the content of a map. In the context of this thesis, a map is generated or updated from sensor data. The mapping requires also a data association of the sensor data to the appropriate map data, so that the map content is updated at the right place. In a probabilistic mapping, the sensor data and the map data contain uncertainty. A data association is ambiguous, if the data association contains discrete variables with multiple possibilities. This is the case for a data association to different tracks. The probabilistic map contains then multiple map hypotheses. Figure 2.5 shows the mapping process from sensor data, data association and the previous map. In the case of a map creation, there is no previous map available. This type of mapping will be called *white-space* mapping in the following. The update of an existing map can be carried out by replacement of the previous map with new information or an information fusion of the previous map with new information. The latter case will be called *prior-map* mapping.

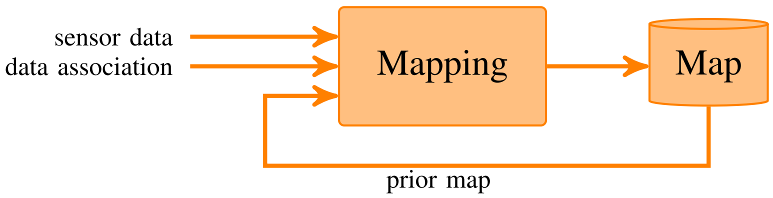


Figure 2.5: Block diagram of a generic mapping process with prior map information.

2.3.3 SLAM: Simultaneous Localization and Mapping

There is a causal problem about what comes first: A localization requires a map and the mapping process requires a location for the data association. Additionally, there are uncertainties from the estimations of the location and the map. The method of simultaneous localization and mapping (SLAM) joins both processes of localization and mapping and addresses the computation with uncertainty. Figure 2.6 shows the linked localization and mapping processes. The two processes are loosely linked and compute the location and the map separately: At first, the location is computed from the previous map and sensor data. Secondly, the map is updated with the location and sensor data.

The development of SLAM focused mainly on a problem in the field of robotics with relative measurements only (see [13]). The classic SLAM problem is a robot with measurements from an onboard laser range scanner on a plane 2-D environment. The range measurements are processed and reduced to landmarks. These landmarks refer to obstacles or walls. The map contains a set of these landmarks including the locations and features of the landmark. The robot's location and orientation is combined and estimated with the robot's pose. The classic robotic SLAM considers only relative measurements, such as range measurements to the landmarks. There are no absolute position measurements involved. As a consequence, the map is not to scale nor rotated to north and only linked to a start position or a zero origin. The classic robotic SLAM works as follows (see [13]): At first, the robot starts to move with an empty map. The robot's pose is estimated from the starting point and from motion measurements, such as wheel turns or wheel speeds. Landmarks are extracted from measurements and put into the map with locations relative to the robot's location. An accumulation of relative motion measurements leads to an increase of uncertainty over time: The uncertainty of the location and the location of all landmarks in the map grow over time and motion. The growing uncertainty of the robot's location transfers directly to the uncertainty of the newly recorded landmark locations. As a consequence, all landmark

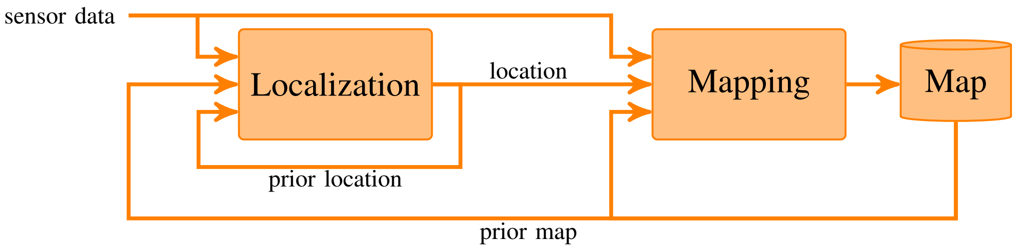


Figure 2.6: Block diagram of a combined localization and mapping process.

locations are correlated with the robot's locations. Additionally, the data association of measurements to a certain landmark is unknown and this data association must be estimated as well. It could be shown, that SLAM can reduce the uncertainty of the pose and the map [13]. An important event therefore is called *loop closure* and describes the return of the robot to a previously mapped area. In the event of a loop closure, the SLAM algorithm with relative measurements only shows its main advantage: The pose and all landmark positions are updated and their uncertainties are reduced. This effect can be explained with the correlations of the pose and the landmark locations [13].

There are two versions of SLAM regarding the posterior: The *online SLAM* uses a filter posterior, and refers to the estimation of the current location state X_k and the map M:

$$\mathrm{p}(X_k, M|Z_{1:k}, U_{1:k}). \tag{2.55}$$

The *full SLAM* posterior refers to the full trajectory of all locations $X_{0:k}$ and the map M:

$$\mathrm{p}(X_{0:k}, M|Z_{1:k}, U_{1:k}). \tag{2.56}$$

There are many approaches published for this classic SLAM problem and a recent overview is found in [51]. The most common are the EKF-SLAM [35] (online SLAM), the GraphSLAM with factor graphs [13] (full SLAM), and the FastSLAM with particle filters [36, 37] that uses the full SLAM approach and can be used as online SLAM.

The EKF-SLAM uses a Gaussian representation for the robot's pose (location and orientation) and the landmark locations. This algorithm solves the unknown data association by the most likely and nearest location [35]. Figure 2.7 shows a simplified block diagram of an EKF-SLAM with already known data association of the measurements to the landmark. Additionally to the location, as depicted in the simplified block diagram, the EKF-SLAM considers also the orientation in a combined robot's pose. The prediction step of the Kalman filter uses a prior location and motion measurements. The update function computes the map and the pose from the predicted location, the landmark measurements and the landmarks of the prior map. If a new landmark is measured, the landmark including its location is put to the map. In the case that an existing landmark is measured, the robot location and the landmark location are simultaneously updated. In contrast to the localization and mapping in Fig. 2.6, the update function of the EKF-SLAM contains a combined location and map update as shown in Fig. 2.7. The landmark estimates are fully correlated

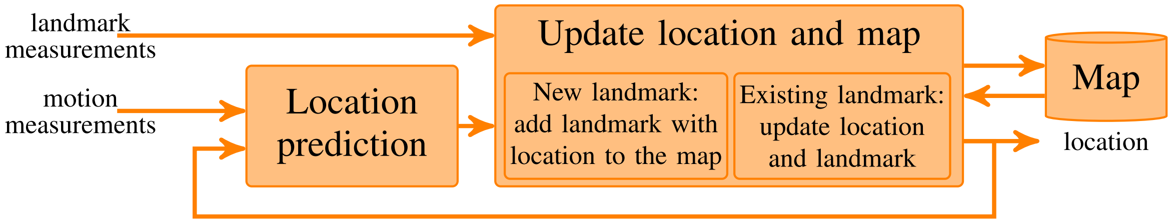

landmark measurements
motion measurements
Location prediction
Update location and map
New landmark: add landmark with location to the map
Existing landmark: update location and landmark
Map
location

Figure 2.7: Simplified block diagram of a the robotic EKF-SLAM with landmark measurements.

with each other in the covariance matrix and linked to the position uncertainty. As a main drawback, the EKF-SLAM is limited to a low number of landmarks. The reason for this is the state vector of the EKF, which contains all landmarks and thus the covariance matrix is growing quadratically with the number of landmarks in the map [13].

Montemerlo et al. proposed "FastSLAM", a SLAM approach with a particle filter [36]. The important advancement is the separation of position uncertainty and map uncertainty as well as the conditional independence of each feature estimate. This is possible, as each particle has its own and independent path history. The particle cloud represents the position uncertainty. The feature estimates are conditionally independent to each other as they are conditioned on the full path $X_{0:k}$.

There are different versions of SLAM in terms of the used sensor data: There is exteroceptive sensor data that are direct measurements of the user's environment. Example sensors are camera, radar or lidar and these sensors can measure also distant features of the environment from the user. An example of exteroceptive SLAM is shown in Figure 2.8 with a commercial robotic vacuum cleaner (XIAOMI Mi Robot Vacuum) that uses a laser scanner. The map shows obstacles and walls, and it is used here to optimize the trajectory (gray line) for the purpose of floor cleaning. Interoceptive sensor data refers to measurements of an inner state of the user, which measure features of environment indirectly over the user's position or motion. An example given with inertial sensors and Figure 2.9 shows an indoor map from an interoceptive SLAM created by the motion of a foot (FootSLAM) [52, 38, 53]. There, the map is the walkable area, that is created from the trajectory of the pedestrian. In the context of this thesis, an interoceptive version of SLAM is used for the dynamic measurement of the track geometry with inertial sensors.

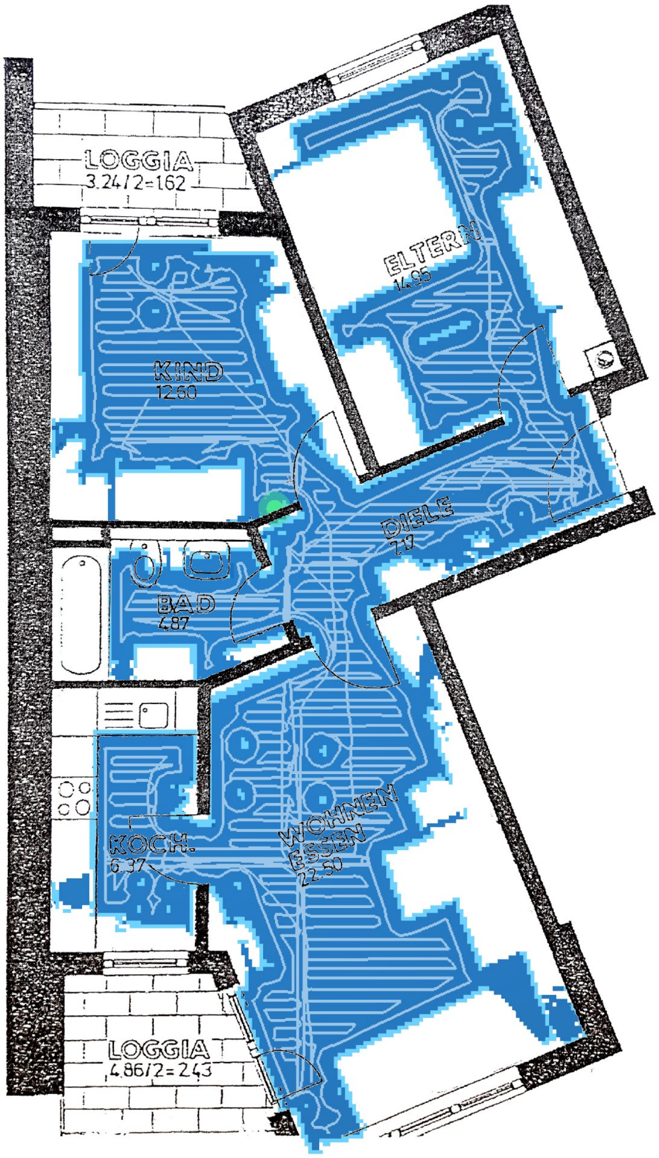


Figure 2.8: Exteroceptive SLAM: Robotic vacuum cleaner map (blue) from laser scanner data with floor plan overlay (black).

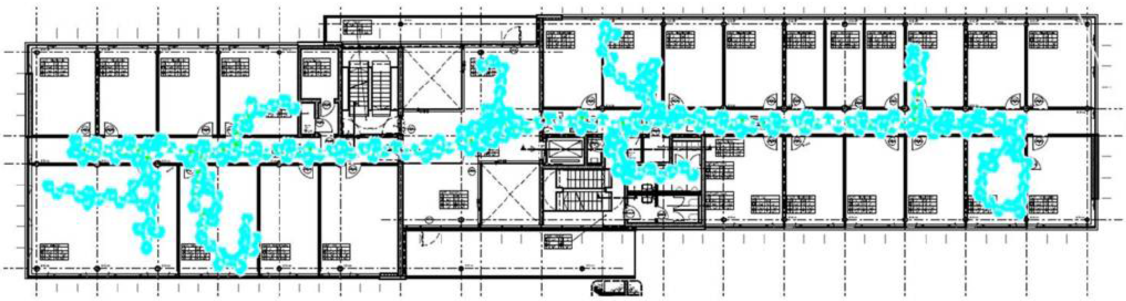


Figure 2.9: Interoceptive SLAM: Indoor map of an office environment map (cyan) with floor plan overlay (black) [53]. Example map of the FootSLAM algorithm from foot mounted IMU data.

Chapter 3

Navigation in the Railway Environment

3.1 Railway Tracks

Trains run always on tracks. This fact is essential for the navigation of trains, because a railway track constrains the positions and trajectory of a train to one degree of freedom in the along-track direction. Multiple tracks are connected to a track network via railway switches or crossings. A railway switch, also known as points, connects tracks and can further change the connection between two possible tracks. Hence, the trajectory of a train is switchable to one of the connected tracks. A plain railway crossing is an overlap of two tracks with no access from one track to the crossing traverse track. There are special crossings with access to the traverse tracks that are called slip switches or diamond crossings, respectively. A single slip switch contains two switches and a double slip contains four inbuilt switches and a crossing (see [15]). Finally, a track can be terminated at one side with no connection to other tracks.

3.1.1 Railway Track Geometry

Railway tracks are planned in railway design offices with the following considerations and can be found in [15]: On the one hand, a straight line or track is desired between two destinations, as it is the shortest and therefore most economical connection. On the other hand, a railway track has to pass around urban areas, private property or terrain obstacles such as mountains, rivers or coast line. Once, curves are implemented in order to pass around these obstacles, there are further dynamics for a running train. Trains can derail or even tilt over if the speed is too high for a certain curve radius due to lateral forces. Therefore, the track design defines also a speed limit for safety reasons. The lateral train dynamics by acceleration and jerk are further limited for passenger comfort and to avoid a shift of cargo.

Railway tracks are fixed to the Earth, so any location on the tracks represent as well a geographic position. In the following, the track geometry is defined along the one dimensional track location s. One track characteristic is also the heading angle ψ at a certain track location s. A change of the heading angle over location s is called curvature (c^{ψ}), or heading curvature more precisely. A curvature is the inverse of a radius and the preferred unit in track design. The reason is a singularity of the radius when it comes to straight tracks. A straight track has a curvature of zero but an infinite radius in theory which results in a large positive or negative radius in practice with tolerances.

A railway track requires at least a C^1 continuity up to the first differentiable because of the inertia of a train. A C^0 continuity defines a continuously connected track without breaks or gaps. Besides a derailment, a C^0 continuity limits the accelerations to finite values of a running train that tries to stay on the track. Additionally for C^1 continuity, the tangents must match from two sides at any track location and thus the turn rates of a running train are limited to finite values. If a straight track connects to a curve with a constant radius, a sudden lateral acceleration acts on the train and causes a theoretical infinite lateral jerk. Practically, there is a high jerk because the undercarriage and suspensions absorb a sudden change of the acceleration. Nevertheless there is a risk of a cargo shift and a high jerk causes discomfort for passengers. A higher continuity is achieved with a linear increase of the curvature between straight and a circular arc with the use of a Clothoid, also known as Euler's spiral. There are also other transition curves with an S-shaped increase of curvature or splines for even higher orders of continuity and a smoother train run.

The superelevation of tracks allows to increase the train speed for a given limit of the lateral acceleration. This is possible, because the gravity compensates parts of the lateral acceleration for a tilted train run in curves. The superelevation described here with the bank angle (ϕ) of the track. This bank angle is also limited, because a train may stop at all possible locations in a curve and then a portion of gravity acts as lateral force. A change of the bank angle over the track location s is the bank curvature c^{ϕ}. A ramp-shaped or S-shaped change of the bank angle satisfy the continuity requirements regarding the bank angle. A smooth train run at a curve entry is achieved with simultaneous changes of the bank angle and the heading curvature, e.g. via Clothoid, followed by a circular curve with constant bank angle and constant curvature. Finally, the slope angle θ is the inclination along a track for changes of the height level. The slope is limited because a train may stop downhills or start towards uphills. The change of a slope angle is the slope curvature c^θ and indicates a crest or a sag (see [15]). All track curvatures are defined as a change of angle over the track location s:

$$\text{bank curvature:} \quad c^\phi(s) = \frac{d\phi(s)}{ds}, \tag{3.1}$$

$$\text{slope curvature:} \quad c^\theta(s) = \frac{d\theta(s)}{ds}, \tag{3.2}$$

$$\text{heading curvature:} \quad c^\psi(s) = \frac{d\psi(s)}{ds}. \tag{3.3}$$

Track design follows usually national limitations and in special cases there are exceptional permissions. As an example, the German limitations for the 1435 mm track gauge are 6.8° for the bank angle from superelevation and 2.3° for the slope (see [15, 54]). The minimum curve radius is above 100 m and above 180 m for new tracks [54], as trains can derail in too tight curves. A maximum lateral acceleration of 1 $\mathrm{m/s^2}$ limits the speed in curves and the maximum lateral jerk is defined with 0.45 $\mathrm{m/s^3}$ [15].

3.1.2 Railway Switches

A standard railway switch with its elements is shown in Figure 3.1. A switch consists of two tracks: the straight track, usually for the main line, and the diverging track, usually for the secondary line. There are negligible and rare examples of switches with three tracks from special railways, which will be disregarded here. In the case that the switch is a curved switch, the least curved track is still denoted as straight track. Beside the standard switch, there are different switch versions as the curved switch, Clothoid switch and the Y-switch with equally curved tracks in opposite directions (see [15]).

A train is guided by the switch blades to the straight or diverging track. These switch blades can be moved so that one blade fits always to an outer rail. The tracks are interrupted by the frog gap in front of the crossing point of the rails from the two tracks. This gap is needed for the wheel rims to pass the crossing rails. The V-shaped crossing rails are known as frog, common crossing, or V-rail. A frog can be rigid or movable in combination with the blades for higher train speeds. During the passage over the gap, the wheels are guided by guide rails. The clearance gauges of the two tracks overlap within the switch. In some countries, the point of the clearance intersection is indicated with a marker. For a collision free operation, only one rail vehicle is allowed to access the switch at a time (see [15]).

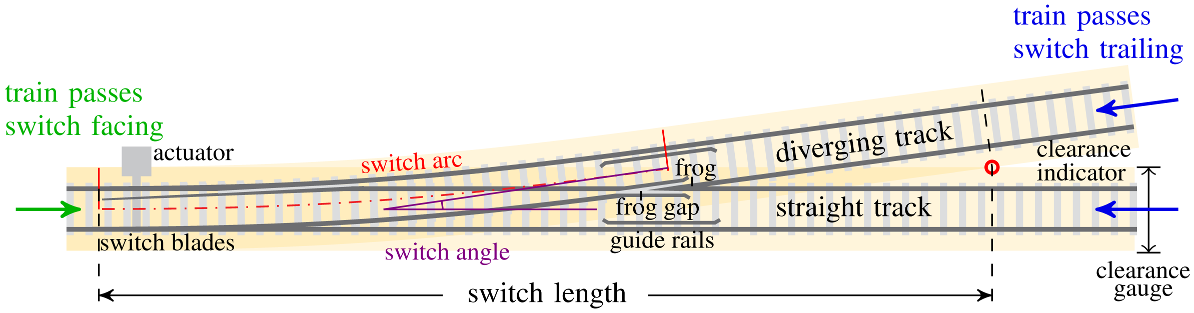


Figure 3.1: Railway switch.

A train passes the switch *facing* in the direction of branching tracks with two possible travel paths. In the opposite direction, the train passes the switch *trailing* with only one possible travel path. From a train's view, the two tracks merge to one track. The travel directions of *left* and *right* of a facing switch run are important for train localization. The resulting path of a left or right switch run will be denoted as *switch way* in the following. There are different ways to define a switch start, switch end and a measure of the length. In the context of train localization, the following definition is beneficial: The switch starts with the blades and also the two possible tracks are defined from that point. The switch ends at the clearance indicator for tolerance reasons of the train localization evaluation. Within the switch, a wrong track identification is uncritical, as only one train can occupy the switch. The track selectivity is evaluated after the switch end.

Table 3.1 shows a list of standard switch examples used in Germany. The radius or curvature of the turning track characterizes the standard switch and limits the train speed. The table presents typical and characteristic switch data with the arc radius, tangent ratio and speed limit [15]. The values of switch curvature, switch angle, arc length, switch length and transit time are calculated from this characteristic data. The switch angle is the heading change caused by the switch arc computed from the tangents ratio. The arc length is calculated from radius and switch angle. The switch length is calculated for a 3.5 m clearance with a continued straight track from radius, ratio and clearance distance (see [15]). Finally, the transit time is here the time of a train run over the curved part of the switch at the speed limit. This time ranges between 1.5 s and 2.8 s and is of interest for the switch-way identification.

switch type		**speed**	**curva-**	**switch**	**arc**	**switch**	**transit**
radius,	tangent	**limit**	**ture**	**angle**	**length**	**length**	**time**
r,	$\tan\alpha$	v_{max}	$c^{\psi} = \frac{1}{r}$	α	$l_{\text{arc}} = r\alpha$	l_{sw}	$t = \frac{l_{\text{arc}}}{v_{\text{max}}}$
[m],	[ratio]	[km/h]	[1/km]	[°]	[m]	[m]	[s]
190	1:9	40	5.26	6.3	21.0	42.0	1.9
”	1:7.5	”	”	7.6	25.2	38.8	2.3
300	1:14	50	3.33	4.1	21.4	59.7	1.5
”	1:9	”	”	6.3	33.2	48.1	2.4
500	1:12	60	2.00	4.8	41.6	62.8	2.5
”	1:14	”	”	4.1	35.7	66.8	2.1
760	1:18.5	80	1.32	3.1	41.1	85.3	1.8
”	1:14	”	”	4.1	54.2	76.1	2.4
1200	1:18.5	100	0.83	3.1	64.8	97.2	2.3
2500	1:26.5	120	0.40	2.2	94.3	139.9	2.8

Table 3.1: Basic designs of German standard switches (see [15]).

An example is presented with a typical scenario of a switch followed by parallel tracks: The curved switch track is part of an S-shaped track that results in a parallel track to the straight track with a distance 4 m. The S-shaped track is composed from the switch arc, followed by a straight track and followed by the same arc turning in opposite direction. Figure 3.2 shows this example for the six representative switches from Table 3.1. A transit time is calculated for a train traveling at the speed limit over the switch and the continued curve until the clearance of 3.5 m is reached. For each parallel track scenario, there are roughly 4 s of transit for the switch-way identification. As a special property of a switch, the two tracks of the competing switch ways differ in geometric characteristics of: curvatures c^{ψ}, headings ψ, and positions (φ, λ). The curvature difference δc^{ψ} can be measured from the switch start until the end of the switch arc. The heading difference $\delta\psi$ is zero at the switch start and increases until the end of the switch arc. The cross distance δ_{CT} increases also from zero at the switch start. Figure 3.2 shows the ideal difference signals of δc^{ψ}, $\delta\psi$, and δ_{CT} between the two tracks of different switch ways. This parallel track scenario includes a second arc that is traveled after the switch. Within

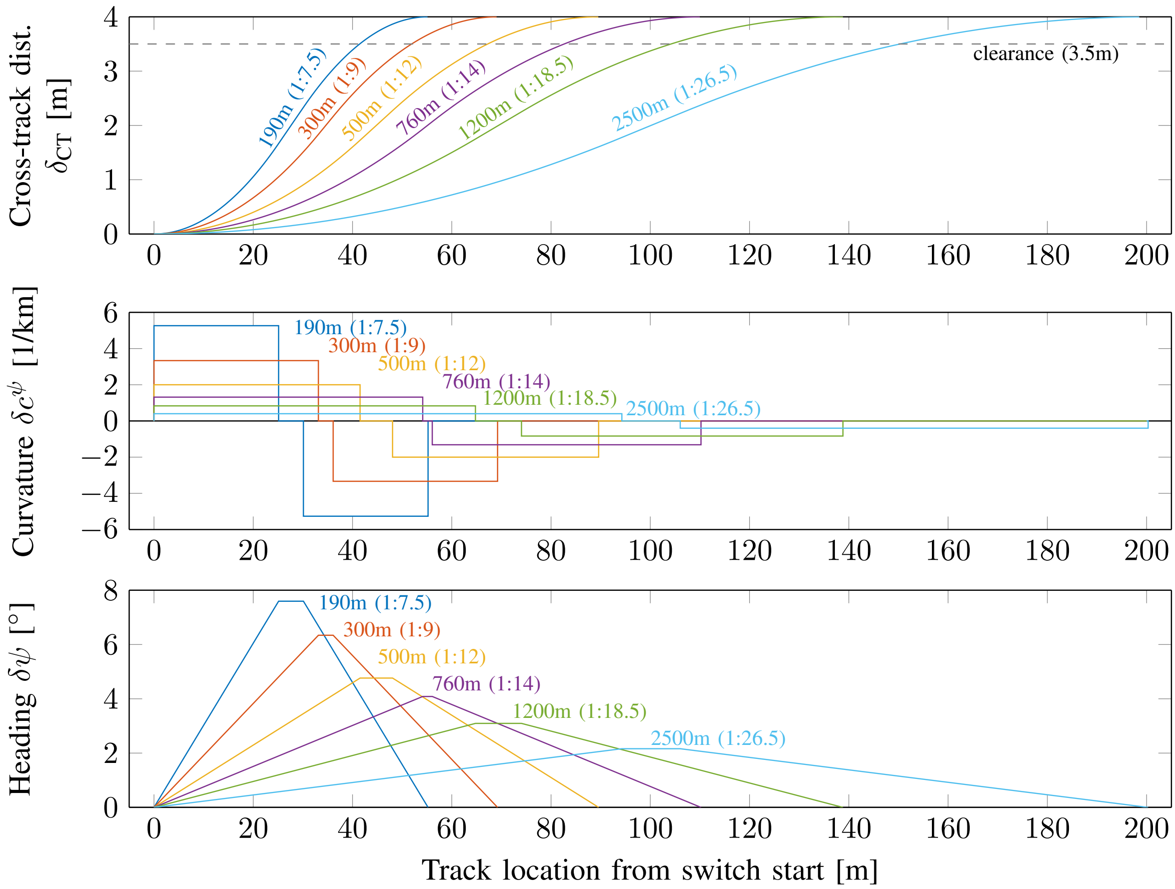


Figure 3.2: Ideal difference signals for a switch-way discrimination of representative switches in a parallel track scenario.

the second arc, the heading difference decreases to zero and a curvature difference can be observed a second time. In this parallel track scenario, the cross-track distance is the only remaining value to identify the tracks after the second arc has passed.

3.2 Map based Navigation in the Railway Domain

3.2.1 Geometric Coordinate Frames

The geometric coordinate frames for railways comprise: sensor frames, train frame, track frame with local navigation frame as well as Earth frame and inertial frame. Sensors measure in their specific *sensor frame*. For further processing, these measurements are converted in the *train frame* according to the sensor mounting. The map with the specific track geometry is expressed in the *track frame* which is linked to a local navigation frame. Frame rotation matrices are defined in the Appendix C.1. A train is constrained and guided by the rails of the track and thus the train frame is linked to the track frame.

Sensor Frame

Inertial sensors, heading sensors and also cameras or other feature detectors measure with respect to their sensor frame (index s). The measurements $\vec{z}^{\mathrm{s}}$ are rotated from sensor to train body frame (index b) with:

$$\vec{z}^{\mathrm{b}} = \mathbf{C}_{\mathrm{s}}^{\mathrm{b}} \vec{z}^{\mathrm{s}}, \tag{3.4}$$

and the direction cosine matrix (DCM) rotation $\mathbf{C}_{\mathrm{s}}^{\mathrm{b}}$ that is defined in Eq. (C.6). For a fixed mounted sensor, this rotation is persistent and can be calibrated and adjusted in advance. The inverse transformation from train body frame to sensor frame is:

$$\mathbf{C}_{\mathrm{b}}^{\mathrm{s}} = \mathbf{C}_{\mathrm{s}}^{\mathrm{b}T}. \tag{3.5}$$

Train Frame

The train frame is the vehicle body frame and defined by a right-handed Cartesian coordinate system. Train frame and body frame are the same in the following. The train axes are specified to the front in along direction (x), to the right side (y) and towards the bottom side (z). A complete train requires different frames for each train car, each bogie and wheel axle. In the following context, the train frame is used only for the cabin of the first car with the mounted onboard sensors. For the train speed, there is only one degree of freedom in the longitudinal x-axis of the body frame. The train speed is defined with: $v_{eb}^{b,x}$ and denoted as train speed v in the following. This notation describes the speed v in the x-axis of the body frame (superscript b, x), with the speed defined between Earth frame e and body frame b (subscript: eb). In the following, a train speed will be also defined in the track frame. The train motion acceleration is mainly caused from the train traction and brakes. This acceleration is denoted with a_{tb} and is also defined in the x-axis of the body frame, between Earth frame and body frame: $a_{\mathrm{tb},eb}^{b,x}$. This train acceleration is also the time derivative of the train speed v. The train attitude angles are defined in the body frame b by right-handed Euler angles between the local navigation frame n with north-east-down (NED) coordinates and the body frame b with the subscript: nb. The train body angles around the x-axis is roll (ϕ_{nb}^{b}), around the y-axis is pitch (θ^b_{nb}) and the orientation to north is yaw (ψ^b_{nb}). The indexes are omitted in the following and the turn rates $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ are defined similar to the angles. The DCM $\mathbf{C}^{\mathrm{b}}_{\mathrm{n}}$ from navigation frame n to train frame b is defined in Eq. (C.6) with the angles of roll, pitch and yaw.

Track Frame

The general track frame is defined on the track at the center between the two rails as depicted in Figure 3.3. The geometric coordinate frame is defined in along-track, cross-track and down direction with the angles of heading, slope and bank. However, this definition is ambiguous, as a track frame can be defined in two different directions with opposite angles and axes directions. Therefore, the track frame is defined within the map with a fixed direction: Each track has a direction definition from its start ($s = 0\,\mathrm{m}$) to its end point with an increasing 1-D track location s. The track start is defined in an arbitrary way at one ending of a track. In the following, a track frame is considered with this fixed definition. The geometric track axes are along-track in the definition direction, cross-track to the right side and downwards as presented in Fig. 3.3. The track frame is defined at any 1-D track location s on the track as a Cartesian coordinate system with index t. The track frame origin at a 1-D track location s can be related to an absolute geographic position. The track attitude is defined in the navigation frame (superscript n) by right-handed Euler angles between the navigation frame n and the track frame t with the subscript nt. Bank (ϕ^n_{nt}) is the angle around the along-track axis, slope (θ^n_{nt}) around cross-track and heading is the angle to north (ψ^n_{nt}). According to the angle definitions, a positive slope angle is defined as uphill, a positive bank angle is leaning to the right and a positive heading angle is defined clockwise from north. Curvature signs are defined equally and a positive heading curvature c^ψ is

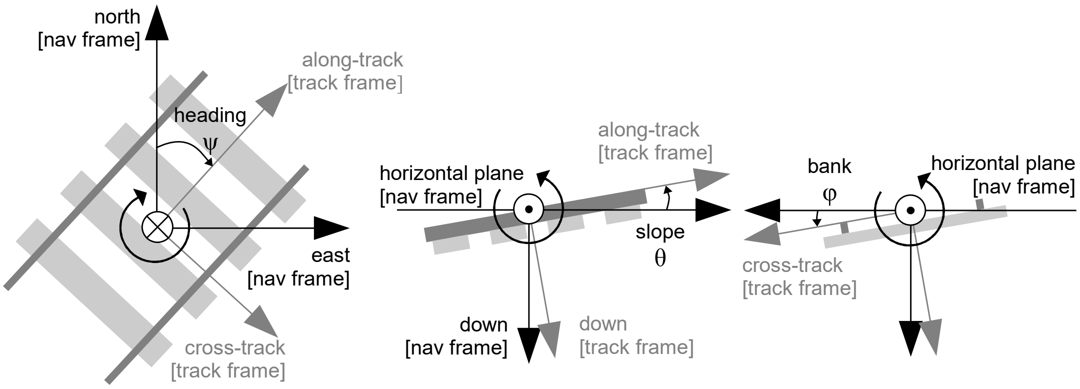


Figure 3.3: Track frame angle and axes definitions. [Left] top view, [Middle] side view, [Right] front view of a track.

a curve to the right, a negative is a curve to the left. The DCM $\mathbf{C}_\mathrm{n}^\mathrm{t}$ from navigation frame n to track frame t is defined in Eq. (C.6) with the angles of bank, slope and heading.

Earth Frame and Inertial Frame

The Earth frame and the inertial frame have the same origin at the center of the Earth. The inertial frame is does not rotate and the inertial measurements are referenced to this frame. The Earth frame is fixed to the rotating Earth. There are two different coordinate systems used for the Earth frame: The Cartesian Earth centered, Earth fixed (ECEF) coordinate system and the latitude, longitude, height (LLH) coordinate system based on an ellipsoid for the approximation of the Earth shape. In this thesis, the world geodetic system of 1984 (WGS84) definition is used for the reference ellipsoid.

Transformations between Earth frame e and navigation frame n are used for train localization, mapping, analysis and evaluation. A position $\vec{p}$ is represented in LLH coordinates with latitude angle φ, longitude angle λ, and a metric height h: $\vec{p} = \begin{pmatrix}\varphi & \lambda & h\end{pmatrix}^T$. Between the two LLH positions $\vec{p_1}$ and $\vec{p_2}$, a distance $\vec{x}$ is defined in metric north-east-down (NED) coordinates: $\vec{x} = \begin{pmatrix}x_\mathrm{n} & x_\mathrm{e} & x_\mathrm{d}\end{pmatrix}^T$.

There are two geodetic problems for the calculations with positions and distances [32]: The direct geodetic problem is defined to determine a position $\vec{p_2}$ from a given position $\vec{p_1}$ and a distance. The inverse geodetic problem is defined to determine a distance between the two positions $\vec{p_1}$ and $\vec{p_2}$.

According to the direct geodetic problem, the LLH position $\vec{p_2}$ can be computed from $\vec{p_1}$ and $\vec{x}$ with:

$$\vec{p_2} = \mathrm{f}_\mathrm{LLH}\left(\mathrm{f}_\mathrm{ECEF}(\vec{p_1}) + \mathbf{C}_\mathrm{n}^\mathrm{e}(\varphi_1, \lambda_1) \cdot \vec{x}\right). \tag{3.6}$$

The transformation from LLH to ECEF coordinates is f_ECEF and the transformation from ECEF to LLH is f_LLH. These transformations are non-linear and solutions can be found in [55]. The rotation matrix $\mathbf{C}_\mathrm{n}^\mathrm{e}(\varphi, \lambda)$ is defined in Eq. (C.9) at a position (φ, λ) and rotates the axes from navigation frame n to ECEF Earth frame e. An alternative transformation uses the arc measure with meridian and transverse Earth radii for the computation of the direct geodetic problem:

$$\vec{p_2} = \vec{p_1} + \begin{pmatrix} \frac{x_n}{(R_n(\varphi_1)+h_1)} \\ \frac{x_e}{(R_e(\varphi_1)+h_1)\cos\varphi_1} \\ -x_d \end{pmatrix}. \tag{3.7}$$

This transformation uses the meridian radius in north direction $R_n(\varphi)$ and the transverse Earth radius in east direction $R_e(\varphi)$ as defined in Appendix Eq. (C.10) and Eq. (C.11). The radii depend on the latitude and change for distances in north-south direction. In this thesis, Eq. (3.7) is used for the propagation of a position with inertial measurements and for calculations with small distances in the vicinity of tracks (e.g. <100 m).

According to the inverse geodetic problem, the NED distance $\vec{x}$ is computed from the two positions $\vec{p_1}$ and $\vec{p_2}$ and the rotation matrix Eq. (C.9):

$$\vec{x} = \mathbf{C}_{\mathrm{e}}^{\mathrm{n}}(\varphi_1, \lambda_1)\left(\mathrm{f}_{\mathrm{ECEF}}(\vec{p_2}) - \mathrm{f}_{\mathrm{ECEF}}(\vec{p_1})\right). \tag{3.8}$$

The alternative calculation of the distance from positions for small distances is based on the arc measure with meridian and transverse Earth radii:

$$\begin{aligned}\vec{x} &= \begin{pmatrix}(R_n(\varphi_1)+h_1) & 0 & 0\\ 0 & (R_e(\varphi_1)+h_1)\cdot\cos\varphi_1 & 0\\ 0 & 0 & -1\end{pmatrix}(\vec{p_2}-\vec{p_1})\\ &= \begin{pmatrix}(\varphi_2-\varphi_1)\cdot(R_n(\varphi_1)+h_1)\\ (\lambda_2-\lambda_1)\cdot(R_e(\varphi_1)+h_1)\cdot\cos\varphi_1\\ -h_2+h_1\end{pmatrix}.\end{aligned} \tag{3.9}$$

Along-track and Cross-track Distance Definitions

The coordinate system in track frame is defined with along-track and cross-track direction. Figure 3.4 shows the different distances between a geographic position $\vec{p}_{\hat{s}}$ on a track at the 1-D track location $\hat{s}$ and a geographic position $\vec{p}$ next to the tracks. The along-track distance d_{AT} is the length of the tangent vector of the track at $\hat{s}$. The cross-track distance d_{CT} is the length of the normal vector to $\vec{p}$. The cross-track distance δ_{CT} is defined with the length of the normal vector between matched geographic track position $\vec{p}_{\tilde{s}}$ at track location $\tilde{s}$ and $\vec{p}$. The along-track distance δ_{AT}, also denoted with Δs, follows the path of the track and is defined between track location $\hat{s}$ and the matched track location $\tilde{s}$. The track location at $\hat{s}$ can be an estimate, a prediction or a true location in the following sections of localization, mapping and evaluation. For straight tracks, the distance δ_{AT} is the same as d_{AT}. If the distance δ_{AT} is small compared to the curve radius r for curved tracks, the distances δ_{AT} and d_{AT} are very similar:

$$\left.\begin{aligned} d_{\mathrm{AT}} &\approx \delta_{\mathrm{AT}},\\ d_{\mathrm{CT}} &\approx \delta_{\mathrm{CT}},\end{aligned}\right\}\text{for curve radius } r \gg \delta_{\mathrm{AT}}. \tag{3.10}$$

This approximation will be used in one following localization method (Section 6.2.4). The computation of $\tilde{s}$ from the geographic position $\vec{p}$ and a map with track segments is also presented in the map-match approach (Section 6.2.2).

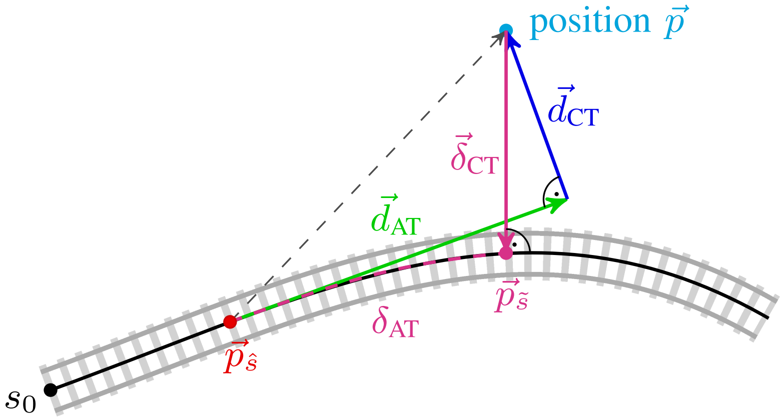


Figure 3.4: Along-track and cross-track distance definitions.

3.2.2 Trains on Tracks

Train-to-track Direction

A train can be placed in two different directions on a track as shown in Figure 3.5. This direction is called train-to-track frame direction and denoted with:

$$o^{\text{t}} \in \{-1, +1\}. \tag{3.11}$$

The map defines the track origin ($s = 0$) at one side of the track and also the direction of the along-track axis in Figure 3.5.

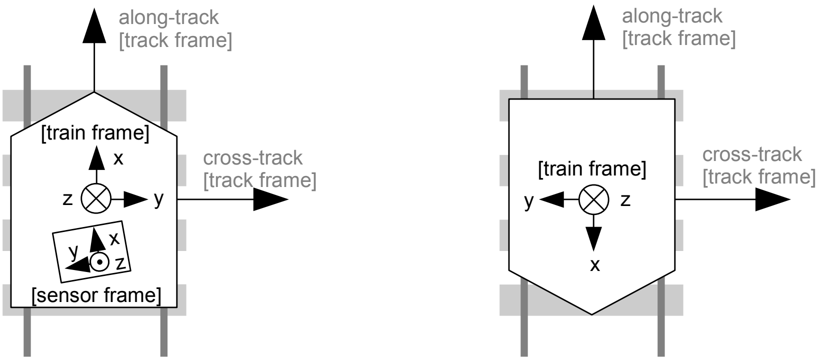


Figure 3.5: [Left] Train with sensor frame in positive train-to-track direction (o^{t} : +). [Right] Train direction in negative train-to-track direction (o^{t} : −).

The rotation from train (body) frame b to track frame t can take two discrete states: both frames are either aligned in the same direction of the tracks or the train frame is rotated in the opposite direction. The rotation matrix $\mathbf{C}(o^{\mathrm{t}})$ depends on the train-to-track direction o^{t} and rotates the train frame or track frame. An opposite train-to-track direction is a rotation of one frame about the z axis with $\psi = \pi$ or $\psi = 180°$, respectively. The train-to-track rotation matrix uses the yaw rotation matrix, as defined in Eq. (C.1), and the train-to-track direction o^{t}:

$$\mathbf{C}(o^{\mathrm{t}}) = \begin{pmatrix} o^{\mathrm{t}} & 0 & 0 \\ 0 & o^{\mathrm{t}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.12}$$

If the train-to-track direction o^{t} is positive, both frames are aligned and $\mathbf{C}(o^{\mathrm{t}})$ is an identity matrix. If the train-to-track direction is negative, the train-to-track rotation matrix is:

$$\mathbf{C}(o^{\mathrm{t}} = -) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.13}$$

For both cases of the train-to-track direction o^{t}, the rotation matrix from track frame t to train frame b is the same as the rotation from train to track frame:

$$\mathbf{C}_{\mathrm{t}}^{\mathrm{b}}(o^{\mathrm{t}}) = \mathbf{C}_{\mathrm{b}}^{\mathrm{t}}(o^{\mathrm{t}}). \tag{3.14}$$

The train speed v is alternatively expressed between train and track frame with $\dot{s}$. This train speed stands for the temporal change of the location s. The absolute velocity of v and $\dot{s}$ are the same but the sign depends now on the definition for the track origin and the motion direction of the train. The sign of $\dot{s}$ indicates either an increasing (+) or decreasing (−) change of the location s of the current track:

$$\dot{s} = o^{\mathrm{t}}\, v : \begin{cases} v, & \text{for } o^{\mathrm{t}} = + \\ -v, & \text{for } o^{\mathrm{t}} = - \end{cases}. \tag{3.15}$$

The acceleration of the train can be expressed in the track frame with $\ddot{s}$ and the conversion is:

$$\ddot{s} = o^{\mathrm{t}}\, a_{\mathrm{tb}}. \tag{3.16}$$

Table 3.2 shows the conversions of the train direction variables: the train-to-track frame direction (o^{t}), the train motion direction by the sign of v (o^{v}) and the direction of motion between train and track frame by the sign of $\dot{s}$ (o^{m}). The motion direction depends on a forward or backward velocity v. Most of the trains are designed for two-way directional operations. The motion direction remains the same during a train run. The direction can change once the train has stopped. As defined in the paragraph about track frame, the orientation of the track frame is fixed and defined in the map. It is possible that two connected tracks in a map, e.g. at a switch, have opposite orientation definitions. This causes an alternating sign of $\dot{s}$ and o^{t} after a change of tracks during a train run. This means, o^{v} is persistent during one train run, while o^{t} and o^{m} can change. The motion direction in track-frame (o^{m}) changes also with the velocity direction. Direction o^{t} remains its value during stand-still and is thus persistent while the other directions are undefined. Table 3.2 contains all possible combinations of the train directions in six different scenarios. With the knowledge of two directions o^{t}, the third can be derived from Table 3.2. These directions are only auxiliary variables and needed for conversions. The directions are tracked in a deterministic way for train localization. However, these directions may be unknown and ambiguous in the initial phase of train localization. They are usually observable soon after the first motion of the train.

train-track frame direction	**train velocity direction**	**train motion direction in track frame**
o^{t}	$o^{\mathrm{v}} = \mathrm{sgn}(v)$	$o^{\mathrm{m}} = \mathrm{sgn}(\dot{s})$
+	forward (+)	+
−	forward (+)	−
+	backward (−)	−
−	backward (−)	+
+	stop (0)	0
−	stop (0)	0

Table 3.2: Train directions for all six scenarios.

Track Geometry and Train Trajectory

For train localization and track mapping, a conversion is required in a common coordinate frame for the map data, the train state and the measurements. All data can either be converted into the *train frame* or in the *track frame*. In this thesis, the train frame is used for train localization as well as for the information fusion of the prior map and measurements in the RailSLAM method. Once the track map is updated, the data is converted to the track-frame.

As previously mentioned, the railway tracks are a hard constraint for the positions, motion and kinematics of a train. In principle, a train bogie adapts the position and attitude of the track with its wheels. The bogie is further linked to the train cabin over a pivot bearing and suspensions. The track attitude angles of bank, slope and heading influence directly the train angles by roll, pitch and yaw. In this thesis, the following set of physical variables are used for train-side measurements as well as for track features in the track map:

$$\{\underbrace{\varphi, \lambda, h}_{\text{position}}, \underbrace{\phi, \theta, \psi}_{\text{attitude}}, \underbrace{c^{\phi}, c^{\theta}, c^{\psi}}_{\text{curvatures}}\}. \tag{3.17}$$

This set of variables is called *train trajectory* in the context of train-side measurements. In the context of a track and the track map, this set of variables is called *track geometry*. As a simplification, the track geometry translates approximately to the train cabin trajectory values in the following. This may induce a small error, as the several parts of the bogie and the undercarriage contains suspension and the cabin is placed on two bogies.

The positions in Eq. (3.17) are not affected from a rotation of the track-to-track direction. Table 3.3 shows the attitude and the curvature conversions for the two track-to-train directions. The attitude angles change if the train direction o^{t} is negative. As mentioned, a curvature describes the change of the attitude angles over the track location parameter s. The bank curvature is independent of the direction definition o^{t}. This property can be explained with the thread of a screw: a thread has the same direction of rotation at both ends of the tread. The clock-wise rotation that is seen from one end of a right-handed thread is also seen as a clock-wise rotation from the other end. An arbitrary or changing definition of the track origin does not affect the direction of rotation in the x-axis and the bank curvature. The slope curvature is also independent of the direction definition o^{t}: the vertical curves of sag or crest do not change with a different definition of the track origin and train-to-track direction. The heading curvature changes its sign with an opposite train-to-track direction. A left curve with increasing track locations turns to a right curve, with respect to increasing track locations, if the track origin changes.

Train Frame	**Track Frame**	**Train-to-track Direction**	
train trajectory	track geometry	$o^{\mathrm{t}} = +$	$o^{\mathrm{t}} = -$
roll ϕ^b	bank ϕ^t	$\phi^b \approx \phi^t$	$\phi^b \approx -\phi^t$
pitch θ^b	slope θ^t	$\theta^b \approx \theta^t$	$\theta^b \approx -\theta^t$
yaw ψ^b	heading ψ^t	$\psi^b \approx \psi^t$	$\psi^b \approx \psi^t + \pi$
bank curvature		$c^{\phi,b} \approx c^{\phi,t}$	
slope curvature		$c^{\theta,b} \approx c^{\theta,t}$	
heading curvature		$c^{\psi,b} \approx c^{\psi,t}$	$c^{\psi,b} \approx -c^{\psi,t}$

Table 3.3: Conversion of attitude angles and curvatures between train and track frame.

3.2.3 Topological Coordinates

The goal of train localization is to estimate the train location in the track network by topological coordinates as well as the train speed. A unique and discrete track ID (id) identifies the track and the track length variable s is the one-dimensional location on that track. Each track is defined with an origin and a direction o^{t} that indicates if a train is oriented towards or against the track origin. The topological pose consists of the topological location and the direction:

$$T^{\mathrm{topo}} = \{\underbrace{id, s}_{\substack{\text{topological}\\ \text{location}}}, o^{\mathrm{t}}\}. \tag{3.18}$$

Tracks are connected by switches, crossings or diamond switch crossings. A track is defined in this thesis with a unique ID between two connections, i.e. a track contains no switch or crossing. This definition ensures, that a track is always one-dimensional with no other access than the track begin or track end. The topological location is a unique address within the railway track network.

Topological Localization Characterization

A suitable analysis for topological localization characterization is the distinction in along-track and track identification. Along-track addresses the continuous 1-D localization on a track. Track identification focuses on the discrete tracks IDs and track selectivity is the ability of a correct localization and identification of the track. Sensors can contribute to along-track and track identification with relative or absolute measurements of track features or train motion. As shown in Table 3.4 and Figure 3.6, measurements can contribute to train localization in four different ways. Measurements may also contribute to absolute along-track or track identification in a local vicinity. The GNSS position measurements are considered as absolute along-track measurements and also useful for the track identification under certain conditions: in scenarios with parallel tracks there is still an ambiguity due to the close tracks and the GNSS measurement accuracy.

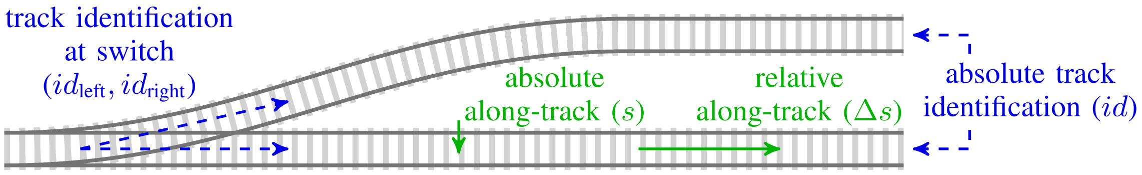


Figure 3.6: Topological characterization of localization.

Localization characterization	**Estimation, observation of**	**Contribution of measurements via**
relative along-track	Δs	velocity integration, displacement, distance measurement
absolute along-track	s	comparison with diverse along-track features, signatures, and landmarks
track identification at switch	id_{left} / id_{right}	comparison with competing switch-way features
absolute track identification	id	comparison with diverse track features, signatures, and landmarks

Table 3.4: Localization characterization with along-track and track identification.

3.2.4 Train Control

The train control, with train driver operation, consists of an along-track control from the train driver and a train route guidance from a train control center:

$$U = \{ \underbrace{U^{\text{sw}}}_{\substack{\text{route guidance,}\\ \text{train control center}}}, \underbrace{U^{\text{acc}}, U^{\text{m}}}_{\substack{\text{along-track control,}\\ \text{train driver}}} \}. \tag{3.19}$$

The route guidance influences the travel path of the train by the selected switch way ($U^{\mathrm{sw}} \in \{\text{straight, turn}\}$ or $\{\text{left, right}\}$). This is usually controlled by a train control center or sometimes by the train driver at shunting yards or industrial tracks. A complete train route consists of multiple and consecutive switch stands. The train driver controls the general train motion U^{m} with a travel direction selector and the acceleration U^{acc} with the traction and brake lever. The general train motion includes the travel direction as well as a stopped train at a railway signal or with activated parking brakes: ($U^{\mathrm{m}} \in \{\text{forward, backward, stop}\}$). The train control center has also influence on the along-track control via railway signaling.

3.3 Topological and Geometric Track Map

The track map contains relevant information for the train localization. This map contains topological information by the track connections as well as the geometric information with the track features. Figure 3.7 shows the two layers of this special map with the topological network part and the linked geometric part. The track map is an abstract model of the railway environment with limited accuracy and limited completeness. An ideal map assumes no errors and exact data. A probabilistic track map contains additionally values for the uncertainty. This uncertainty can be represented as a deviation, e.g. a Gaussian variance or covariance for the geometry values. A probabilistic track map can also contain weighted and multiple versions of one track.

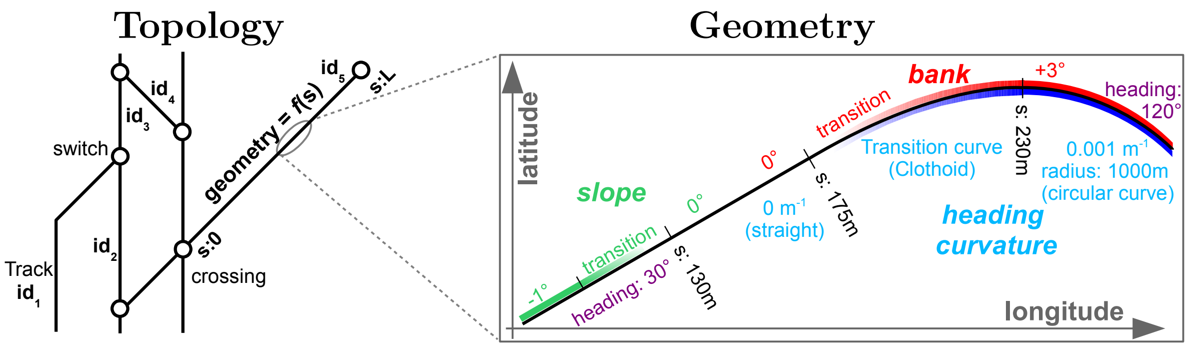


Figure 3.7: Railway map with topological and geometric data (see [P4]).

3.3.1 Railway Track Features

Track features are characteristics of the track and continuously present over position. Discrete features occur only at a particular position and are called landmarks. Examples for landmarks are switches, significant changes of the geometry as start and end of curves, bridges, a tunnel entrance or railway signals. A suitable track feature or landmark is a location dependent and unchanging property of the track which can be measured. A sequence of track features over location is called track signature in this thesis. In the context of landmarks, a signature is a set of signals or properties which are needed for the detection and identification of the landmark (see [13]). Over different locations, there are unique track features as well as repeated and ambiguous features.

In this thesis, the track geometry consists of geographic position, attitude and curvatures. The geographic positions describe the center line of a railway track. It consists of latitude, longitude and height (φ, λ, h) is a unique feature in combination as a position. The track attitude contains bank ϕ, slope θ, heading ψ, and the track curvatures are bank curvature c^{ϕ}, slope curvature c^{θ}, and heading curvature c^{ψ}. Attitude and curvature are not unique features and can be locally constant over s or repeated on other locations. Additional features can be extended, provided that there is a different signal over different locations and sensors can measure these features. Examples of extended features are the vibratory signatures [P5], magnetic signatures [P6], metal objects for eddy current sensor and visual features for cameras.

3.3.2 Model of the Track Map

The track map model stores the map information of topology and track features and provides interfaces to access this information. An example of a data model and exchange format for a track map is the *Railway Markup Language railML* [56]. The track map model defined here is confined to the needs of train localization and defines also interface functions. Additional information of speed limits, track load, electrification and other railway properties can extend this map.

The track map is organized with a set of the different tracks. Every track is identified with a unique track ID *id* and defined with one start and one end point. A track is always one-dimensional with no other access than the start or track end. The geometric data of each track is parametrized to the one-dimensional location and stored by samples over the location s. A continuous representation of track features is achieved by interpolations between these points. Figure 3.8 shows an example of three track features with standard deviations. Each track feature shows nine sample points and an interpolation in between.

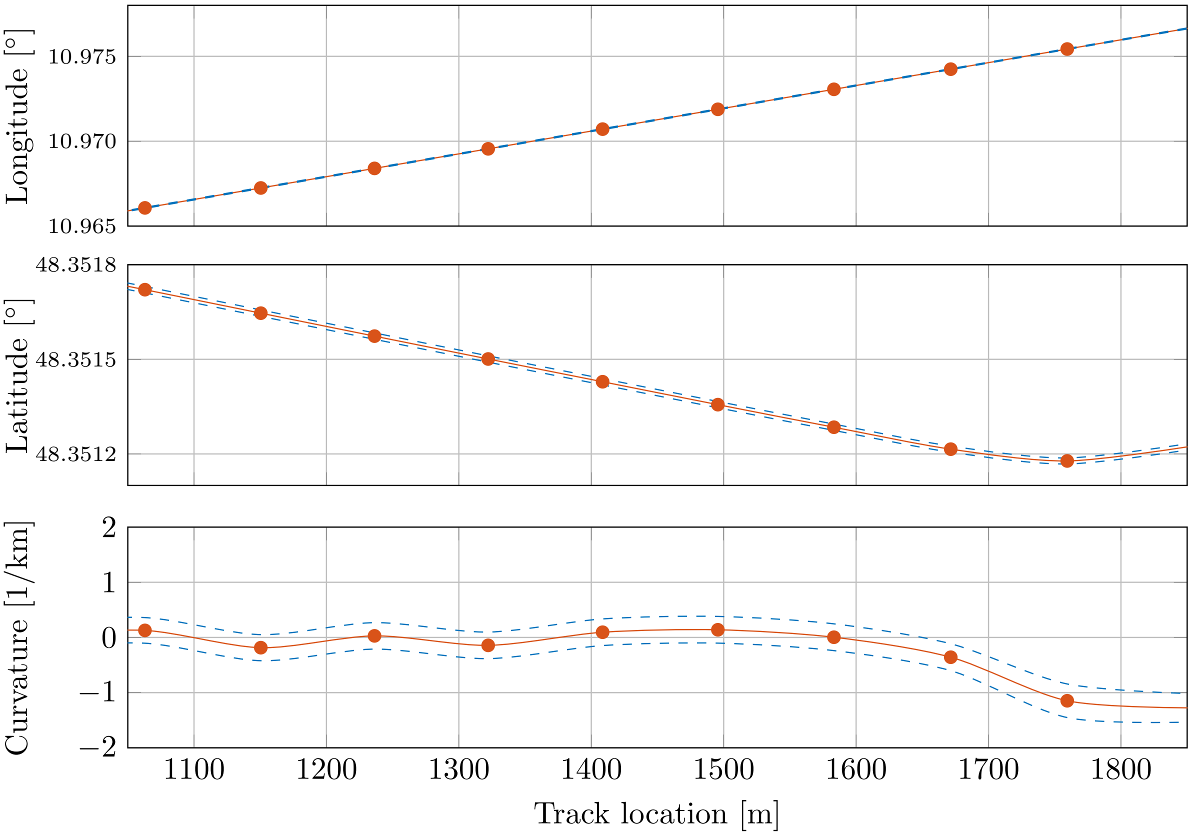
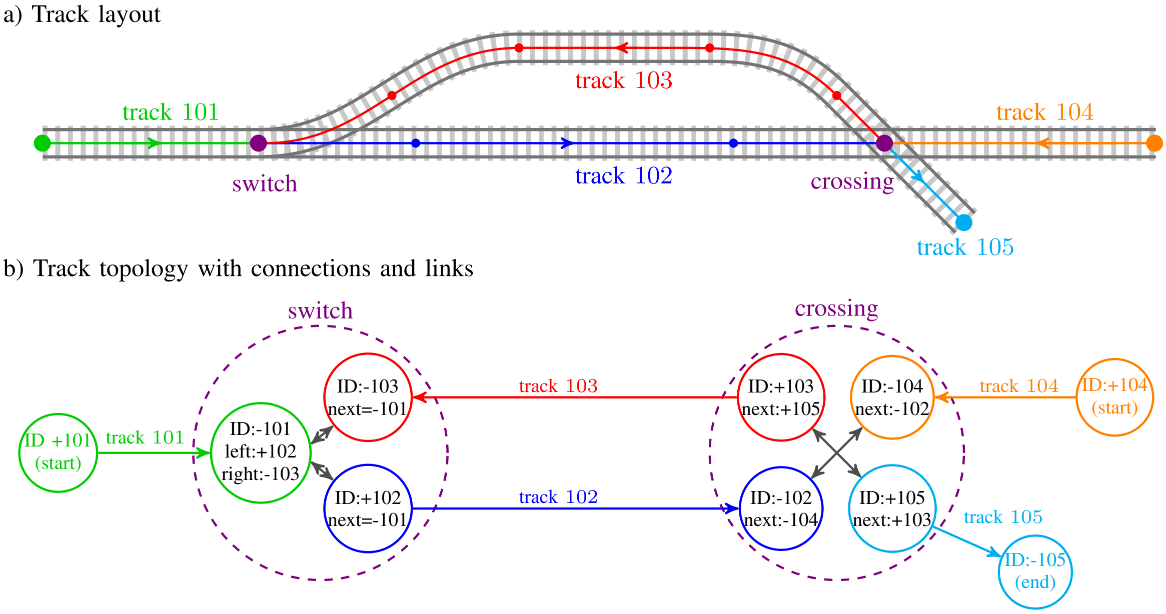


Figure 3.8: Parametrized track features over 1-D track locations: interpolated samples (red) of latitude, longitude and heading curvature with standard deviations (blue).

The track map model provides interface functions for the applications of train localization and mapping. One important interface function is a transition of a track location (topological location) with a defined distance, called transition interface function. Another interface functions accesses the track features, that are sampled over the 1-D track location. An interpolation enables a continuous access of the track feature and track geometry data. Possible data interpolation methods can be linear interpolation (C^0 continuity), cubic interpolation (C^1) or cubic spline interpolation (C^2) or similar. For a comparison of data from the track map and measurements, the track feature vector is converted to the train frame, as described in Table 3.3.

Figure 3.9 visualizes a small railway network of tracks with one switch and one crossing. The topological connections are shown with the linked IDs to the other connections. Each connection has an own ID consisting of the track ID and a sign. A positive sign refers a track start, a negative refers to a track end. In Figure 3.9, the crossing displays only the accessible IDs while the traverse link IDs are not shown. Switches and crossings are nodes of the topology and a switch connects three tracks, a crossing four tracks. The topological information is stored in the connections of the track endings, respectively the track start point and track end point. A connection links another connection with its ID while the other connection references also the first connection ID. There are six types of connections considered: buffer stop, open end, single connection, switch merge, switch branch and crossing. The endings of *buffer stop* and *open end* terminate the track or leave it open to an unknown track. The *single connection* connects only one track to another one. The *switch merge* connects one track ending with a track ending of the type switch. There is no access from a merging track to the other merging track of a switch. The *switch* connects to two other track connections and both tracks are accessible with a train. The *crossing* points to one accessible connection and the two traverse tracks are inaccessible. In this way, the track map limits the access of certain tracks, which is a vital information for the track map transition function (see Eq. (6.5)).


a) Track layout

b) Track topology with connections and links

Figure 3.9: Map with linked connections of a switch and crossing.

3.3.3 Errors of the Track Map

There are several types of map errors: incomplete data, incorrect data and inaccurate data between the map data and the truth. A track map may usually focus on a region or a specific railway line. Missing tracks that are out of the region are not considered as a map error. If a train runs on a track that is not part of the track map, this map is insufficient.

Some of the map errors can be detected with checks and the map data itself. Examples for incomplete data are missing track connections, missing connection references or missing track feature data. Missing references can be detected from rules, as a switch requires two connections with references to other tracks, for example. A missing track feature can be detected if other track features are present. Examples of incorrect data that can be detected are inconsistent connections, inconsistent cross references, or a non-continuity of positions at a track change. The track geometry data can be further checked against reasonable limits or limits for changes.

The geometry data is a representation of the true tracks with a specific accuracy. Hence, this data deviates from the true track geometry with a certain error. In the following, different maps are evaluated with a cross-track map bias $\delta_{\mathrm{CT,map}}$. This bias in lateral direction is the systematic offset between a position on the true tracks and the corresponding position in the map model.

3.3.4 Data Sources of the Track Map

In this thesis, the RailSLAM method estimated the track map from measurements. As an alternative, the track map can be also generated from existing geo-position data. This data may originate from public or private geo databases, from land surveys or from sampled positions of geo-referenced aerial images.

In the following, Open Street Map (OSM) [57] is used as representative example of an existing geo database. OSM is an open source database from crowd sourced data and contains 2-D position samples of many railway tracks. The major advantage is the good coverage of railway tracks in certain areas. In contrast, a train measurements contain only those tracks and switches that are part of the train runs. These covered tracks are only a fraction of the track network. Fig. 3.10 shows the comprehensive tracks of Augsburg main station from OSM position data with interpolations between the track sample positions. The disadvantage of the OSM based map is the coarse accuracy of the position samples in some parts of the environment, an incompleteness in some areas, and the lack of other geometry data than 2-D positions.

Geo-referenced aerial images, also known as digital orthophotos (DOP), are created by land surveying offices from surveying flights or from satellite images. This type of image is also used by a virtual globe software, such as Google Earth [58] or NASA WorldWind [59]. A geo-referenced image has geo position information associated to the pixel positions of the image. Railways tracks can be identified inside the image and sampled either manually or with specific algorithms. The geo positions are obtained from these

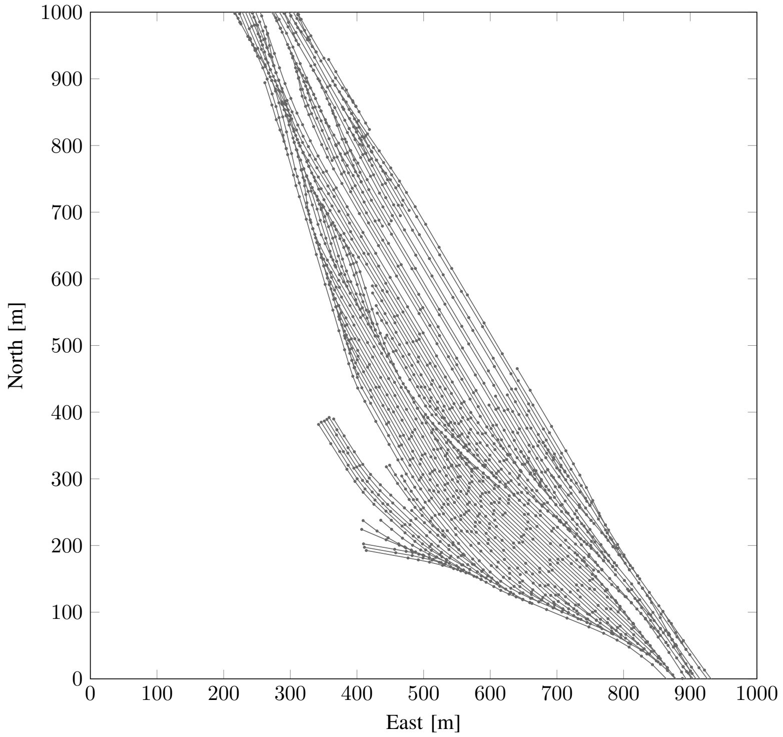


Figure 3.10: Railway track positions of Augsburg main station from Open Street Map (OSM) data.

samples, as the geo positions are linked to the image. Depending on the quality of the images, there is a limited accuracy of the positions with offsets and image distortions. Figure 3.11 shows a geo-referenced DOP from the local land surveying office [60]. The reference track is sampled from this image and shown with white points and and interpolation in between. The track from OSM position data is shown in magenta. In this example, the cross-track map bias $\delta_{\mathrm{CT,map}}$ between the OSM based track map and the reference track is between 1.7 m and 2.4 m over a length of 1.7 km.

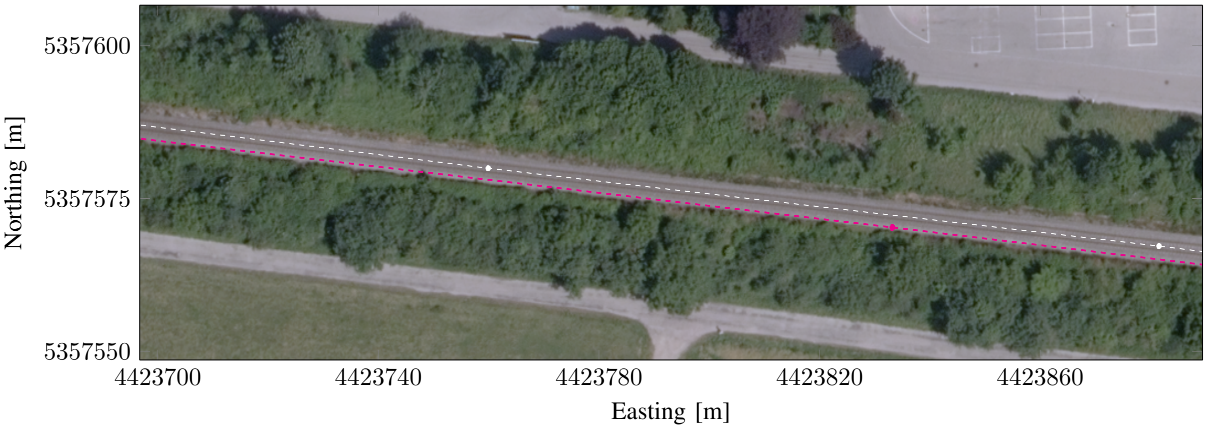


Figure 3.11: Geo-referenced digital orthophoto with a reference track (white) and OSM based track (magenta).

The raw position samples of railway tracks are not sufficient for a track map, because the topological layer is missing. Therefore, a track map requires further processing of the position samples. At first, the positions are grouped to tracks and labeled with an unique track ID *id*. For every track, the 1-D location s is computed from the Euclidean distances between these position samples. An origin is defined for each track at an arbitrary ending. The last step is a connection of track connections. The track geometry information may be extended with heading angle and heading curvature, which can be calculated from the 2-D position samples: The heading may be computed with the metric distances in north (Δx_n) and east (Δx_e) direction from two consecutive position samples:

$$\psi = \arctan\left(\frac{\Delta x_n}{\Delta x_e}\right). \tag{3.20}$$

The curvature is the change of heading of two consecutive headings over the arc length Δs of the track between the samples:

$$c^{\psi} = \frac{\Delta\psi}{\Delta s}. \tag{3.21}$$

3.3.5 Memory Requirements of the Track Map

The memory requirements are assessed for a track map for train localization. The plain data volume is a rough estimate of uncompressed map information. Each value of the geometric data is considered with a double precision floating point representation of 8 data bytes. The topology requires for each track integers for the track ID (4 B), connection types of start and end (1 B) and linked IDs. These linked IDs point to other connections with a track ID and a start and end connection identifier (sign) with a signed integer (4 B). There are no linked IDs for a termination (buffer stop or open end), one for single connections and merging switch tracks, two for switches and three for crossings.

A representative track map is analyzed for memory consumption with different formats. This track map covers all tracks of and near the operating area of the regional transportation cooperation BRB in Augsburg (Germany) in 2014. The track map contains 1563 tracks over a total length of 568 km with 14886 track samples, 915 switches, 124 crossings, 2087 endings and an average segment length of 42.6 m. Every track sample contains four variables by 1-D track location, 2-D geographic position and curvature data. For the given example, the uncompressed map information is 465 kB for the track features and 16.2 kB for the topology. Table 3.5 contains the estimated data volumes of all German tracks over 63 839 km [61] with different sample spacings and number of track features. The large map of German railways is 112 times larger than the example map. The topology data volume of 1.8 MB is estimated from the example map and scaled up.

Track features	**Average sample spacing**			
	5 m	10 m	20 m	50 m
4 values $(s, \varphi, \lambda, c^{\psi})$	391 MB	197 MB	99 MB	41 MB
10 values $(s, \varphi, \lambda, h, \phi, \theta, \psi, c^{\phi}, c^{\theta}, c^{\psi})$	976 MB	489 MB	245 MB	99 MB
10 values + variances	1950 MB	976 MB	489 MB	197 MB

Table 3.5: Estimated and raw data volumes for a complete track map of Germany (63 839 km [61])

The railway networks of 27 European countries sum up to 330 892 km [61] and are more than five times larger than the German network. The raw data volume of the topology is roughly 9.2 MB. The raw memory requirements for the very large EU27 railway map with sample points of 5 m and ten variables with variances are about 10 GB.

A data reduction and state-of-the-art compression can further decrease the data volume of a track map. Data reduction can be achieved with a sample point reduction on straight tracks, for example. This is possible if sample points are redundant and can be interpolated from surrounding samples within a defined tolerance. Splines are advantageous to approximate curves and reduce further sample points. Hasberg [62] has shown a spline approximation for railways tracks with 2-D position data.

This memory analysis shows, that even uncompressed and very large maps are manageable with off-the-shelf storage devices.

Chapter 4

Onboard Sensors

This chapter presents the sensors that are used for an onboard train localization and track mapping. The algorithmic implementations in Chapter 6 use a GNSS and an IMU. Section 4.1 presents the GNSS with an overview, theoretic accuracy and a train measurement analysis. Section 4.2 presents the IMU with an overview, a state-of-the art inertial navigation system method, an analysis of train measurements, and an analysis of a direct track feature measurement method with inertial sensors. Furthermore, two novel measurement methods are presented with the passive magnetic measurement method in Section 4.3.1 and the vibration method in Section 4.3.2. A summary of possible onboard train measurements is given in Section 4.3.3.

4.1 GNSS: Global Navigation Satellite System

A global navigation satellite system (GNSS) enables a user to determine world-wide its absolute geographic position, velocity and time. The first and most widely used GNSS is the NAVSTAR-GPS (USA). Other systems are the Russian GLONASS system, the European GALILEO system and the Chinese BeiDou (Compass) system. A GNSS has three major system components: A space segment with the satellite constellation, a ground segment and the GNSS receiver as user segment. Figure 4.1 shows a simplified GNSS receiver block diagram with a focus on the data interfaces.

A simple layout of a GNSS receiver consist of several elements, such as a radio frequency (RF) front end with connected antenna, a signal processing unit and a navigation computer. Further explanation of signal processing and position computation can be found in [63, 64]. The three

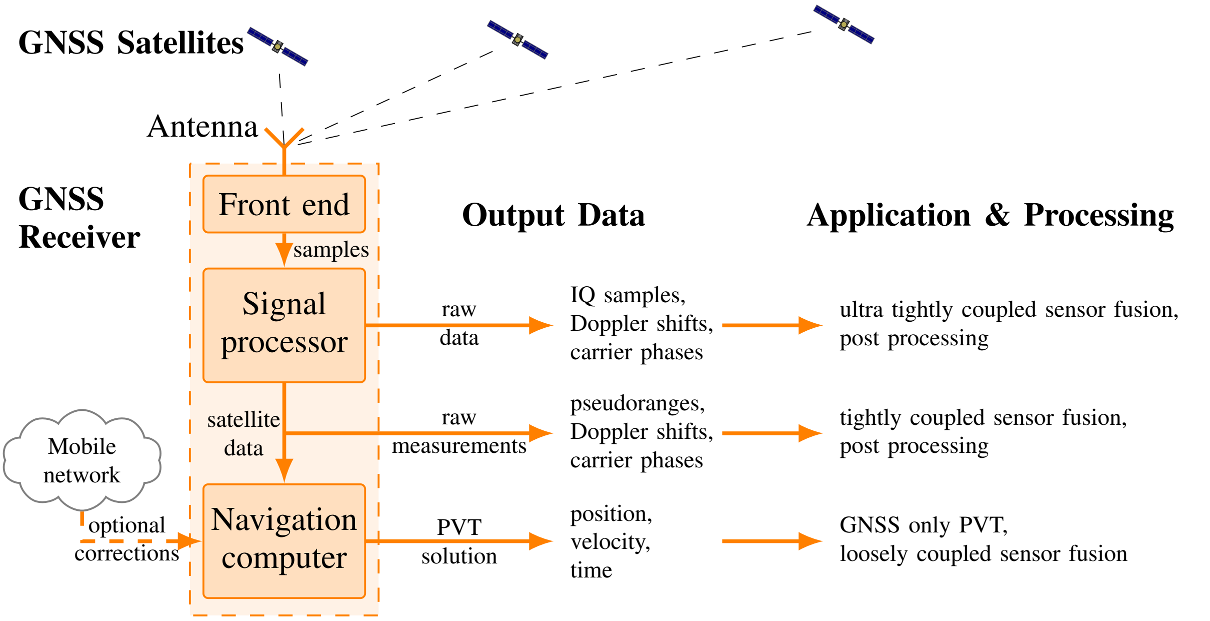


Figure 4.1: Simplified GNSS receiver diagram with different data outputs.

data interfaces, as depicted in Fig. 4.1, represent different stages of the data processing. The GNSS data can be processed in combination with additional external information. This external data can be measurements from an inertial measurement unit (IMU), a barometer, speed sensors or a map of the environment. The coupling level denotes how tight this external data is coupled to the GNSS data. A post-processing considers a computation of the PVT data from recorded raw data or raw measurements. This post-processing comprises for example non-causal signal filters and additional information, such as precise satellite orbit information or corrections from nearby reference stations. First, the position, velocity, time (PVT) solution is the standard output that is available on every commercial receiver. This PVT data is based on GNSS measurements and optional correction data. A further information fusion with other measurements will be denoted as loosely coupled positioning. Second, a tightly coupled positioning requires raw measurements that consist of pseudoranges, Doppler and phase measurements of each satellite. Third, an ultra tight coupled positioning uses inphase and quadrature (IQ) samples that enables an enhanced tracking of the satellite signals with additional information from other sensors.

The scope of this thesis considers a standard COTS receiver with a single antenna and a PVT solution as output for a loosely-coupled integration. Advanced GNSS approaches are briefly presented with suggestions for improvements in Section 8.1.5. The following presents the theoretical accuracies of GNSS positions with a measurement error model (Section 4.1.1) and ideal measurements (Section 4.1.2). A useability analysis for train localization is given in Section 4.1.3 with an evaluation of measurements recorded on a train in a railway environment.

4.1.1 Measurement Errors and Corrections

The navigation computer processes measurements of several satellites. A pseudorange is a range measurement from satellite s to the receiver u. A standard measurement model for the pseudorange ρ is given by [64]:

$$\rho_u^s = \underbrace{||\vec{r}^s - \vec{r}_u||}_{\substack{\text{range between} \\ \text{satellite and user}}} - \underbrace{c \cdot \delta^s}_{\substack{\text{satellite} \\ \text{clock offset}}} + \underbrace{c \cdot \delta_u}_{\substack{\text{user clock} \\ \text{offset}}} + \underbrace{I_u^s + T_u^s}_{\substack{\text{Ionospheric and} \\ \text{Tropospheric delay}}} + \underbrace{\epsilon_\rho}_{\substack{\text{remanent} \\ \text{errors}}} \quad . \tag{4.1}$$

The navigation computer estimates the user position $\vec{r}_u$ and the user clock offset δ_u from multiple satellite measurements. As shown in Eq. (4.1), there are several error sources which decrease the position accuracy. Some of them can be corrected online at the receiver. Errors of the ephemeris data cause errors in satellite position $\vec{r}^s$ and therefore in the range estimate. An error from the satellite clock causes a bias in the range measurement. The atmospheric effects of ionosphere and troposphere cause biases on the measurements. The parameters for pseudorange corrections of ephemeris, satellite clock and atmospheric errors for each satellite are transmitted by several ways: First, the satellite navigation message of each satellite contains coarse parameters with a global scope. More accurate parameters as well as integrity information for an increased reliability are transmitted by satellite based augmentation system (SBAS) service with a continental scope. More accurate corrections with a local scope can be achieved with differential GNSS (DGNSS), that requires a mobile network and reference stations with known positions (see [63, 64]).

Multipath is another error source: the satellite signal is reflected in the environment near the receiver and part of the remanent errors ϵ_ρ. The reflected signal path is different to the direct line-of-sight path and affects the range and Doppler measurements.

4.1.2 Theoretical GNSS Accuracy

This theoretic analysis shows the challenge of track-selective accuracy with GNSS position measurements. Therefore, the horizontal GNSS accuracy is compared to typical distances of parallel railway tracks. A parallel track distance depends mainly on the railway gauge, route types by main or secondary line and the maximum speed. A common parallel track distance in Germany is 4.0 m for train speeds up to 200 km/h [15]. A simple map-matching by a nearest neighbor approach (see 6.2.2) uses inherently half of the parallel track distance as a decision threshold.

The GNSS accuracy is monitored and periodically reported by the wide area augmentation system (WAAS) stations in North America and by European geostationary navigation overlay service (EGNOS) stations in Europe. Figure 4.2 shows the histogram and cumulative error probability of the horizontal navigation system error of the EGNOS open service. There, the probability of a horizontal error below 2.0 m is shown with over 99.9 % from 27 monitor stations [65].

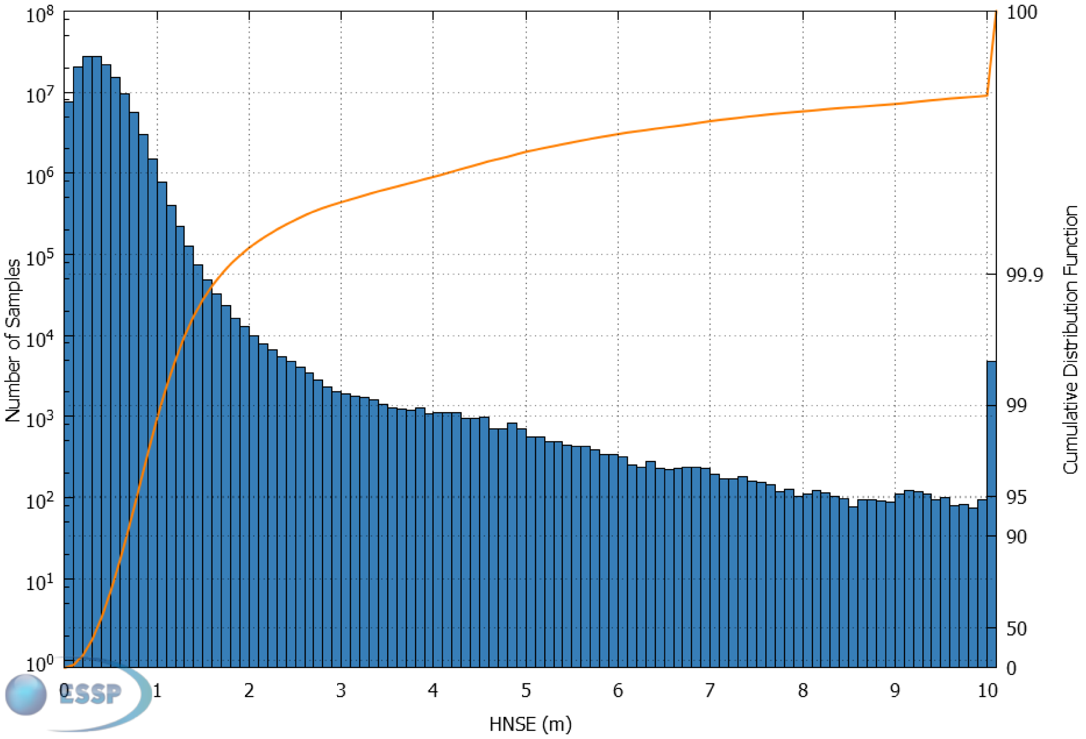


Figure 4.2: EGNOS open service horizontal navigation system error (HNSE) histogram and cumulative probability of August 2016. Courtesy of ESSP and European GNSS Agency, produced under a program funded by the European Union [65].

The measurement conditions of a train mounted GNSS receiver are different compared to the ideal conditions of the monitor stations. The major differences are multipath and obstructed line-of-sight paths to satellites in the railway environment. A train driver's view of the railway environment can be seen in Fig. 4.3. There are many possible signal reflectors, such as metal poles, catenary, bridges, other trains, railway tracks and sound walls. Other possible sources of multipath are buildings, trees or station roofs next to the tracks. The own train can also cause multipath by the pantograph or with roof structures such as the casings of air-conditioning.

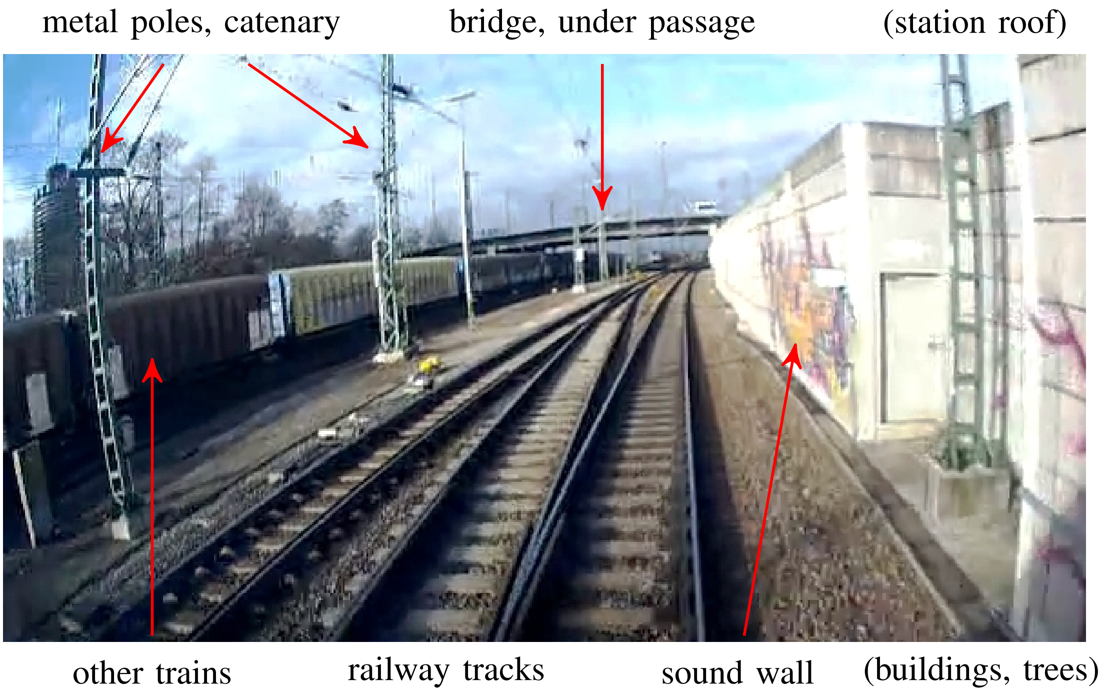


Figure 4.3: Possible multipath causes in railway environment.

Besides multipath, signals from low elevated satellites are likely obstructed by the railway environment. With higher latitudes of the GNSS receiver position, the satellite augmentation signals from geostationary satellites, such as EGNOS, have lower elevations and are also more likely obstructed by the railway environment. This causes a degradation of the satellite geometry. Additionally, GNSS signals are not available in tunnels or below station roofs. As a consequence, there is a degraded accuracy performance expected in the railway environment compared to the ideal conditions of the monitor stations. A localization based on GNSS positions and a nearest neighbor method (simple map-match) is likely to fail in a parallel track scenario.

4.1.3 Analysis of GNSS Accuracy with Train Measurements

The following measurement analysis encompasses a comparison of GNSS position measurements of several runs with an example track map and a reference map. Furthermore, the GNSS measurements of position and motion direction are analyzed at a railway switch. The presented analysis uses filtered PVT data (position, velocity, time) from a single frequency (L1) GPS receiver (u-blox LEA 6T) with one patch antenna. This GNSS setup represents a low-cost COTS setup. The measurements were recorded on a regional passenger train, as described in Section 7. The following shows, how the GNSS measurements of position, motion direction and speed contribute to the train location estimation.

GNSS Position Measurements

Figure 4.4 shows typical GPS position measurements of ten different train runs over the same railway track. The train repeats in every run the same trajectory given by the positions of the rails. It can be seen that each GPS trace (orange) has a nearly constant offsets to each other trace and the track map. The example track map from OSM data (magenta) shows also an offset to the intended center line between the rails, represented with a white reference line.

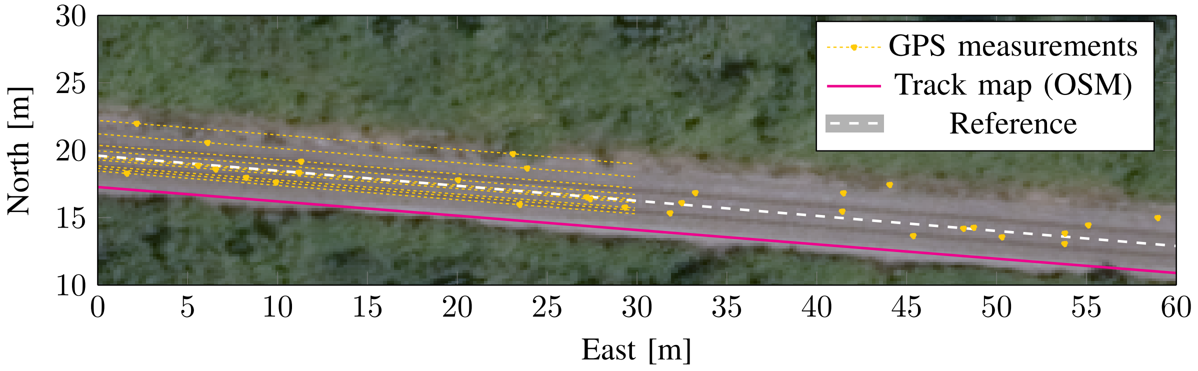


Figure 4.4: GPS position traces (orange) of ten runs on a single track scenario and a biased track map (magenta) and a reference trace (white).

The position measurements will be associated to the track map in the train localization, later on. An error between the position measurements and the track map can be observed in cross-track direction, with the cross-track distance δ_{CT} (see Section 3.2.1). This error encompasses a map bias and the error between measured positions and true positions. The along-track error is not observable. As mentioned in Section 3.3.3, a cross-track map bias $\delta_{\mathrm{CT,map}}$ defines the offset between the true center line of the tracks and the track map.

Figure 4.5 shows the cross-track error distribution between GNSS position measurements and a reference track map. The measurement setup is explained with more details in Section 7.1, the reference track map is explained in Section 7.2.2, and the cross-track error computation is explained in Section 7.3.3. The cross-track evaluations comprise position measurements with 1 Hz and a total length of 4.4 h from 21 train runs (see Table 7.1 and Table E.2). These evaluations are limited to the extent of the reference track map between Augsburg main station and station Friedberg. The axes

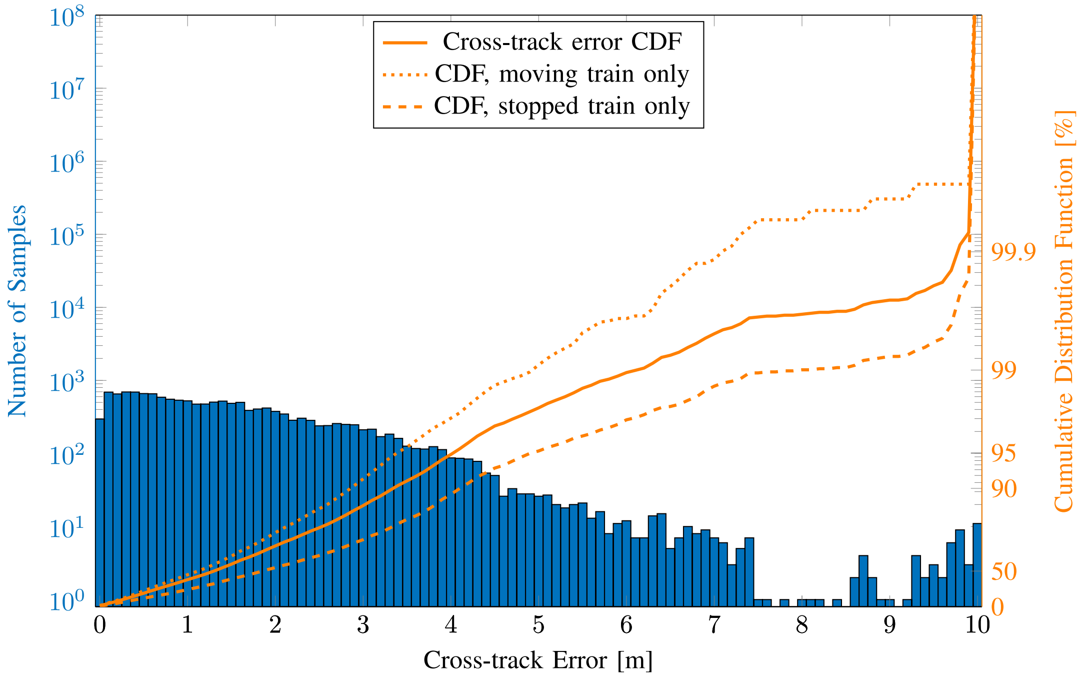


Figure 4.5: Cross-track accuracy evaluation of a COTS GNSS receiver in a railway environment. Cross-track error histogram and cumulative distribution function (CDF) between GNSS position measurements and the reference track map. The last histogram bin contains all errors $\geq$ 10 m.

scales are the same as in Fig. 4.2, and the last histogram bin in both figures contains all remaining errors equal or greater than 10 m. The cross-track error distribution of Fig. 4.5 shows a lower accuracy compared to the ideal horizontal error deviation of a monitor station in Fig. 4.2. However, the two figures are not directly comparable. The data set of Fig. 4.5 comprises less samples with 16006 measurement samples. As a consequence, the possible satellite constellations are not complete. Additionally, the train routes were mostly in east-west direction and thus the cross-track errors are in north-south direction. The reference track map may be close to the true positions of the tracks, but this track map contains still position errors. Furthermore, the type and mounting position of the GNSS antenna has influence on the cross-track error. Improvements regarding the antenna setup are discussed in Section 8.1.5. Nevertheless, the following train localization and track mapping methods are evaluated with these measurements. A train localization computes and uses this cross-track distance as part of the track identification. The error histogram of Fig. 4.5 shows still a lower accuracy compared to the expected performance figures on GNSS position accuracy from Fig. 4.2. Figure 4.5 shows also the error CDFs when the train is stopped and in motion. A stopped train shows larger errors than a train in motion because a multipath changes with a moving train more quickly and the receiver filters a fast changing multipath. A deterministic track identification of parallel tracks with a track separation of 4 m has a discrimination threshold at 2 m. A dynamic track identification at parallel tracks or at switch with a moving train shows 76.3 % below 2 m (see Fig. 4.5). A static track identification with a stopped train at parallel tracks is more difficult with a lower accuracy of 53.3 % below 2 m. The ideal monitor station showed over 99.9 % of the deviations below 2 m (Fig. 4.2). Section 8.2.2 shows more distributions of cross-track errors with different track maps and positions from GNSS only as well as positions from an integrated navigation system (INS/GNSS).

GNSS Position Measurements at a Switch

The switch is a crucial point for a track-selective train localization. The switch way in left or right direction is resolved from the measurements. Figure 4.6 and Fig. 4.7 illustrate the GPS positions of three runs to the left switch way and three to the right. The 2-D positions are shown in along and cross-track coordinates relative to the start of the switch. The cross-track bias δ_{CT} between measurements (blue and orange) and the map (gray) can be seen in Fig. 4.6.

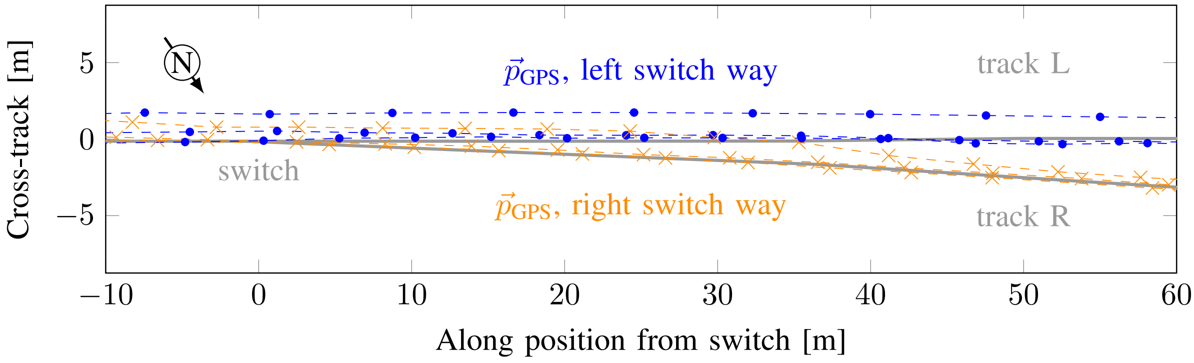


Figure 4.6: GPS position traces over left (blue) and right (orange) switch way.

A cross-track bias is unfortunate for a correct switch-way estimation. On the other hand, this bias can be observed from measurements and the map just before the switch, for the case that the bias does not change. The switch way estimation can then be calibrated by the observed cross-track bias $\delta_{\mathrm{CT}}(s_{\mathrm{switch}})$. As a consequence, only the difference signal is evaluated by the estimator or switch-way detector. In Fig. 4.7, the last cross-track bias is stored before the switch start, and then it is subtracted from the measurements.

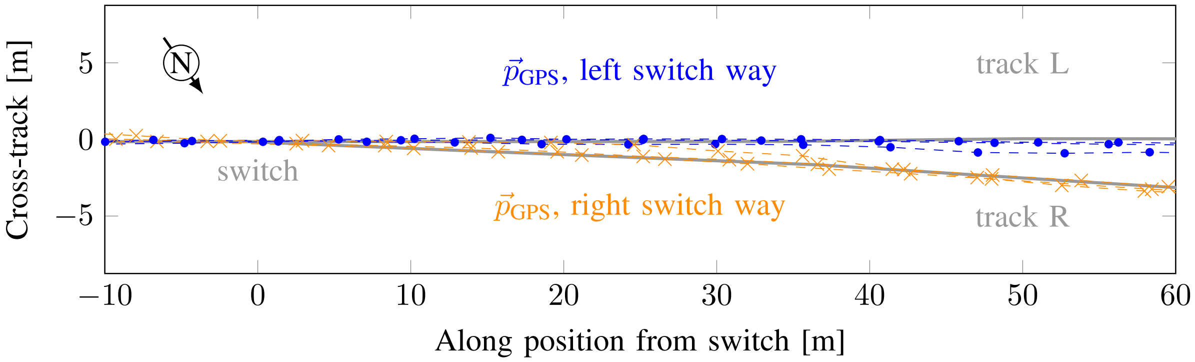


Figure 4.7: GPS position traces over switch with cross-track bias compensation.

Figure 4.8 shows the cross-track difference signals of the switch-way measurements. With three left and three right position traces, there are three difference signals between the same switch way of left (blue) and and three between the same of right (orange). Ideal measurements of the same switch way have no differences and should appear as a zero line. There are nine difference signals between a left and a right switch way (green). These differences are computed between all six measured position traces:

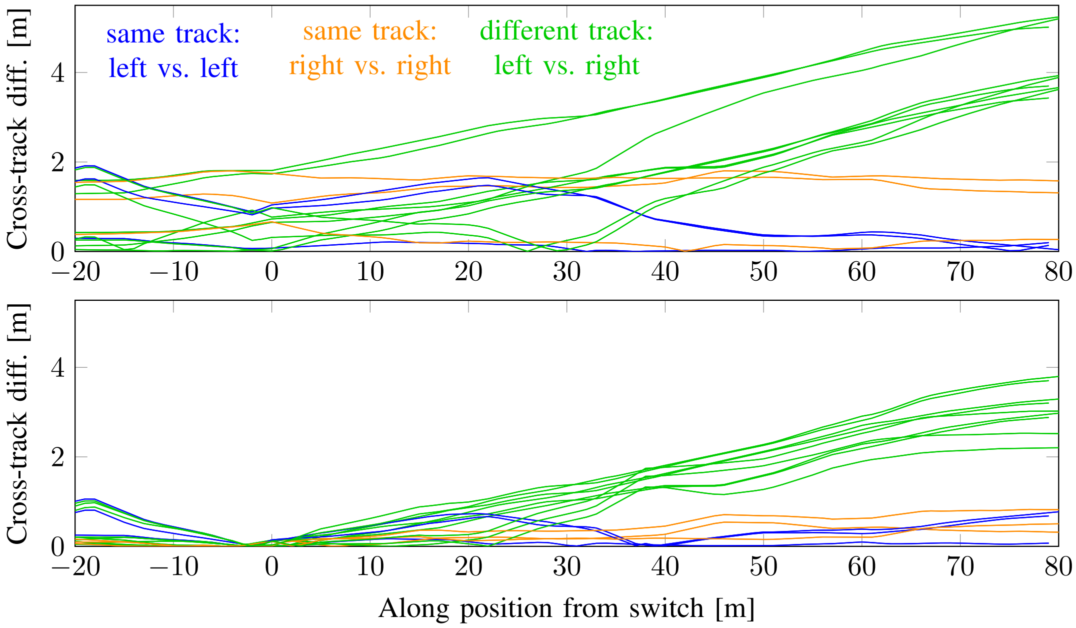


Figure 4.8: [Top] Signals of cross-track distance differences between GPS position traces. [Bottom] Signals of cross-track distance differences with bias compensation.

Therefore, the cross-track deviations are computed between the trace positions and a track map with a simple map-match method (Section 6.2.2). The difference signals are computed at defined along-track positions between interpolated deviation curves of two different traces. The upper plot of Figure 4.8 uses directly the GPS measurements while the lower plot subtracts the cross-track bias. The lower plot shows a better separability between the signals of different switch ways and the same switch way. The signal separation can be enhanced with the cross-track bias compensation for the case that this bias is constant during the train run over the switch. A better signal separation is beneficial for the switch-way estimation in order to resolve the correct switch way.

GNSS Heading Measurements at a Switch

Apart from position measurements, a GNSS receiver measures and estimates the motion of the antenna. A heading angle ψ_m of the train motion is defined between the horizontal motion direction and north. The heading estimate of a single antenna receiver requires a movement and can be computed from position differences or from the velocity vector $\vec{v}_{\mathrm{NED}}$ of the receiver in the navigation frame. The heading angle is computed from the north (v_{n}) and east (v_{e}) component of the estimated velocity vector:

$$\psi_{\mathrm{m}} = \arctan(\frac{v_{\mathrm{n}}}{v_{\mathrm{e}}}). \tag{4.2}$$

Figure 4.9 shows the heading from GPS measurements of the train experiment. There are six runs over the switch and the switch is passed facing. The blue lines show the headings of three runs over the 1-D track locations for the left switch way. The orange lines show the of three runs to the right switch way. It can be seen, that the switch ways can be separated against each other from the motion heading measurements. For a train localization, the GNSS heading measurements contribute mainly to the track identification at a switch. Further, the heading measurements may also contribute to an absolute along-track and track identification, if the different location hypotheses show heading differences.

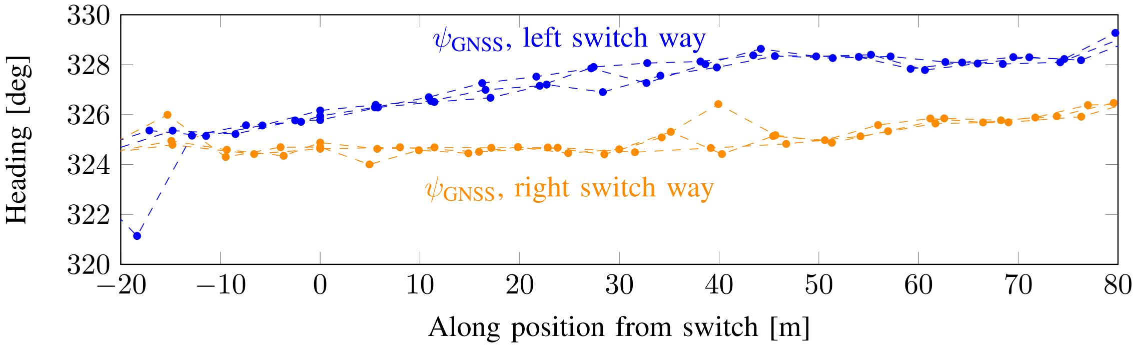


Figure 4.9: GPS heading measurements of different switch ways.

The orientation of a train is denoted with the yaw angle ψ and is defined as the angle between the train frame and north (see Section 3.2.1). Modern trains are designed as two directional vehicles and operate equally in both directions. A forward moving train has the same yaw angle (train frame) than the heading of the motion direction. In the case that a train moves backwards, the yaw angle of the train frame is shifted to the motion heading by π, or 180° respectively. As a consequence, the measured heading angle needs to be converted to the train frame or track frame with Table 3.2 for further processing.

GNSS speed

The GNSS speed measurement contributes to the train localization in terms of relative displacement estimation. The speed measurement of a GNSS receiver is computed from the length of the horizontal velocity vector in the navigation frame:

$$|v|_{\mathrm{GNSS}} = \sqrt{v_{\mathrm{n}}^2 + v_{\mathrm{e}}^2}. \tag{4.3}$$

A single GNSS antenna is invariant for horizontal rotation, i.e. the speed and also heading measurements are independent of the angle between an antenna frame and the train frame. As a consequence, the GNSS speed is always positive and points in the direction of motion. Similar to the heading measurements, the speed measurement is converted from the motion direction to the train frame, or track frame if required. As defined in Section 3.2.1, the speed in train frame of a forward running train has a positive sign, while a backward running train has a negative sign. The further conversion to the track frame was defined in Eq. (3.15). An example of a speed measurement from GPS is illustrated in Figure 4.10 with a typical passenger train run of 66 km from Augsburg to Ingolstadt and eight train stops in between.

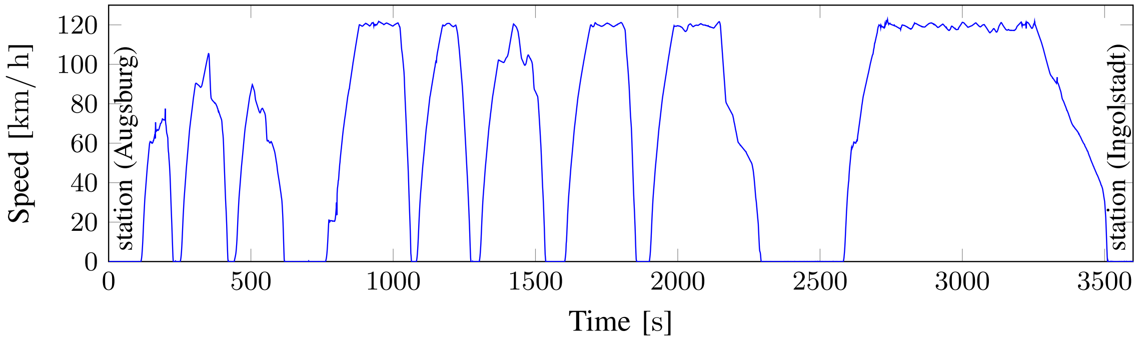


Figure 4.10: GPS speed measurements of a train-run from Augsburg to Ingolstadt.

4.2 IMU: Inertial Measurement Unit

An inertial measurement unit (IMU) measures accelerations and turn rates in three dimensions. Section 4.2.1 starts with an overview of the measurement principle and errors and continues with the common use-case of an IMU as part of an integrated navigation system (INS) in Section 4.2.2. Besides the INS approach, other methods for train localization with an IMU are presented. These methods measure railway track features, in particular the geometric characteristics of a track. The theoretical expected measurements of track features with an IMU are presented in different models in Section 4.2.3. Finally, IMU measurements of train runs are analyzed in Section 4.2.4 and the track feature methods are analyzed for applicability in a train localization.

4.2.1 Measurement Principle and Errors

An inertial measurement unit (IMU) combines three turn rate sensors, named gyroscopes, and three acceleration sensors, named accelerometers. Figure 4.11 shows a typical block diagram of an IMU. This IMU measures a motion in six degrees of freedom (6 DoF) with three dimensional acceleration measurements $a_{x,y,z}$ (translations), and three dimensional turn rate measurements $\omega_{x,y,z}$ (rotations). Therefore, an IMU contains a sensor triad of orthogonal acceleration sensors and a triad of orthogonal gyroscopes. Most IMUs contain additionally a controller and measure a temperature for corrections. An internal or external clock signal triggers measurements at a constant frequency. The controller corrects the measurements with calibration data for the correction of biases, scales, misalignment, and temperature. Finally, the IMU outputs the six measurements at a certain clock rate (see [66]).

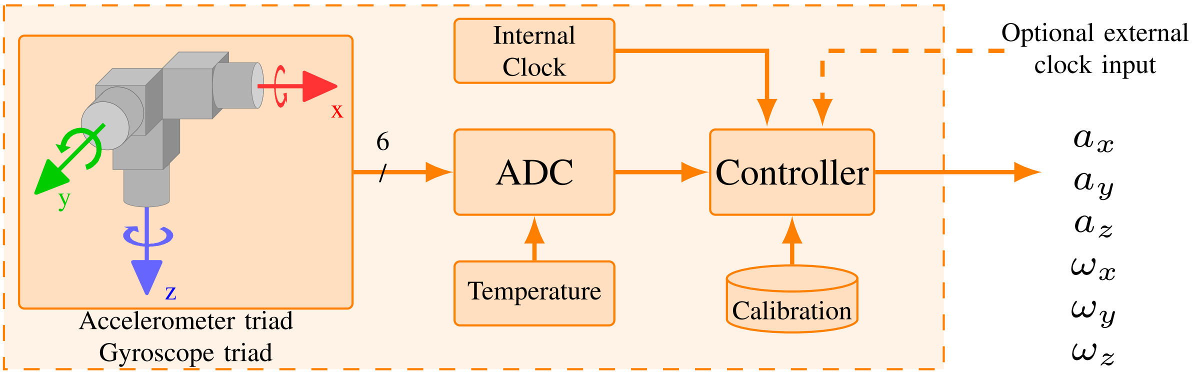


Figure 4.11: Typical design of an IMU.

There are several types of accelerometers and gyroscopes which differ in design, measurement principle, and accuracy ([66]). The train measurements of Chapter 7.1 comprises a data set with a micro electro mechanical system (MEMS) IMU from Xsens (Xsens MTi) and also a high-end fiber optical gyroscope (FOG) IMU from KVH (1750 IMU) as reference.

Accelerometer

Both types of IMUs from the train experiment contain MEMS acceleration sensors. There a different ways to design an acceleration sensor and alternatives can be found in [66]. MEMS acceleration sensors are widely used in the automotive domain and in consumer products, such as smartphones. An acceleration sensor measures a translational acceleration a via inertial force f which acts on a well-known mass m:

$$a = \frac{f}{m}. \tag{4.4}$$

This mass is attached to a spring with known spring characteristics. A force or acceleration causes a displacement of the mass, which is measured and digitized. A typical MEMS acceleration sensor combines the spring with mass, the displacement sensing, and signal processing in silicon. A displacement of the mass changes the distances of electrodes of differential capacitors, which can be translated to a signal (see [66]).

Gyroscope

The turn rate is measured with a gyroscope. There are different ways to built a gyroscope, and an overview is given in [66] as well. In the following, a MEMS gyroscope and a FOG gyroscope are explained briefly.

A MEMS gyroscope measures an angular rate via the Coriolis acceleration. Therefore, a small and known mass oscillates linearly in one axis and an applied turn rate causes a Coriolis force. The oscillation axis, the turn rate sensing axis and the axis of the Coriolis force are perpendicular to each other. The Coriolis force is measured similar to the explained MEMS acceleration sensor (see [66]).

Another type of gyroscope is the fiber optical gyroscope (FOG). The measurement principle of optical gyroscopes is based on the Sagnac effect. This effect refers to a difference in optical length of two light beams, which travel in opposite directions through an optical ring. A FOG uses a fiber coil with multiple windings. At first, a light beam is split and coupled into the opposite endings of the fiber coil. The beams are combined afterward and inference fringes are observed using a photo detector (see [66]). The sensor noise and bias stability of a FOG gyroscopes is typically lower compared to MEMS gyroscopes.

IMU Measurement Errors

Some inertial measurement errors, such as scale errors, non-linearity, misalignment, biases and temperature dependencies can be determined in a calibration procedure. The internal controller can adjust for these known errors as long as these errors are persistent over time. However, there are also measurement errors, that are not persistent and change over time. A significant measurement error of inertial sensors is the changing sensor bias over time, also known as drift. Table 4.1 shows the major sensor noise characteristics of MEMS acceleration sensors of the IMUs used in the experiment. The bias stability is the characteristic value for the changing bias during operation of the IMU (in-run) and includes the changes caused by temperature over the specified temperature range. The value of the velocity random walk defines the expected variance of the velocity estimate after one hour caused by a white noise. It can been seen, that the KVH MEMS sensors have better performance characteristics for the bias stability and noise by a factor of two compared to the Xsens acceleration sensors.

Acceleration	**MEMS** **Literature** [47]	**MEMS** **Xsens MTi** [67]	**MEMS** **KVH 1750** [68]
measurement range ±[m/s^2]	up to 1000	50	100
bias stability (in-run, over temp.) σ [m/s^2]	0.01-0.1	0.02	0.01
noise: velocity random walk [m/s/$\sqrt{\text{h}}$]	down to 0.04	0.12	0.07

Table 4.1: Theoretical accelerometer characteristics.

The rotational turn rates are measured with gyroscopes. Table 4.2 shows the major sensor noise characteristics of MEMS and FOG gyroscopes used in the train experiment. The value of the angular random walk defines the expected variance of an angle estimate after one hour caused by a white noise.

The gyroscope values of Table 4.2 shows the difference of FOG and MEMS: The used MEMS gyroscope biases have a deviation of 60 °/h due to the drift and temperature effects, and a 3° standard deviation after one

Gyroscope	**MEMS** **Literature** [47]	**MEMS** **Xsens MTi** [67]	**FOG** **Literature** [47]	**FOG** **KVH 1750** [68]
measurement range ±[°/s]	100-6000	300	up to 1000	490
bias stability σ [°/h] (in-run, over temp.)	5-135	60	0.1-50	1
noise: $[°/\sqrt{\mathrm{h}}]$ angle random walk	down to 1	3	down to 0.1	0.013

Table 4.2: Theoretical gyroscope characteristics.

hour due to integrated noise. The FOG gyroscope bias stability is smaller than 1 °/h and 0.013° angle random walk after one hour.

4.2.2 Inertial Navigation System

An inertial navigation system (INS) contains an IMU and computes the position, attitude and velocity of a vehicle from acceleration and turn rate measurements. The following contains a brief overview of use-cases, state-of-the-art functional principle, errors, and solutions that can be found in [66, 47]. The INS will be used in the measurement processing (Section 6.1) for one localization method (Section 6.2.4) and in the track map creation method of the track geometry (Section 6.3).

One way to built an INS is to process inertial measurements from a fixed mounted and strapped-down IMU. The processing method is hence called strapdown method [66, 47]. In order to compute the attitude, the velocity and the position, three differential equations need to be solved. The equations for the strapdown method can be found in the Appendix D. The strapdown method (SD) computes for each discrete time step k the attitude, the velocity and finally the position:

$$\vec{x}_k^{\mathrm{SD}} = (\underbrace{\varphi, \lambda, h}_{\text{position}}, \underbrace{v_n, v_e, v_d}_{\text{velocity}}, \underbrace{\phi, \theta, \psi}_{\text{attitude}})^T. \tag{4.5}$$

The strapdown method, defined as a function, uses as input the previous result at $k-1$, the measurements of acceleration $\vec{a}$ and turn rates $\vec{\omega}$, and computes a new result for the current time step k:

$$\vec{x}_k^{\mathrm{SD}} = \mathrm{f}_{\mathrm{SD}}(\vec{x}_{k-1}^{\mathrm{SD}}, \vec{a}_k, \vec{\omega}_k). \tag{4.6}$$

The main problem of an INS is the growing error over time. An INS computes only relative values and requires therefore an initial information about position, velocity and attitude. The position, for example, is computed via dead reckoning from this initial position in the strapdown method (see [47],[66]). An update with inertial measurements is a relative increment of attitude, velocity, position and added to the previous result. As a consequence, measurement errors sum up and the errors of position, velocity and attitude grow over time. The accuracy of an INS with IMU depends strongly on the drift and noise of the gyroscopes: At first, uncompensated biases of the turn rate measurements result in a growing angular error over time of the attitude due to temporal integration. Afterward, the attitude estimate is used for the gravity compensation of the acceleration measurements. The erroneous attitude causes an erroneous acceleration that is further integrated two times for the position estimate. Secondly, a constant or growing error of the heading angle influences also the position accuracy of a moving vehicle. The position error grows over time because the vehicle moves in a wrong direction. An INS with solely IMU measurements can only provide a defined accuracy for a certain time. This time duration determines the quality and grade of an INS. As mentioned in the previous section, there are different types of IMUs, or gyroscopes in particular, with different figures in terms of noise and drift.

A long-term stable INS requires additional measurements. An INS is usually combined with GNSS measurements to an INS/GNSS [66, 47]. GNSS measurements solve the problem of the initial starting point and limit the error growth. An INS/GNSS estimates also the inertial measurement biases from GNSS measurement updates in order to compensate the drift. An INS/GNSS is often used to enhance positioning data compared to GNSS only. The INS/GNSS increases the position accuracy and availability in short GNSS outages and in environments with degraded signal reception over limited periods of time. Further, an attitude estimate is achieved that is stable with a limited error over time. A loosely coupled version of an INS/GNSS is defined for the GNSS and IMU data fusion in the implementation Section 6.1.

4.2.3 Theoretical Measurements of Track Features with an IMU

A train localization compares measurements with information from the track map. Accelerations and turn rates can not be stored in a map, because these quantities are dynamic and depend on the train speed. Instead, the geometry of the tracks can be stored in the track map. The rails of a track constrain the trajectory of a train. When a train runs over a curved track with curvature c^ψ, the train changes its heading ψ with a constant turn rate $\dot{\psi}$. At the same time, a centripetal acceleration a_{lat} acts on the train in lateral direction. The relation between the turn rate, the speed and curvature is [69]:

$$\dot{\psi} = \tfrac{d\psi}{dt} = \tfrac{d\psi}{ds} \cdot \tfrac{ds}{dt} = c^\psi \dot{s}. \tag{4.7}$$

The relation between lateral acceleration, turn rate, train speed and curvature is also described in [69]:

$$a_{\text{lat}} = \dot{\psi}\dot{s} = c^\psi \dot{s}^2. \tag{4.8}$$

An IMU, that is mounted on a moving train, measures these turn rates and accelerations. The curvature c^ψ is an important track feature for the switch-way identification and also suitable for an observation of the along-track location at curvature changes. A curvature has been also used for car localization in [69] and for train localization in [18, 19]. An advanced filtering and classification of a railway track curvature into standard geometric elements, e.g. straights and curves, have been analyzed in [70, 71]. Figure 4.12 shows theoretical turn rates and lateral accelerations of the curved track of six different German standard switches, as presented in Section 3.1.2. Each railway switch has a different curve radius and a speed limit. The diagram shows the linear speed dependency of the turn rate and the quadratic speed dependency of the lateral accelerations.

The horizontal curvature can be measured in several ways with an IMU, which was theoretically analyzed in [C3]. The first method uses directly the IMU measurements of the turn rate ω_z and the lateral acceleration a_{lat} in horizontal direction:

$$c_1^\psi = \frac{\omega_z^2}{a_{\text{lat}}}. \tag{4.9}$$

The second variant uses the turn rate measurement and the estimated speed $\dot{s}$ in track frame:

$$c_2^\psi = \frac{\omega_z}{\dot{s}}. \tag{4.10}$$

The third variant depends on the lateral acceleration measurement and the train speed:

$$c_3^\psi = \frac{a_{\text{lat}}}{\dot{s}^2}. \tag{4.11}$$

These three methods are theoretically analyzed in [C3] with three different types of IMU (consumer, automotive, tactical) and their typical sensor noise. This study showed for consumer and automotive grade IMUs, that

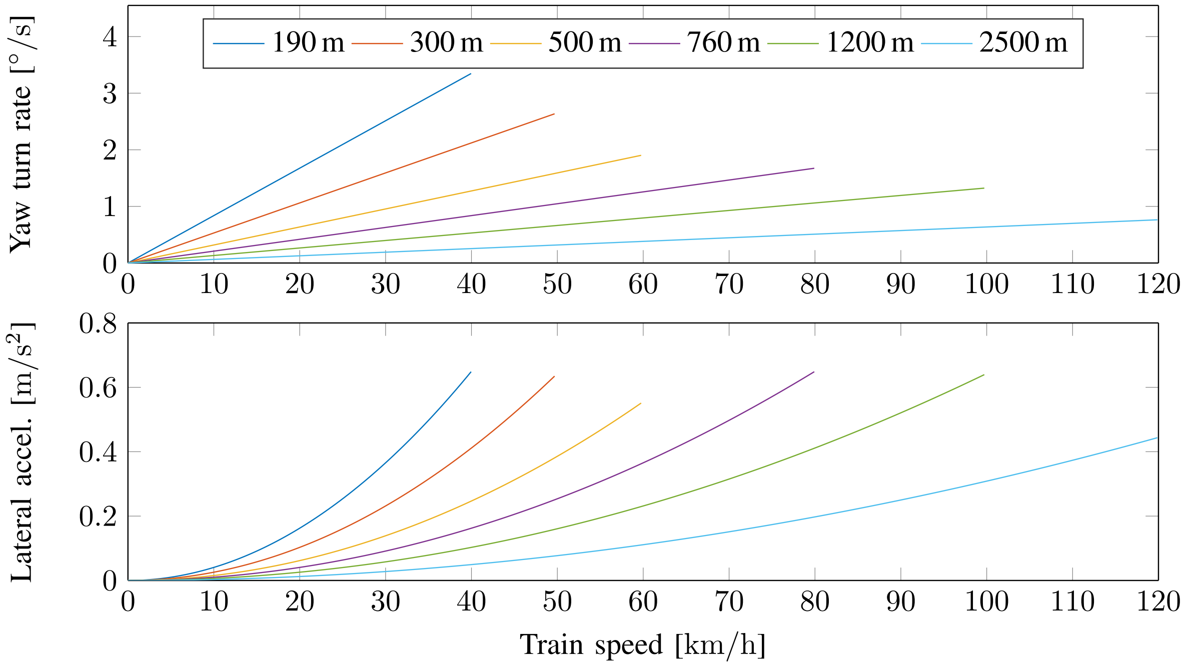


Figure 4.12: Theoretical yaw turn rates and lateral accelerations over train speed of selected railway switches.

c_2^ψ is best for lower velocities and c_3^ψ is best for higher velocities above 50 km/h. For tactical grade IMUs, c_2^ψ is always the best, followed by c_1^ψ.

A 3-D kinematic model for the inertial measurements is defined in the following. This model relates the IMU measurements to the dynamic train states of train speed v and acceleration a_{tb} to the track geometry at a particular track location (id, s) with the train-to-track direction o^{t}.

There are two different possibilities for the curvature definition: The first one defines the curvatures on planes that are defined about the horizontal axes of the navigation frame in along-track, cross-track, and vertical down axes. The second curvature definition defines curvatures in the track frame about the track frame axes in along-track, cross-track, and vehicular down direction. The track frame axes may be tilted or inclined to the horizontal axes. The first curvature definition has been used in [P1] and is not followed up. The second curvature definition with the track frame is more practical and is used in the following.

The acceleration model is defined in the train body frame from track geometry, train speed and acceleration:

$$\underbrace{\mathbf{C}_{\mathrm{s}}^{\mathrm{b}}\vec{a}^{s}}_{\substack{\text{measurements}\\ \text{in train frame}}} = \underbrace{\mathbf{C}_{\mathrm{n}}^{\mathrm{b}}(\phi^b,\theta^b)\begin{pmatrix}0\\0\\-g\end{pmatrix} + \begin{pmatrix}a_{\mathrm{tb}}^b\\ a_{\mathrm{lat.}}^b\\ a_{\mathrm{vert.}}^b\end{pmatrix}}_{\substack{\text{predicted measurements}\\ \text{from map, location, speed and acc.}}} . \tag{4.12}$$

The lateral acceleration and the vertical acceleration are centrifugal accelerations. The IMU measures in the sensor frame and the values are rotated to the train frame. The final acceleration model uses Eq. (4.8) with all values in train frame:

$$
\begin{aligned}
a_x^b &= g \sin \theta^b + a_{\text{tb}}^b, && (4.13)\\
a_y^b &= -g \sin \phi^b \cos \theta^b + c^{\psi,b} \cdot v^2, && (4.14)\\
a_z^b &= -g \cos \phi^b \cos \theta^b - c^{\theta,b} \cdot v^2. && (4.15)
\end{aligned}
$$

The track geometry values of bank ϕ, slope θ, slope curvature c^θ, and heading curvature c^ψ are obtained from the map in the track frame. For an opposite track-to-train direction o^{t}, the angles ϕ, θ and the curvature c^ψ change signs, as defined in Section 3.2.2, Table 3.3.

The turn rate measurements from the gyroscopes $(\omega_x, \omega_y, \omega_z)$ are also rotated from sensor frame to the train frame. The relation between measurements and the predicted turn rates from the track geometry of the map are given by Eq. (4.7) with curvatures in the body frame and the train speed v:

$$
\begin{aligned}
\omega_x^b &= c^{\phi,b} \cdot v, && (4.16)\\
\omega_y^b &= c^{\theta,b} \cdot v, && (4.17)\\
\omega_z^b &= c^{\psi,b} \cdot v. && (4.18)
\end{aligned}
$$

For an opposite track-to-train direction o^{t}, the heading curvature $c^{\psi,t}$ from the map changes its sign, see Table 3.3.

4.2.4 Measurement Analysis of a Train-mounted IMU

The following figures Fig. 4.14 and Fig. 4.15 show IMU data and the frequency spectrogram over a 10 minutes run of a regional train with two station stops in between. The IMU data was recorded with 200 Hz in the train cabin with a Xsens MTi (MEMS). The axes of the IMU measurements are aligned with the train frame. The spectrograms use a window and FFT length of 256 samples and visualize the signal over time in the frequency domain with frequencies up to the Nyquist frequency of 100 Hz. These spectrograms use a decibel scale to visualize the signals with the logarithmic ratio to 1 m/s^2 for the acceleration measurements and to 1 °/s for the gyroscope measurements. Figure 4.13 shows the speed profile of a typical train run for the IMU measurement analysis. There are two extra train stops between 175 s-200 s and 375 s-400 s.

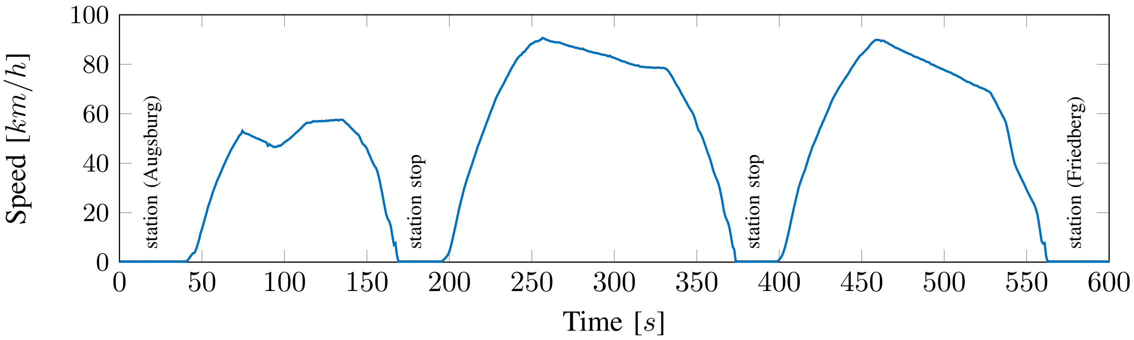


Figure 4.13: Speed profile of the train run for the IMU analysis.

The spectrograms of Fig. 4.14 show vibrations in all axes, which can be noticed by the higher frequency components between 1 Hz and 100 Hz. The stops of the train can be identified in the acceleration measurements. The vibrations of a stopped train arise mainly from the diesel engine and fans. A moving train causes different vibrations from the engine, gearbox, wheels and the track (see [P1, P5]). More details about vibrations are described in Section 4.3.2. The train acceleration or deceleration is measured in the x-axis. These accelerations are limited to approx. 1 m/s^2 due to limited engine power and a limit of the maximum transferable force over the rail-wheel contact [15]. This traction limited force leads to low dynamics of the x-axis acceleration in combination with inertia of a relative high train mass. Figure 4.14 indicates these low dynamics and changes at 1 Hz and below in the spectrogram of the x-axis. The y-axis in shows accelerations due to the banked track between 80 s and 250 s as well as lateral accelerations of higher switch curvatures at 75 s and 350 s. The z-axis shows vibrations only.

The gyroscope values of Fig. 4.15 are less affected by vibrations than the accelerations. The turn rate of the gyroscope x-axes measures the change of the roll angle caused by the bank curvature and the actual speed. The gyroscope y-axis measures the low changes of a slope angle. However, the diagrams of the y-axes do not visualize these low signals. The gyroscope z-axes measures the change of the yaw angle caused by the heading curvature and the actual speed. As seen in the diagrams, the heading changes are the strongest signal from the tracks. Some of these signals indicate curvatures from switches that can be also seen in the y-axis diagram of the acceleration measurements. All curvature signals have a low dynamic below 1 Hz and the remaining signal components are vibrations and measurement noise.

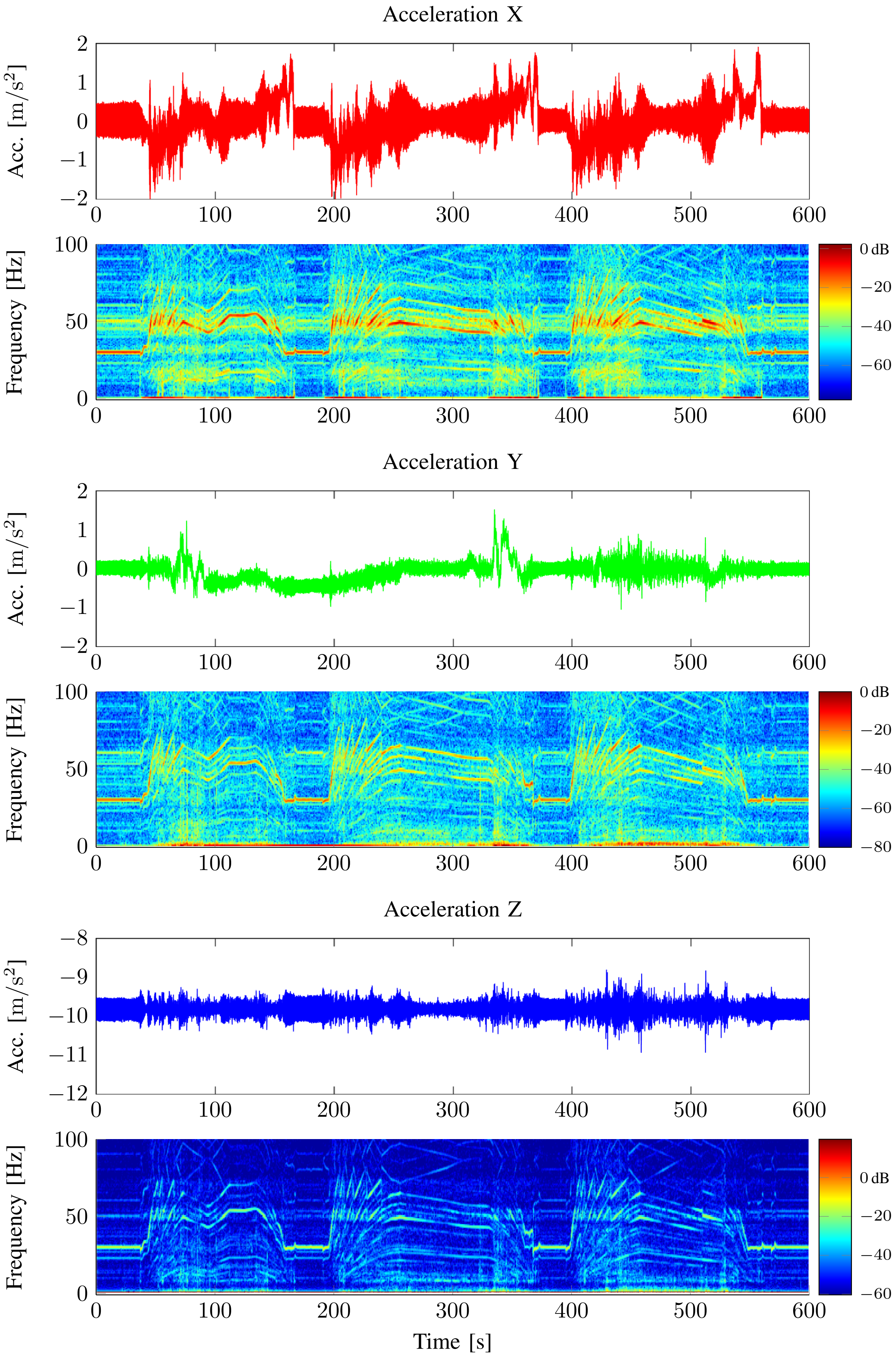


Figure 4.14: Acceleration measurements over time and spectrum.

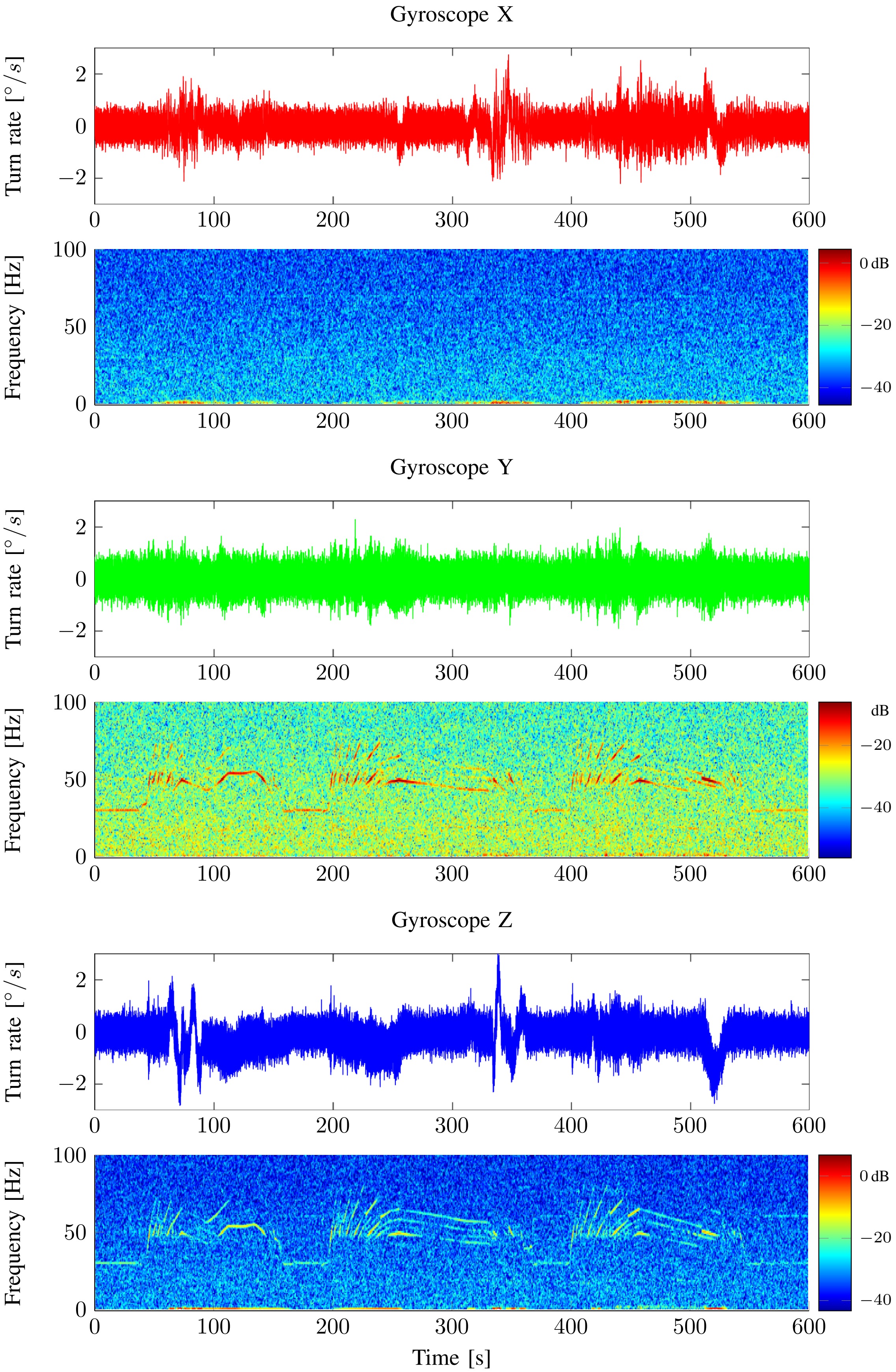


Figure 4.15: Gyroscope measurements over time and spectrum.

Train Acceleration Measurement

A typical train acceleration measurement a_{tb} of the reference run is shown in Fig. 4.16. Therefore, the signal of the x-axis acceleration (Fig. 4.14) is filtered with a 6'th order Butterworth filter at a cut-off frequency of $f_{(-3\,\mathrm{dB})} = 1\,\mathrm{Hz}$. As a reference, the change of GPS speed measurements between two consecutive time steps are shown for comparison reasons. The train moves backwards to the train frame in this example, and thus the train accelerates with negative accelerations and positive deceleration. The IMU accelerations and GPS speed changes match in most cases, but differences can be seen around 100, 130, and 140 s. There, the train runs below bridges and the GPS signals are disturbed. The accelerating maneuvers can be seen in the figure with several steps from the different gears. The brake maneuvers contain different steps for speed reduction and a final spot brake down to standstill with the strongest deceleration signals. Between acceleration and deceleration, the train is coasting with a low deceleration (e.g. 475 s-525 s). The IMU acceleration measurements can be used to estimate the velocity and the 1-D location on the track combined with GNSS velocity measurements.

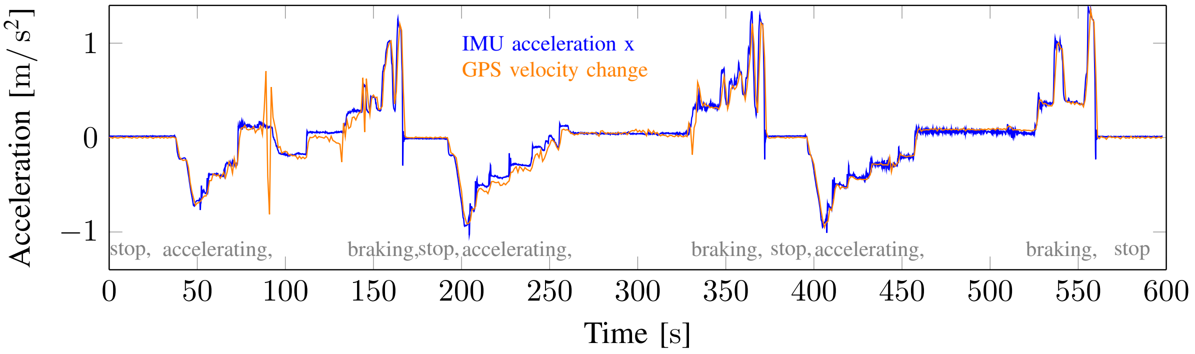


Figure 4.16: Typical train accelerations of a train run.

Measurements of the Attitude

An INS/GNSS estimates the attitude of the train from IMU measurements amongst other measurements. Figure 4.17 depicts bank ϕ, slope θ and yaw ψ estimates over the 1-D track location s. The example scenario shows six runs in the same direction on a straight and single track with a uphill slope. After 200 m the train passes a banked curve, changes its yaw direction and passes a switch at 470 m. The train passes three times the switch on the left switch way (blue) and three time on the right switch way (orange). The last part of the train runs are on parallel tracks of a station. The attitude signatures show a low discrimination characteristic in along-track direction. At the switches, especially the yaw angle shows a beneficial separability for the track identification. In this case, the bank angle can be also used to discriminate the switch. There, the train is excited in the sharp left turn of the switch. The cabin swings in a rolling motion and the IMU was mounted inside cabin. The different switch ways with their geometry have influence on the excitation of the cabin. A bogie mounted IMU does not show this rolling effect.

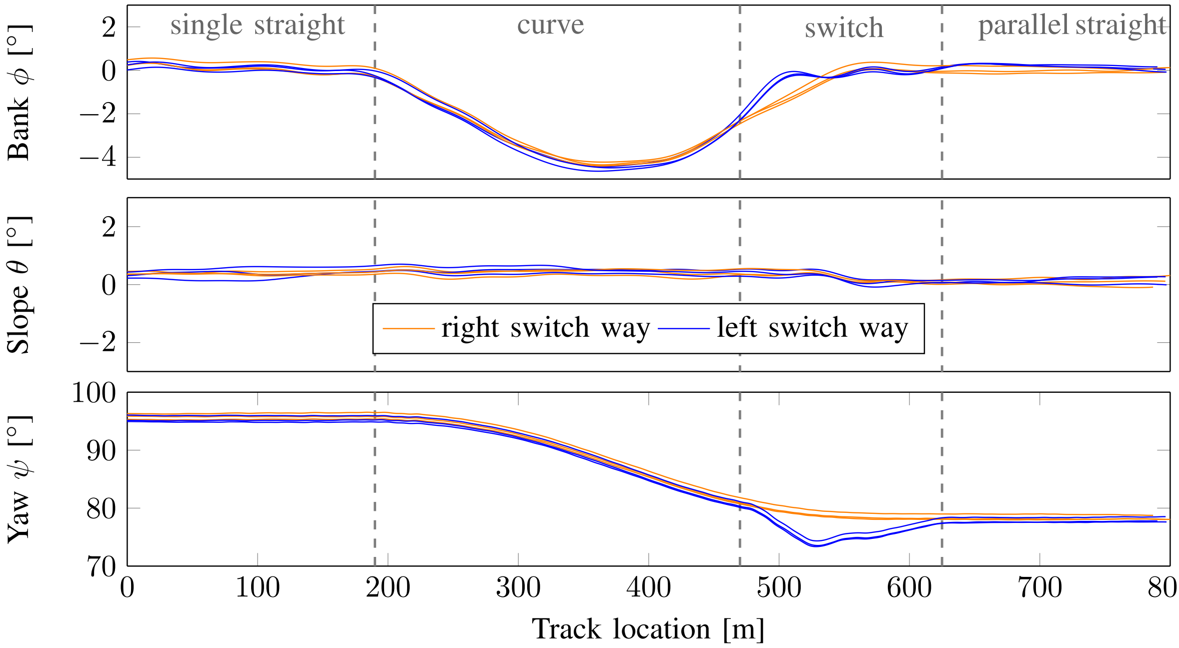


Figure 4.17: Attitude measurements from INS/GNSS measurements.

Measurements Analysis of Switch-way Curvatures

The different curvature computation variants of 4.2.3 are analyzed with recorded IMU data. In the theoretical analysis of 4.2.3, the curvature computations were free of systematic errors. Additionally, the curvature measurements are dynamic and the magnitudes of the IMU measurements depend on the train speed. The systematic errors are the IMU measurement biases. Further, the lateral acceleration of variant 3 (Eq. (4.11)) depends also on an exact estimate of the roll and pitch angle for corrections. The lateral acceleration (see Eq. (4.14)) is aligned to the cross-track axis of the track frame and is calculated from the bias corrected measurements:

$$a_{\text{lat}} = a_y - b^{\text{ay}} + g \sin\phi \cos\theta. \tag{4.19}$$

The acceleration measurements are further disturbed by a shaking train cabin or undercarriage.

The first curvature variant (Eq. (4.9)) uses solely a_{lat} and ω_z measurements from the IMU. There is a singularity when the acceleration measurement is close to zero in the denominator. This happens on straight tracks with lateral acceleration measurements near and around zero. The second and third variant (Eq. (4.10), Eq. (4.11)) contain the train speed in the denominator and require a certain minimum speed.

The following analysis contains typical measurements of several runs over two different switches. Each switch is overrun three times to the left and three times to the right switch way. The measurements are from the MEMS IMU inside the train cabin and the train speed, biases and roll angle are estimated with an INS/GNSS. The results from the first variant (Eq. (4.9)) were not useful due to the singularity issue and are not shown.

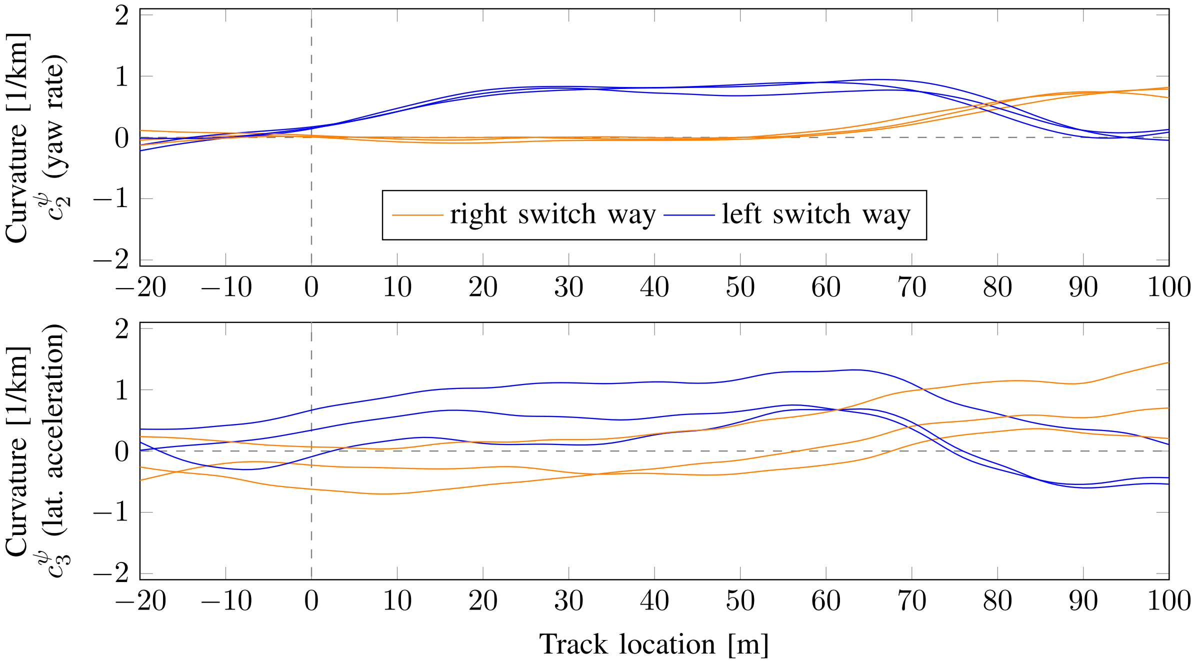


Figure 4.18: Switch A ($\approx 20\,\mathrm{km/h}$): Curvatures over track locations from cabin MEMS IMU. [Top] Curvature c_2^{ψ} (yaw rate), [Bottom] Curvature c_3^{ψ} (lat. accel.)

Figure 4.18 shows the first switch that is passed facing (see Fig. 3.1) for six times at low train speeds of approximately 20 km/h. It can be seen, that the differences of the curvature between left and right switch way are more significant with the second curvature variant by the yaw rate (Eq. (4.10)). The third variant by the lateral acceleration (Eq. (4.11)) shows different

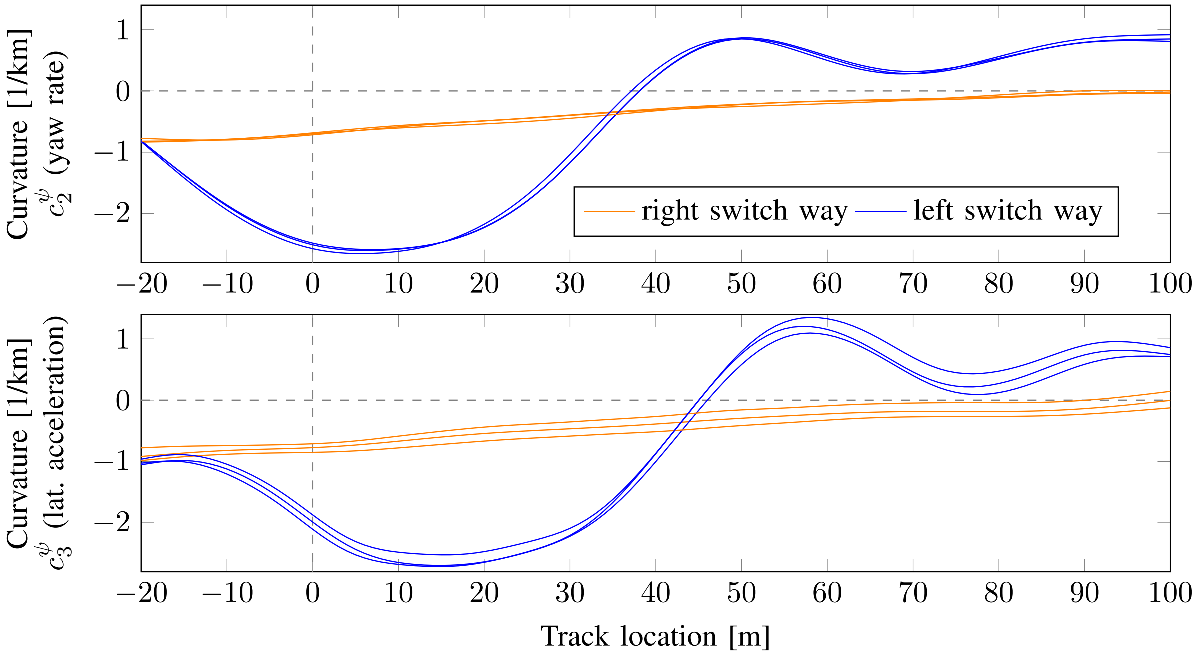


Figure 4.19: Switch B (≈ 55 km/h): Curvatures over track locations from cabin MEMS IMU. [Top] Curvature c_2^ψ (yaw rate), [Bottom] Curvature c_3^ψ (lat. accel.)

offsets and little variations between the runs. The offsets might be caused by an inaccurate estimate of the roll angle.

The second switch in Fig. 4.19 is passed six times at higher train speeds of 55 km/h in average. At higher train speeds, the third variant with acceleration shows a better signal-to-noise ratio and separability of the two different switch ways compared to Fig. 4.18 with a lower train speed. The curvature signature from the yaw rate (second variant, Eq. (4.10)) is still better in repeatability of the same switch way and differentiability of the competing switch way.

The train approaches the first switch in forward direction and the second switch in backward direction according to the train frame definition. The curvature signatures of the second switch are measured ahead of the switch start at track location 0 m in Fig. 4.19. This effect stems from the GPS antenna mounted on the train front, while the rear side of the wagon enters the switch at first. Further, the curvature from lateral acceleration in Fig. 4.19 shows a delayed signature of approx. 10 m in comparison to the turn rate based curvature. The reasons are: First, the wagon starts to turn, once the first wheel set enters the curved track and because the wagon can be considered as a rigid body. At this time, the GPS antenna is about one wagon length (20 m) away. Second, the measurement of the lateral acceleration depends on the mounting position of the sensor on the train and the delay is caused from lever arm effects. The IMU was also mounted near the train front.

Compared to Section 4.2.3 and the theoretic results in [C3], the turn rate based variant (variant 2, Eq. (4.10)) was preferred for lower train speed with automotive grade IMUs and for all velocities with tactical grade IMUs. The measurement analysis with a MEMS IMU showed also the most repeatable signatures of the same switch way with the turn rate based variant (Eq. (4.10)) at two different train speeds. The average train speed at the second switch (55 km/h) is near the threshold of the lower speed definition in the theoretic analysis (50 km/h). Nevertheless, the theoretic analysis did not consider a limited accuracy of bias or gravity corrections, which is additionally unfortunate for the variant 3 with the lateral acceleration (see Eq. (4.19)). Therefore, the variant 2 (turn rate, Eq. (4.10)) is used in the following for the curvature measurements. The variant 3 (lateral acc., Eq. (4.10)) may be still considered as a valuable and additional signal for the switch-way discrimination for train speeds above 50 km/h.

Measurements of the Curvature Signatures

The three-dimensional curvatures, as defined in Eq. (4.16) - (4.18) are analyzed for repeatability. The curvatures of bank, slope and heading are calculated from bias corrected turn rates and the estimated speed:

$$\begin{pmatrix} c^{\phi} \\ c^{\theta} \\ c^{\psi} \end{pmatrix} = \begin{pmatrix} \omega_x - b^{\omega \mathrm{x}} \\ \omega_y - b^{\omega \mathrm{y}} \\ \omega_z - b^{\omega \mathrm{z}} \end{pmatrix} \frac{1}{\dot{s}}. \tag{4.20}$$

The previous presented curvature computation variant 2 (Eq. (4.10)) is contained in the last row (heading curvature). The biases $b^{\omega \mathrm{x}}$, $b^{\omega \mathrm{y}}$, and $b^{\omega \mathrm{z}}$ are either estimated from the INS/GNSS or observed at standstill, where the turn rates are assumed to be zero.

Figure 4.20 shows repeated measurements of three curvature signatures from three different runs over the same track route. The turn rates are measured from the MEMS IMU inside the cabin and the INS/GNSS estimates the gyroscope biases and the train speed. It can be seen, that the bank and heading curvature signatures show a repeatable signature over three different runs. The noise and other disturbances of the raw turn rate measurements of Fig. 4.15 could be removed with low-pass filters. The slope signature is with a maximum amplitude of 0.1 in units of 1/km very low compared to the other curvature signatures and the disturbances.

The curvature signatures from turn rates are not sensitive to drift, as the biases or errors do not sum up in integrations of the inertial mea-

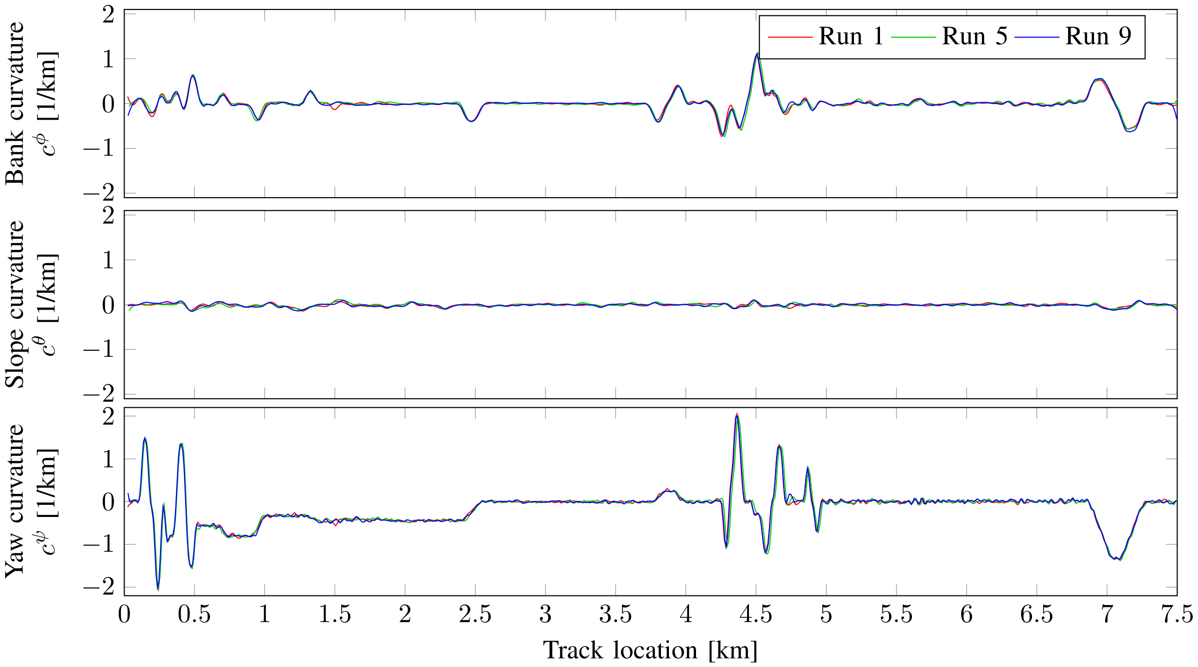


Figure 4.20: Repeated curvature signature measurements over track locations from three different train runs with a MEMS IMU inside the cabin. [Top] Bank curvatures, [Middle] slope curvatures, [Bottom] heading curvatures.

surements. This property is advantageous for the use of low-cost MEMS gyroscopes. Similar results with repeatable signatures could be achieved without the INS bias and speed estimates. The train speed was directly obtained from GPS. The gyroscope biases are calibrated during stand-still, when the vehicle turn rates are known to be zero. As described in the train experiment (see Section 7.1), there is a FOG IMU placed next to the MEMS IMU. A further comparison to curvature signatures from a FOG IMU showed also similar results in terms of repeatability of signatures. The slope curvature signatures from the FOG IMU showed slight improvements with a more repeatable pattern of the curvature signals over different runs. The advantage of a FOG IMU for the curvature measurements is less significant than for the inertial navigation system (INS). The repeatability of the curvature signals, especially the heading curvature for the switch-way identification, is similar for MEMS and FOG gyroscopes.

4.3 Alternative Measurements for Train Localization

This section presents further sensors and measurements for train localization. Two novel measurements with methods for train localization are presented: The passive magnetic measurement method and the vibration measurement method have been discovered from the train measurements in parallel to the work for the localization and mapping algorithms. However, in this thesis, these measurement methods are not used in the implementations of the train localization and mapping approaches. Nevertheless, the methods with magnetic and vibration measurements show very promising results.

4.3.1 Passive Magnetic Measurement Method

A passive magnetic field sensor measures the current magnetic field as it appears at the sensor. Passive means without an active field generation and is used to discriminate from inductive sensors. The magnetic field can be measured with inexpensive solid-state magnetometer sensors that are widely used in navigation equipment, electronic compasses and smartphones. Example magnetometers sensors are Hall-effect sensors, anisotropic magnetic resistive (AMR) sensors and fluxgate sensors.

The measured magnetic field at the sensor is a result of a superposition of one or more magnetic fields. Furthermore, the direction and density of a magnetic field can be changed in the vicinity of a ferromagnetic object. The railway environment contains many components from ferromagnetic material, such as steel rails, wheels, sign posts, catenary and other steel constructions (see Figure 4.3). The resulting magnetic field at the sensor is a superposition of the Earth field, ferromagnetic effects including magnetized components and electric currents:

$$\vec{B}_{\text{sensor}} = \vec{B}_{\text{Earth}} + \sum \vec{B}_{\text{ferromag.effects}} + \sum \vec{B}_{\text{currents}}. \tag{4.21}$$

In [P6], in a co-supervised master thesis [M4], and in [J2], it could be shown that the magnetic field measurements are related to train location and speed. The magnetic field shows a characteristic and location-dependent signature as well as periodic characteristics with a speed dependency. A measurement model for train localization with the magnetic field is defined in [J2]:

$$\vec{B}_{\text{sensor}} = \underbrace{\vec{B}_{\text{env.}}(id, s)}_{\substack{\text{track location}\\ \text{dependent}}} + \underbrace{\vec{B}_{\text{Earth}}(\psi)}_{\substack{\text{heading}\\ \text{dependent}}} + \underbrace{\vec{B}_{\text{wheel}}(v)}_{\substack{\text{speed}\\ \text{dependent}}} + \underbrace{\vec{B}_{\text{rest}}}_{\substack{\text{currents,}\\ \text{train, other}}} . \tag{4.22}$$

It could be shown that these terms are separable with signal filters, such as low-pass and band-pass filters in time and spatial domain [J2]. The results of the heading dependent, speed dependent and localization dependent are briefly summarized. The measurement analysis was carried out with the train measurements of Section 7.1 with a bogie and a cabin magnetometer.

The heading dependent part of Eq. (4.22) is known as the classical compass use-case. In [P6], a railway compass was analyzed from measurement data for two different measurements positions at the bogie and inside the train cabin. A calibration method was used in [P6] based on GPS heading measurements. It could be shown that the compass angle measurements are not very accurate with up to 59° of deviation for the cabin mounted sensor. Nevertheless, this accuracy is sufficient for a discrimination of the train-to-track frame direction. The bogie compass instead showed worse results and is not suitable for a rough orientation discrimination [P6].

The speed dependent part of Eq. (4.22) is called odometry and first results were presented in [P6]. Figure 4.21 shows magnetic signals in frequency domain of multiple train runs over train speed. This figure is created from a power spectral density (PSD) that is computed over a sequence of 512 samples from the magnetic measurements in combination with a speed measurement from GPS [P6].

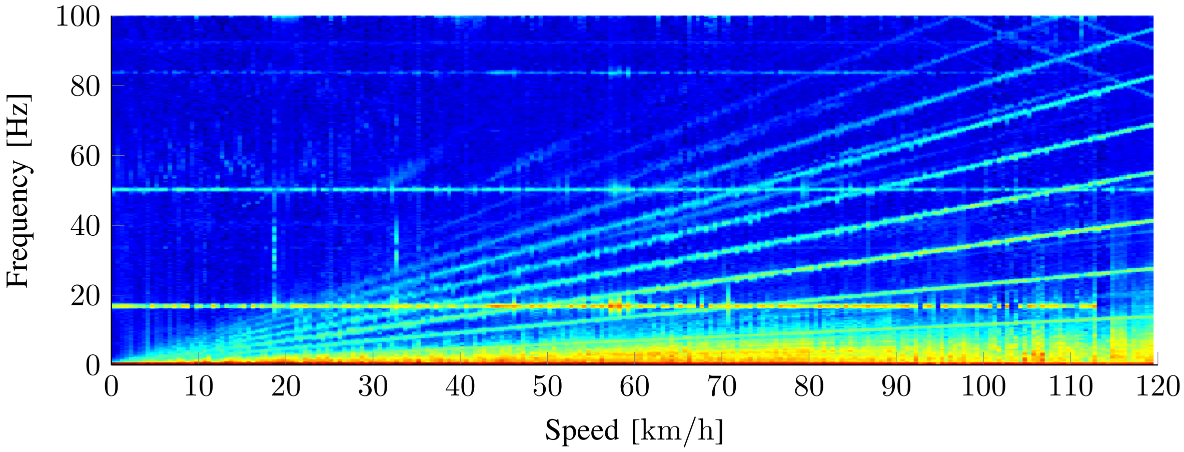


Figure 4.21: Magnetic frequency spectrum over train speed.

It can be seen, that multiple lines with different slopes arise from the origin. These lines correspond to harmonic frequencies that are generated from a turning wheel. The following model relates measured frequencies to the wheel diameter d, speed v and the harmonic n:

$$f_n = \frac{v}{d \cdot \pi} \cdot (n+1). \tag{4.23}$$

Two different methods for the speed measurement have been evaluated, a wheel turn method based on Eq. (4.23) and a speed signature method. The speed signature method incorporates a database over different speeds, each with frequency signatures and a corresponding speed. For the speed estimation, this method compares the current frequency signature with the prior known frequency signatures. The train speed is derived from the best matching frequency signature and its corresponding speed. The wheel turn method showed better results with 43.4 % below an error of 1 km/h, compared to the speed signature method with 21.9 % $<$ 1 km/h. The magnetic measurements are from a bogie mounted magnetometer and the speed reference is from GPS measurements. A concept for speed estimation with a Kalman filter in combination with acceleration measurements has been shown in [P7].

The third use-case is the location dependent part with the magnetic signatures of a track. A magnetic signature is a signal of the localization dependent distortions and magnetized components over the 1-D track location. This signature is generated from a filtered sequence of magnetic measurements that is resampled with equidistant samples over the track locations. A second filter in spatial domain mostly separates the location dependent signatures from other parts in Eq. (4.22). Fig. 4.22 shows magnetic signatures of three runs to the left switch way and three runs to the right switch way. First, it can be seen that on same along-track locations,

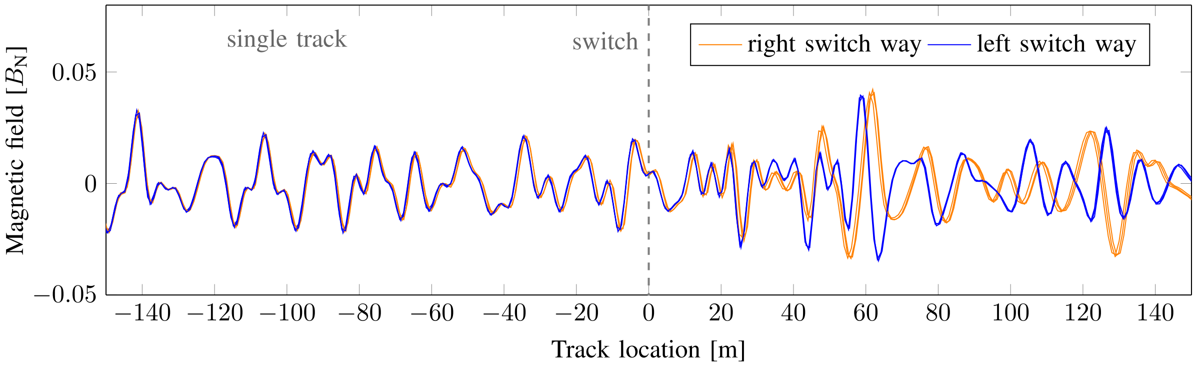


Figure 4.22: Magnetic signatures of three left and three right runs at a switch.

the signatures are very similar and at different locations, the signatures are also different and therefore distinguishable. It it possible to measure magnetic signatures also inside the train cabin. Compared to the bogie mounted sensor, the cabin sensor showed lower amplitudes. Nevertheless the repeatability of the signatures on same track locations and the separability of different locations and track were equally good: The evaluation results over 45.6 km with a 50 m signature were 97.8 % below an error of 4 m for a bogie mounted sensor and 97.4 % for a cabin mounted sensor [J2]. GPS position measurements were used as reference. Secondly, the figure shows different signatures for different switch ways. The measurement analysis has shown excellent switch-way identification characteristics: the signatures are repeatable for the same way and separable from different ways. Furthermore, it has been shown in [J2] that magnetic signatures of parallel tracks are distinguishable. A parallel track separation is a useful property in contrast to the inaccuracies of GNSS measurements, as shown in Section 4.1.3.

Figure 4.23 shows the magnetic signatures in a tunnel from five independent runs. The signatures show a high repeatability between the train runs. This repeatability proves the location dependency of the signatures. Hence, the magnetic signature method is suitable for a localization in tunnels. This is an important result, as a GNSS based train localization is not suitable in tunnels.

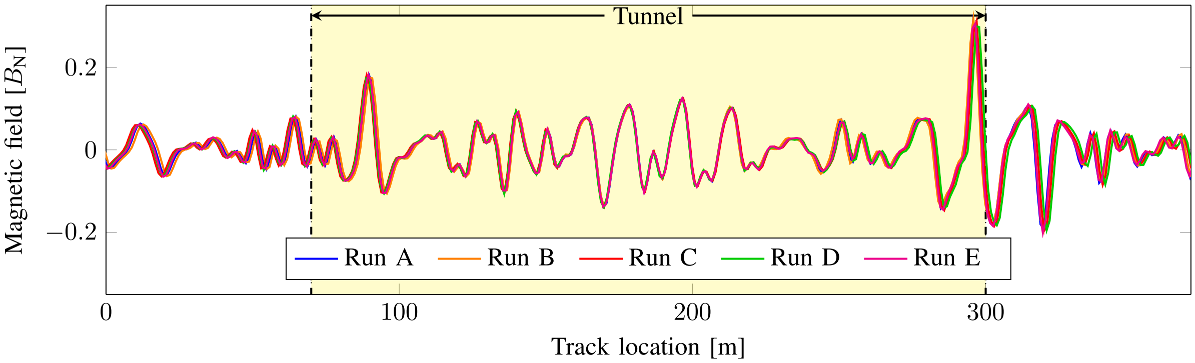


Figure 4.23: Magnetic signatures of a 230 m long railway tunnel.

In summary, the passive magnetic measurements contribute to the train localization estimation in all four ways: to relative along-track estimation with the speed measurements, to absolute along-track estimation, to a track identification at switches and also to an absolute track identification at parallel tracks. The passive magnetic measurement method has a high potential to increase accuracy and add redundancy to a GNSS based train localization approach. Moreover this passive magnetic approach is also suitable in GNSS denied environments and especially for tunnels.

4.3.2 Vibration Measurement Method

Vibrations are alternating accelerations and turn rates that can be measured with inertial sensors. This vibration method extends the presented methods with the IMU. The vibration method is independent from a slowly changing drift. Therefore, inexpensive MEMS sensors or piezoelectric acceleration sensors are suitable. One advantage is, that the vibration method requires no extra sensor hardware if an IMU is already part of the navigation hardware. The spectrograms of accelerations (Fig. 4.14) and turn rates (Fig. 4.15) visualize the vibrations with specific frequency components over time. Similar to the passive magnetic field sensor, a train speed and location measurement is possible from speed dependent and location dependent signatures. The vibrations of interest arise while the train is moving from exited wheels, the engine and from the track. First results have been presented in [P5], in a co-supervised master thesis [M4] and in [P7]. A general measurement model for a train localization with vibrations can be defined with the IMU measurements of acceleration $\vec{a}$ and turn rate $\vec{\omega}$:

$$\vec{a} = \underbrace{\vec{a}_{\text{tb}}}_{\substack{\text{train}\\ \text{traction/brake}}} + \underbrace{\vec{a}_{\text{geo.}}(id, s, v)}_{\substack{\text{track geometry}\\ \text{location dependent}}} + \overbrace{\underbrace{\vec{a}_{\text{vib.}}(id, s)}_{\substack{\text{track vibrations}\\ \text{location dependent}}} + \underbrace{\vec{a}_{\text{vib}}(v)}_{\substack{\text{speed}\\ \text{dependent}}}}^{\text{vibration}} + \underbrace{\vec{a}_{\text{rest}}}_{\substack{\text{other vib.}\\ \text{Coriolis}}}, \tag{4.24}$$

$$\vec{\omega} = \underbrace{\vec{\omega}_{\text{geo}}(id, s, v)}_{\substack{\text{track geometry}\\ \text{location dependent}}} + \overbrace{\underbrace{\vec{\omega}_{\text{vib.}}(id, s)}_{\substack{\text{track location}\\ \text{dependent}}} + \underbrace{\vec{\omega}_{\text{vib}}(v)}_{\substack{\text{speed}\\ \text{dependent}}}}^{\text{vibration}} + \underbrace{\vec{\omega}_{\text{rest}}}_{\substack{\text{other vibrations,}\\ \text{Earth rate}}} . \tag{4.25}$$

This model contains also the accelerations and turn rates from the track geometry and train motion, as defined in Eq. (4.13) - Eq. (4.15) and Eq. (4.16) - Eq. (4.18). The location dependent accelerations from track geometry contain the gravity portions depending on the track attitude as well as the centripetal accelerations from curve runs. In Section 4.2.4, the train dynamics have been analyzed with frequencies of 1 Hz and below. Therefore, the vibrations can be separated with a high-pass filter.

The measurement analysis was carried out from measurements of a bogie mounted and a cabin mounted MEMS-IMU. Figure 4.24 shows z-axis acceleration in frequency domain of multiple train runs over train speed. This figure is created from a power spectral density (PSD) from acceleration measurements. The PSDs are sorted according to the corresponding speed measurement of a GPS. There are visible lines in Figure 4.24 that point to the origin. These lines arise from periodic excitations of turning

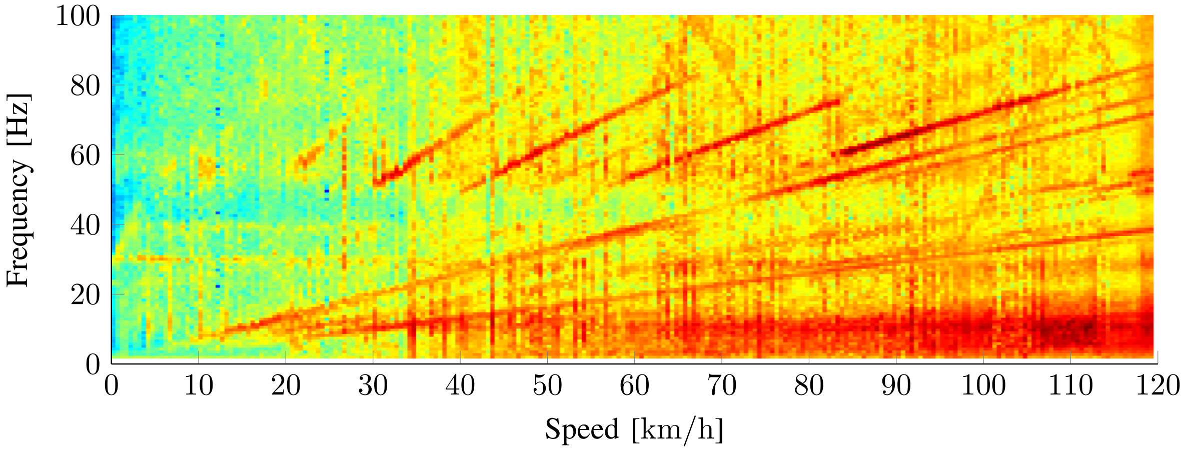


Figure 4.24: Vibrations: Frequency spectrum of z-axis acceleration over train speed.

wheels, gear-box or other speed dependent vibrations of the train. Above 50 Hz, there is a saw-tooth signal visible when the train is accelerating that is caused by the diesel engine and the gear shifts.

The train speed and the location measurement with vibrations are summarized in the following. The train speed observations have been analyzed and presented in [P5], [M4], and in [P7]. A speed signature method has been used to observe the speed from vibrations. Same as for the magnetic speed analysis, the speed signature method finds a speed with a comparison of prior known frequency signatures with associated speeds. The best results for the bogie sensor are 46.1 % below a speed error of 1 km/h with the acceleration y-axis. The speed observations are further combined in a Kalman filter with the train acceleration measurements and an outlier rejection [M4], [P7]. In [P7], an IMU-only odometry has been presented with a bounded speed error as shown in Figure 4.25. The speed observations from vibrations show often large deviations: the red crosses indicate the rejected speed observations while the green circles indicate accepted speed observations from vibrations.

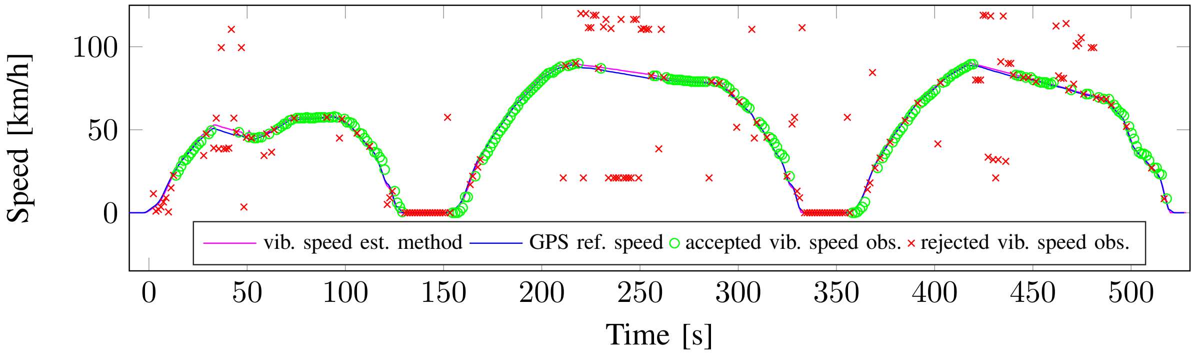


Figure 4.25: Speed estimation with acceleration and vibration measurements [P7].

The location dependent vibration signatures are used for the train localization. A vibration signature is a normalized vibration intensity over the 1-D track location. Therefore, the high-pass filtered acceleration or turn rate signal is transformed into spatial domain with speed information. The result is a resampled signal with constant distances over the 1-D track locations, called signature. Afterward, the signal envelope is computed that represents a vibration intensity over 1-D track locations. Finally the signature is normalized with the quadratic speed. This ensures same amplitudes of localization dependent vibration features when the train runs with different speeds over the same track. Figure 4.26 shows the vertical acceleration vibrations of a bogie mounted IMU from six runs over a railway switch. It can be seen that the signature has repeatable patterns on the single track, and mostly separable patterns after the switch start according to the switch way. The along-track error was analyzed over 59.7 km with a search window of 200 m. The bogie sensor showed an error below 4 m in 89 % of the cases while the cabin sensor showed an error below 4 m in 74 % of the cases. Initial results of a switch-way analysis showed a successful switch-way estimation with all six inertial signals for the bogie sensor, and with five inertial signals for the cabin sensor. The bogie sensor performs better than the cabin sensor in the localization dependent vibrations.

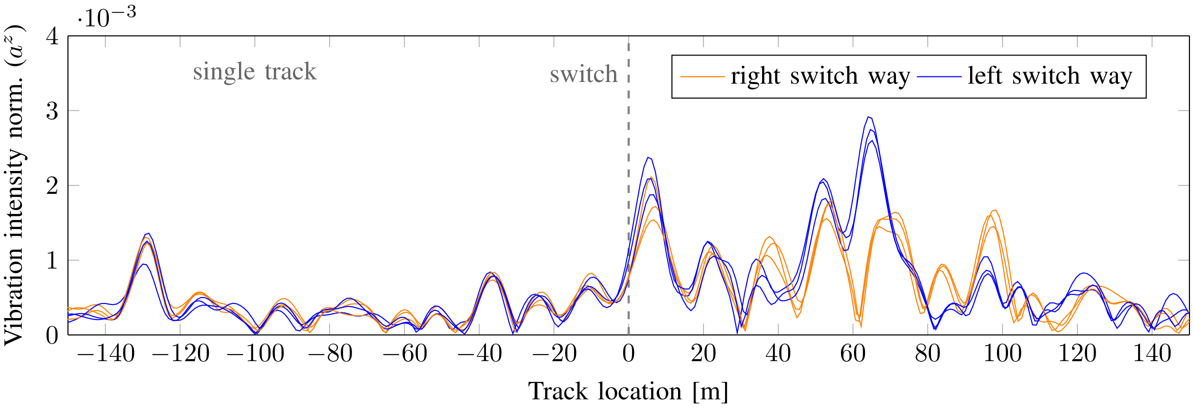


Figure 4.26: Vibration signatures of three left and three right runs at a switch.

The vibration method in combination with IMU methods presented in Section 4.2 are very beneficial for a train localization in terms of relative and absolute along-track localization as well as a track identification, especially at switches.

4.3.3 Overview of Onboard Measurements for Train Localization

The proposed methods for train localization and track mapping can be extended with additional sensors. Beside GNSS and IMU, there are various other approaches and scientific examinations with different sensors for train localization. Multiple and independent sensor measurements can be used to increase reliability and availability of a train localization. Furthermore, different combinations of sensors are advantageous for the along-track accuracy and the correct track identification.

Train mounted sensors and systems	**Along-track** relative displacement, absolute along-track	**Track-identification** switch-way discrimination parallel tracks discrimination
GNSS receiver	speed, positions, motion heading	switch: position, heading parallel tracks: position
Inertial sensor, IMU	train accel., curvature & attitude	switch: curvature & attitude signatures
Passive magnetic signature sensor	speed: periodic signature, magnetic signature	discriminative magnetic signature
Vibration signature sensor	speed: periodic signature, vibrational signature	discriminative vibrational signature
Train speed sensors wheel turns, Doppler radar	speed, distance integration	–
Imaging sensors camera, radar, lidar	speed: image displacement, image or feature signature	image or feature signature of switch elements, other tracks
Inductive sensor	speed: periodic signature	signature of switch-way elements
Add. infrastructure balise, cable loop, beacon	fixed positions of balise, cable crossing, beacon	transmission of IDs to train on every track

Table 4.3: Overview of train mounted sensors and usability for train localization.

Table 4.3 lists the usability of different sensor types for an onboard train localization. The first two, GNSS and IMU, are used in the implementations in Chapter 6. The measurement methods with magnetic signatures and vibration signatures are described in this thesis and suggested for further use (Section 8.1.5). The general train speed sensors, imaging sensor and inductive sensors are briefly described in the following. For completeness, the last row includes train mounted sensors, that require additional infrastructure. The usability and contribution to the train localization is shown in the categories of along-track and track identification (see Section 3.2.3). A contribution to along-track estimation can be achieved with a relative displacement estimate from speed or with an absolute along-track feature that identifies a 1-D location on a track. The track identification can be archived while passing a switch and measuring competitive track features of a particular switch way. Alternatively, a measurement may contribute to a track identification that directly discriminates a track without a switch.

Train Speed Sensors

Train speed sensors are usually installed on trains for speed and distance estimation. Current train protection systems monitor the speed and traveled distance near railway signals and block section boundaries. This sensor contributes to relative displacement estimation of a train localization. In general, a wheel turn sensor counts the revolutions of a train wheel. One drawback is the wheel slip when the train is accelerating or decelerating. This wheel slip causes erroneous speed measurements and distance integrations [72]. Another type of speed sensor is a Doppler radar mounted below a train [73, 74]. The advantage is a slip-free measurement of speed for the estimation of traveled distance.

Imaging Sensors

Imaging sensors are considered widely here with cameras, lidar and also synthetic aperture radar (SAR). Some of these sensors are able to generate an image with depth information, such as stereo cameras, lidar and radar. The imaging sensors are exteroceptive sensors and measure characteristics in the railway environment. A further signal processing detects and classifies specific objects and features. The train speed or displacement may be measured from changes of consecutive images or scans. A positioning is achieved with a comparison of measured features and map features (see [13]). For the railway domain, a switch-way detector or a parallel track detector can discriminate and detect the correct track. Examples of vision based railway track and switch detection can be found in [23, 24]. Lidar has been used for railway localization with track and switch detection in [26, 75]. As a disadvantage, optical sensors have usually a decreased performance with dirt, dust, rough weather, and direct or reflected sunlight. Current research in imaging with SAR is conducted extensively for the automotive domain in a front facing direction (e.g. [76]) as well as in a ground facing direction [77], also known as ground penetration radar. Nevertheless, imaging sensors can provide useful information for clearance monitoring and obstacle detection.

Inductive Sensors

An inductive sensor measures primarily a change of inductance of a coil. Among other parameters, the inductance depends on conductive material within the magnetic field of a coil. Railway tracks, fastenings and switches are made of the conductive material steel. Comprehensive research has been conducted with the "Eddy-Current Sensor" (ECS) for train localization in [21, 78, 79]. This ECS sensor contains one coil for the generation of a modulated magnetic field and two other coils for a differential measurement of the resulting field affected from metallic structures within the sensing range. It senses changes in the metallic structure of the tracks while the train and the sensor move over the tracks. Sleeper fastenings can be detected and the train speed is derived form continuously passed sleepers with known or calibrated distance. Furthermore, a train run over a certain switch way is characterized by specific switch elements. Classification methods can discriminate switch parts and detect a switch and also the switch way a train has taken [21]. In [80], a difference inductance sensor is defined for speed and displacement measurement. This sensor evaluates the inductance measurements of two separated coils in along-track direction.

Infrastructure based Train Sensors

Infrastructure based train sensors consist of components on the train and in the railway infrastructure. These infrastructure components are artificial landmarks near or on the tracks, which can be detected by a train mounted unit. Examples are beacons, markers, cable loops, and balises. The Eurobalise, for example, is used for train localization in ETCS [14, 81]: On the train side, there is a balise transmission module (BTM) with an antenna at the undercarriage. During a passage over a track-mounted balise, a data telegram with location information is sent from the balise to the train. For track-selective train localization, an appropriate placing of the balises at switches and parallel tracks is necessary.

Chapter 5

Probabilistic Localization and Mapping for Railways

The following presents the theoretic part for the probabilistic train localization and track mapping methods. Section 5.1 presents a general dynamic Bayesian network (DBN) for railway navigation and a realized DBN with GNSS and IMU measurements. The latter DBN is used in the factorization of the train localization posteriors (Section 5.2) and of the RailSLAM posterior (Section 5.3). The posterior distributions can be seen as general estimation problem definitions and the factorization divides the estimation problem into smaller parts. The further estimation implementations are based on these factorized posterior distributions.

5.1 Bayesian Theory Applied to Railways

The generic DBN of a train equipped with onboard sensors is shown in Fig. 5.1 for two time steps. This DBN extends the presented DBN in [P4]. At first, a train state T_k is defined as a random variable at the time step k. This train state contains the train related physical quantities, such as position, attitude, and velocity. The train control input U is the train driver control of traction and brakes. The railway environment consists of time invariant railway tracks, denoted with M, and a network control input V. This network control comprises the control of signals and switches that influence the train trajectory. Several onboard sensors are mounted on the train and measure physical quantities of the train and the environment. These measurements are differentiated into intrinsic measurements

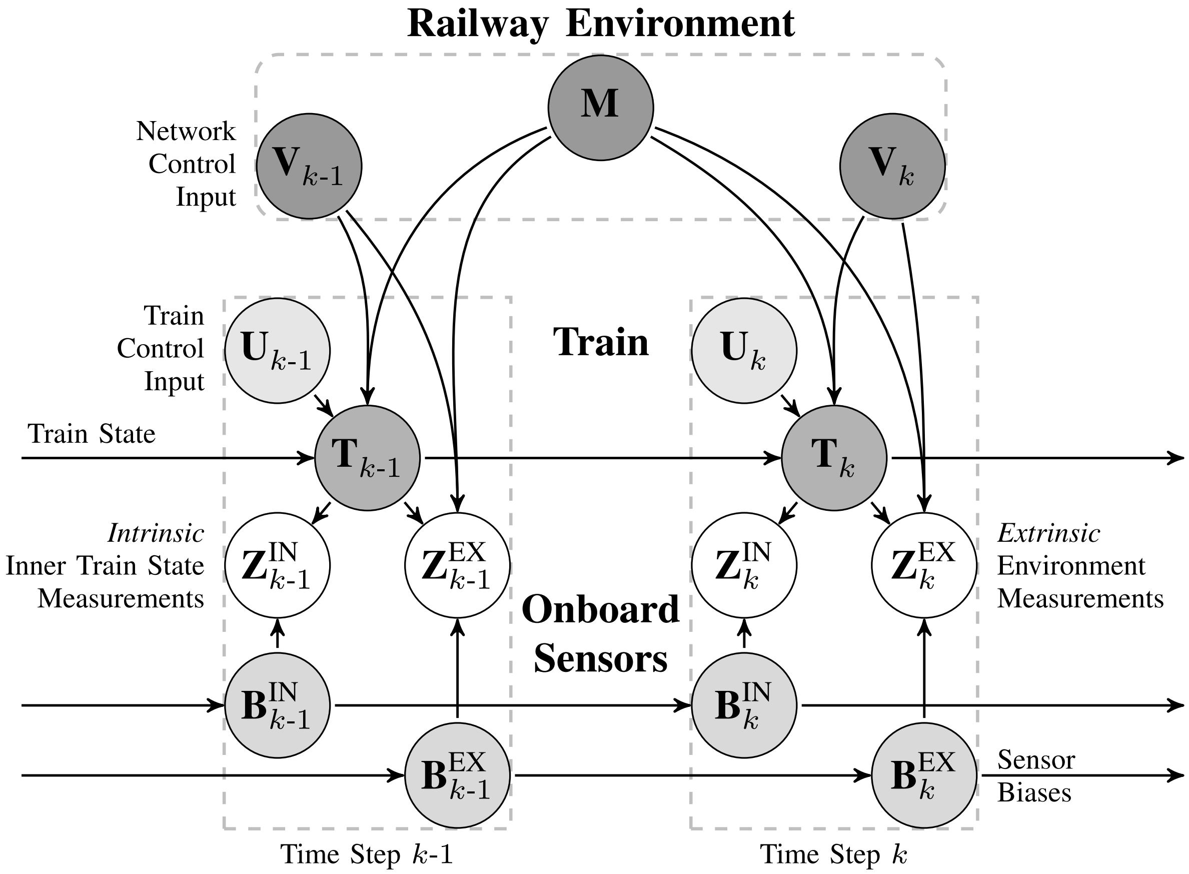


Figure 5.1: General DBN for railway navigation.

Z^{IN} and extrinsic Z^{EX} measurements. The extrinsic sensors measure a characteristic of the surrounding environment apart from the train. An intrinsic sensor measures characteristics related solely to the train such as attitude or position. The measurement biases of the onboard sensors are denoted with B^{EX} and B^{IN} respectively.

This generic DBN shows the dependencies between the random variables with arrows: The train state depends strongly on the railway tracks M due to the constraint train kinematics that is caused by the geometry of the railway tracks. The train states are also conditional dependent on its predecessor because of the inertia. A train has a significant mass and a change of the velocity is limited by the wheel-rail force transmission and the engine power. Additionally, the railway tracks are designed to cause limited and slow changing dynamics in terms of jerk, accelerations and turn-rates for safety reasons. The train control input U and network control input V are defined conditionally independent from its predecessor. This conditional independence means that the input can change randomly between time steps. The intrinsic sensor measurements depend on the train state and there is only an indirect dependency of the railway tracks from the constrained train states. For example, a gyroscope measures a turn rate of the train, which depends on the train velocity and the curvature of the railway track. For the extrinsic measurements, there is a direct dependency of the environment and the train states. A train mounted camera, for example, measures a distant feature in the railway environment as well as the camera's perspective depends on the position and attitude of the train. Measurements Z have no dependency over time, but the correlated sensor errors, such as the biases B, are dependent over time.

This DBN is also a hidden Markov model (HMM) [42, 82], as only the hidden states (T, B) posses a time transition and depend only on their last time step (Markov condition). In contrast, the sensor measurements are independent from the last measurement.

The implementations in this thesis consider a simplified DBN: First, only intrinsic measurements from GNSS and IMU are used. Second, only biases for the IMU are estimated. The railway environment contains railway tracks with a time invariant geometry. The network control that controls the switches is combined with the train control input in the control input U. The special DBN is shown in Fig. 5.2, and it represents the causal dependencies for the localization and mapping estimations in the following.

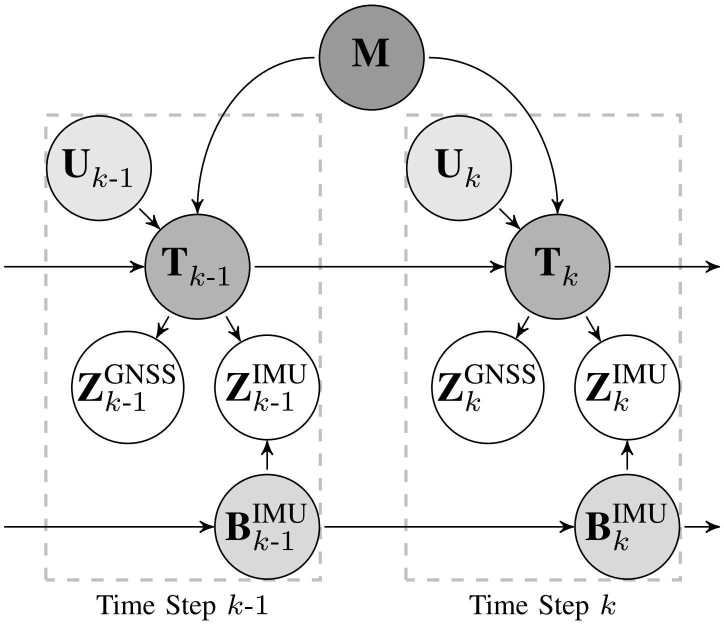

$\mathbf{M}$
$\mathbf{U}_{k\text{-}1}$
$\mathbf{U}_{k}$
$\mathbf{T}_{k\text{-}1}$
$\mathbf{T}_{k}$
$\mathbf{Z}^{\text{GNSS}}_{k\text{-}1}$
$\mathbf{Z}^{\text{IMU}}_{k\text{-}1}$
$\mathbf{Z}^{\text{GNSS}}_{k}$
$\mathbf{Z}^{\text{IMU}}_{k}$
$\mathbf{B}^{\text{IMU}}_{k\text{-}1}$
$\mathbf{B}^{\text{IMU}}_{k}$
Time Step k-1
Time Step k

Figure 5.2: DBN for railway navigation with IMU and GNSS.

5.2 Train Localization

5.2.1 Localization Estimation with Onboard Sensors

Train localization is the procedure to estimate the topological location in the railway network. This location is estimated by means of a track map and onboard sensor measurements.

Estimation State

The goal for a train localization algorithm is to estimate and resolve T_k and the sensor biases B_k. The train location is the topological pose T^{topo} by track ID, track location and direction, as defined in Eq. (3.18). Further, the train speed v is of interest and estimated in the train localization. In the following posterior factorizations, T^v will be used as the random variable for train speed in order to assign the affiliation to the train state. These values are combined in the train localization state vector, or simply, train state T. The basic train state for one discrete time step k is:

$$T'_k = \{\underbrace{id, s, o^{\text{t}}}_{T^{\text{topo}}}, \underbrace{v}_{T^{\text{v}}}\}_k. \tag{5.1}$$

This basic train state is augmented with random variables from the train trajectory:

$$T_k = \{T^{\text{topo}}, T^v, T^{\text{traj}}\}_k. \tag{5.2}$$

The trajectory describes the path of the train over time. The values of the trajectory depend on the track map content and the available sensor measurements. A trajectory for a measurement setup with a GNSS receiver contains the geographic position:

$$T^{\text{traj}'} = \{\varphi, \lambda\}. \tag{5.3}$$

A 2-D extension with heading and curvature results in:

$$T^{\text{traj}''} = \{\varphi, \lambda, \psi, c^{\psi}\}. \tag{5.4}$$

A full 3-D trajectory with curvatures consists of nine variables:

$$T^{\text{traj}'''} = \{\varphi, \lambda, h, \phi, \theta, \psi, c^{\phi}, c^{\theta}, c^{\psi}\}. \tag{5.5}$$

The estimation states of a train localization is formulated with the random variables and summarized with continuity properties in Table 5.1.

The track ID is a discrete variable, and the track location is continuous within the track. At a track change is a discontinuity of the track location s. As explained in 3.2.2, the direction o^{t} is an auxiliary state that is unknown at the beginning, but its transition is deterministic with the use of the track map. Depending on the onboard sensors, there are unknown measurement biases b. These biases b can change over time due to a random drift, which can not be calibrated in advance.

State variable	**Symbol**	**Continuity**
track ID	id	discrete ID, changes at track change
track location	s	continuous within a track, discontinuous at track change
train direction	o^{t}	binary, can change at track change
train speed	v	continuous
train trajectory	$\varphi, \lambda, h, \phi, \theta, \psi, c^{\phi}, c^{\theta}, c^{\psi}$	continuous
sensor biases	b	continuous

Table 5.1: Estimation states for train localization.

Train State Estimation and Challenges

The initial estimate for train localization is an unknown location and train direction. Trains may be parked, shut down, and reactivated on side tracks with parallel tracks in the vicinity. This fact may cause ambiguities in terms of the correct track and train direction. On the other hand, this initial uncertainty will be resolved shortly after moving and can be further avoided with a permanent power supply or a storage of the last estimate before shutdown.

The discontinuities of certain state values in Table 5.1 arise from the track definitions and can be handled by the track map transition function (Eq. (6.5)). The transition of the location is mostly one-dimensional with one outcome. A transition over a switch passed facing results in two possible locations.

The actual ambiguities and uncertainties of train localization arise from missing or inaccurate measurements of absolute along-track locations and measurements for a track identification. Sensor measurements are disturbed by random noise, biases, and unresolved systematic errors. The train location accuracy suffers from low signal-to-noise ratios in terms of track feature measurements and ambiguities of parallel track features for an absolute track identification.

A dead-reckoning of the train location with relative along-track measurements result in a growing position uncertainty and error. This uncertainty and error is not only limited to along-track: As soon as switches are passed, this uncertainty has an impact on the correct assignment of measurements to a switch way and can cause an incorrect track identification. A lack of an absolute track identification measurement can be compensated with a location tracking, an adequate estimation of the along-track location and an identification of the switch way. Nevertheless, an absolute track identification reduces uncertainty, especially for the initial localization.

5.2.2 Localization Posterior Factorizations

The localization posterior estimates the train state T and the sensor biases B from the measurements Z, the control input U, and the map M. The DBN of Fig. 5.2 visualizes the random variables and their dependencies for the posterior factorizations. A factorization splits the posterior into factors, which are feasible for computation. In this thesis, a particle filter and a multi hypothesis tracker (MHT) are implemented. There are two posteriors: The first posterior is for the particle filter and estimates all train states $T_{0:k}$ and biases $B_{0:k}$. The second is the filter posterior, which estimates only the current states of T_k and B_k.

Localization Full Posterior for Particle Filter

The *full posterior* of train localization with GNSS and IMU measurements is factorized in the Appendix B.3 and consists of:

$$\begin{aligned}
\mathrm{p}(T_{0:k}, B_{0:k} | Z_{1:k}, U_{1:k}, M) \propto\, & \underbrace{\mathrm{p}(Z_k^{\mathrm{GNSS}} | T_k) \cdot \mathrm{p}(Z_k^{\mathrm{IMU}} | T_k, B_k)}_{\text{measurements}} \cdot \underbrace{\mathrm{p}(B_k | B_{k-1})}_{\text{bias transition}} \cdot \\
& \underbrace{\mathrm{p}(T_k^n | T_k^l, T_{k\text{-}1}^n, U_k, M)}_{\text{non-linear train transition on tracks}} \cdot \underbrace{\mathrm{p}(T_k^l | T_{k\text{-}1}^l, U_k)}_{\text{linear train transition}} \\
& \underbrace{\mathrm{p}(T_{0:k\text{-}1}, B_{0:k\text{-}1} | Z_{1:k\text{-}1}, U_{1:k\text{-}1}, M)}_{\text{prior}} .
\end{aligned} \tag{5.6}$$

The different parts can be assigned to specific functions and computations. The train transition consists of two parts: a linear train transition for the linear train states and a non-linear part for the non-linear train states.

The linear part is the velocity and distance estimate, while the non-linear part contains the track map. This posterior is used in the particle filter implementation in Section 6.2.3.

Localization Filter Posterior for MHT Filter

The localization *filter posterior* is factorized in the Appendix B.4 and consists of:

$$\begin{aligned}\mathrm{p}(T_k, B_k|Z_{1:k}, U_{1:k}, M) \propto\ & \underbrace{\mathrm{p}(Z_k|T_k, B_k)}_{\text{measurement update}} \int \underbrace{\mathrm{p}(T^n_k|T^l_k, T^n_{k\text{-}1}, U_k, M)}_{\text{non-linear train transition on tracks}} \cdot \\ & \underbrace{\mathrm{p}(T^l_k|T^l_{k\text{-}1}, U_k)}_{\text{linear train transition}} \cdot \underbrace{\mathrm{p}(B_k|B_{k\text{-}1})}_{\text{bias transition}} \cdot \underbrace{\mathrm{p}(T_{k\text{-}1}, B_{k\text{-}1}|Z_{1:k\text{-}1}, U_{1:k\text{-}1}, M)}_{\text{prior}}\ d(T_{k\text{-}1}, B_{k\text{-}1}).\end{aligned} \tag{5.7}$$

This posterior is used in the MHT implementation in Section 6.2.4.

5.2.3 Pre-processed Measurements

The pre-processed measurements Z^{P} contain position, velocity, attitude, and curvature measurements and is defined with:

$$Z^{\mathrm{P}} = (\underbrace{\varphi, \lambda, h,}_{\text{position}} \underbrace{v,}_{\text{velocity}} \underbrace{\phi, \theta, \psi,}_{\text{attitude}} \underbrace{c^{\phi}, c^{\theta}, c^{\psi}}_{\text{curvatures}}). \tag{5.8}$$

These measurements are actually estimates from a sensor fusion of GNSS and IMU measurements. Nevertheless, the pre-processed measurements will be handled as measurements of the train trajectory and also the track geometry in the following. In order to distinguish from sensor measurements, these measurements are named pre-processed measurements. The filter posterior describes the estimation of the pre-processed measurements Z^{P}_k and also the IMU biases B_k of the acceleration and turn rate measurements:

$$\begin{aligned}\mathrm{p}(Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k|Z^{\mathrm{GNSS}}_{1:k}, Z^{\mathrm{IMU}}_{1:k}) \propto\ & \underbrace{\mathrm{p}(Z^{\mathrm{GNSS}}_k|Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k)}_{\text{GNSS measurement update}} \cdot \\ & \int \underbrace{\mathrm{p}(Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k|Z^{\mathrm{P}}_{k\text{-}1}, B^{\mathrm{IMU}}_{k\text{-}1}, Z^{\mathrm{IMU}}_k)}_{\text{estimate with IMU input}} \cdot \underbrace{\mathrm{p}(Z^{\mathrm{P}}_{k\text{-}1}, B^{\mathrm{IMU}}_{k\text{-}1}|Z^{\mathrm{GNSS}}_{1:k\text{-}1}, Z^{\mathrm{IMU}}_{1:k\text{-}1})}_{\text{prior}}\ d(Z^{\mathrm{P}}_{k\text{-}1}, B^{\mathrm{IMU}}_{k\text{-}1}).\end{aligned} \tag{5.9}$$

The factorization is presented in the Appendix B.5. The IMU measurements are used as input in this estimation. A GNSS measurement updates and corrects the estimate. The pre-processed measurements will be used in the MHT train localization and in the RailSLAM method. The particle filter instead uses directly the IMU and GNSS measurements.

5.3 RailSLAM Simultaneous Localization and Mapping for Railways

The simultaneous localization and mapping for railways (RailSLAM) is an automated track map generation and track map enhancement method. This method uses measurements from onboard sensors and creates either new track maps, or augments, corrects and enhances existing tracks maps instantaneously during the train run.

5.3.1 Learning of the Track Map

The track map is created and enhanced during train runs. There are two main states for the mapping process: the *white-space* phase and the *prior-map* phase. These mapping phases or mapping states are named phases in order to distinguish from the estimation states with random variables. The mapping phases are denoted with ξ and used as auxiliary values in a state machine of the RailSLAM algorithm.

White space means that there is no information known about the tracks, in analogy to white and empty paper maps. In this phase, new tracks are estimated from measurements of the onboard sensors.

In the prior-map phase, the train is localized on the previous track map with measurements and the track geometry of the track map is updated. In this way, the track map with its features is enhanced by every revisit and the uncertainty of the track geometry estimate is reduced. The prior-map phase is valid as long as the train stays on known tracks. These known tracks are either previously visited or a preliminary track map has been used. A preliminary track map can be augmented if it contains less information about tracks and track geometry before.

Figure 5.3 explains the mapping phases and transitions with a typical example of a two-way passenger train operation between the stations A and B. The train starts at station A and travels towards station B. In the beginning, there are no tracks known and the track map is newly created in the white-space phase. In Figure 5.3(b), the train departs from B back to station A. The train reverses on the previous mapped track, and the mapping process is now in the prior-map phase. At this time, the track map contains one track. Railway lines contain often two tracks with dedicated travel directions. After leaving the platform track, the train branches to the reverse track of the railway line. At this branch, a switch is observed from the trajectory of the train and inserted in the track map. This switch splits the current track in two tracks and a new track is created. The train runs now again on an unknown track and the mapping process is in the white-space phase. In Figure 5.3(d), the train merges to the known track and a second switch is observed. This switch splits the known track in two tracks and links all connected tracks in the switch. The mapping process continues in the prior-map phase and the track map contains now four tracks and two switches.

RailSLAM State Machine for the Mapping Phases

Figure 5.4 shows the state machine for RailSLAM with the two main mapping phases and five transitions. Within the boxes of white-space phase and prior-map phase there are the elementary steps shown for the map processing. In RailSLAM there are also multiple hypothesis on different switch ways, tracks or new tracks and switches possible. Each hypothe-

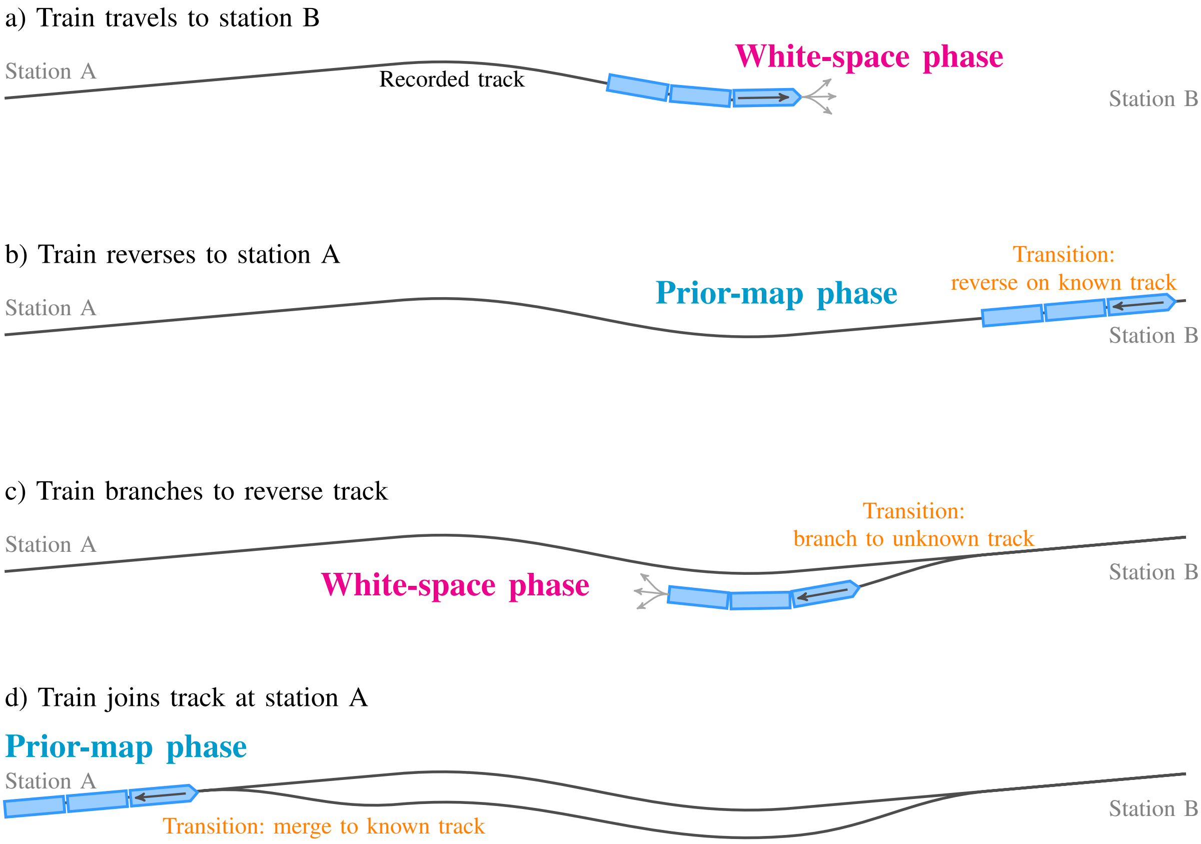


Figure 5.3: A typical learning sequence of a track network with train runs.

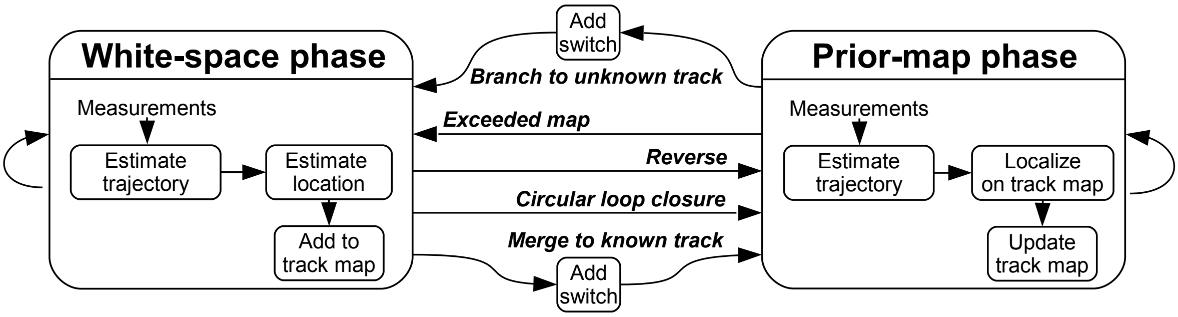


Figure 5.4: State machine of RailSLAM mapping phases with corresponding track map processing.

sis contains this state machine and mapping phases and processes. The conditions for a transition are checked and decided before or within the localization part. In a probabilistic algorithm, these decisions can be random or result in multiple hypothesis. The transitions of the mapping phases can be derived from the different scenarios:

Reversing train: The reversing train is depicted in Fig. 5.3(b). There, the train changes the travel direction and the mapping phase changes from white-space phase to prior-map phase.

Branch via switch: Fig. 5.3(c) shows the transition from prior-map phase to white-space phase after branching from a known track with a new switch. The switch is passed facing.

Merge via switch: The returning train in Figure 5.3(d) merges with a new switch to the known track and the mapping phase changes from white-space phase to prior-map phase.

Exceeded map: This transition happens when a train exceeds the known track map. The train travels further than the track map is known and there no new switch involved. The mapping phase changes from prior-map phase to white-space phase.

Circular loop closure: In a circular loop closure, the train returns to known tracks without a new switch. One example are circular networks, where the train arrives at its initial location without changing travel direction. The mapping phase changes from white-space phase to prior-map phase.

5.3.2 RailSLAM Posterior Factorization

The presented version of RailSLAM estimates the *online* SLAM posterior with intrinsic measurements from a GNSS and an IMU. A *full* SLAM posterior with particle filter has been presented in [P4] and implemented in [M3].

The RailSLAM version of this thesis contains additionally a delay for the track map update. This delay allows some time to identify tracks and new switches over certain time steps without generating multiple maps. Some time after a passed switch or a newly estimated switch, the number of multiple train state hypotheses reduces again and most of the multiple maps become obsolete. This delayed mapping is possible if solely intrinsic measurements are used and the train does not revisit the same track locations within this delay period. In the implementation, the mapping will be delayed until the train stops again. Other possibilities of delays are a fixed time lag or a fixed distance. The time step of the last track map update is denoted with k'.

The main factorization for the online SLAM posterior results in a mapping part and a train state estimation part:

$$\mathrm{p}(T_k, B_k, M_k | Z_{1:k}, U_{1:k}) \overset{\substack{\text{prod. rule} \\ \text{cond. indep.}}}{=} \underbrace{\mathrm{p}(M_k | T_k, Z_{1:k})}_{\text{track map estimation}} \cdot \underbrace{\mathrm{p}(T_k, B_k | Z_{1:k}, U_{1:k})}_{\text{train state estimation}} . \tag{5.10}$$

This posterior factorization is for both mapping cases: white-space and prior map. The train state estimation is different for each mapping case and will be explained in the following. The track map estimate M is now indexed with a time step k in order to separate the different times of previous track map update and actual time step. The train state contains the topological pose T^{topo}, train speed v and the train trajectory values T^{traj}, as defined in Eq. (5.2). The train location and track map are estimated from the pre-processed measurements.

The first factor in Eq. (5.10) is the track map estimation. There, the track map is updated with information from the train state T_k. As a specialty of RailSLAM, the train trajectory is also the estimate of the track geometry. The topological pose of T_k is the data association of the track geometry for the track map update. This means that the train state contains all information for an incremental track map update. As long as the train runs on different track locations, the current train state does not depend on a track map that includes the last updates. Therefore, a series of consecutive train state estimates can be temporarily buffered for a collective track map update. The track map estimation is delayed and computed some time after a series of train state estimations. The actual data fusion of prior track map information and measurements is processed in the train state estimation.

The second factor in Eq. (5.10) is the train state estimation. In the white-space case, the trajectory is estimated from the pre-processed measurements that contain measurements of the trajectory alone. The white-space train state estimation is similar to Eq. (5.9) and the track location is updated from location increments. In the prior-map case, the train state estimation of Eq. (5.10) is more complex: Now, the train state trajectory is estimated from measurements and geometry data of the previous map. This previous map is missing in the train state estimation of Eq. (5.10). Therefore, the train state estimation is extended and marginalized (see Eq. (A.15)) with the previous map $M_{k'}$ from a past time step k':

$$\underbrace{\mathrm{p}(T_k, B_k | Z_{1:k}, U_{1:k})}_{\text{train state estimation}} \overset{\substack{\text{extension,}\\ \text{margin. } M_{k'}}}{=} \int \underbrace{\mathrm{p}(T_k, B_k | Z_{1:k}, U_{1:k}, M_{k'})}_{\substack{\text{localization and trajectory}\\ \text{estimation (Eq. (5.12))}}} \cdot \underbrace{\mathrm{p}(M_{k'})}_{\text{previous map}} \; dM_{k'}. \tag{5.11}$$

This extension and marginalization of a random variable has been described in Eq. (2.13), and used also in the prediction step of the filter posterior Eq. (2.20) of a Kalman filter, for example. This marginalization inserts known map information with uncertainty into the train state estimation. The localization and trajectory estimation in Eq. (5.11) is different to the train localization of Eq. (5.7) because the trajectory is also estimated and updated with measurements. In contrast to Eq. (5.7), the localization and trajectory estimation is split into a trajectory estimation part and a train localization part:

$$\begin{aligned} &\mathrm{p}(\overbrace{T_k^{\mathrm{traj}}, T_k^{\mathrm{topo}}, T_k^{\mathrm{v}}}^{T_k}, B_k | Z_{1:k}, U_{1:k}, M_{k'}) \overset{\substack{\text{prod. rule}\\ \text{cond. indep.}}}{=} \\ &\qquad \underbrace{\mathrm{p}(T_k^{\mathrm{traj}} | T_k^{\mathrm{topo}}, Z_{1:k}, M_{k'})}_{\substack{\text{trajectory estimation}\\ \text{(Eq. (5.14))}}} \cdot \underbrace{\mathrm{p}(T_k^{\mathrm{topo}}, T_k^{\mathrm{v}}, B_k | Z_{1:k}, U_{1:k}, M_{k'})}_{\substack{\text{train localization}\\ \text{(Eq. (5.13))}}}. \end{aligned} \tag{5.12}$$

In a first step, the train localization is computed with the estimation of T_k^{topo}, train speed T_k^{v} and an additional bias B_k. The factorization of the train localization step is similar to the localization filter posterior in Eq. (5.7):

$$\begin{aligned}
&\mathrm{p}(T_k^{\mathrm{topo}}, T_k^{\mathrm{v}}, B_k | Z_{1:k}, U_{1:k}, M_{k'}) = \\
&\quad \underbrace{\mathrm{p}(Z_k | T^{\mathrm{topo}}, B_k, M_{k'})}_{\text{measurement update}} \cdot \int \underbrace{\mathrm{p}(T_k^{\mathrm{topo}} | T_k^{\mathrm{v}}, T_{k\text{-}1}^{\mathrm{topo}}, U_k, M)}_{\text{non-linear train transition on tracks}} \cdot \underbrace{\mathrm{p}(T_k^{\mathrm{v}} | T_{k\text{-}1}^{\mathrm{v}}, U_k)}_{\text{linear train transition}} \cdot \\
&\quad \underbrace{\mathrm{p}(B_k | B_{k\text{-}1})}_{\text{bias transition}} \cdot \underbrace{\mathrm{p}(T_{k\text{-}1}^{\mathrm{topo}}, T_{k\text{-}1}^{\mathrm{v}}, B_{k\text{-}1} | Z_{1:k\text{-}1}, U_{1:k\text{-}1}, M)}_{\text{prior}} \; d(T_{k\text{-}1}, B_{k\text{-}1}).
\end{aligned} \tag{5.13}$$

The measurement update compares the predicted trajectory values from the predicted location and the map. This update corrects the along-track location and compares the predicted trajectory with the measured value for a track identification.

The second step of the trajectory estimation (Eq. (5.12)) is computed in the same time step, but after the computation of the train localization step. The trajectory update computes the trajectory from the measurement and the map information at the estimated location:

$$\mathrm{p}(T_k^{\mathrm{traj}} | T_k^{\mathrm{topo}}, Z_{1:k}, M_{k'}) \overset{\substack{\text{Bayes} \\ \text{cond. indep.}}}{=} \frac{\overbrace{\mathrm{p}(Z_k | T_k^{\mathrm{traj}})}^{\text{trajectory update}} \cdot \overbrace{\mathrm{p}(T_k^{\mathrm{traj}} | T_k^{\mathrm{topo}}, M_{k'})}^{\substack{T^{\mathrm{traj}} \text{ prediction} \\ \text{from estimated } T^{\mathrm{topo}}}}}{\mathrm{p}(Z_k | Z_{1:k\text{-}1})}. \tag{5.14}$$

The presented factorization is used in the RailSLAM implementation with an MHT filter and a delayed mapping in Section 6.3.

5.3.3 Comparison to Robotic SLAM

There are some differences between the classical robotic SLAM, as described in [13] and the proposed RailSLAM method. These differences are listed in Table 5.2.

The map in robotic SLAM is either location-based or feature-based and locations are addressed with Cartesian coordinates (see [13]). A location-based map holds information for every location of the map grid and an example is the occupancy grid map. The feature-based map contains only objects at specific locations of the environment, called landmarks. In contrast, RailSLAM contains feature sample points of each railway track and assigns these with a track ID and a 1-D location on the track. The track map in RailSLAM represents railway tracks as possible train trajectories

	Robotic SLAM (see [13])	**RailSLAM**
Map Content	obstacles, accessible free space	railway tracks: constrained trajectories
Map Type	location-based: occupancy grid feature-based: landmarks	features over 1-D location for each railway track
Map Coordinates	2-D/3-D: $x, y, (z)$, relative to origin	track ID (id), track location (s)
Motion Model	3 degrees of freedom (2-D position and yaw)	constrained to 1-D track, two paths at switches
Motion Meas.	speed, steering angle	speed, acceleration
Feature Measurements	laser scanner, camera, sonar (extrinsic)	GNSS, IMU (intrinsic)
Features, Signatures	depends on sensors e.g. edges, patterns, colors	track features: position, attitude, curvatures
Association Problem	corresponding features, landmarks	track and 1-D location, also switch way
Loop Closure	required for conversion	uses absolute positions, no requirement for conversion
Simultaneous	localization with previous map followed by map update in the same time step	delayed mapping possible, *Simultaneous* mapping in the same train run

Table 5.2: Differences between classical robotic SLAM and RailSLAM.

instead of distant features and free space for an arbitrary motion as used in the robotic SLAM. Furthermore, the features of the railway track consists of absolute positions, attitude, curvatures, and additional signatures such as magnetic of vibration patterns. A robot moves freely in a 2-D space and the motion model has three degrees of freedom. Trains are constrained to 1-D tracks and only at switches, there are two possible paths. The measurements are divided into motion measurements and feature measurements for localization and mapping. The feature measurements of the robotic SLAM are extrinsic because laser scanners or cameras measure distant features, which additionally persist over several time steps. In contrast, the RailSLAM measurements of GNSS and IMU are intrinsic, as only the position, acceleration and turn rate of the train is measured and not a direct track feature. RailSLAM does not require loop closures for the convergence of location and map, because the absolute position updates from GNSS decouple the track sample points from each other. Finally, the simultaneous property of RailSLAM can be relaxed with a delayed mapping for some circumstances. A delayed mapping is possible if solely intrinsic measurements are used and trains do not revisit the same track immediately. With exteroceptive measurements, the same distant landmark is seen over several time steps and should be directly integrated in the map. In the context of robotic or pedestrian SLAM, a robot or person can quickly turn and close a loop. Trains instead move in a more deterministic way, for example a direction change is only possible after a stand still and a switch is needed to join to known tracks. The mapping delay is used to reduce the creation and deletion of multiple maps for each hypotheses after passed switches or possible new switches. Nevertheless, RailSLAM can be still considered as a simultaneous mapping method in the sense of the same train run.

Chapter 6

Implementation

This chapter presents the algorithms for the train localization and track mapping as well as a GNSS and IMU data fusion in a pre-filter. There are three different localization methods and the RailSLAM method.

6.1 GNSS and IMU Data Fusion

The GNSS and IMU measurements can be processed in the localization or mapping estimation or in a pre-processing, as defined in Section 5.2.3. This pre-processing keeps the sensor fusion and bias estimation separate from the localization or map estimation algorithms. The pre-processing combines GNSS and IMU measurements with an inertial navigation system (INS). The output is the pre-processed measurement vector Z^{P} (see Eq. (5.8)) composed of position, velocity, attitude and curvatures of the time step k:

$$Z_k^{\mathrm{P}} = (\underbrace{\varphi, \lambda, h}_{\text{position}}, \underbrace{v,}_{\text{velocity}} \underbrace{\phi, \theta, \psi}_{\text{attitude}}, \underbrace{c^{\phi}, c^{\theta}, c^{\psi}}_{\text{curvatures}})_k. \tag{6.1}$$

Fig. 6.1 shows the pre-processing with INS/GNSS, and the curvatures are computed from the INS data and the IMU turn rates. The INS/GNSS is implemented in a loosely-coupled version with the strapdown algorithm and an error-state Kalman filter (ESKF), as defined in [47]. The main steps of the pre-processing are shown in the algorithm Fig. 6.4.

The INS state vector comprises 15 values with 3-D positions (latitude, longitude, height), velocity in north-east-down navigation frame, attitude in roll-pitch-yaw angles as well as the biases of acceleration and gyroscope measurements:

$$\boldsymbol{x}^{\mathrm{INS}} = (\underbrace{\varphi, \lambda, h}_{\text{position}}, \underbrace{v_n, v_e, v_d}_{\text{velocity}}, \underbrace{\phi, \theta, \psi}_{\text{attitude}}, \underbrace{b^{\mathrm{ax}}, b^{\mathrm{ay}}, b^{\mathrm{az}}}_{\text{accel. bias}}, \underbrace{b^{\omega \mathrm{x}}, b^{\omega \mathrm{y}}, b^{\omega \mathrm{z}}}_{\text{gyro. bias}})^T. \tag{6.2}$$

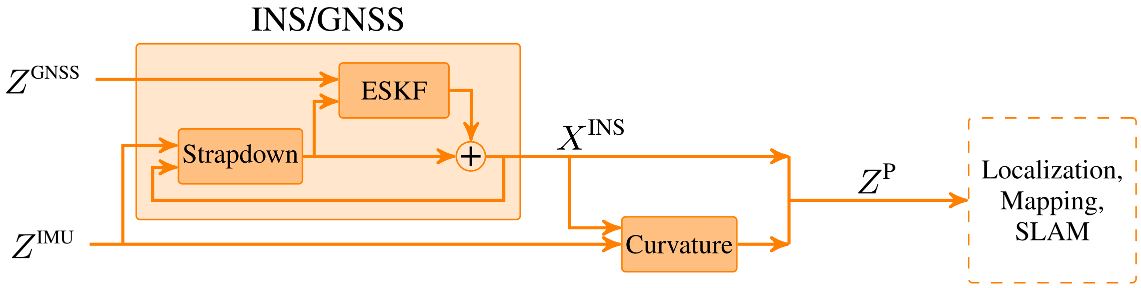


Figure 6.1: GNSS and IMU data fusion for pre-processed measurements Z^{P}.

A strapdown algorithm tracks the position, velocity and attitude from inertial measurements as described in Section 4.2.2. The differential equations are in Appendix D and more details of the algorithm and derivations are in [47]. The algorithm is showed in Fig. 6.2.

The INS propagates the position, velocity and attitude with the strapdown method (Fig. 6.2, line 4):

$$\boldsymbol{x}^{\mathrm{INS}}_{k|k\text{-}1} = \mathrm{f}_{\mathrm{SD}}(\boldsymbol{x}^{\mathrm{INS}}_{k\text{-}1}, \hat{\boldsymbol{z}}^{\mathrm{IMU}}_{k}). \tag{6.3}$$

```
    Function: Pre-processed measurements
    Input: GNSS measurements (PVT) $Z^{\mathrm{GNSS}}$, IMU measurements $Z^{\mathrm{IMU}}$
    Output: pre-processed measurements $Z^{\mathrm{P}}$
 1: loop
 2:     if new IMU measurement $Z^{\mathrm{IMU}}$ then
 3:         correct inertial meas. with estimated biases: $\hat{\boldsymbol{z}}^{\mathrm{IMU}}_k = \boldsymbol{z}^{\mathrm{IMU}}_k - \boldsymbol{x}^{\mathrm{INS,bias}}_{k\text{-}1}$
 4:         propagate INS state via strapdown Eq. (6.3)
 5:         propagate ESKF cov.$\boldsymbol{\Sigma}^{\Delta}_{k|k\text{-}1}$ with meas. noise and bias noise Eq. (2.31)
 6:         update low-pass filter of curvature computation with corrected turn rates
 7:     else if new GNSS measurement $Z^{\mathrm{GNSS}}$ then
 8:         compute innovation between GNSS measurements and INS state $\boldsymbol{x}^{\mathrm{INS}}_{k|k-1}$
 9:         update ESKF error estimate $\Delta\boldsymbol{x}_{k|k}$ Eq. (2.30) and $\boldsymbol{\Sigma}^{\Delta}_{k|k}$ Eq. (2.31)
10:         correct the INS state $\boldsymbol{x}^{\mathrm{INS}}_{k|k-1}$ with the estimated errors $\Delta\boldsymbol{x}_{k|k}$
11:     end if
12:     if time for output then
13:         compute horizontal velocity from : $v = \sqrt{v_{\mathrm{n}}^2 + v_{\mathrm{e}}^2}$
14:         compute curvatures from filtered turn rates and velocity $v$ (Eq. (4.20))
15:         output vector $Z^{\mathrm{P}}$ with position, velocity, attitude and curvatures
16:     end if
17: end loop
```

Figure 6.2: Algorithm: Pre-processed measurements from loosely-coupled INS.

The ESKF has been described in Eq. (2.27)-Eq. (2.32) in Section 2.2.1 and the error state vector comprises 15 error states in the loosely-coupled version [47]:

$$\Delta\boldsymbol{x} = \\ (\Delta x_{\mathrm{n}},\Delta x_{\mathrm{e}},\Delta x_{\mathrm{d}},\Delta v_{\mathrm{n}},\Delta v_{\mathrm{e}},\Delta v_{\mathrm{d}},\Delta\alpha,\Delta\beta,\Delta\gamma,\Delta b^{\mathrm{ax}},\Delta b^{\mathrm{ay}},\Delta b^{\mathrm{az}},\Delta b^{\omega\mathrm{x}},\Delta b^{\omega\mathrm{y}},\Delta b^{\omega\mathrm{z}})^{T}. \tag{6.4}$$

The position error is in the metric north-east-down format and the attitude error consists of Euler angle increments. The ESKF system matrix is derived from differential equations for the errors of position, velocity, attitude, and inertial sensors [47]. In line 5, the ESKF covariance is propagated with measurement noise of the inertial measurements input and with process noise for the bias estimation [47]. The curvatures are computed with Eq. (4.20) from the INS velocity estimate and from the low-pass filtered and bias-corrected turn rate measurements (lines 6,13-14).

6.2 Train Localization

This section introduces the most common track map interface functions and three train localization methods. The three localization methods are the simple map-match method, a probabilistic particle filter method and a probabilistic multi hypothesis tracking (MHT) method.

6.2.1 Track Map Interface Functions

The train localization implementations and also RailSLAM use common interface functions of the track map:

The track map transition function is needed for the propagation of a train on tracks. A topological pose of a train is shifted with a distance Δs and the transition function computes a new topological pose:

$$\{\ \underbrace{\hat{id},\ \hat{s},\ \hat{o}^{\mathrm{t}}}_{\text{new topo pose(s)}}\ \} = \mathrm{f}_{\mathrm{map,trans}}(\ \underbrace{id,\ s,\ o^{\mathrm{t}},}_{\text{prev. topo pose}}\ \Delta s). \tag{6.5}$$

In the simplest case, the topological pose stays on the same track and Δs is just added or subtracted from the location s according to the orientation and the sign of the distance. In the case that a train changes the track, the location s faces a jump discontinuity at the changeover of the tracks and the track ID is also changed. Additionally, the new track may have a different track orientation definition. In this case, the transition function changes the track-to-train frame direction o^{t} of the new topological pose. If a switch passed facing, there are two possible switch ways. For this case, the track map transition function outputs two topological poses.

The second track map interface function is a query of the track geometry $m_{id,s}$ defined in track frame t at the topological location (id, s):

$$\underbrace{\{\boldsymbol{\mu}^{\mathrm{m}}, \boldsymbol{\Sigma}^{\mathrm{m}}\}_{id,s}}_{\substack{\text{track sample}\\ \text{point } m_{id,s}}} = \mathrm{f}^{\mathrm{t}}_{\mathrm{map}}(\underbrace{id, s}_{\substack{\text{topo.}\\ \text{location}}}). \tag{6.6}$$

The track sample $m_{id,s}$ of the probabilistic track map contains a mean vector $\boldsymbol{\mu}^{\mathrm{m}} \in \mathbb{R}^{9\times 1}$ and a covariance matrix $\boldsymbol{\Sigma}^{\mathrm{m}} \in \mathbb{R}^{9\times 9}$. The vector $\boldsymbol{\mu}^{\mathrm{m}}$ contains the track geometry values with positions, attitude and curvatures. The track geometry values are interpolated at the location s from the stored geometry sample points of the track.

A train localization requires a comparison of the measurements and the track geometry. In this thesis, the track geometry is converted to the train frame for the this comparison. Therefore, the function Eq. (6.6) is extended and the track geometry is converted to the train frame, as described in Table 3.3. The trajectory function of the track map returns the track geometry converted to the train frame b and requires additionally the train-to-track frame direction o^{t}:

$$\underbrace{\{\boldsymbol{\mu}^{\mathrm{traj}}, \boldsymbol{\Sigma}^{\mathrm{traj}}\}_{id,s,o^{\mathrm{t}}}}_{\substack{\text{trajectory}\\ \text{point } T^{\mathrm{traj}}}} = \mathrm{f}^{\mathrm{b}}_{\mathrm{map}}(\underbrace{id, s, o^{\mathrm{t}}}_{\substack{\text{topo.}\\ \text{pose}}}). \tag{6.7}$$

A train trajectory point T^{traj}, as defined in Eq. (5.5) and Section 5.2.1, contains the mean vector $\boldsymbol{\mu}^{\mathrm{traj}} \in \mathbb{R}^{9\times 1}$ with the trajectory values and the covariance matrix $\boldsymbol{\Sigma}^{\mathrm{traj}} \in \mathbb{R}^{9\times 9}$.

6.2.2 Simple Map-Match Method

The simple map-match method uses a nearest neighbor approach. This method searches the nearest track (id) and the track location s from position measurements without prior information:

$$\underbrace{\{id, s\}}_{\text{topo. location}} = \mathrm{f}_{\text{map-match}}(\underbrace{\varphi\ ,\ \lambda}_{\text{geo position}}). \tag{6.8}$$

Figure 6.3 illustrates the block diagram with the position measurement input, the track map and the train location output. The simple map-match method is a deterministic function and does use a past result. The location computation involves a search of tracks and segments in the vicinity of the position measurement, an orthogonal projection of the position to the track segments, and the output of the nearest matched track ID and track location. This method is used as a reference localization approach as well as in the initialization of the probabilistic localization methods. The simple map-match method would be sufficient if the position measurement are continuously available in combination with an accuracy that is always better than half of the distance of parallel tracks. It will be shown in the results, that this method has accuracy problems in terms of track selectivity with real GNSS position measurements, recorded on a train.

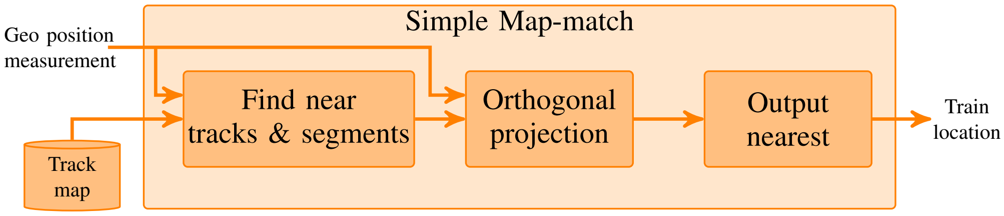


Figure 6.3: Block diagram of a train localization with the simple map-match method.

The matched position on the track map minimizes the distance between a measured position and the track model. The track model is defined with position sample points and straight segments in between. Figure 6.4 shows a Voronoi diagram of a switch scenario with linear track segments. This diagram shows areas around the segments, where any position within a specific area would be matched by shortest distance to the current segment and track, respectively. The areas around different tracks are separated with different colors.

The orthogonal projection is the method to find a position on a track segment and a track location from a 2-D position measurement. This method calculates a point on the track segment which defines an orthogonal

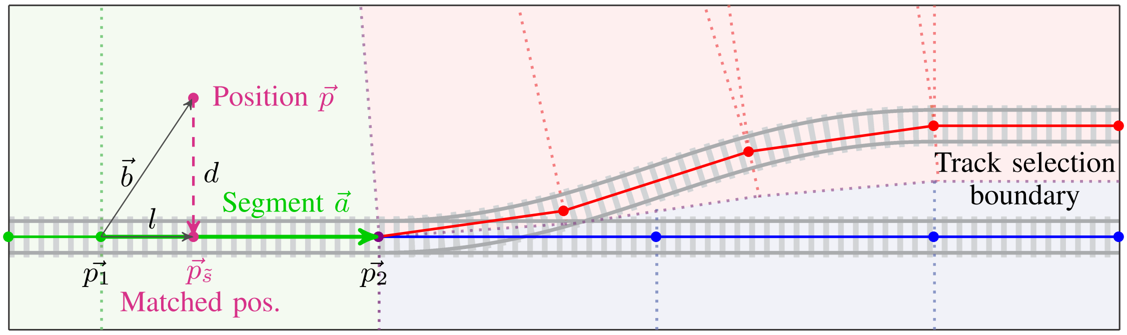


Figure 6.4: Voronoi diagram for line segments and orthogonal projection.

vector $\vec{d}$ between segment $\vec{a}$ and position $\vec{p}$, as depicted in Figure 6.4. The segment $\vec{a}$ is defined between the track sample positions $\vec{p_1}$ and $\vec{p_2}$. The vector $\vec{d}$ is an approximation of the cross-track distance δ_{CT} (see Section 3.2.1) because of the track geometry approximation with straight segments. The vector $\vec{b}$ points from the first segment point $\vec{p_1}$ to the position measurement $\vec{p}$. A length l is calculated with the scalar dot product ($\vec{a} \bullet \vec{b} = |\vec{a}||\vec{b}| \cos \alpha$):

$$l = \frac{(\vec{a} \bullet \vec{b})}{|\vec{a}|}. \tag{6.9}$$

The projected segment point is calculated with the unit vector of $\vec{a}$:

$$\vec{l} = l \cdot \frac{\vec{a}}{|\vec{a}|}, \tag{6.10}$$

and the absolute matched position is $\vec{p_{\tilde{s}}} = \vec{p_1} + \vec{l}$. The distance d is defined between $\vec{l}$ and $\vec{p}$. The train location is composed by the ID of the track with the minimum distance d to a segment and the track location, which is calculated with the track location of the first track sample s_{p_1} and the length l:

$$\tilde{s}_{\mathrm{match}} = s_{p_1} + l. \tag{6.11}$$

The algorithm summary of the simple map-match method is presented in Fig. 6.5. This method is entitled with *simple* because it uses no feedback of previous results and it does not need any parameters or noise assumptions. Different projection methods of positions to railway tracks regarding covariances can be found in [16].

```
Algorithm: Train Localization (Simple Map-Match)
Input: GNSS or INS position data ($\vec{p}$), track map
Output: topological location ($id, s$)
loop
    if new position measurement $\vec{p}$ available then
        get track IDs in vicinity of search area
        for all near tracks do
            find nearest track segment with $\vec{p_1}$ and $\vec{p_2}$
            compute orthogonal projection with $\vec{p}, \vec{p}_1, \vec{p}_2$
            save track $id$ with cross-track distance $d$ and length $l$
        end for
        $id$: select nearest track with minimum distance $d$
        $\tilde{s}$: compute 1-D location with length $l$ and nearest segment
    end if
end loop
```

Figure 6.5: Train localization algorithm with simple map-match method.

6.2.3 Particle Filter Method

The train localization method with a particle filter is depicted in Figure 6.6. The sensor measurements of GNSS and IMU are directly used as an input for the particle filter method. The particle filter is a probabilistic estimation method because it computes the train location with probability distributions. This particle filter is a Bayesian sequential filter because it updates with measurements for each time step and it uses the output probability distribution of the past time step as a prior information. As defined in Section 2.2.3, the particle filter consists of a proposal function, a weight function, a resample function and an output estimate. The proposal and weight functions are defined from the factorized full posterior in the following. The train localization full posterior, as defined in Section 5.2.2 Eq. (5.6), is now estimated with particles and their associated weights:

$$\mathrm{p}(T_{0:k}, B_{0:k}|Z_{1:k}, U_{1:k}, M) \approx \{x^i_{0:k}, w^i\}^{N_p}_{i=1}. \tag{6.12}$$

This train localization approach with particle filter has been published initially in [P2], with extensions and simulations in [P3], and in the final version with train measurement evaluations in [J1].

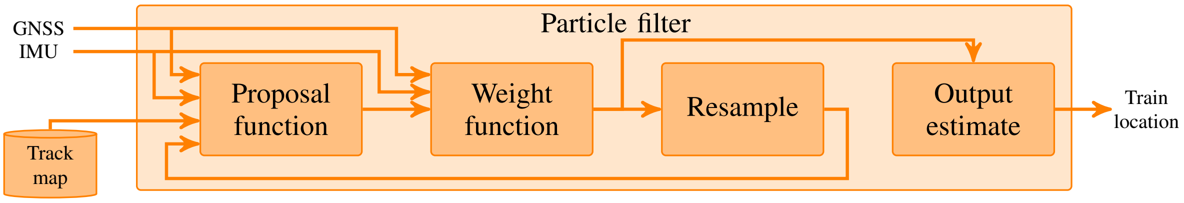


Figure 6.6: Block diagram of the train localization with particle filter method.

Proposal Function

The proposal function or importance density q() is designed with the linear train transition, the bias estimation, the non-linear transition on tracks and the prior:

$$\mathrm{q}(T_{0:k}, B_{0:k}|Z_{1:k}, U_{1:k}, M) =$$
$$\underbrace{\mathrm{p}(T^l_k|T^l_{k\text{-}1},U_k)}_{\substack{\text{KF: odometry,}\\ \text{linear transition}}}\cdot\underbrace{\mathrm{p}(B_k|B_{k\text{-}1})}_{\substack{\text{KF}\\ \text{bias transition}}}\cdot\underbrace{\mathrm{p}(T^n_k|T^l_k,T^n_{k\text{-}1},U_k,M)}_{\text{non-linear train transition}}\cdot\underbrace{\mathrm{p}(T_{0:k\text{-}1},B_{0:k\text{-}1}|Z_{1:k\text{-}1},U_{1:k\text{-}1}, M)}_{\text{prior}}. \tag{6.13}$$

The linear and one dimensional train transition, called odometry in the following, is estimated with a Kalman filter and updated with GNSS speed and longitudinal IMU acceleration measurements. The odometry state (x^{odo}) contains the estimate of the speed v, displacement Δs and acceleration measurement bias b^{ax}. The state transition of the odometry prediction step is defined here as a 1-D transition model of the type discrete white noise constant acceleration (DWPA) [40]. The linear 1-D train transition and the acceleration bias are estimated with the discrete model:

$$\underbrace{\begin{pmatrix} \Delta s \\ v \\ a \\ b^{\mathrm{ax}} \end{pmatrix}_k}_{x_k^{\mathrm{odo}}} = \begin{pmatrix} 0 & \Delta t & \frac{\Delta t^2}{2} & 0 \\ 0 & 1 & \Delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta s \\ v \\ a \\ b^{\mathrm{ax}} \end{pmatrix}_{k-1} \tag{6.14}$$

The non-linear train transition is estimated with particles and the displacement result of the odometry is used as an input. Therefore, the displacement of the i'th particle is sampled from the distance estimate of the odometry:

$$\Delta s_k^i \sim \mathcal{N}\left(\Delta s_k^{\mathrm{odo}}, \sigma_{\Delta s}^2\right). \tag{6.15}$$

Afterward, the train transition function on the track (Eq. (6.5)) is processed for each particle. This transition ensures, that the estimates, or particles respectively, exist and stay exclusively on tracks. The transition function on tracks computes a new topological coordinate from the linear displacement Δs_k^i, the previous coordinate, a randomly chosen switch way and the track map. The transition function of the track map considers the discontinuity at a track change. In the case that the train passes a switch facing, the switch way is sampled from a discrete and uniform distribution. Finally, the track geometry is obtained with Eq. (6.7) from the track map and defined in the train frame. The bias of the gyroscope ($b^{\omega \mathrm{z}}$) is observed and updated only during train stops:

$$b_k^{\omega \mathrm{z}}|_{v=0} \approx \omega_z. \tag{6.16}$$

The turn rates are assumed to be zero at stand-still and the low Earth turn rate is neglected. The proposal function contains a Kalman filter for odometry and gyroscope bias. Therefore the particle filter estimates actually only two state dimensions with the track ID and 1-D location:

$$x^{\mathrm{PF}} = (id, s)^T. \tag{6.17}$$

In combination with the estimates from the Kalman filter and the track map, the particle state x is extended with: the train direction o^{t}, the velocity v and the gyroscope bias $b^{\omega z}$ as well as positions, heading and curvature of the current track location in the train frame. The complete state is:

$$x = (\underbrace{id, s}_{\text{PF}}, o^{\text{t}}, \underbrace{v, b^{\text{ax}}}_{\text{odometer}}, b^{\omega z}, \underbrace{\varphi, \lambda, \psi, c^{\psi}}_{\text{from track map}})^T. \tag{6.18}$$

As mentioned in Section 3.2.2, the train-track direction o^{t} is an auxiliary state that can be tracked in a deterministic way, once this direction is known. This tracking is handled in the train transition function of the track map Eq. (6.5).

Weight Function

The weight function is computed from the posterior Eq. (5.6) and the proposal function Eq. (6.13) as defined in Eq. (2.41):

$$w_k = \eta \frac{p(T_{0:k}, B_{0:k} | Z_{1:k}, U_{1:k}, M)}{\mathrm{q}(T_{0:k}, B_{0:k} | Z_{1:k}, U_{1:k}, M)}. \tag{6.19}$$

Hence, the weight function for the i'th particle with IMU and GNSS likelihoods is:

$$w_k^i = \eta \cdot \mathrm{p}(Z_k^{\text{GNSS,pos}} | T_k^i) \cdot \mathrm{p}(Z_k^{\text{GNSS},\psi} | T_k^i) \cdot \mathrm{p}(Z_k^{\text{IMU},\omega_z} | T_k^i, B_k^i) \cdot w_{k-1}^i. \tag{6.20}$$

In this approach, the GNSS and IMU data is directly used without any pre-processing such as an INS. In practice, the weighting function with the measurement likelihoods is based on a comparison of measurements with expected measurements from the track map. The following sensor models implement a Gaussian likelihood: The measurement model for GNSS positions includes 2-D positions from the extended state Eq. (6.18) and an additional white noise n^{pos}:

$$Z_k^{\text{GNSS,pos}} = \begin{pmatrix} \varphi \\ \lambda \end{pmatrix}_k + n_k^{\text{pos}}, \qquad n_k^{\text{pos}} = \mathcal{N}(0, \Sigma^{\text{pos}}). \tag{6.21}$$

The GNSS heading measurement of a single antenna represents the direction of motion, while the state ψ in Eq. (6.18) considers a heading in train frame. For this reason, the measurement model turns the estimated heading ψ with π, if the train motion is backward:

$$Z_k^{\text{GNSS},\psi} = \begin{cases} \psi_k + n_k^{\psi}, & \text{if } v > 0, \\ \psi_k + \pi + n_k^{\psi}, & \text{if } v < 0, \end{cases} \qquad n_k^{\psi} = \mathcal{N}\left(0, \sigma_{\psi}^2\right). \tag{6.22}$$

Finally, the yaw rate measurement model is defined from the curvature c^{ψ}, the train speed in track frame $\dot{s}$, a gyroscope bias $b^{\omega \mathrm{z}}$ and a white noise $n^{\omega z}$:

$$Z_k^{\mathrm{IMU},\omega \mathrm{z}} = c_k^{\psi} \cdot \dot{s}_k + b_k^{\omega \mathrm{z}} + n_k^{\omega z}, \qquad n_k^{\omega z} = \mathcal{N}\left(0, \sigma_{\omega z}^2\right). \tag{6.23}$$

The train speed $\dot{s}$ in track frame can be obtained from the estimation state Eq. (6.18) with the speed in train frame v and the direction o^{t}, as defined in Table 3.2.

Output Estimate

For most applications, a particle distribution is not convenient because a single mode or most likely output is required. As described in Eq. (2.47), a minimum mean square estimate (MMSE) is computed from the particle distribution. An alternative is the output of the most likely particle with the highest weight. The MMSE takes the particle distribution also into account and filters over particles because the sampling step added noise. Internally, the particle filter keeps its particle distribution for the next update. As a specialty of this filter, the particles are on the tracks. It is not feasible to compute an MMSE for a switch scenario with particles on both switch ways. The output estimate decides first for the more likely 1-D path of tracks and computes then an MMSE on this path. A track path contains one or more sequential tracks on a 1-D path, where a train is able to run over sequentially. For further computations, a track path has a continuous 1-D coordinate frame compared to discontinuities at the joints of tracks. A topological pose can be translated into path coordinates as well as translated from path coordinates. In a switch scenario, particles can be distributed before the switch, on the left switch way and on the right switch way. This particle distribution would result in two possible track paths: $id_p^1 = \{id_{\mathrm{before}}, id_{\mathrm{left}}\}$ and $id_p^2 = \{id_{\mathrm{before}}, id_{\mathrm{right}}\}$. The output estimate is calculated with the following four steps as defined in [J1]:

a) Track path: all track path hypotheses are identified with at least one particle.
b) Most likely (ML) path: the sum of weights are calculated for each path and the most likely path has the highest cumulative weight.
c) Mean square estimate on path: A 1-D location and standard deviation is calculated from a weighted mean and weighted sample variance of the selected particles, which belong to the ML path.
d) Translation to topological pose: the most likely path and the mean location on path are converted back into topological coordinates with track ID and 1-D location.

Algorithm Summary

The algorithm of the sequential Bayesian filter with a particle filter, GNSS, IMU, and a track map is shown in Fig. 6.7. The particle filter is initialized from the first GNSS position measurement: Therefore, the initial particles are sampled from a 2-D Gaussian distribution at the first position with an enlarged covariance [J1]. For each particle, the sampled positions are map-matched by Eq. (6.8) for the initial topological location.

```
Algorithm: Train Localization (Particle Filter)
Input: GNSS and IMU sensor data, track map, parameter
Output: topological coord. $(id, s, o^{\mathrm{t}})$ and train speed $v$
initialize all $N_p$ particles by first GNSS position
loop
    if new measurement(s) available then
        time step: $k = k + 1$, $\Delta t = t_k - t_{k-1}$
        for all $N_p$ particles do
            predict odometry KF Eq. (6.14)
            update KF with speed / acceleration measurement
            if train is moving then
                sample displacement from odometry Eq. (6.15)
                compute map transition Eq. (6.5)
                get geometry from track map (train frame) Eq. (6.7)
                compute likelihoods Eq. (6.21)/Eq. (6.22)/Eq. (6.23)
                multiply particle weight by likelihoods Eq. (6.20)
            else  (train is stopped)
                observe and filter gyroscope bias Eq. (6.16)
            end if
        end for
        normalize weights Eq. (2.46)
        compute most likely output estimate
        if resampling necessary then
            perform resampling
        end if
    end if
end loop
```

Figure 6.7: Train localization with particle filter, GNSS, IMU, and track map.

6.2.4 Multiple Hypotheses Tracking Method

The multiple hypothesis tracking (MHT) method estimates the train localization filter posterior. The MHT is a probabilistic Bayesian filter method that computes different hypothesis with probabilities and probability distributions for the train location estimate. The MHT method computes sequentially the train location from measurements and from the last estimate. Figure 6.8 shows the block diagram of the MHT method with the pre-processed measurements and the track map as inputs and the train location as output. There are two functions that estimate the train location: the along-track estimation function and the track identification function. The pruning step removes hypotheses with a low weight below a defined value. The MHT differs from the particle filter method with a range of changes. One difference is the estimation of the filter posterior instead of the full posterior. Other differences are a pre-filter of sensor data via INS, a separate along-track estimation, and an explicit focus on switch-way detection and track identification. Same as the particles, the hypotheses contain a weight and an estimate of the train state and biases. The main difference is, that there is only one hypothesis for each track in along-track direction. The multiple hypotheses arise from the multiple tracks after passing a switch facing, for example.

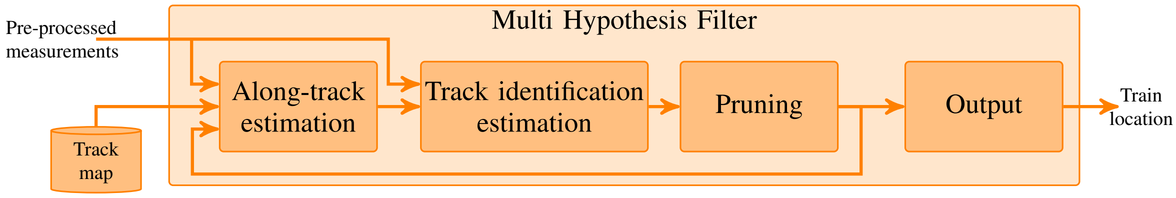


Figure 6.8: Block diagram of the train localization with the MHT method.

The filter posterior of the train localization (Eq. (5.7)) is extended with multiple hypothesis and associated weights (see Section 2.2.2, Eq. (2.33)-Eq. (2.35)). The filter posterior factorization with the hypotheses extension is:

$$\mathrm{p}(T_k, B_k | Z_{1:k}, U_{1:k}, M) \propto \sum_j \underbrace{w_k^j}_{\substack{\text{mixture weight of} \\ j\text{'th hypothesis}}} \cdot \underbrace{\mathrm{p}(Z_k^{\mathrm{AT}} | T_k^j, B_k)}_{\substack{\text{measurement update} \\ \text{for along-track}}} \cdot$$

$$\int \underbrace{\mathrm{p}(T_k^j | T_{k\text{-}1}^j, U_k^j, M)}_{\text{train transition}} \cdot \underbrace{\mathrm{p}(B_k | B_{k\text{-}1})}_{\text{bias transition}} \cdot \underbrace{\mathrm{p}(T_{k\text{-}1}^j, B_{k\text{-}1} | Z_{1:k\text{-}1}, U_{1:k\text{-}1}^j, M)}_{\text{prior}} \; d(T_{k\text{-}1}^j, B_{k\text{-}1}). \tag{6.24}$$

The pre-processed measurements contain already a sensor fusion of IMU and GNSS measurements. The bias vector B considers the correlated errors of the pre-processed measurements and is the same for all hypotheses. The MHT posterior of Eq. (6.24) shows a mixture weight w that is used for the track identification estimation. The remaining factorization of Eq. (6.24) is used for the along-track estimation. The measurements Z^{AT} are suitable measurements for an along-track estimation, according to Section 3.2.3. The number of hypotheses is dynamically changed, e.g. after passing a switch. A switch passed facing at time step k results in two hypotheses of the train control input: $\mathcal{H}_k^1 = U_k^{\text{sw,right}}, \mathcal{H}_k^2 = U_k^{\text{sw,left}}$. For a switch passed facing with an unknown switch way, there are two hypothesis created with equal weights. The mixture weight update of the j'th hypothesis is (see Eq. (2.35)):

$$w_k^j = \eta \cdot w_{k\text{-}1}^j \cdot \mathrm{p}(Z_k^{\mathrm{TI}} | \underbrace{T_k^j,\ B_k^j}_{\text{j'th hypothesis } \mathcal{H}^j}). \tag{6.25}$$

The measurements Z^{TI} are all suitable measurements for a track identification, especially at a switch, according to Section 3.2.3 and Z^{TI} is defined in the following.

A pruning step reduces hypotheses with low weights or after a switch was passed merging with two hypotheses. The pruning is suspended, as long as the hypotheses are inside a defined switch section. Finally, the hypothesis with highest weight is selected as output.

Measurement of Along-track and Cross-track Distance

A position measurement contributes to the along-track estimation as well as to the track identification. Therefore, an along-track distance d_{AT} and a cross-track distance d_{CT} are computed from an estimated track position and a measured position, as depicted in Fig. 3.4, Section 3.2.1. At first, a distance vector in NED coordinates is computed between a position measurement and a position on the tracks with Eq. (3.8) or Eq. (3.9). The predicted position on the tracks is retrieved from the map with Eq. (6.6) at the predicted location $(id, s, o^{\mathrm{t}})_{k|k-1}$. The following computes a NED vector in 2-D from the position measurement $\boldsymbol{z}^{\varphi\lambda}$ and the predicted trajectory vector $\boldsymbol{\mu}_{k|k-1}^{\mathrm{traj}}$ with Eq. (3.9):

$$\begin{pmatrix} x_{\mathrm{n}} \\ x_{\mathrm{e}} \end{pmatrix} = \begin{pmatrix} (R_n + h) & 0 \\ 0 & (R_e + h)\cos\varphi \end{pmatrix} \left(\boldsymbol{z}^{\varphi\lambda} - \boldsymbol{H}^{\varphi\lambda} \boldsymbol{\mu}_{k|k-1}^{\mathrm{traj}} \right). \tag{6.26}$$

The conversion matrix and also the meridian radius R_n (Eq. (C.10)) and the transverse radius R_e (Eq. (C.11)) depend the measurements of latitude and height. The matrix $\boldsymbol{H}^{\varphi\lambda}$ selects the latitude and longitude values from the predicted trajectory:

$$\boldsymbol{H}^{\varphi\lambda} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{6.27}$$

The NED vector of Eq. (6.26) in north (x_{n}) and east (x_{e}) direction is rotated with the heading angle ψ to the along-track and cross-track axes:

$$\begin{pmatrix} d_{\mathrm{AT}} \\ d_{\mathrm{CT}} \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} x_{\mathrm{n}} \\ x_{\mathrm{e}} \end{pmatrix} . \tag{6.28}$$

The measurement covariance of the position measurement is $\mathbf{\Sigma}^{\varphi,\lambda}$. The variances of along-track and cross-track distances are computed from the rotated measurement covariance with the heading angle ψ:

$$\begin{pmatrix} \sigma^2_{\mathrm{AT}} & \cdot \\ \cdot & \sigma^2_{\mathrm{CT}} \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix} \mathbf{\Sigma}^{\varphi,\lambda} \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix}^T . \tag{6.29}$$

This transformation does not include the Earth radius as in Eq. (6.26) because the covariance for the position is defined in meters.

Along-track Estimation

The along-track estimation contains a 1-D Kalman filter. However, the 1-D track location is discontinuous at track changes and also the track ID changes, too. This non-linearity requires to integrate the map functions Eq. (6.5) and Eq. (6.7) in the Kalman filter. The prediction step shifts the previous track location with the distance Δs as input, uses the map transition Eq. (6.5), and enlarges the 1-D track location uncertainty σ^2_s:

$$\begin{aligned} \Delta s_k &= v_k \cdot \Delta t, && (6.30) \\ (id, s, o^{\mathrm{t}})_{k|k-1} &= \mathrm{f}_{\text{map-trans}}((id, s, o^{\mathrm{t}})_{k-1}, \Delta s_k), && (6.31) \\ \sigma^2_{s,k|k-1} &= \sigma^2_{s,k} + \sigma^2_{\Delta s,k}. && (6.32) \end{aligned}$$

Afterward, the track geometry is retrieved from the track map in train frame with Eq. (6.7). In the case that a switch has been passed facing, one hypothesis gets duplicated after the map transition function, each with its own Kalman filter. The track ID and the train direction o^{t} are automatically tracked with the map transition function. The next step is an update with measurements. The 1-D location s is an arbitrary definition and hence it is not possible to measure this value directly. Special functions are required to process a track location or a distance increment from measurements and map geometry. The update step of the Kalman filter processes a distance on tracks $\Delta s^{\mathrm{meas}}_k$, which is the innovation between a measurement and expected measurement. The Kalman filter estimates then the distance $\Delta s'_{k|k}$ in the update step with the Kalman gain K, the predicted uncertainty $\sigma^2_{s,k|k-1}$, innovation Δs^{meas}_k, and a measurement uncertainty σ^2_{meas}:

$$\begin{aligned} K &= \frac{\sigma^2_{s,k|k-1}}{\sigma^2_{s,k|k-1} + \sigma^2_{\text{meas}}}, && (6.33)\\ \Delta s'_{k|k} &= K \cdot \Delta s^{\text{meas}}_k, && (6.34)\\ \sigma^2_{s,k|k} &= (1-K) \cdot \sigma^2_{s,k|k-1}. && (6.35) \end{aligned}$$

The along-track distance d_{AT} of Eq. (6.28) is a suitable measurement for the along-track estimation: $\Delta s^{\text{meas}}_k = d_{\text{AT}}$ including the measurement variance $\sigma^2_{\text{meas}} = \sigma^2_{\text{AT}}$, as defined in Eq. (6.29).

Finally, the predicted location $(id, s, o^{\text{t}})_{k|k-1}$ is shifted with the distance on tracks $\Delta s'_{k|k}$ via the map transition function Eq. (6.5) and the track geometry in train frame is retrieved with Eq. (6.7), once again.

Switch Way and Track Identification Estimation

The switch way or track identification estimation is realized with the weight update of the multiple hypotheses, as defined in the posterior Eq. (6.24) and in the general weight update Eq. (6.25). If a switch is passed facing, the along-track estimation creates two hypotheses with the same weight. The switch way is now estimated with a comparison of measurements and the different track geometries from the hypotheses of the left and right switch way. This switch-way estimation evaluates several measurements over a certain range from the switch start. This range is defined either with switch length (see 3.1.2) or a fixed distance from the switch start. The pruning is suspended in the switch range. A switch-way decision occurs, once the pruning step decides for a hypotheses elimination after that distance and if one switch-way hypothesis has a sufficient low weight.

The weight function Eq. (6.25) of each hypothesis is computed with a likelihood. This likelihood evaluates a difference between measurements and the trajectory of the j'th hypothesis. This difference is called track identification innovation Δ^{TI} and is computed from a measurement vector $\boldsymbol{z}^{\text{TI}}$ and the trajectory vector $\boldsymbol{\mu}^{\text{traj},j}$ of the j'th hypothesis. A Gaussian likelihood evaluates a probability (see Eq. (2.15)) from the track identification innovation vector $\boldsymbol{\Delta}^{\text{TI},j}$ and the innovation covariance matrix $\boldsymbol{\Sigma}^{\Delta\text{TI},j}$ of the j'th hypothesis:

$$p(Z^{\text{TI}}_k | T^j_k, B^j_k) \approx \mathcal{N}\left(\boldsymbol{\Delta}^{\text{TI},j}, \boldsymbol{\Sigma}^{\Delta\text{TI},j}\right). \tag{6.36}$$

Three measurements are used for the track identification: the cross-track distance d_{CT}, the heading ψ and the heading curvature c^{ψ}. The cross-track distance requires the 2-D position measurements, as defined in Eq. (6.28) and Eq. (6.26), and the measurement model is non-linear. The track identification measurements Z^{TI} with the measurement vector $\boldsymbol{z}^{\mathrm{TI}}$ and the measurement covariance $\boldsymbol{\Sigma}^{\mathrm{TI}}$ are obtained from the pre-processed measurements Z^{P} and defined with:

$$\boldsymbol{z}^{\mathrm{TI}} = \begin{pmatrix}\varphi & \lambda & \psi & c^{\psi}\end{pmatrix}^{T}. \tag{6.37}$$

The matrix $\boldsymbol{H}^{\mathrm{TI}}$ selects the track identification values from the train trajectory:

$$\boldsymbol{H}^{\mathrm{TI}} = \begin{pmatrix}1&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&1\end{pmatrix}. \tag{6.38}$$

The innovation $\boldsymbol{\Delta}^{\mathrm{TI},j}$ depends on the j'th hypothesis and contains a conversion to NED coordinates and a rotation to the cross-track axis with the heading ψ:

$$\underbrace{\begin{pmatrix}d^{j}_{\mathrm{CT}}\\ \Delta\psi^{j}\\ \Delta c^{\psi j}\end{pmatrix}}_{\boldsymbol{\Delta}^{\mathrm{TI},j}} = \begin{pmatrix}-\sin\psi(R_n+h) & \cos\psi(R_e+h)\cos\varphi & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\left(\boldsymbol{z}^{\mathrm{TI}} - \boldsymbol{H}^{\mathrm{TI}}\boldsymbol{\mu}^{\mathrm{traj},j}\right). \tag{6.39}$$

The innovation covariance $\boldsymbol{\Sigma}^{\Delta\mathrm{TI},j}$ is computed with the measurement covariance $\boldsymbol{\Sigma}^{\mathrm{TI}}$, the trajectory covariance $\boldsymbol{\Sigma}^{\mathrm{traj},j}$, and a rotation to the cross-track axis:

$$\boldsymbol{\Sigma}^{\Delta\mathrm{TI},j} = \begin{pmatrix}-\sin\psi & \cos\psi & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\left(\boldsymbol{H}^{\mathrm{TI}}\boldsymbol{\Sigma}^{\mathrm{traj},j}\boldsymbol{H}^{\mathrm{TI}^{T}} + \boldsymbol{\Sigma}^{\mathrm{TI}}\right)\begin{pmatrix}-\sin\psi & 0 & 0\\ \cos\psi & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}. \tag{6.40}$$

Eq. (6.37) to Eq. (6.40) describe the matrix definitions for the multivariate Gaussian likelihood. Alternatively, the Gaussian likelihood for the track identification can be defined with three one-dimensional Gaussian likelihoods:

$$\begin{aligned} p(Z^{\mathrm{TI}}_{k}|T^{j}_{k}, B^{j}_{k}) \approx &\mathcal{N}\left(d^{j}_{\mathrm{CT}}, (\sigma^{2}_{\mathrm{CT}} + \sigma^{2}_{\mathrm{CT},j})\right)\cdot\mathcal{N}\left((\psi - \psi^{j}), (\sigma^{2}_{\psi} + \sigma^{2}_{\psi,j})\right)\\ &\cdot\mathcal{N}\left((c^{\psi} - c^{\psi j}), (\sigma^{2}_{c^{\psi}} + \sigma^{2}_{c^{\psi},j})\right). \end{aligned} \tag{6.41}$$

The notation in Eq. (6.41) contains simplifications for clarity reasons: The estimates from $\boldsymbol{\mu}_k^{\mathrm{traj},j}$ are indexed only with j, the measurements are denoted without an index and the time step index k is omitted. The cross-track distance is computed with Eq. (6.28), and the variances are computed with Eq. (6.29). The heading variance is σ_ψ^2, and the heading curvature variance is $\sigma_{c^\psi}^2$.

During a passage of a switch, certain biases can be compensated. As described in 4.1.3, the cross-track position bias may be observed with the last position measurement before the switch and subtracted for the switch-way identification. In the ideal case, there are only two hypotheses after passing a switch facing, and a single hypothesis after the switch range has been left.

Algorithm Summary

The MHT train localization algorithm with pre-processed measurements is shown in Fig. 6.9. The set of hypotheses $\mathcal{H}$ is denoted after the first map

```
Algorithm: Train Localization (MHT)
Input: pre-processed measurements $Z^{\mathrm{P}}$, track map, parameter
Output: topological coord. $(id, s, o^{\mathrm{t}})$ and train speed $v$
initialize all $\mathcal{H}_0$ hypotheses by first position measurement
loop
    if new pre-processed meas. $Z^{\mathrm{P}}$ available then
        time step: $k = k + 1$, $\Delta t = t_k - t_{k-1}$
        if train is moving then
            compute map transitions of each $\mathcal{H}$ from velocity displacement $\rightarrow \mathcal{H}'$
            update absolute along-track filters of each $\mathcal{H}'$ with measurements
            compute map transitions of each $\mathcal{H}'$ from along-track correction $\rightarrow \mathcal{H}''$
            if $\mathcal{H}''$ contains only one hypothesis then
                output estimate: $\hat{\mathcal{H}} = \mathcal{H}''$
                keep hypothesis for next measurement update: $\mathcal{H} = \mathcal{H}''$
            else
                compute track identification weights of multiple hypotheses $\mathcal{H}''$
                pruning: remove merged and low weight hypotheses of $\mathcal{H}'', \rightarrow \mathcal{H}'''$
                choose most likely output estimate $\hat{\mathcal{H}}$ of $\mathcal{H}'''$
                keep remaining hypotheses for next measurement update: $\mathcal{H} = \mathcal{H}'''$
            end if
        end if
    end if
end loop
```

Figure 6.9: Train localization with MHT filter, pre-processed measurements, and a track map.

transition with $\mathcal{H}'$, and with $\mathcal{H}''$ after the second transition. The number of hypotheses changes at each switch passed facing in the map transition. A pruning step keeps or removes hypotheses and the set of hypotheses is now denoted with $\mathcal{H}'''$, which is $\mathcal{H}$ for the next time step.

6.3 RailSLAM Method

The RailSLAM algorithm estimates the most probable train location in combination with the track map. Figure 6.10 shows the block diagram of the RailSLAM algorithm. The train localization and trajectory estimation computes a train location and updates the trajectory with information from the track map and pre-processed measurements. This estimation will be also referred to as train state estimation. The mapping function generates or updates the track map from the trajectory estimate with the associated train location. The mapping function contains further a buffer for a delayed mapping in order to resolve certain hypothesis after switches before the track maps of each hypothesis is updated. The output of the RailSLAM method is the topological train location and an updated track map. The implementation of the train localization and trajectory estimation comprises several parts and functions: a train localization with and without a prior track map, a trajectory estimation with and without prior track map and an estimation of the topology with a switch discovery detector and a mapping-phase estimation.

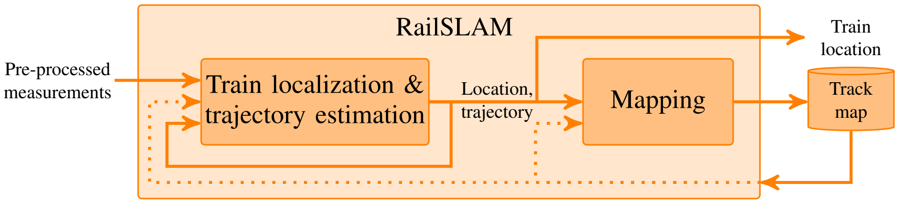

RailSLAM
Pre-processed measurements
Train localization & trajectory estimation
Location, trajectory
Mapping
Train location
Track map

Figure 6.10: Block diagram of the RailSLAM method.

The estimation states of train state, trajectory point, track map geometry and measurements are recapitulated for completeness. The train state T contains topological pose, velocity and trajectory variables (cf. Eq. (5.1), Eq. (5.2), Eq. (5.5)) at the time step k:

$$T_k = \{ \underbrace{id, s, o^{\mathrm{t}}}_{\text{topo. pose } T^{\text{topo}}}, v, \underbrace{\varphi, \lambda, h, \phi, \theta, \psi, c^{\phi}, c^{\theta}, c^{\psi}}_{\substack{\boldsymbol{\mu}^{\text{traj}}, \boldsymbol{\Sigma}^{\text{traj}}, \\ \text{trajectory point } T^{\text{traj}}}} \}_k. \tag{6.42}$$

A trajectory point contains the geometric variables with position, attitude angles and curvatures in the train frame. Each trajectory point T_k^{traj} is represented with a mean vector $\boldsymbol{\mu}_k^{\text{traj}}$ and covariance matrix $\boldsymbol{\Sigma}_k^{\text{traj}}$. In the following, a sequence of these points is called a trajectory. A track map sample $m_{id,s}$ at a certain topological location contains the track geometry in the track frame (cf. Eq. (6.6)):

$$m_{id,s} = \underbrace{\{\varphi, \lambda, h, \phi, \theta, \psi, c^{\phi}, c^{\theta}, c^{\psi}\}}_{\substack{\boldsymbol{\mu}^{\text{m}}, \boldsymbol{\Sigma}^{\text{m}}, \\ \text{track geometry at location } id, s}}. \tag{6.43}$$

Each track geometry point $m_{id,s}$ is represented with a mean vector $\boldsymbol{\mu}_{id,s}^{\text{m}}$ and covariance matrix $\boldsymbol{\Sigma}_{id,s}^{\text{m}}$. A track geometry point is accessed from the track map at a topological location with Eq. (6.6). The measurement Z_k^{P} is represented with a mean vector $\boldsymbol{z}_k^{\text{P}}$ and covariance matrix $\boldsymbol{\Sigma}_k^{P}$. The pre-processed measurements are defined in Section 6.1, Eq. (6.1) and these measurements match with most of the variables of T_k:

$$Z_k^{\text{P}} = \{v, \underbrace{\varphi, \lambda, h, \phi, \theta, \psi, c^{\phi}, c^{\theta}, c^{\psi}}_{\substack{\boldsymbol{z}^{\text{traj}}, \boldsymbol{\Sigma}^{\text{z,traj}}, \\ \text{trajectory measurement } Z^{\text{traj}}}}\}_k. \tag{6.44}$$

In Section 5.3.2, the online SLAM posterior factorization has been presented. The following implementation uses a multiple hypothesis tracking (MHT) approach for the estimation of the online SLAM posterior. The main factorization with the mapping and train state estimation part (Eq. (5.10)) is extended with hypotheses and a mixture weight for each hypothesis j:

$$\text{p}(T_k,B_k,M_k|Z_{1:k},U_{1:k}) \propto \sum_j \underbrace{w_k^j}_{\substack{\text{mixture weight of} \\ j\text{'th hypothesis}}} \cdot \underbrace{\text{p}(M_k^j|T_k^j, Z_{1:k})}_{\substack{\text{track map} \\ \text{estimation}}} \cdot \underbrace{\text{p}(T_k^j,B_k|Z_{1:k},U_{1:k}^j)}_{\substack{\text{train state} \\ \text{estimation}}}. \tag{6.45}$$

The track map estimation updates the track map with information from the train state T_k, and is explained in Section 6.3.3. A hypothesis $\mathcal{H}^j$ of RailSLAM contains a mixture weight w_k^j, a train state hypothesis T_k^j, a mapping phase ξ_k^j, and a track map hypothesis M_k^j:

$$\mathcal{H}_k^j = \{w_k^j, T_k^j, \xi_k^j, M_k^j\}. \tag{6.46}$$

The mapping phase ξ is an additional and auxiliary variable and can be associated to the train state. This variable indicates either the white-space phase or the prior-map phase. The train state estimation contains the train localization and trajectory estimation. Depending on the mapping phase, the implementations of the localization and trajectory estimation are different. Hypotheses are needed to estimate switches and switch ways. Hypotheses arise from a decision of the control input U^{sw} for a particular switch way. If a switch is known in the track map, it is similar as in the MHT localization one hypothesis for each switch way. If a switch is unknown in the track map, there is one hypothesis with a switch and one other without.

	Expansion stages	Init. map	Data association of measurements
1	White space mapping	no	1-D location only
2	Along-track SLAM	yes	along-track localization, known switch ways
3	Prior-map SLAM	yes	full train localization, new track features
4	RailSLAM	opt.	full train loc., new tracks, known switches
5	Full RailSLAM	opt.	full train loc., new tracks, new switches

Table 6.1: Expansion stages of RailSLAM.

Table 6.1 shows five expansion stages of the RailSLAM algorithm. The first expansion stage considers a simple white-space mapping with no initial prior map. The second stage uses an initial map and a known switch-way path. This stage is suitable for an update of track data in a mapping process if the train route and switch ways are known in advance. The third stage is similar to the second stage with the difference, that the switch ways are unknown now. There, the localization estimates the switch way. Second and third stage can enhance the initial map with new and improved track features. The forth stage contains white-space and prior-map estimation, as the initial map is optional and new tracks can be estimated. The detectors of new switches are bypassed with true information. The last expansion stage is the full RailSLAM with switch detectors for the estimation of undiscovered and new switches. There, the switches can be detected if a new path is chosen from a known track. The initial map is again optional. RailSLAM (stage 4) and the full RailSLAM (stage 5) can start without a track map or with an incomplete track map in terms of missing tracks and switches.

6.3.1 Algorithm

The algorithm of RailSLAM with a delayed mapping is shown in Fig. 6.11 and explained in the following. As defined in Eq. (5.10) and Eq. (6.45),

```
    Algorithm: Online RailSLAM with delayed mapping
    Input: pre-processed measurements $Z^{\mathrm{P}}$, optional: prior track map $M$
    Output: most likely topological location ($\hat{T}^{\mathrm{topo}}$), updated track map $M$
 1: initialize track map or load track map, if available
 2: initialize all $\mathcal{H}_0$ hypotheses by first position
 3: loop
 4:     if new pre-processed meas. $Z^{\mathrm{P}}$ available then
 5:         time step: $k = k + 1$, $\Delta t = t_k - t_{k-1}$
 6:         if train is moving then
 7:             if train starts moving then
 8:                 memorize time step when train starts moving: $k_0 = k$
 9:                 if train is reversing then
10:                     change all white-space hypotheses of $\mathcal{H}$ to prior-map phase
11:                 end if
12:             end if
13:             predict with $Z^{\mathrm{P}}$ for each white-space hypot. in $\mathcal{H}$ (Fig. 6.12)
14:             localize & predict with $Z^{\mathrm{P}}$ for each prior-map hypot. in $\mathcal{H}$ (Fig. 6.13)
15:             assign trajectory from $Z^{\mathrm{P}}$ for each white-space hypot. in $\mathcal{H}'$ (Eq. (6.47))
16:             update trajectory with $Z^{\mathrm{P}}$ for each prior-map hypot. in $\mathcal{H}'$ (Eq. (6.48))
17:             update mixture weights with meas. $Z^{\mathrm{P}}$ of each hypothesis in $\mathcal{H}'$
18:             pruning: remove low weighted hypotheses of $\mathcal{H}', \to h''$
19:             buffer for mapping: add remaining hypotheses to sequence $\mathcal{H}''_{k_0:k}$
20:             compute most likely location hypothesis ($\hat{T}^{\mathrm{topo}}$ estimation) from $\mathcal{H}''$
21:             keep remaining hypotheses for next measurement update: $\mathcal{H} = \mathcal{H}''$
22:         else if train stops moving then
23:             compute track map $M_k$ of each hypothesis sequence in $\mathcal{H}''_{k_0:k}$ (Fig. 6.14)
24:             clear buffered data and all track map variants of obsolete hypotheses
25:         end if
26:     end if
27: end loop
```

Figure 6.11: Algorithm: Online RailSLAM with delayed mapping.

the algorithm splits in a train state estimation part and a mapping part. The train state is estimated while the train is moving in the lines 7-17 in Fig. 6.11. The map is updated once the train stops moving in line 19.

At first, the condition of a reversing train is checked in line 7 from the motion data. If the train is reversing, all white-space hypotheses are changed to the prior-map phase (line 8) because the train runs now on a previously recorded track. The train state estimation with localization and trajectory estimation is in lines 10-14. The train state estimation depends on the mapping phase ξ because the prediction and update functions are different with or without a known map. The train state prediction of all white-space hypotheses in line 10 contains a transition with a switch estimation and is presented in Figure 6.12. The localization and prediction of the trajectory for all prior-map hypotheses is in line 11, and presented in Figure 6.13. Within these prediction steps (line 10-11), the mapping phase ξ can change or additional hypotheses can be generated at switches. Therefore the hypothesis set is denoted with $\mathcal{H}'$ in the next line. Line 12 contains the trajectory assignment from measurements for all white-space hypotheses in $\mathcal{H}'$ (see Eq. (6.47)). Line 13 contains the trajectory update function with measurements for each prior-map hypothesis in $\mathcal{H}'$ (see Eq. (6.48)). In line 14, the mixture weights are updated with appropriate measurements in the case that there are multiple hypotheses. Unlikely hypotheses with a low weight are removed in line 15 and the remaining set of hypotheses is denoted with $\mathcal{H}''$. A mapping buffer in line 16 stores the trajectories of each hypotheses for the delayed mapping. This buffer, represented with the set of hypotheses series $\mathcal{H}''_{k_0:k}$, contains a sequence of data for each remaining hypothesis. A sequence contains the mapping phase, topological locations and estimated trajectory from the point in time, when the train starts moving (k_0) until the current time step k. Finally, the most likely train location is computed in line 17.

The track map is updated once at stand-still in this implementation. The benefit of a delayed mapping is higher computational efficiency. Many hypothesis will be deleted after a certain time and the delayed mapping computes only track maps of the remaining hypothesis. Ideally, there is only one hypothesis left. The mapping function of line 19 for one hypothesis is shown in Figure 6.14. An additional map maintenance deletes obsolete track maps and clears the buffer after the mapping process (line 20).

6.3.2 Train State Estimation

This section presents more details on the train state estimation from the RailSLAM algorithm (lines 11-14 in Fig. 6.11). The train state estimation for the white-space and the prior-map case are each presented with a prediction and update function.

Prediction of White-space Phase

A white-space hypothesis has no information about a prior or initial map. Figure 6.12 shows the white-space prediction function for one hypothesis.

The white-space prediction function is called in line 10 of the RailSLAM algorithm (Fig. 6.11) for each white-space hypothesis. The topological location is estimated in along-track direction from the last topological pose $T^{\text{topo}}_{k\text{-}1}$ and the velocity measurement (line 1, Fig. 6.12). The track ID remains the same, as long as the RailSLAM hypothesis remains in the

```
    Function: White-space Prediction
    Input: pre-proc. measurements $Z^{\mathrm{P}}$, white-space hypothesis $\mathcal{H}^j$
    Output: updated hypothesis $\mathcal{H}^j$
 1: predict $T^{\mathrm{topo}}$ with velocity measurement input
 2: get results of merge detector with $Z^{\mathrm{P}}_{k'':k}$, $T^j_{k'':k}$, and prior map $M^j_{k'}$
 3: if new merging switch detected then
 4:     $\xi^j_k$ changes to prior-map phase
 5:     memorize new switch
 6:     assign new switch path $T^{\mathrm{topo},j}_{k'':k}$ from detector
 7: else if circular loop closure detected then
 8:     $\xi^j_k$ changes to prior-map phase
 9:     assign new path $T^{\mathrm{topo},j}_{k'':k}$ from detector
10: end if
```

Figure 6.12: White-space prediction function.

white-space phase. A hypothesis transition of the mapping phase ξ from white-space to prior-map phase can occur with a reversing train, a merging switch or a circular loop closure as described in Fig. 5.4. The reverse motion is already handled in the main RailSLAM algorithm (Fig. 6.11). More challenging are the merging switch and the loop closure. A merging switch detector evaluates possible merging scenarios from a sequence of time steps between k'' and k and decides for a change of the topological location and mapping phase ξ of the current hypothesis (line 2). In the case that a switch or a closed loop is detected, the path of the hypothesis history $T^{\mathrm{topo}}_{k'':k}$ is changed between the time step of the switch start or closed loop and the current time step k.

Localization and Trajectory Prediction of Prior-map Phase

Figure 6.13 shows the prior-map localization and trajectory prediction function for one hypothesis. This function is called in line 11 of the RailSLAM algorithm (Fig. 6.11) for each prior-map hypothesis in $\mathcal{H}$.

In line 1 of Fig. 6.13, the location is similarly estimated as in the MHT localization: A new track location is computed from a measured displacement and the map-transition function Eq. (6.5). At this point, the track map can be exceeded to an unknown area. This event is called *exceeded map* and is easily detected in the train transition function when the known track in the track map is exceeded. A continued prior-map phase will instead update the along-track location (line 3) and recompute the transition on the track map (line 4), followed with a check if the track map is

```
    Function: Prior-Map Localization and Trajectory Prediction
    Input: pre-proc. meas. $Z^{\mathrm{P}}_{k':k}$, prior-map hypothesis $\mathcal{H}^j$
    Output: hypothesis $\mathcal{H}^j$ or mult. hypotheses $\mathcal{H}$
 1: compute track map transition of $\mathcal{H}^j$ with velocity measurement input
 2: if track map is not exceeded then
 3:     update along-track Kalman filter of each $\mathcal{H}'$ with measurements
 4:     compute track map transitions of $\mathcal{H}'$ from along-track correction
 5:     if track map is not exceeded then
 6:         get trajectory $\boldsymbol{\mu}^{\mathrm{traj},j}_{k|k\text{-}1}$ and cov. $\boldsymbol{\Sigma}^{\mathrm{traj},j}_{k|k\text{-}1}$ from $M^j_{k'}$ and predicted $T^{\mathrm{topo},j}_k$
 7:         get results of switch discovery detector with $Z^{\mathrm{P}}_{k'':k}$, $T^j_{k'':k}$, and $M^j_{k'}$
 8:         if new branching switch detected then
 9:             $\xi^j_k$ changes to white-space phase
10:             memorize new switch
11:             assign new switch path $T^{\mathrm{topo},j}_{k'':k}$ from detector
12:         end if
13:     end if
14: end if
15: if track map is exceeded then
16:     $\xi^j_k$ changes to white-space phase
17:     predict $T^{\mathrm{topo},j}_k$ with remaining displacement
18: end if
```

Figure 6.13: Prior-map localization and trajectory prediction function.

exceeded. The predicted trajectory and covariance are received from the track map and from the predicted topological pose with Eq. (6.7) in line 6. In line 7, the switch detector evaluates a possible switch from a sequence of measurements and past train states. If the switch estimator decides for a *branch to unknown track*, a new switch is created and a new path $T^{\mathrm{topo}}_{k'':k}$ is assigned that includes the switch start. The new switch with its location and connections are memorized in the mapping buffer. For the exceeded map scenario, the mapping phase changes and the remaining distance is predicted in the same way as in the white-space case (lines 16-17).

Trajectory Assignment in White-space Phase

The white-space trajectory assignment function assigns the trajectory from measurements Z^{P} without a prior map. This function is called in line 12 of the RailSLAM algorithm (Fig. 6.11). The trajectory measurements Z^{traj}_k are directly assigned to trajectory values $T^{\mathrm{traj},j}_k$. The measurement vector $\boldsymbol{z}^{\mathrm{traj}}$ and the measurement covariance $\boldsymbol{\Sigma}^{\mathrm{z,traj}}$ contains all values of the pre-processed measurements Z^{P} except the velocity. The assignment of the trajectory for the j-th hypothesis is:

$$\boldsymbol{\mu}_k^{\mathrm{traj},j} = \boldsymbol{z}_k^{\mathrm{traj}}, \tag{6.47a}$$

$$\boldsymbol{\Sigma}_k^{\mathrm{traj},j} = \boldsymbol{\Sigma}_k^{\mathrm{z,traj}}. \tag{6.47b}$$

Trajectory Update in Prior-map Phase

The second function in the prior-map phase is the prior-map trajectory update function and is called in line 13 of the RailSLAM algorithm (Fig. 6.11). This function contains the update step of a Kalman filter from a prior known trajectory and measurements. The measurement model $\boldsymbol{H}$ is an here an identity matrix, see Eq. (6.47). As an option, a different measurement matrix can exclude certain variables from an update of $T_k^{\mathrm{traj},j}$ if Z_k^{traj} contains only limited set measurements.
The Kalman filter update of the trajectory is given by:

$$\boldsymbol{K} = \boldsymbol{H}\boldsymbol{\Sigma}_{k|k\text{-}1}^{\mathrm{traj}}\boldsymbol{H}^T(\boldsymbol{\Sigma}_{k|k\text{-}1}^{\mathrm{traj}}\boldsymbol{H}^T + \boldsymbol{\Sigma}_k^{\mathrm{z,traj}})^{-1}, \tag{6.48a}$$

$$\boldsymbol{\mu}_{k|k}^{\mathrm{traj}} = \boldsymbol{\mu}_{k|k\text{-}1}^{\mathrm{traj}} + \boldsymbol{K}(\boldsymbol{z}_k^{\mathrm{traj}} - \boldsymbol{H}\boldsymbol{\mu}_{k|k\text{-}1}^{\mathrm{traj}}), \tag{6.48b}$$

$$\boldsymbol{\Sigma}_{k|k}^{\mathrm{traj}} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{\Sigma}_{k|k\text{-}1}^{\mathrm{traj}}. \tag{6.48c}$$

The predicted trajectory $\boldsymbol{\mu}_{k|k\text{-}1}^{\mathrm{traj}}$ and covariance $\boldsymbol{\Sigma}_{k|k\text{-}1}^{\mathrm{traj}}$ are obtained from the track map at the predicted location in the prior-map localization and prediction function, see line 6 of Fig. 6.13.

6.3.3 Track Map Generation and Update

The RailSLAM track map generation and update function is called in the RailSLAM algorithm (line 19, Fig. 6.11) and is shown in Fig. 6.14. This function incorporates information from the train location and trajectory estimation and updates the track map. The train state buffer contains J hypotheses at the time of a track map update. For each remaining hypothesis $\mathcal{H}^j$, there is a sequence of train state estimates $T_{k_0:k}^j$ since the last track map update at time step k_0.

In Fig. 6.14, the first step of the track map function is the topology update with new tracks and new switches (line 1). If a new switch is inserted into an existing track, this track is split in two tracks and a third track is connected. Therefore, new track IDs are generated, new connections are created, and existing connections are updated with new links. New tracks from a white-space mapping are generated.

```
   Function: Track Map Generation and Update
   Input: sequence of one hypothesis $\mathcal{H}^j_{k_0:k}$, with train states $T^j_{k_0:k}$ and track map $M^j$
   Output: updated track map $M^j$
1: topology update: add new tracks (white space) and new switches
2: for all track IDs $id$ in $\mathrm{T}^j_{k_0:k}$ do
3:     get trajectory $T^{\mathrm{traj},j}_{k':k''}$ and 1-D locations $s^j_{k':k''}$ that are associated to $id$
4:     convert trajectory to track geometry with Eq. (6.49)
5:     interpolate track geometry with new sample grid (Eq. (6.50))
6:     update track $id$ of track map $M^j$: save track samples $s^{\mathrm{m}}, \boldsymbol{\mu}^{\mathrm{m}}, \boldsymbol{\Sigma}^{\mathrm{m}}$
7:     update common sample points of connected tracks
8: end for
```

Figure 6.14: Track map generation and update function.

The second step is the track geometry update or generation for each track. Therefore, the function determines the sequence of trajectory estimates $T^{\mathrm{traj}}_{k':k''}$ and the 1-D locations $s_{k':k''}$ that are associated to the current track (line 3). The trajectory values are in the train frame and a function converts the values to the track geometry in the track frame with the current track-to-train direction from the topological pose (line 4), according to Table 3.3:

$$\underbrace{\{\boldsymbol{\mu}^{\mathrm{m}}_k, \boldsymbol{\Sigma}^{\mathrm{m}}_k\}}_{\substack{\text{track geometry}\\ \text{track frame}}} = \mathrm{f}_{\text{train-to-track.}}\big(\underbrace{\boldsymbol{\mu}^{\mathrm{traj}}_k, \boldsymbol{\Sigma}^{\mathrm{traj}}_k}_{\substack{\text{train trajectory,}\\ \text{train frame}}}, \underbrace{o^{\mathrm{t}}_k}_{\substack{\text{track to train}\\ \text{frame orient.}}}\big). \tag{6.49}$$

A train state sequence contains samples at a constant frequency, e.g. at 1 Hz, in contrast to a track geometry sequence with 1-D location dependent and equidistant samples, e.g. every 10 m. Therefore, new track geometry samples are generated with an interpolation function from the time-dependent sample points $s_{k':k''}$, time-dependent track geometry values $\boldsymbol{\mu}^{\mathrm{m}}_{k':k''}$ and a new sampling grid $s^{\mathrm{m}}_{s':s''}$:

$$\underbrace{\boldsymbol{\mu}^{\mathrm{m}}_{s':s''}}_{\substack{\text{new geometry samples}\\ \text{over 1-D locations}}} = \mathrm{f}_{\text{interpol.}}\big(\underbrace{s_{k':k''}}_{\substack{\text{1-D locations}\\ \text{over time}}}, \underbrace{\boldsymbol{\mu}^{\mathrm{m}}_{k':k''}}_{\substack{\text{geometry samples}\\ \text{over time}}}, \underbrace{s^{\mathrm{m}}_{s':s''}}_{\substack{\text{constant sample}\\ \text{grid of 1-D locations}}}\big). \tag{6.50}$$

The indexes indicate either samples from location s' to s'' or a sequence of time steps from k' to k'' within the current track *id*. The sequence of covariances $\boldsymbol{\Sigma}^{\mathrm{m}}_{s':s''}$ are interpolated with a nearest neighbor method from the time-dependent covariances $\boldsymbol{\Sigma}^{\mathrm{m}}_{k':k''}$. Line 5 contains the track geometry interpolation. The sample grid is either a new grid with constant distance spacings or a reused sample grid from a prior map. Afterward, the track samples are stored to the map in line 6. In the prior-map phase, the previous track samples are simply overwritten, because the information fusion of prior map and new measurements has been carried out in the train state estimation. The last step corrects the common sample points of connected tracks with switches or crossings (line 7).

Figure 6.15 shows the results of the track map generation and update function for an example scenario with a track map update of existing tracks and new tracks.

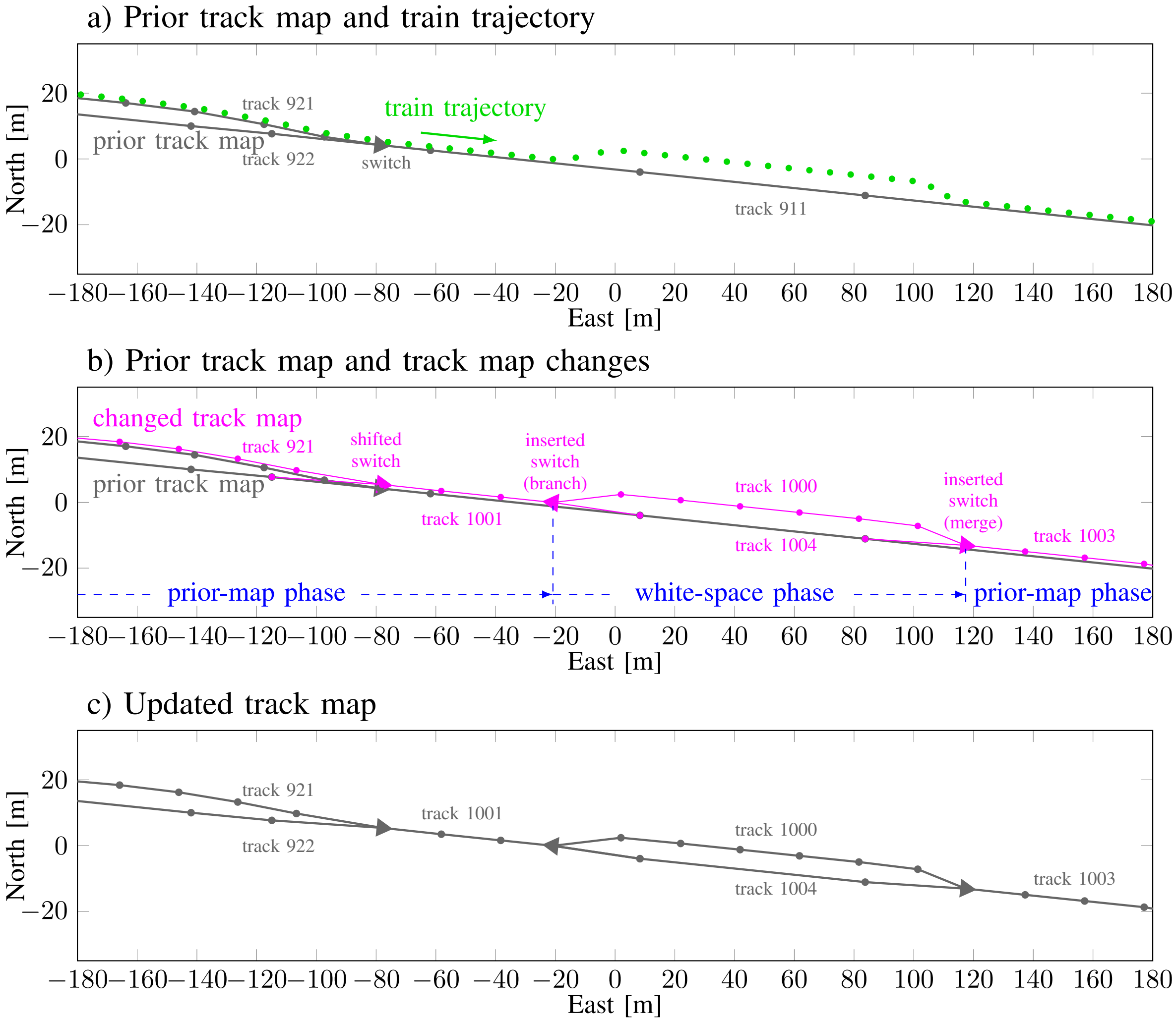


Figure 6.15: RailSLAM track map update from location referenced train trajectory.

In Figure 6.15 a), the 2-D positions are shown for an existing track map in gray and the train trajectory sequence with green dots. This example contains a branching switch, a short new track, and a merging switch back to the prior-known track 911. Figure 6.15 b) shows the changed parts of the updated track map (magenta) next to the previous track map (gray).

It can be seen, that the new track map is slightly shifted and there are two new switches. Further, the shifted switch causes also a change of the first sample point of those tracks that are not overrun, as for the track 922 for instance. The final result of the updated map is shown in Fig. 6.15 c).

Chapter 7

Experiment Setup and Evaluation Methods

This chapter presents the train measurements with the measurement equipment, sensors and installation on the train, as well as a description of the train route of several runs. Furthermore, the following presents ground-truth references for the switch ways and the track map. The evaluation methodology comprises methods for a track-selective evaluation, a cross-track evaluation for position measurements and a track map evaluation.

7.1 Train Measurements

7.1.1 Measurement Setup

The train measurements were carried out with a specially designed measurement box. This box is shown in Figure 7.1 and records synchronized and time-stamped sensor data. The measurement box starts recording data after power-up automatically. The components of the box comprise a railway certified power supply unit for the 24 V train power, several sensors as well as an electronic board with the data logger and storage. The sensors are a GNSS receiver (u-blox LEA 6T), a box-mounted MEMS IMU with magnetometer (Xsens MTi), a box-mounted FOG IMU (KVH 1750) and an external MEMS IMU with magnetometer (Xsens MTi). The electronic board (Figure 7.1) was designed as an extension for a microcontroller board with a 32-bit ARM microcontroller. This extension board contains an SD-card slot, a 10 MHz temperature stabilized clock and hardware interfaces for the sensors.

All measurements are stored in combination with a timestamp. The timestamps are vital for the temporal alignment of the multi-sensor data in

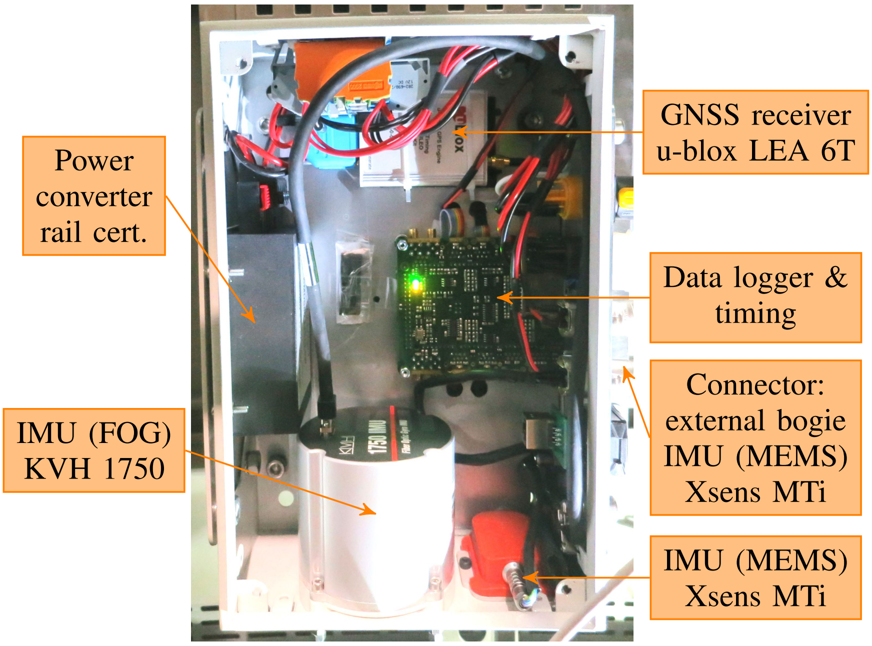

Power converter rail cert.
GNSS receiver u-blox LEA 6T
Data logger & timing
Connector: external bogie IMU (MEMS) Xsens MTi
IMU (FOG) KVH 1750
IMU (MEMS) Xsens MTi

Figure 7.1: Components of the train measurement box.

the post-processing and evaluation methods. Therefore, the electronic extension board has a common clock with a 10 MHz frequency and a counter register within the microcontroller. The frequency is divided for the 100 Hz and 200 Hz trigger signals for the IMU measurements. The 1 PPS (pulse-per-second) signal of the GNSS receiver is captured with the microcontroller. The IMU measurements are triggered from the common clock and each IMU sends a message to the microcontroller. Once this message is received by the controller, the timestamp of the trigger event is linked with the message and stored on SD-card. A post-processing step computes absolute timestamps from the 10 MHz counter values, the 1 PPS observations and the absolute time and date information from the GNSS messages. The absolute timestamps are expressed in the Unix timestamp format, which is a millisecond counter from the 1.1.1970 at 0:00:00 UTC time. The measurement data set comprises 1 Hz GNSS data (position, velocity, time from GPS), two MEMS IMUs with 200 Hz data rate, a FOG IMU with 100 Hz data rate, and video data.

The train setup is shown in Fig. 7.2 with the measurement box mounted in the rack behind the train driver (upper right picture). The external IMU was mounted on top of the front bogie, as shown in the lower right picture. The GNSS antenna was mounted below the fiberglass roof for electrical safety reasons, because the train route contained electrified parts with a high-voltage catenary. The antenna placement was centered in the cross direction of the train. A special camera (dash-cam) with GPS time-stamped video was installed behind the front windshield for the switch-way reference.

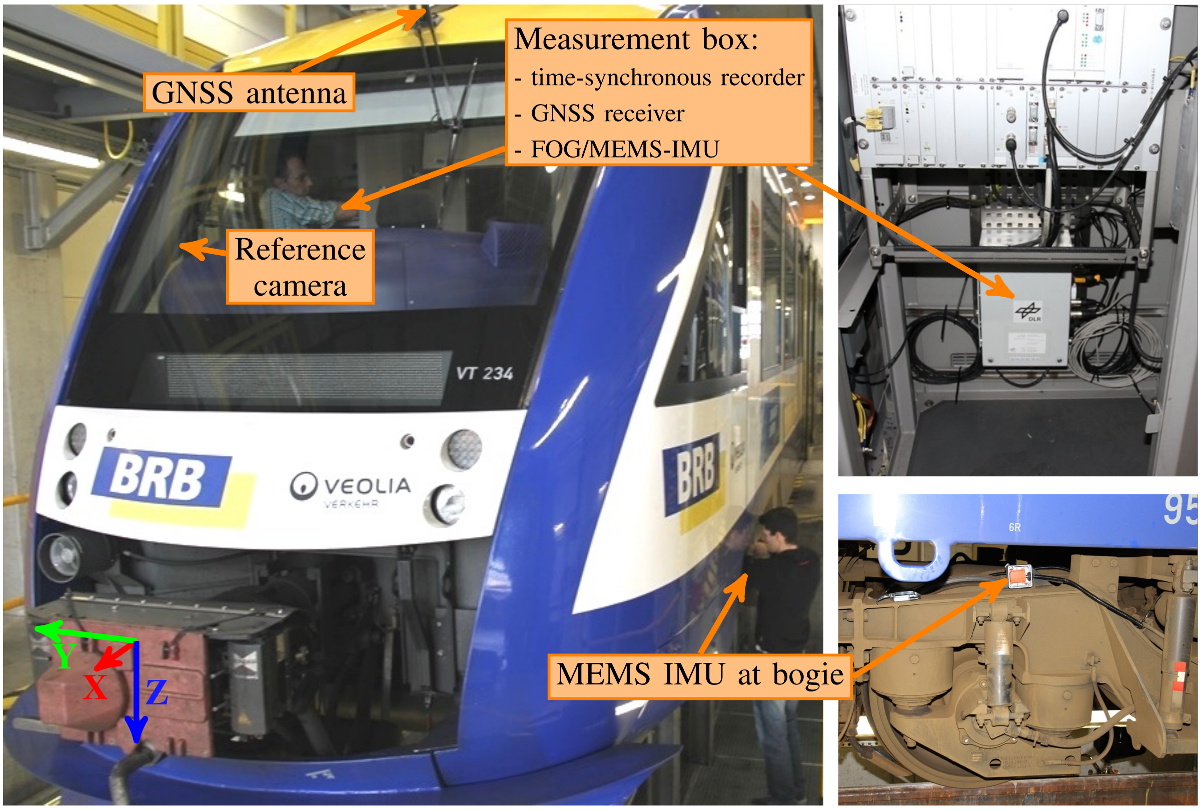


Figure 7.2: Measurement setup on the regional train.

7.1.2 Train Experiment

The measurements were recorded on the regional train ”Alstom Coradia Lint41” under regular passenger service conditions. This train can travel up to 120 km/h and has two driver’s cabs for a two-side operation. The Lint41 is 41 m long with a mass of 67.5 t, has 140 seats and is powered by two diesel engines [83]. The train runs in a two-way commuting service between stations on the railway line Augsburg - Ingolstadt. Fig. 7.3 shows the map of the train route and the stations along the line.

Figure 7.4 shows the topological schematic of the train route and the travel directions for ten example runs. This track topology is later estimated by the RailSLAM algorithm. The train runs on some tracks in one direction and on some tracks in both directions. Stations have often departure or destination tracks in a specific direction and some of the tracks are directional tracks. Most of the tracks with other and parallel tracks in the vicinity are between station Augsburg and station Friedberg. A single

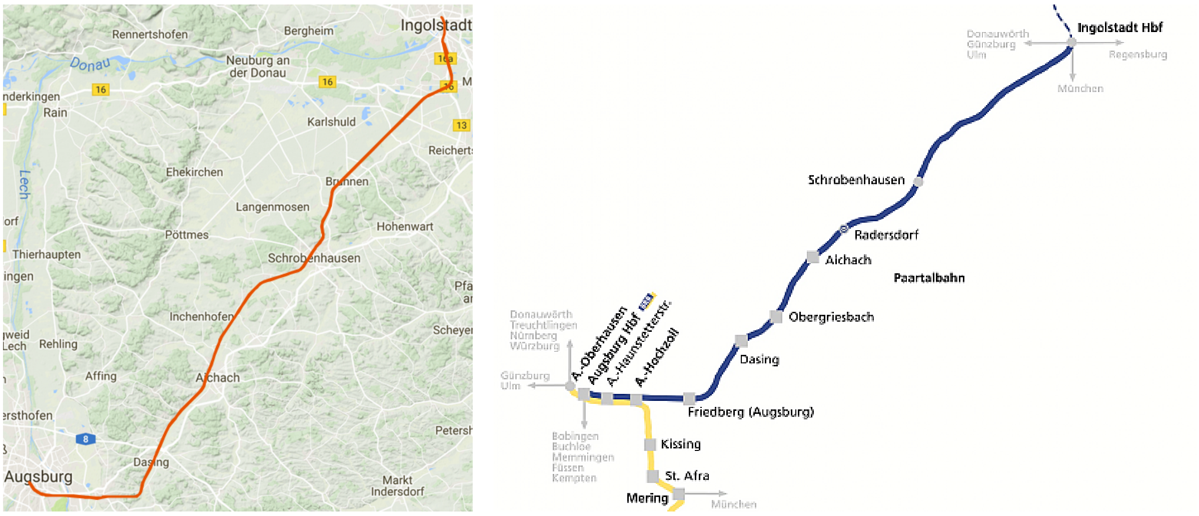


Figure 7.3: Train route from Augsburg to Ingolstadt. (sources left: Google maps, Kartendaten 2017, GeoBasis-DE/BKG 2009, right: Bayerische Regio Bahn 2014)

track scenario is indicated in Figure 7.4 near Friedberg station. All train runs pass this single track section and each iteration of the RailSLAM track map is evaluated within this single track.

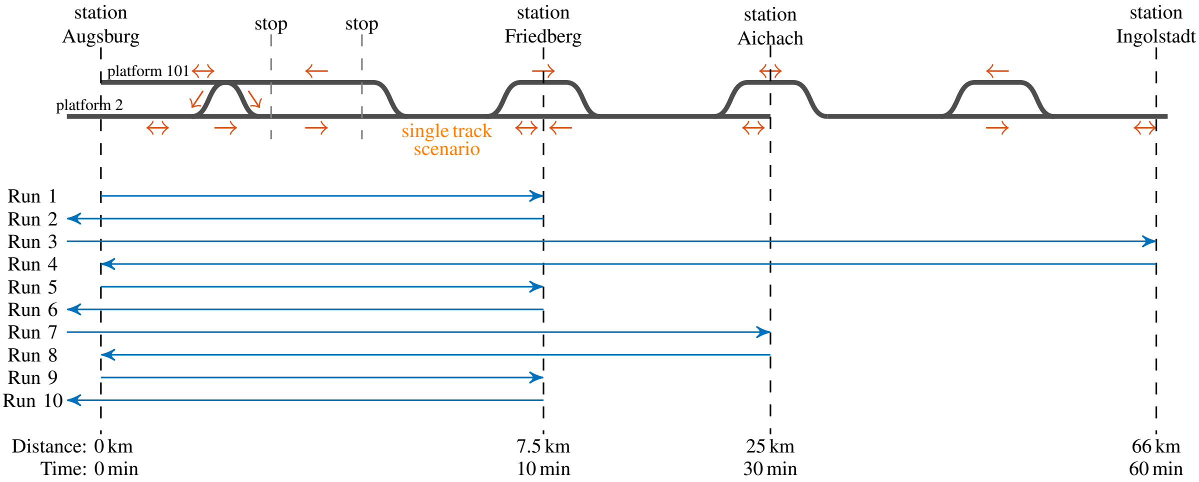


Figure 7.4: Train route from Augsburg to Ingolstadt.

Table 7.1 shows the schedule of the train runs with motion direction, travel time, distance, and the number of branching switches. All runs start or finish in Augsburg and pass all stations to Friedberg. Some runs continue to Aichach or Ingolstadt. A second data set with several train runs is listed in the Appendix E.

Run	**Departure station**	**Arrival station**	**Motion direction**	**Branching switches**	**Time**	**Distance [km]**
1	Augsburg	Friedberg	backward	8	10 min	8
2	Friedberg	Augsburg	forward	7	10 min	8
3	Augsburg	Ingolstadt	backward	22	1 h	66
4	Ingolstadt	Augsburg	forward	20	1 h	66
5	Augsburg	Friedberg	backward	8	10 min	8
6	Friedberg	Augsburg	forward	7	10 min	8
7	Augsburg	Aichach	backward	11	30 min	25
8	Aichach	Augsburg	forward	9	30 min	25
9	Augsburg	Friedberg	backward	8	10 min	8
10	Friedberg	Augsburg	forward	7	10 min	8
				107	4 h	230

Table 7.1: Train routes (data set 1)

7.2 Evaluation References

7.2.1 Labeled Train Route

The track-selective analysis requires known switch ways. A switch-way reference is obtained from a recorded video of the train run. Figure 7.5 shows one still picture of the video while the train passes a switch facing. The possible switch ways are either left or right and manually extracted from the video for each passed switch. For every run, a reference travel path can be computed with the map from a start position, the motion direction, and a series of true switch ways.

The upper plot of Figure 7.6 shows the geographic view of all tracks in gray, the traveled path of the train in orange and the positions of the branching switches in blue. The lower plot visualizes the train speed and the true switch ways of Run 1 over time. The train runs backwards to Friedberg and turns five times to the right switch way, three times to the left way and seven switches are passed trailing (merge). The train stops can be identified from the train speed. The branching switches are shown with the true switch ways of left and right over time in the lower plot. This combined visualization with the geographic plot and temporal plot is used in the analysis of the localization methods.

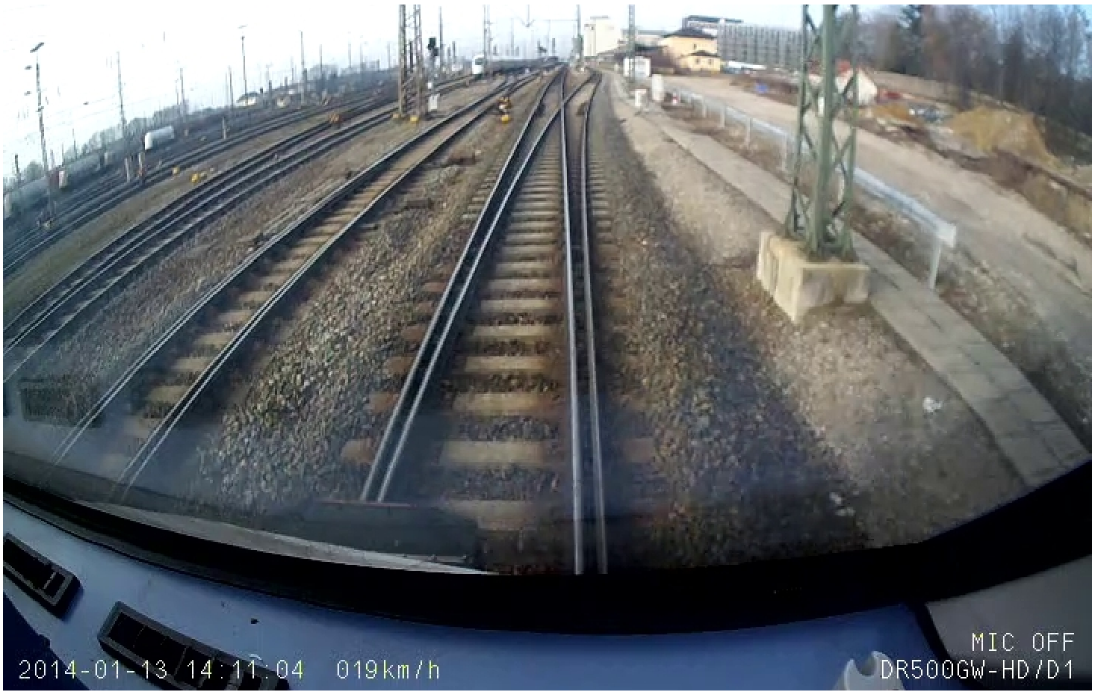


Figure 7.5: Still picture of a switch run from the reference video.

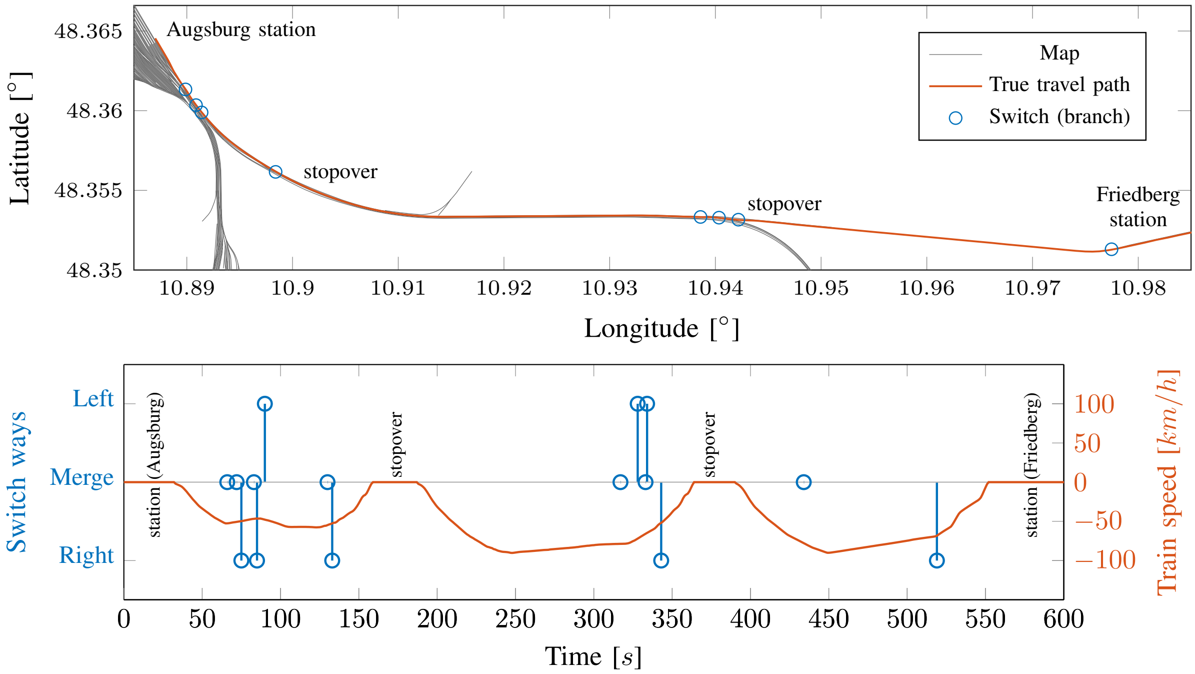


Figure 7.6: Run 1 over time from Augsburg main station to Friedberg station.

7.2.2 Reference Track Map

The estimated track map from the RailSLAM algorithm is evaluated with geo-referenced images from the local land surveying office *"Bayerische Vermessungsverwaltung"* [60], as described in Section 3.3.4. A geo-referenced image has position information associated with the center of each pixel. The quality of the geo-referenced image is defined by the metric resolution of one pixel as well as by the accuracy of the positions. The used areal images had a position accuracy of 50 cm and a pixel resolution of 20 cm [60]. The next step is a manual sampling of the railway tracks from that image. Figure 7.7 shows a section of the DOP image with Gauß-Krüger coordinates and the manually sampled reference line of the track. The localization methods are evaluated with a different track maps, but also with the reference track map. The reference track map exists only in the area between Augsburg and Friedberg. However, the evaluation with the reference track map considers only 77 passed switches on 80 km of travel distance for the ten runs. Nevertheless, this reference track map contains all parallel tracks and switches in this area.

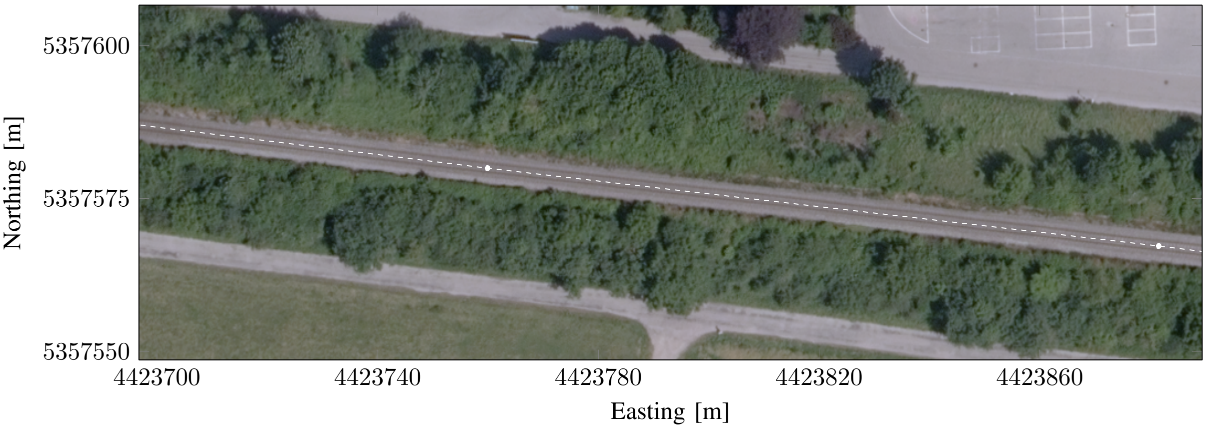


Figure 7.7: Geo-referenced digital orthophoto with a sampled track (white) of the reference track map.

7.3 Evaluation Methodology

7.3.1 Evaluation Method for Track-Selective Accuracy

The track-selective performance is a measure for the correct track estimation of the train localization, especially in switch scenarios or parallel track scenarios. This performance evaluation is used in a comparison with different localization methods, different parameters, different sensor data and different track maps. Therefore, an evaluation procedure compares a known reference route with the estimated track IDs. This reference route are the true track IDs that are obtained from the map and the switch-way reference.

The critical scenario for track selectivity is a switch passed facing with the two possible switch ways of left and right way. Fig. 7.8 shows a switch with different evaluation sections. The track estimate is evaluated as correct, if the estimated track ID is part of the reference route. At a switch, there is a tolerance range in order to allow a certain time or traveled distance for the switch-way estimation process. Some of the discriminative switch signatures are hard to separate at the switch start. Once the end of the switch is reached, the switch-way decision is evaluated with correct or wrong. This tolerance range is also shown in the figure between the switch start and the clearance point of the two different switch ways. A wrong estimate in the switch section or tolerance range is less severe, as only one train can occupy the switch at a time. Alternatively, a fixed tolerance range can be defined for simplicity reasons. In this thesis, a fixed tolerance range of 50 m is used. The possible outcomes of the track-selective evaluation ϵ at a time step k are:

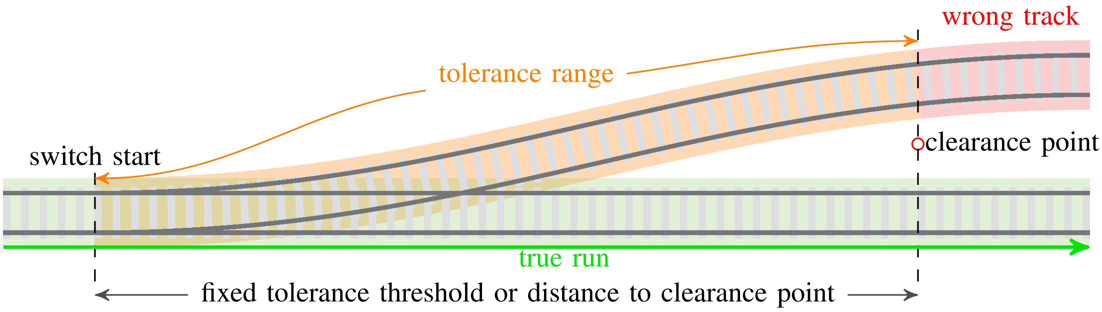


Figure 7.8: Track-selective evaluation at a switch with tolerance range and wrong track section for a true right (straight) run.

$$\epsilon_k = \begin{cases} \epsilon_k^{\text{ok}} \text{ (correct)} \\ \epsilon_k^{\text{tol}} \text{ (wrong, but tolerated before clearance point)} \\ \epsilon_k^{\text{err}} \text{ (wrong)} \end{cases} \tag{7.1}$$

The performance of a train localization method for track selectivity is evaluated statistically over different train runs. In a run analysis, the track-selective result can be visualized with the correct, wrong or tolerated outcome over time. The evaluation is computed from the localization output for each time step of a moving train and if switches or parallel tracks are in the vicinity: The number of correct-track evaluations N_{OK} is counted if the train is moving and other tracks are in the vicinity. The tolerated-wrong-track counts N_{tol} and wrong-track counts N_{err} are counted when the train is moving. N_ϵ is the combined count of N_{OK}, N_{tol} and N_{err}.

More compact performance figures are defined for performance conclusions and a comparison of different methods, parameters, and a comparison of track maps. There are three track-selective performance figures:

- The *track-selective accuracy* A_{TS} is a ratio of the accepted track-selective results and indicates how well a train localization method identifies the correct tracks. Accepted evaluations are correct outcomes and tolerated wrong-track outcomes, and a perfect evaluation reaches a ratio of 100%. The track-selective accuracy is defined with:

$$A_{\mathrm{TS}} = \frac{N_{\mathrm{OK}} + N_{\mathrm{tol}}}{N_\epsilon}. \tag{7.2}$$

- The *track-selective error ratio* E_{TS} counts all errors including the tolerated wrong tracks. A perfect evaluation with no wrong estimates reaches 0%. The error ratio is defined with:

$$E_{\mathrm{TS}} = \frac{N_{\mathrm{err}} + N_{\mathrm{tol}}}{N_\epsilon}. \tag{7.3}$$

- The *switch-way-detection accuracy* A_{SW} compares the number of a correct detected switch ways versus all passed branching switches. A switch way is correct, if the correct track is identified after the tolerance range. Any change to a wrong track afterward is also counted as a wrong switch-way detection.

The track-selective accuracy A_{TS} may be used to evaluate a design goal for the development of a train localization. The error rate E_{TS} considers also the tolerated errors and is more suitable for comparisons or the confirmation of improvements in a development or parameter adjustment process. The switch-way-detection accuracy A_{SW} with the number of branching switches indicates also the statistical size of the evaluation.

The track-selective performance is only evaluated in scenarios with parallel tracks in the vicinity, because track-selective errors occur only in the presence of switches and parallel tracks. A train run on a route with many single track scenarios would distort a comparative evaluation in favor of a better track-selective evaluation. The vicinity is defined in this thesis with a radius of 20 m around each estimated location. Additionally, the performance is only evaluated for times where the train is in motion because long standing periods distort the performance figures as well.

7.3.2 Evaluation Process for Train Localization Methods

The evaluation of the different localization methods is carried out with recoded measurements and also different track maps. Figure 7.9 depicts the functional elements and interfaces of the train localization and evaluation for track selectivity. There are several configurations possible: The pre-processing involves a processing stage with INS/GNSS and curvature computation or alternatively the direct sensor data. Additionally, IMU data of three different IMUs can be selected. The train localization methods involve the simple map-match method, particle filter and the MHT method.

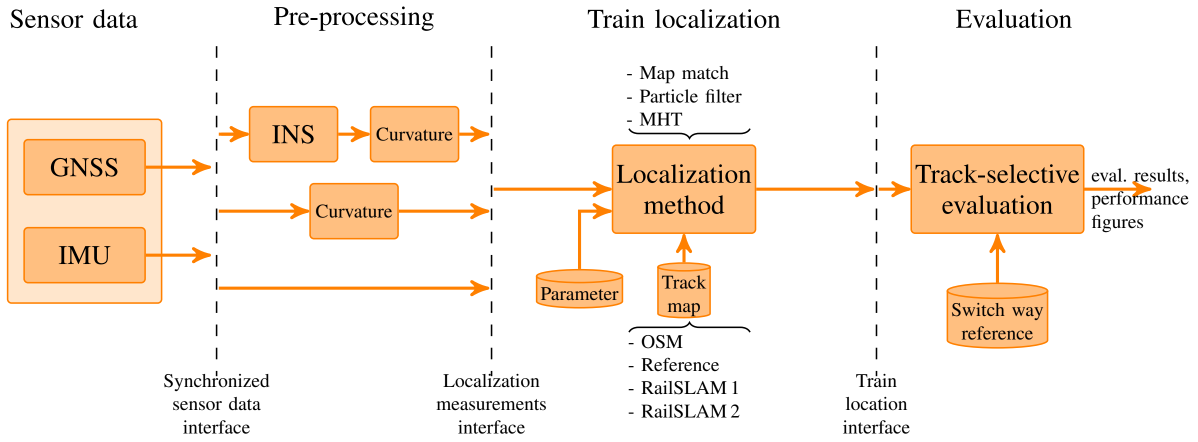


Figure 7.9: Architecture of the train localization evaluation with recoded measurements, pre-processing, different methods and different track maps.

For the presented train measurements, there are four different track maps available: An OSM (Open Street Map) based track map, a reference track map and two track maps generated with the RailSLAM method from recorded data. The first one (RailSLAM 1) is created from the same data set as used in the evaluations, the second map (RailSLAM 2) is computed from different data. Table 7.2 shows the differences of the track maps. The first two track maps of the table are created from 2-D position samples (φ, λ). The heading ψ and curvature c^{ψ} are calculated from the 2-D positions (see Section 3.3.4), and the height h is computed from height models. The OSM track map and reference track map are compa-

Track Map	Data Source	Geometry	Coverage
OSM	Open street map, position samples	$\varphi, \lambda, (h, \psi, c^{\psi})$	Augsburg-Ingolstadt, all tracks near route
Reference	geo-referenced aerial image, pos. samples	$\varphi, \lambda, (h, \psi, c^{\psi})$	Augsburg-Friedberg, all tracks near route
RailSLAM 1	RailSLAM method, meas. from runs 1-10	$\varphi, \lambda, h, \phi, \theta, \psi, c^{\phi}, c^{\theta}, c^{\psi}$	Augsburg-Ingolstadt, traveled tracks only
RailSLAM 2	RailSLAM method, meas. from other runs	$\varphi, \lambda, h, \phi, \theta, \psi, c^{\phi}, c^{\theta}, c^{\psi}$	Augsburg-Ingolstadt, traveled tracks only

Table 7.2: Track maps for the localization evaluation.

rable, as both track maps contain the same track network topology with the same amount of parallel tracks and switches along the train run. The RailSLAM track maps consider only the tracks where the train was running in the mapping phase. These track maps contain less parallel tracks and switches than the OSM track map or reference track map from aerial images. RailSLAM track map 1 is considered as an ideal map, because it is created from the same data as used for the localization evaluation. RailSLAM track map 2 is created from measurements of different runs and represents the more realistic use-case.

7.3.3 Evaluation Methods of the Track Map

A map-based train localization requires small deviations between measurements and the stored geometry of the track map. A deviation is called error if a ground-truth reference is used. The reference track map is used as ground-truth in this thesis. Further, a deviation will be called a residual, if the deviation is computed between measurements and a track map that is estimated from the same data. In this thesis, residuals are evaluated with the track map RailSLAM 1.

Mean Track Map

The mean track map can be considered as a simple and alternative track map generation approach compared to the SLAM based approach. This track map will be used to compare the errors of different iterative mapping approaches and the plain measurements. The mean track map is generated as follows: The measurements are pre-processed for a trajectory and associated to topological tracks. For same track IDs, the trajectories of all runs are interpolated to a common sample grid of 1-D locations. For each sample, a mean value is computed for all trajectory values. The main difference to the update of the RailSLAM track map is the storage of each trajectory and the processing of all stored trajectories. RailSLAM updates the track map incrementally and deletes the trajectory estimate afterward.

Cross-track Evaluation

Measurements and track maps are evaluated with the cross-track error. This error is the cross-track distance δ_{CT} between the reference track map and a measured position or a track map position. The cross-track error is computed in the following for the measured position and the track map position. A measured position is matched on the reference track map with a function that returns the nearest position on the tracks:

$$\vec{p}_{\mathrm{ref}} = \mathrm{f}_{\mathrm{matched,pos}}(\vec{p}). \tag{7.4}$$

This function involves the simple map-match method and the reference track map. Afterward, the cross-track error e is computed between every position measurement $\vec{p}$ and the corresponding nearest position on a reference track $\vec{p}_{\mathrm{ref}}$.

$$e = \underbrace{||\vec{p} - \vec{p}_{\mathrm{ref}}||}_{\delta_{\mathrm{CT}}}. \tag{7.5}$$

As a compact performance figure, the root mean squared error (RMSE) for measurements (E_{meas}) is defined between measurements and reference over all time steps k:

$$E_{\mathrm{meas}} = \sqrt{\frac{\sum_k^N e_k^2}{N}}. \tag{7.6}$$

The number of samples for the error statistics is N. In this case, N is the number of all time steps.

Track maps are also evaluated and compared with the cross-track error. For each track sample of the candidate map, the error is computed between the position of the sample $\vec{p}_{id,s}$ and the matched position of the reference track map $\vec{p}_{\mathrm{ref}}$:

$$\vec{p}_{\mathrm{ref}} = \mathrm{f}_{\mathrm{matched,pos}}(\vec{p}_{id,s}). \tag{7.7}$$

The cross-track error of a track map sample ($e_{id,s}$) is computed in the same way as in Eq. (7.5):

$$e_{id,s} = ||\vec{p}_{id,s} - \vec{p}_{\mathrm{ref}}||. \tag{7.8}$$

The RMSE for a candidate track map (E_{map}) is computed over all track IDs and locations with:

$$E_{\text{map}} = \sqrt{\frac{\sum_{id} \sum_{s} e_{id,s}^2}{N}}. \tag{7.9}$$

The track map error and RMSE are used in the analysis of the RailSLAM track maps over different iterations counts.

Combined Measurement and Track Map Evaluation

This evaluation analyzes the statistics of deviations, errors and residuals between measurements and a track map. Therefore, a cumulative distribution function (CDF) of the deviations is used with the measurement data over several runs. The analysis comprises the cross-track deviations and additionally the deviations of height, attitude and curvatures. The CDF $F(\hat{x})$ is defined in general with:

$$F(\hat{x}) = \mathrm{p}(x \leq \hat{x}), \tag{7.10}$$

and determines the probability that a value of the random variable x is less or equal than $\hat{x}$ [84]. The analysis of the deviation, error or residual uses a discrete set of errors and a discrete CDF. This CDF is represented with a graph of the probability over a range of increasing errors $\hat{e}$. For a specific value of $\hat{e}$, the discrete CDF is the number of all errors that are smaller than $\hat{e}$ over the count of all error values N:

$$F_e(\hat{e}) = \frac{\sum_k \chi(e_k \leq \hat{e})}{N}. \tag{7.11}$$

The indicator function, also called characteristic function χ, returns 1 if e_k is equal or less than $\hat{e}$:

$$\chi(e_k \leq \hat{e}) = \begin{cases} 1, \text{ if}\,(e_k \leq \hat{e}), \\ 0, \text{ otherwise.} \end{cases} \tag{7.12}$$

Additionally to the CDF, the results for combined measurement and track map evaluation show also the RMSE E_{meas} (Eq. (7.6)) as a compact performance figure of a statistics with many measurements of several train runs.

Chapter 8

Results and Discussions of the Algorithms

8.1 Localization Methods

In the following, the train localization methods of simple map-matching, particle filter and the multi hypothesis tracker are evaluated with measurement data of the regional train experiment. The track-selective analysis results are shown with visual representations and performance figures from statistics over ten runs. The results of each localization method are discussed and the different methods are compared.

8.1.1 Simple Map-Match Method

Figure 8.1 shows the track-selective evaluation of the simple map-match method for Run 1 between the stations Augsburg and Friedberg. The evaluation results over time are shown in Figure 8.2 with the true switch way and train speed. The correct estimated track is marked with OK in green, a yellow triangle marks a tolerated error in the vicinity of the switch, and a wrong track is marked with a red cross. The track-selective evaluation is suspended on the single track before Friedberg station, between 430 s and 520 s, because track selectivity is always given in single track scenarios. There is also no evaluation of the track selectivity while the train is standing. The train speed is indicated in gray in Fig. 8.2. The train speed is negative because the train runs reversed in Run 1, according to the train frame definition. The switches are shown in blue, as described in Section 7.2.1, Fig. 7.6. Branching switches are shown with the true switch way of left and right. As seen from the figures, the matched positions are several times on the wrong tracks. The identification of the tracks can

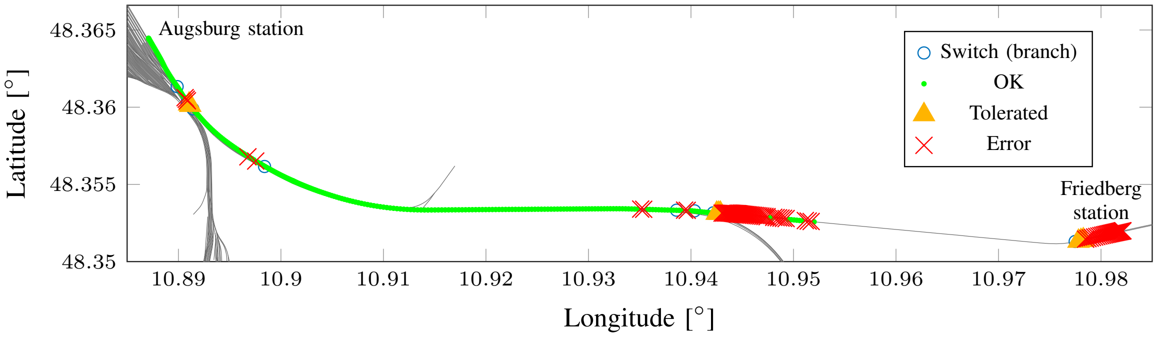


Figure 8.1: Track-selective evaluation of the simple map-match method.

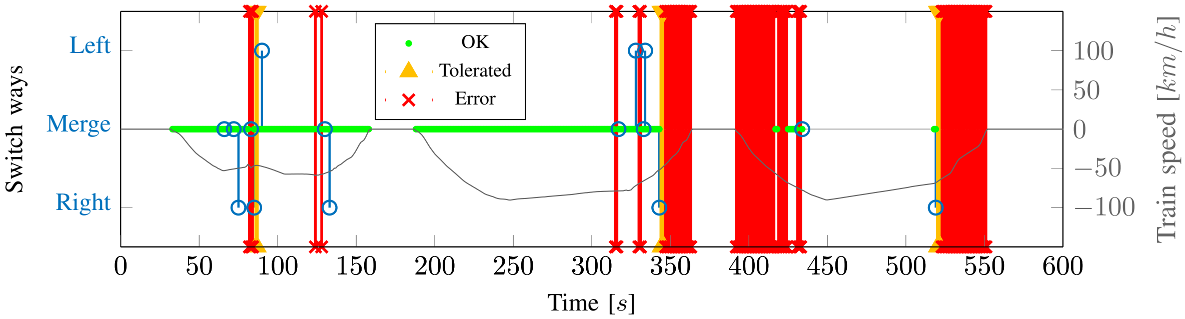


Figure 8.2: Track-selective evaluation of the simple map-match method over time.

change instantly, because the simple map-match method uses no prior information of past track estimates. Every measurement is matched to the nearest track in a snapshot manner and this method does not estimate a switch way.

The performance figures over ten runs over 230 km between Augsburg and Friedberg are shown in Table 8.1 with different track maps. The track selectivity is appropriately evaluated on tracks with parallel tracks in the vicinity. The number of error events represents how many times a transition to a wrong track occurred.

Method: Map-match **Measurements**	**Track** **map**	**Track** **selectivity**	**Error** **rate**	**Error** **events**
INS/GNSS positions	OSM	83.2%	18.7%	61
	Reference	91.2%	11.3%	44
	RailSLAM1	95.5%	5.6%	37
	RailSLAM2	93.9%	7.1%	33

Table 8.1: Track-selective performance figures of the simple map-match method.

The reference track map shows better results than the OSM track map with a higher track selectivity value and a lower error rate. The track maps created with RailSLAM show better results than the OSM and reference track. The RailSLAM track map 1 with the same data performs slightly better in track selectivity and error rate than the RailSLAM track map 2 from different data. This track map is not necessarily more accurate, because common errors are neglected with the use of the same data for mapping and localization. However, the evaluation with the ideal map shows still matches on wrong tracks. The track selectivity of this method depends on small cross-track distances to the correct tracks. In general, a higher accuracy of position measurements and a higher accuracy of the geographic track positions in the track map are beneficial for a better track selectivity.

8.1.2 Particle Filter Localization

The particle filter (PF) is analyzed for track selectivity in the following. The measurements are directly used for the particle weighting without a pre-processing stage. The most relevant parameters are the number of particles, the resample threshold and the probability distribution parameters for the sampling noise and the measurement noise.

Track-Selective Results

Figure 8.3 shows the track-selective evaluation of the particle filter output for Run 1 and Figure 8.2 shows the evaluation over time. It can be seen, that the particle filter estimates the tracks more accurately than the simple map-match approach. At one switch, the filter estimates an incorrect track within the tolerance region. This result represents one parameter setting with 100 particles, an OSM track map and GNSS position and heading measurements. The parameter adjustment process uses fixed random seeds for the pseudo-random number generators. Different random seeds alter the results of the evaluation even with the same data and parameters. The results are sensitive to the parameters such as different random seeds, measurement noise, the noise ratio between different measurements, resampling occurrence, process and sampling noise as well as the track map quality.

Table 8.2 shows an evaluation with the OSM track map and direct measurements from GNSS and IMU over ten runs and 230 km of traveled distances. These results have been also presented in [J1]. For the switch-way evaluation, 'failed' means that a track was not correctly identified. A 'late' switch way means that the estimate was wrong on the first

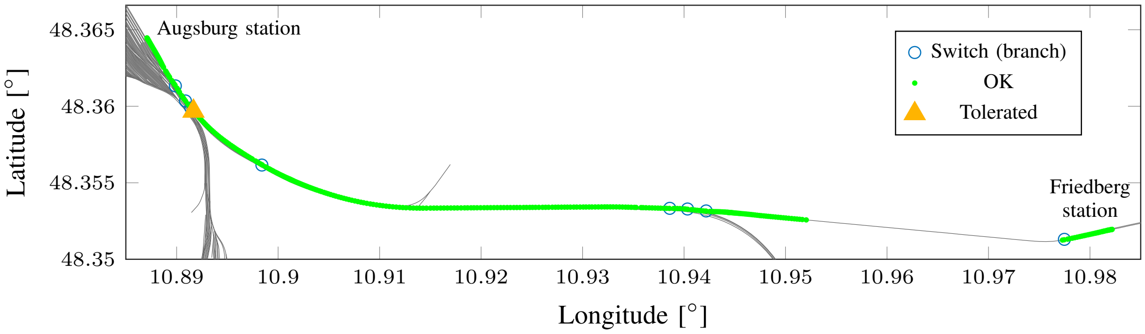


Figure 8.3: Track-selective evaluation of the particle filter method of Run 1.

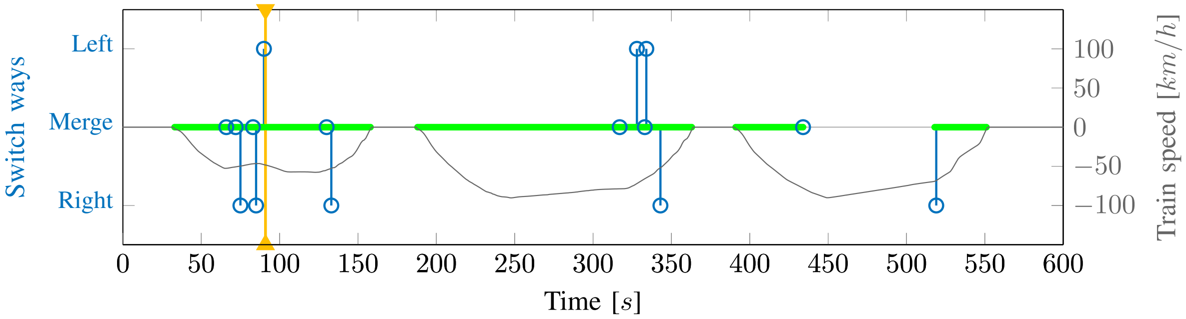


Figure 8.4: Track-selective evaluation of the particle filter method over time.

hand, but correct in the long run after the tolerance range of a switch. The results of this table were processed with parameter adjustments and fixed random seeds for the best track-selective performance. The parameters of the sample variance, measurement variances, and the resample threshold are different for each combination and random seed. Therefore, the track-selective results of Table 8.2 show mainly tendencies of improvements for different combinations and no general results on track selectivity. The combination of GNSS positions and heading is better than positions only in terms of track selectivity, error rate or correct estimated switch ways. The best combination in Table 8.2 is combination 3 with GNSS positions and IMU yaw rate. The combination of all likelihoods with GNSS positions, heading, and IMU yaw rate shows similar results and no improvement towards perfect track selectivity. The imperfect results and this lack of improvement can be mainly explained with the imprecise positions of the OSM track map and the derived heading and curvature values (see Section 3.3.4). The track map quality is a very important factor for the track selectivity with the particle filter. The reference track map with more accurate geographic track positions is beneficial for the comparison with GNSS position measurements in the weighting function. The RailSLAM track map contains better geometry values with advantages for the weightings with GNSS heading and IMU yaw rate.

Method: particle filter **Measurements**	**Track** **map**	**Track** **selectivity**	**Error** **rate**	**Correct** **switch ways**
1) GNSS positions	OSM	84.0 %	20.2 %	74 of 107
2) GNSS positions, GNSS heading from motion	OSM	90.2 %	10.8 %	103 of 107 (3 failed, 1 late)
3) GNSS positions, IMU yaw rate	OSM	99.3 %	1.0 %	104 of 107 (1 failed, 2 late)
4) GNSS positions, heading IMU yaw rate	OSM	99.2 %	1.2 %	103 of 107 (1 failed, 3 late)

Table 8.2: Track-selective performance figures of the particle filter method.

Table 8.3 shows results over 100 repeated evaluations with different seeds, 1000 particles, the RailSLAM map 2 and the switch-way estimation with the IMU yaw rate. The switch ways have been correctly estimated in all cases except some accepted errors in the switch sections. There are little variations of the error rate due to different results over different random seeds.

Method: PF **Measurements**	**Track** **map**	**Track** **selectivity**	**Error** **rate**	**Correct** **switch ways**
GNSS positions, IMU yaw rate	RailSLAM 2	100 %	min. 0.04 %, max. 0.21 %	26 of 26 (0 failed, 0 late)

Table 8.3: Track-selective performance figures of the particle filter (PF) method with over 100 repeated evaluations of different seeds.

Discussion

The proposed particle filter estimates the 1-D track location and the track ID with particles and has thus a low state dimensionality of two. The speed and displacement are estimated with a Kalman filter. As a result, the number of particles can be relatively low, as particles are limited and constraint on the tracks. In the evaluations, particle sizes of 100 to 1000 are used.

The particle distribution in along-track requires a certain spread across the 1-D location domain and must be regarded in the proposal function design and parameter adjustment process. On the one hand, if the particles are distributed too narrow, divergence can happen and all particles are away from the true values. In order to avoid this situation, more sampling noise is added. On the other hand, if the distribution is too large, there might be ambiguities of a repeated geometry signature of the track. The particle distribution in along-track direction is expanded with the sampling step and gets reduced with the resampling step. The particles follow the position measurements in the usual case because the particles are weighted according to their distance to the position measurements. In fact, this is a weak effect compared to a Kalman filter update. A measurement update from a Kalman filter changes and drags the estimate in the direction of the measurement. The particle filter instead changes only the particle weights in a first step. The estimated values of the particle state are not directly changed with the position measurement update. A change of the particle distribution is achieved with the resampling step. Secondly, there is a reduced impact of the along-track weighting due to the multiplication of several likelihoods from different sensors. Some measurements, such as the heading or turn-rate measurements, contain rare and ambiguous along-track information. Without this drag and a reduced impact of the along-track weighting, the particle distribution can escape from the actual true position in some scenarios. The consequence is a wrong and divergent result of the particle filter.

The parameter adjustment of the different likelihoods is crucial and affects the particle distribution in along-track as well as the cross-track estimation. The parameter adjustment is complex because it requires a Monte Carlo evaluation over different random seeds and the adjustment is sensitive to small changes of the sampling variance, measurement covariances, the ratio between the different measurement likelihoods and the resampling threshold.

The particle filter has three processes with random generators. The sampling of the velocity, the sampling of the switch way, and the resampling process require random numbers. These random generators can be controlled with random seeds for a pseudo random sequence of the same numbers. A set of same random seeds can be used for the analysis of parameter impact in order to have comparable results. However, the particle filter estimates different results for different seeds or shuffled random generators.

Particle filters use typically more computational resources than other estimation approaches due to the much higher number of hypotheses. Nevertheless, an evaluation of a MATLAB implementation with measurement data and 1000 particles was more than 150 times faster than real-time on a contemporary business notebook (Intel i7-5600 CPU, 2.6 GHz).

8.1.3 Multi Hypothesis Tracker Localization

The multi hypotheses tracker (MHT) estimates a switch way and keeps the estimate of an identified track similar to the particle filter. The switch-way estimation is carried out with the evaluation of likelihoods that weight the different hypotheses. Beneficial likelihoods for the switch-way estimation are the cross-track distance to the position measurement d_{CT}, the heading ψ, and the horizontal curvature c^{ψ}. A pruning process eliminates all hypotheses with weights below a threshold. In the ideal case, there are two hypotheses after the train has passed a switch facing, and one hypothesis after the train has left the switch area.

Track-Selective Results

The results for the exemplary Run 1 with the OSM track map are also similar to the particle filter results in Section 8.1.2, as seen in Fig. 8.5 over time. The number of hypothesis varies at each switch passed facing and the correct switch way is estimated shortly after (Fig. 8.5, bottom). A scenario with two nearby switches passed facing and three hypotheses can be seen at 90 seconds. There, the train passes the second switch within the switch section of the first switch.

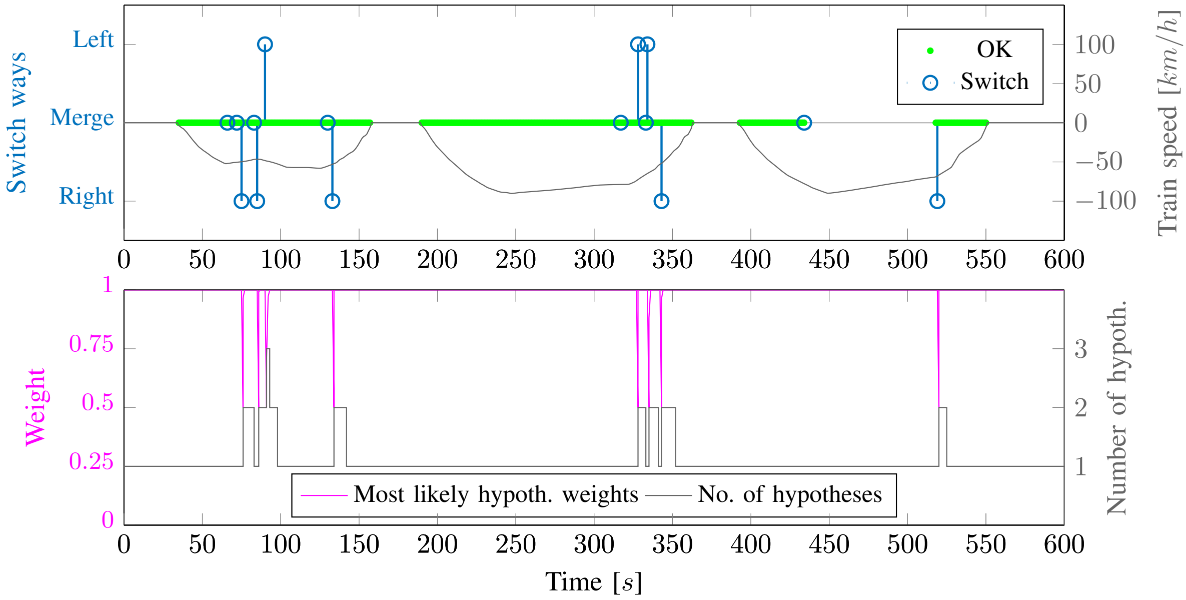


Figure 8.5: Train localization with MHT method: [Top] Track-selective evaluation over time of Run 1. [Bottom] Number of hypotheses and weight over time.

The track-selective performance figures of the MHT filter over 230 km are shown in Table 8.4. The MHT method reached almost full track selectivity and two late resolved switch ways. Nevertheless, the map quality is also very important for the MHT filter. Evaluations with the RailSLAM track map 2, based on different data, showed a full track selectivity over a larger range of different parameters. Additionally, evaluations showed also full track selectivity with the measured heading curvature as exclusive weight for the switch-way estimation.

Method: MHT **Measurements**	**Track** **map**	**Track** **selectivity**	**Error** **rate**	**Correct** **switch ways**
INS/GNSS, (d_{CT}, ψ, c^{ψ})	OSM	99.9%	0.46%	105 of 107 (0 failed, 2 late)
INS/GNSS, (d_{CT}, ψ, c^{ψ})	RailSLAM 2	100%	0.13%	26 of 26 (0 failed, 0 late)

Table 8.4: Track-selective performance figures of the MHT method.

Discussion and Comparison to Particle Filter

The MHT method was taken into consideration after the evaluations of the particle filter with measurement data and the OSM map. Several ideas for improvements were realized with the MHT method, such as a separation of the along-track estimation and cross-track estimation. This separation is achieved by the along-track Kalman filter with position measurement updates and the hypotheses weights are exclusively used for the cross-track estimation. Further, the GNSS and IMU data fusion is achieved with an INS in a pre-filter stage. This separation into smaller elements is beneficial from an engineering point of view, as the different elements can be improved, tested and adjusted individually.

The non-linearities of the train localization estimation problem are the switch ways and the track features that are rather deterministic and known in a track map. Furthermore, the train localization estimation is separable to a 1-D parameter estimation. As a consequence, the particle filter can not show its full capability in terms of non-linear estimation.

Compared to the particle filter method, the MHT method achieves also a full track selectivity for the RailSLAM track map 2. If the coarse OSM track map is used, the MHT method outperforms the particle filter method. The particle filter results of Table 8.2 appear similar, but these results are only valid for one random seed realization. The particle filter method is a non-deterministic algorithm and produces different outcomes for the same input data. The track-selective results of the MHT method are repeatable because the MHT method is a deterministic algorithm in the sense that this algorithm produces the same results from the same input data. The MHT train localization is still a probabilistic algorithm in the sense that this method implements an estimation of the train location with probabilities and a Bayesian framework. The parameter adjustment process for the MHT method is less complex, because there is no need for a Monte Carlo evaluation as well as the along-track tuning and track identification tuning is independent from each other. The presented measurement likelihoods depend on the quality of the track map. The particle filter uses inherently also unfortunate likelihoods for the along-track estimation such as the coarse heading or coarse curvature of the OSM track map. The track identification with imprecise GNSS position measurements is difficult with imprecise geographic track positions of the OSM track map. The MHT along-track estimation uses instead position measurements only. The MHT track identification uses other parameters for the position measurements in order to achieve a low and less significant weight from the cross-track position likelihood.

8.1.4 Related Work

There are few studies with a similar evaluation scope of more than 100 km or 100 switches, as presented in Section 1.4. The advantage of a probabilistic estimation approach over deterministic map-matching in terms of track selectivity was also shown by [22] with a GNSS and a velocity sensor. Böhringer [16] received similar track-selective results (99.78 %) with a different train, different location and a sensor setup with INS/GNSS in combination with an eddy current sensor as switch-way detector. Finally, it can be reasoned from the literature and the present results, that most of the gain in accuracy can be achieved by using a probabilistic estimation filter as well as using sensors which can measure the competing switch ways.

8.1.5 Suggestions for Improvements

The localization results depend on the accuracy of the measurements as well as on the track map. First, position accuracy improvements are expected from advanced multi-constellation and multi-frequency GNSS receivers compared to the used GPS-L1 receiver. Second, the GNSS measurement setup may be improved with an advanced antenna and an optimized placing. Further improvements are expected from positioning methods using the carrier-phase measurements, railway specific GNSS positioning methods [C5], [85] and integrity monitoring [86], [87], [88]. A promising approach for the train localization with a particle filter has been shown in a supervised master thesis [M2] and in [C4]. This approach used pseudorange and Doppler measurements of a GNSS receiver and showed a successful GNSS based train localization with less than four satellites in view. Further, beneficial satellite constellations for along-track and track identification were analyzed. The particle filter implemented a bias estimation for each pseudorange between satellite and track map position. This bias estimation showed enhanced results for the switch-way identification.

For the sensor fusion, a tightly-coupled INS [89], [90] or an ultra-tightly coupled INS may be chosen in favor of the loosely-coupled INS/GNSS. An increased availability and track-selective accuracy is also expected with a train localization based on multiple GNSS antennas, e.g. for two or more GNSS on the same train.

In the case that GNSS measurements are absent, the INS/GNSS can bridge a limited amount of time with position data. A continued along-track localization with other measurements than GNSS positions is important for a continued accuracy at GNSS outages, an increased availability and for redundancy reasons. The presented MHT method uses only position measurements for the along-track estimation. Further, this along-track estimation is computed at single track location, which is the estimated mean of one hypothesis. The particle filter instead samples the track geometry over a range of along-track locations and uses also other measurements for the along-track estimation. Suitable extensions for the MHT method may involve additional measurements for the along-track estimation, a quality evaluation for the measurement and the track feature, and a sampling of a track feature over a range of track locations. Moreover, a GNSS redundant approach for train localization is desired for longer GNSS interruptions in tunnels and for a cross-check verification of measurements. The proposed magnetic and vibration based train localization (Section 4.3.1) are promising extensions and can be used independently from GNSS.

8.2 RailSLAM Method

This section contains results of the RailSLAM method with the experimental data set. The following plots show the results with INS data from the cabin mounted MEMS IMU. The RailSLAM method estimates a trajectory of the train, that consists of nine geometric values and a topological data association to the track ID and the 1-D track locations. This trajectory is estimated from measurement data and from a prior track map, in the case that the actual track is prior known and available in the track map. A white-space mapping adds new tracks, a prior-map SLAM replaces previously estimated geometry data.

8.2.1 Estimation of the Topological and Geometric Track Map

The track map is now estimated from the experimental data set, as depicted in Fig. 7.4. The following results show the creation of a new track map.

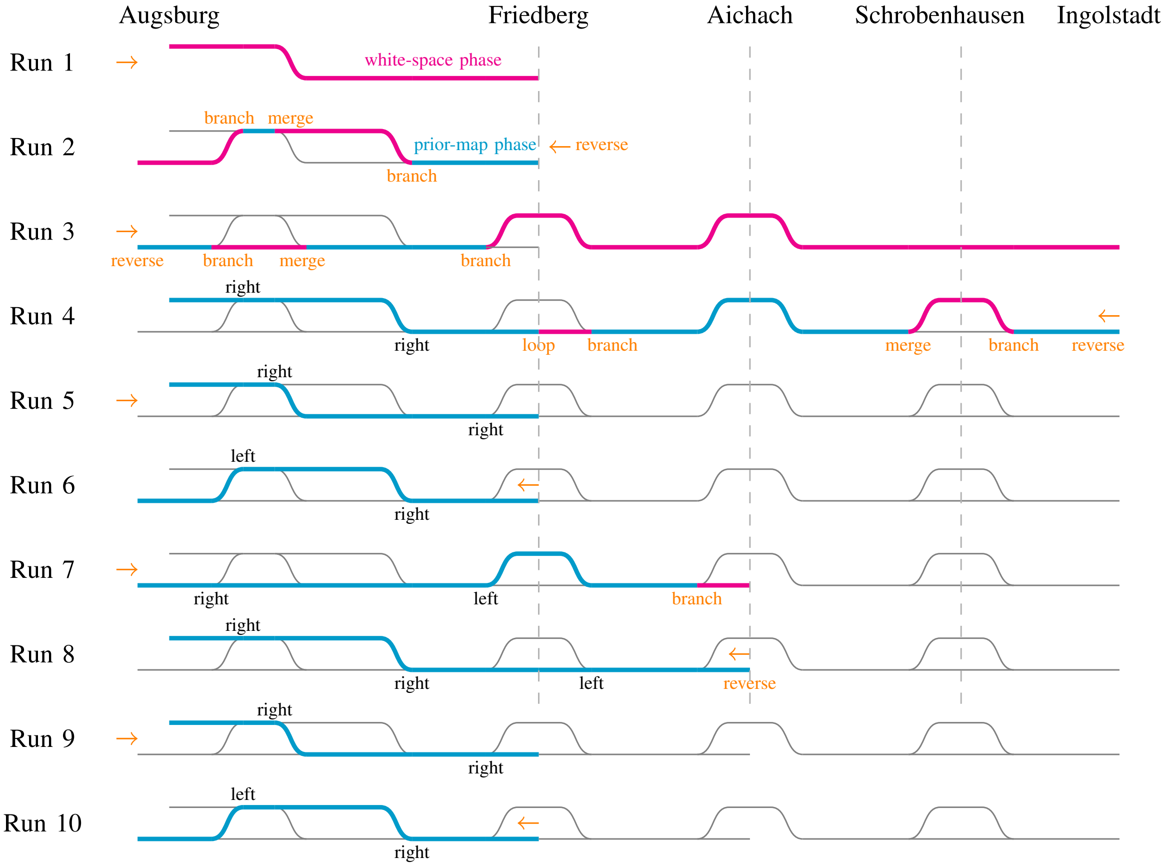


Figure 8.6: Topology evolution of the iterative track map creation for ten runs.

Figure 8.6 shows the evolution of the topological track network over ten runs. A new and unknown track (magenta) is recorded in the white-space phase. An existing track is updated in the prior-map phase (cyan). Some tracks are known but unchanged (gray) because the route of the current train run does not cover these tracks. Changes between white-space phase and prior-map phase are shown in orange with reverse, loop, branch and merge switches. There are 15 known switches passed facing indicated with the true switch ways in right or left (black). It can be seen, that most of the track network is explored in the first four runs during the white-space phases. Later runs update the track map with the prior-map phase.

The final RailSLAM track map after ten runs is shown in Figure 8.7. The left figure shows the 2-D positions of the track map with stations. The colors indicate the number of runs on a particular track, which is also the iteration count of the map update. However, parallel tracks and some switches are very close compared to the geometric dimension of the complete track map. Hence, the right plot shows a topological schematic with iterations in different colors as well as the track IDs next to the tracks. The track with the most iterations is the single track near Friedberg with ten runs while each track of the station in Schrobenhausen are only recorded once.

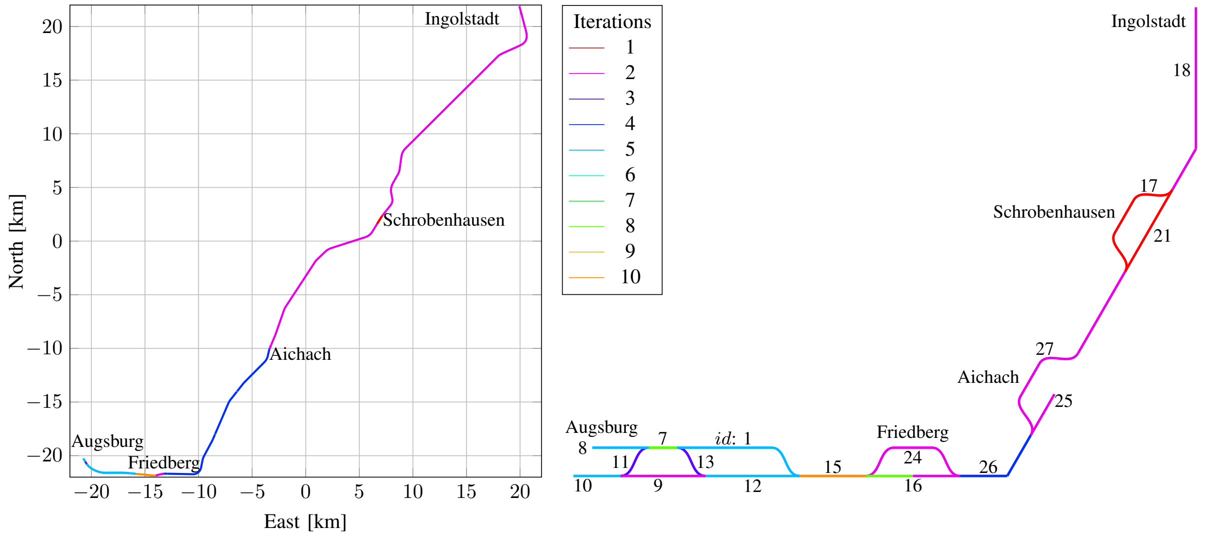


Figure 8.7: RailSLAM track map after ten runs. [Left] Geometric plot of 2-D positions. [Right] Schematic plot of the recorded track network with track IDs.

Figure 8.8 shows the geometry of track 1 with a length of 4.8 km after five iterations. Each variable is shown with the uncertainty in terms of a one-sigma deviation. It can be seen from the figure that the track geometry has a low variation in along-track direction over many stretches of track 1. The geometry values of the attitude angles and the corresponding curvatures are mathematically linked and changes are correlated. Banked curves allow higher velocities and therefore the bank or bank curvature is also linked with horizontal curves of heading angle and heading curvature. The height is linked with slope and slope curvature.

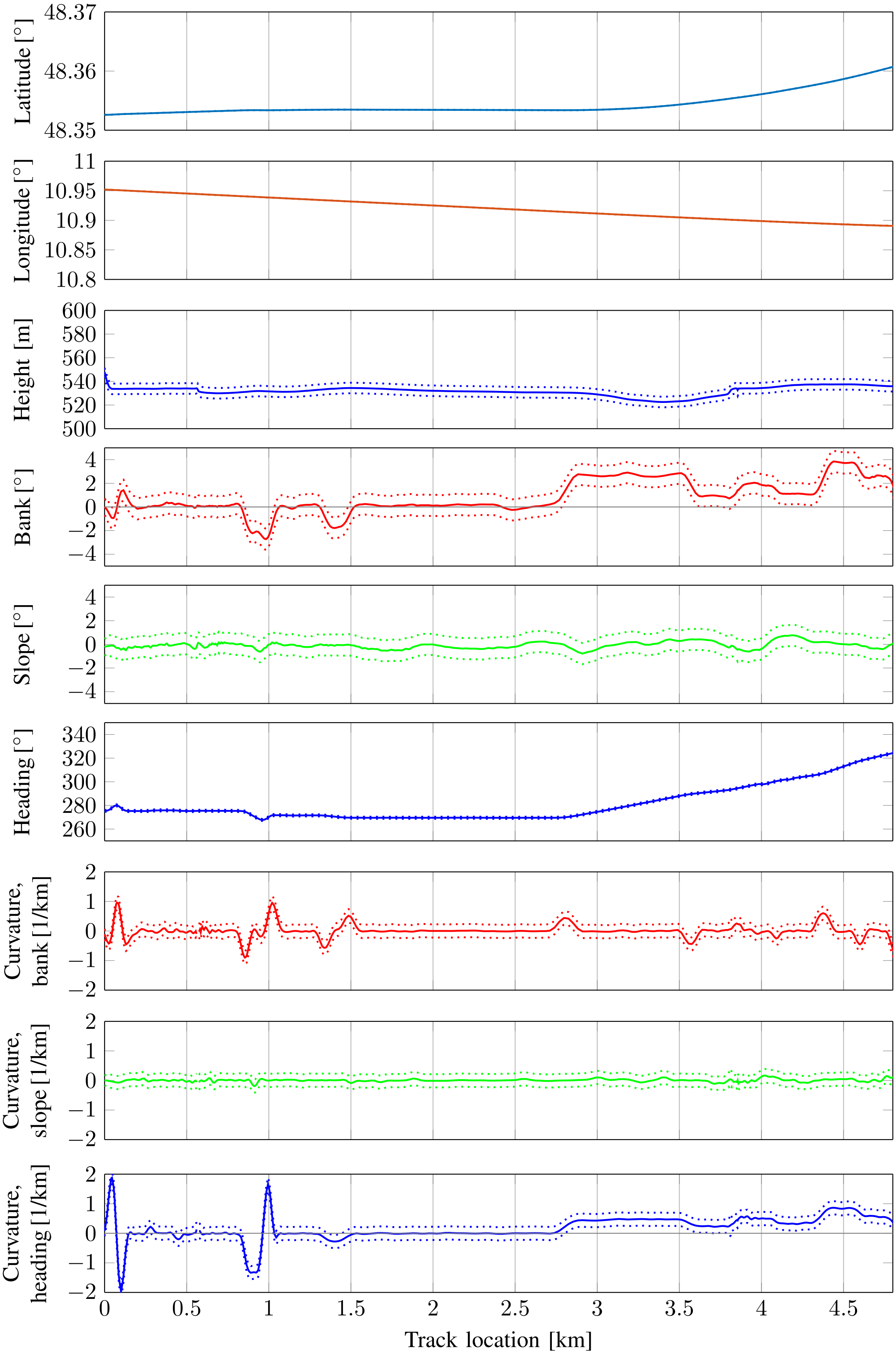


Figure 8.8: Geometry and uncertainty of track 1 from the RailSLAM track map.

Figure 8.9 visualizes 2-D positions of the RailSLAM track map 1 near Augsburg station and shows four switches with parallel tracks. The parallel tracks in the center are track 7 and track 9, as defined in Fig. 8.7. It can be seen that the RailSLAM track map matches the tracks of the geo-referenced aerial image [60] in the background. However, the areal image depicts more tracks and switches that are not covered with the train runs. RailSLAM can only map the tracks that are passed with the train. The track map contains 17 tracks with a total length of 74.9 km after ten runs and each track contains nine geometric variables. These variables are sampled over the 1-D track location and contain the positions, attitude and curvatures.

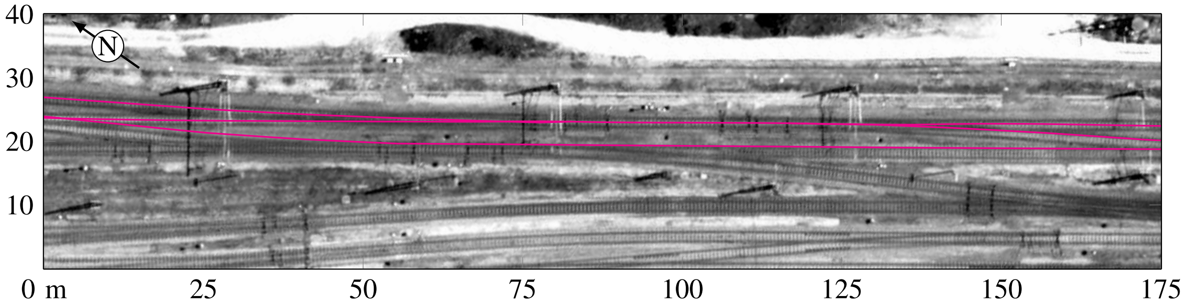


Figure 8.9: RailSLAM track map with switches near Augsburg station.

8.2.2 Track Map Analysis and Comparison of Track Maps

The accuracy of the track map in lateral direction is analyzed in the following. Figure 8.10 shows a 100 m section of track 15 between Augsburg and Friedberg with the trajectories of all ten runs. It can be seen, that

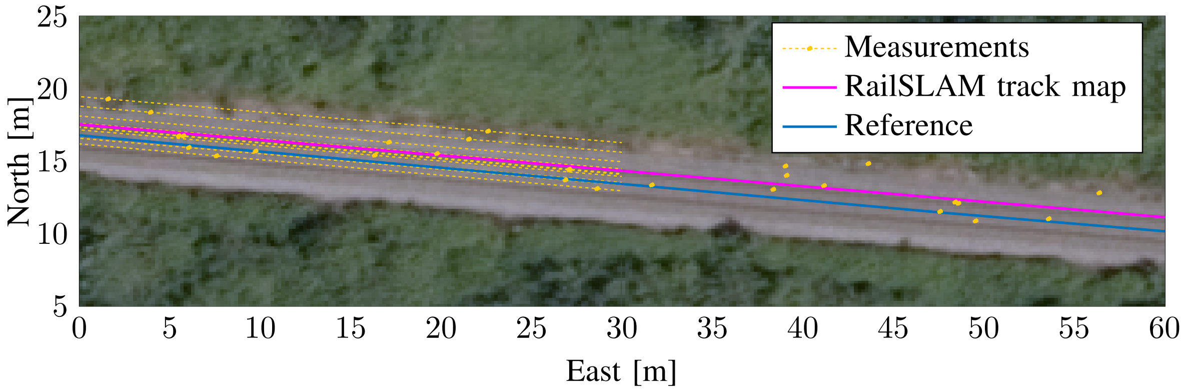


Figure 8.10: Single track scenario with a geo-referenced areal image, the RailSLAM track map after 10 iterations (magenta), 10 trajectories (orange) and a reference (blue).

each trajectory shows a different, but constant deviation to the reference line. The resulting RailSLAM track map can be seen after ten iterations (magenta).

The single track scenario (see Fig. 7.4) of track 15 with 1.7 km between Augsburg and Friedberg is now analyzed for the cross-track position errors over ten runs and ten map iterations. The box plot in Fig. 8.11 sketches the cross-track error distribution of the plain measurements, the RailSLAM track map, and the mean track map. The box plot shows the mean error values ('+'), a thick line represents the error range between the 25 % and the 75 % percentiles and the thin lines represent the range between the 10 % and the 90 % percentiles.

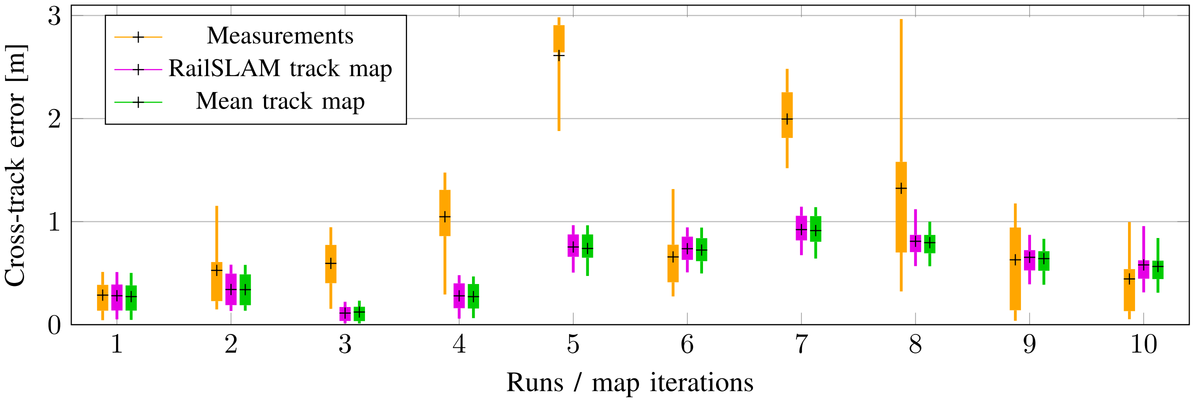


Figure 8.11: Evolution of the cross-track error over ten runs (single track scenario).

In Fig. 8.11, the *measurements* are the cross-track error between preprocessed, INS/GNSS position measurements and the reference and shown in orange color. The errors between the RailSLAM track map 1 and the reference are shown in magenta and denoted with *RailSLAM track map*. Finally, the mean track map of the n-th run is created from an average of n measurement trajectories. The cross-track errors of the *mean track map* is shown in green and computed between the positions of the track map and the reference.

Two characteristics can be observed in Fig. 8.11: The first one is the comparison of the errors from the RailSLAM track map and the mean track map. The second characteristic is the evolution of the cross-track accuracy over iterations.

The comparison of the mean track map and the RailSLAM track map show very similar results in terms of cross-track errors. The mean map approach does not respect any variances for the iterations, so all measurement weights of every iteration are equal. RailSLAM instead has an inbuilt way of respecting the covariances and thus adjusting the weights. However, the correct weighting of the measurements is difficult due to unknown biases and inadequate estimated measurement covariances. With every iteration, the RailSLAM algorithm reduces by default the estimated covariance or uncertainty, respectively. There is a potential risk of overconfidence and the accuracy is overestimated. This overconfidence leads to an error in terms of a bias when the track map converges over iterations.

An expectation for an iterative mapping approach, such as the RailSLAM method, is an improvement of the accuracy over several iterations. The evolution of the cross-track errors over iterations shows neither a monotonic decrease nor a decrease of the error with a trend in Fig. 8.11. This result has to be seen carefully with respect to the limited statistics of ten iterations and to a position error of the reference with 0.5 m [60]. The final RailSLAM track map error, i.e. the mean cross-track error of the single track section after ten iterations, is in the range of the accuracy of the reference. There are further accuracy evaluations in the following with CDFs for the cross-track error as well as for seven other geometry values.

A comprehensive analysis with measurements and different track maps with cumulative distribution functions (CDFs) is presented next. Therefore, the deviations are analyzed between the track geometry and measurements over ten runs, as defined in Table 7.1. This deviation between track map and measurement is also processed in a likelihood for train localization. The deviations of cross-track, heading and heading curvature are computed for all four track maps. Figure 8.12 shows the distribution of the cross-track deviation between different maps and position measurements from INS/GNSS or GNSS only. The deviations between measurements and the RailSLAM track map 1, which was estimated from the same data, are the residuals. The deviation evaluation is limited to the tracks between Augsburg and Friedberg, because of the limitations from the reference track map. This limitation of the evaluation is also RailSLAM track maps have more iterations over the tracks compared to the complete range from Augsburg to Ingolstadt.

The GNSS position deviations to the reference track map in Fig. 4.5 are slightly different, because there are 21 train runs analyzed (see Table 7.1 and Table E.2). Table 8.5 extracts the performance values of the GNSS-only position deviations as well as the deviations with INS/GNSS positions. The performance values are the RMSE of the cross-track error, the error probability below 1 m and 2 m as well as the 95% and the 99% error. In the case that the parallel track separation is 4 m, there are 85.3 % below and 14.7 % above the discrimination distance of 2 m for the best and realistic case with the RailSLAM track map 2. More than half of the measurements are above 2 m in the worst case with OSM track map and GNSS positions only.

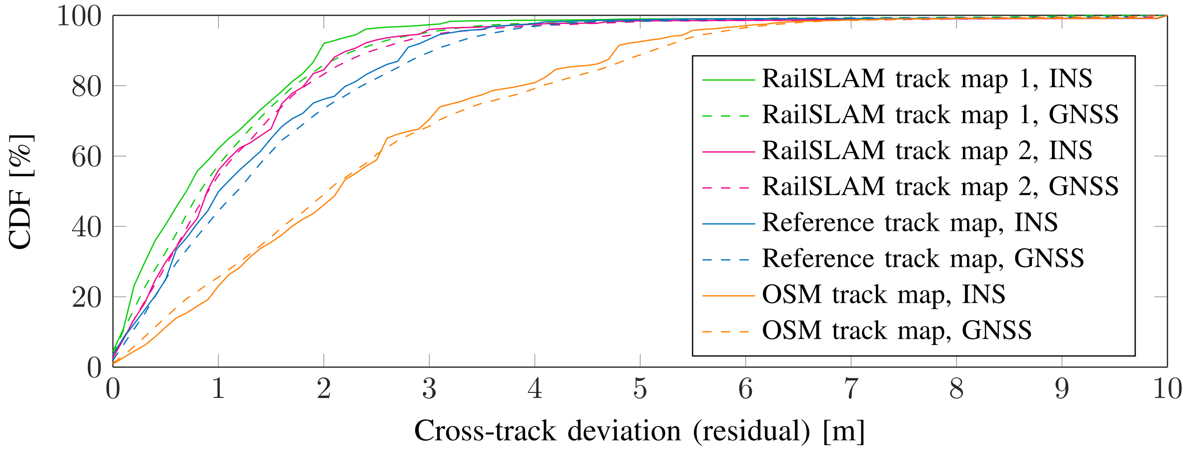


Figure 8.12: Cumulative cross-track deviations over different track maps.

Positions	**Track Map**	**RMSE**	**Err.** $<$ **1 m**	**Err.** $<$ **2 m**	**95%**	**99%**
GNSS	OSM	3.08 m	25.6%	49.1%	5.7 m	7.8 m
	Reference	2.00 m	44.6%	73.6%	3.6 m	6.3 m
	RailSLAM 2	1.85 m	53.9%	83.9%	3.1 m	8.2 m
	RailSLAM 1	1.66 m	57.1%	85.5%	3.0 m	6.0 m
INS/GNSS	OSM	2.99 m	23.4%	46.2%	5.4 m	7.5 m
	Reference	1.93 m	49.5%	76.0%	3.2 m	5.6 m
	RailSLAM 2	1.86 m	54.7%	85.3%	2.9 m	7.4 m
	RailSLAM 1	1.56 m	60.8%	90.7%	2.3 m	5.5 m

Table 8.5: Cumulative cross-track deviations and residuals of GNSS positions, INS/GNSS positions and four different track maps.

The track map with the largest cross-track deviations is the OSM track map, as seen from the table and the figure. Further, there is an improvement with the reference track map and the second best is the RailSLAM track map 2, which was created from a different data set than the measurements used for this evaluation. An explanation for the better performance of the RailSLAM track map 2 may be a limited accuracy of the geo-referenced aerial images for the reference track map. The best results are the residuals from RailSLAM track map 1. This result does not surprise, as the same data is used and is not a realistic use-case. Nevertheless, the residual distribution of this track map demonstrates how repeatable the measurements are with this ideal track map in terms of the cross-track deviation. A higher repeatability of measurements is a quality indicator that can be used to qualify different sensors, sensor setups or pre-processing approaches. Finally, there is a small gain in terms of lower errors for the INS/GNSS measurements compared to the GNSS only measurements.

The following shows the deviations of attitude and curvature measurements to all four track maps with CDF curves and selected values in tables. The measurements are the pre-processed measurements and only a moving train is evaluated because the curvature measurement require a motion. The following deviation evaluations comprise 5400 deviations for each track feature between the tracks of Augsburg and Friedberg. The track features of heading ψ and curvature c^{ψ} are important for the switch-way estimation and the track-selective accuracy. Figure 8.13 shows the cumulative deviations and residuals for all four track maps. Both RailSLAM track maps show an improved heading and curvature compared to the reference and OSM track map. The latter two track maps calculate the heading and curvature from consecutive position samples, as explained in Section 3.3.4.

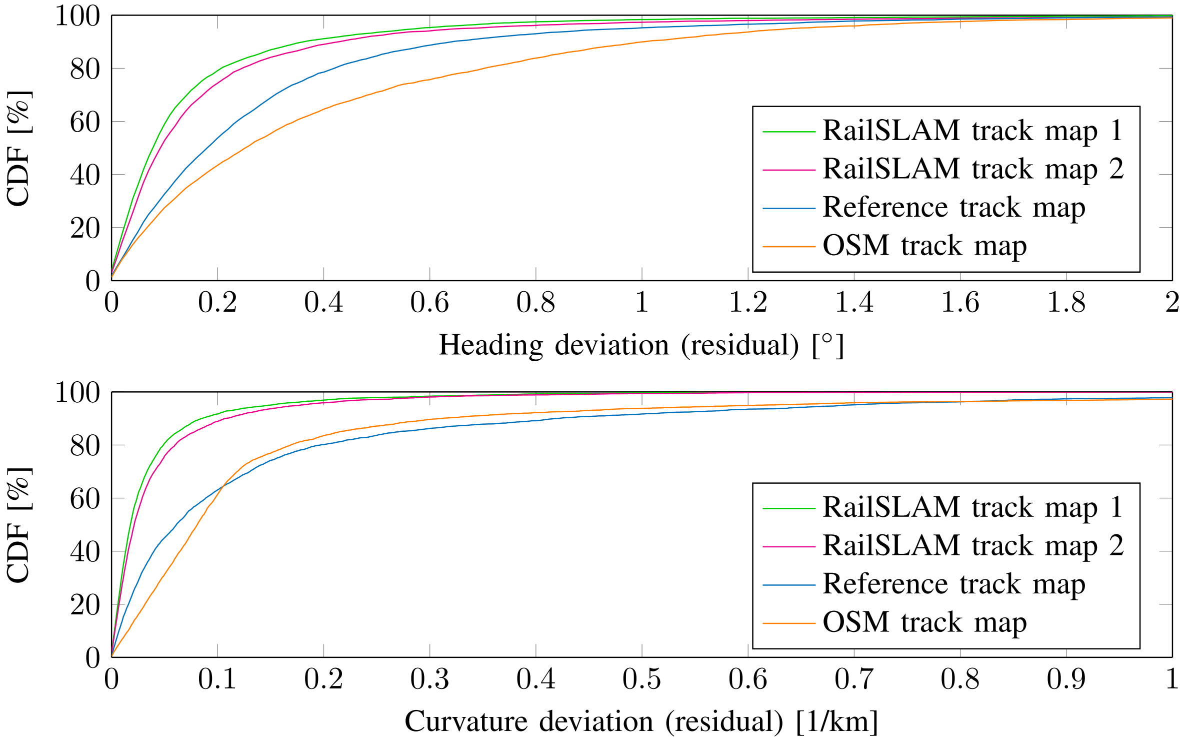


Figure 8.13: Cumulative distribution function of the heading and curvature deviation between track maps and measurements over ten runs.

Table 8.6 shows the performance values from Figure 8.13. The improvement in terms of heading deviation from the worst track map (OSM) to the best track map (RailSLAM 1) are approximately a factor of two. The

Measurements	Track Map	RMSE	95% error	99% error
Heading ψ	OSM	0.60°	1.3°	2.0°
	Reference	0.46°	0.98°	1.8°
	RailSLAM 2	0.36°	0.66°	1.7°
	RailSLAM 1	0.29°	0.58°	1.3°
Curvature c^{ψ}	OSM	$0.35\,{}^{1}/_{\text{km}}$	$0.61\,{}^{1}/_{\text{km}}$	$1.6\,{}^{1}/_{\text{km}}$
	Reference	$0.33\,{}^{1}/_{\text{km}}$	$0.70\,{}^{1}/_{\text{km}}$	$1.4\,{}^{1}/_{\text{km}}$
	RailSLAM 2	$0.09\,{}^{1}/_{\text{km}}$	$0.18\,{}^{1}/_{\text{km}}$	$0.42\,{}^{1}/_{\text{km}}$
	RailSLAM 1	$0.08\,{}^{1}/_{\text{km}}$	$0.15\,{}^{1}/_{\text{km}}$	$0.37\,{}^{1}/_{\text{km}}$

Table 8.6: Cumulative deviations and residuals of heading and curvature for four different track maps.

curvature deviations from the RailSLAM track maps are 3 to 4 times lower than the curvature deviations from OSM and reference track map.

Figure 8.14 shows the deviations and residuals of the attitude angles and curvatures of bank and slope for the two RailSLAM track maps.

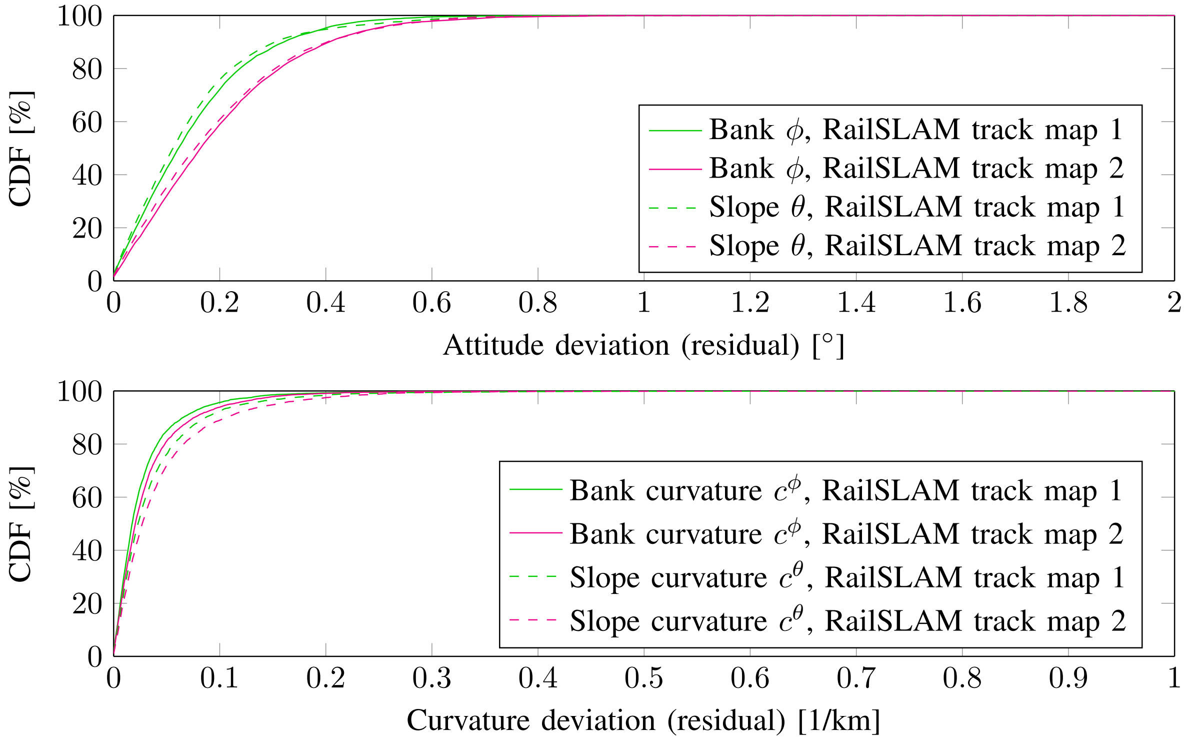


Figure 8.14: Cumulative distribution function of the attitude and curvature deviations between track maps and measurements over ten runs.

The OSM or reference track maps do not contain these track features, because the bank and slope angles as well as the bank and slope curvatures

Measurements	**Track Map**	**RMSE**	**95% error**	**99% error**
Height h	RailSLAM 2	8.3 m	16.5 m	21.8 m
	RailSLAM 1	8.8 m	17.1 m	20.9 m
Bank ϕ	RailSLAM 2	0.25°	0.50°	0.70°
	RailSLAM 1	0.20°	0.40°	0.56°
Slope θ	RailSLAM 2	0.25°	0.50°	0.65°
	RailSLAM 1	0.20°	0.41°	0.65°
Bank curvature c^{ϕ}	RailSLAM 2	$0.07\,{}^1/_{\mathrm{km}}$	$0.15\,{}^1/_{\mathrm{km}}$	$0.26\,{}^1/_{\mathrm{km}}$
	RailSLAM 1	$0.06\,{}^1/_{\mathrm{km}}$	$0.12\,{}^1/_{\mathrm{km}}$	$0.23\,{}^1/_{\mathrm{km}}$
Slope curvature c^{θ}	RailSLAM 2	$0.05\,{}^1/_{\mathrm{km}}$	$0.11\,{}^1/_{\mathrm{km}}$	$0.19\,{}^1/_{\mathrm{km}}$
	RailSLAM 1	$0.05\,{}^1/_{\mathrm{km}}$	$0.09\,{}^1/_{\mathrm{km}}$	$0.18\,{}^1/_{\mathrm{km}}$

Table 8.7: Cumulative deviations and residuals of height, bank, slope, bank curvature, slope curvature for two different RailSLAM track maps.

can not be computed from 2D positions. The residuals of the RailSLAM track map 1 are again slightly smaller than the deviations between measurements and RailSLAM track map 2.

Table 8.7 shows the performance values from Figure 8.14 and additionally the track feature of height h. The performance values show relative large height deviations compared to the horizontal cross-track deviations, even for the residuals. The larger height deviations can be explained with the unfortunate the satellite constellations for the observation of the height. Bank and slope angle deviations show similar results with a deviation lower than 0.5° for 95 % of all evaluations. The slope curvature shows slightly lower deviations with $0.11\,{}^1/_{\mathrm{km}}$ (95 % error) than the bank curvature with $0.15\,{}^1/_{\mathrm{km}}$ for the realistic use-case with the RailSLAM track map 2 and different data for track map and measurements.

8.2.3 Discussion and Suggestions for Improvements

A train localization benefits from small deviations between measurements and map values, especially for the estimation of switch ways. An increased measurement accuracy is advantageous for the localization, as discussed in Section 8.1.5, especially in terms of position measurements. An iterative mapping can benefit from an increased measurement accuracy in two ways:

in terms of data association and the track geometry accuracy of the track map.

The track geometry values show a low variation in along-track direction over many stretches. Most variables change in the curve entrance or curve exit. In straights or constant curves, there are less significant variations of the geometry values that can be used for an along-track estimation. In the proposed implementation with the MHT localization, the along-track estimation, and hence the data association, depend on a position propagation from train speed and a position update.

A different RailSLAM approach was investigated in [P4] and in the two supervised master-theses [M1], [M3]. The sensor fusion and track geometry estimation was different and a particle filter was used for the localization in combination with a random switch generation. The track geometry estimation in [P4] used a model to predict the geometric track values of position, attitude and geometry, and GNSS and IMU measurements were directly used to update the track geometry estimation. The changes in terms of sensor fusion approach, track estimation and localization filter were motivated in order to compute an improved RailSLAM track map from a more significant amount of recorded measurements in combination with a comparison to ground truth references. The pre-processed measurements decouple the trajectory estimation from the RailSLAM estimation and solve the estimation of IMU sensor biases in a more effective way than the track geometry estimation in [P4]. However, the pre-filter introduces correlations over time of the pre-processed measurements. Nevertheless, the advantages of the pre-filter predominate in terms of bias estimation, the separated and independent development, and optimization possibilities. Further optimization potential of the pre-processing has been discussed in Section 8.1.5. The benefits of the MHT method over the particle filter for train localization has been also discussed in Section 8.1.3 and Section 8.1.5.

The estimation of new switches showed undesired results, especially for scenarios with long-stretched merging switches with low train speeds and the given measurements. Therefore, the presented RailSLAM track map of Fig. 8.7 used prior known information about a new and undiscovered switch. Hasberg has also described challenging scenarios for the switch-way detection in [62]. Nevertheless, Hasberg could identify 117 out of 120 switches correctly with his approach and his measurement data set. The main differences to the SLAM approach from Hasberg are a simpler map interpolation in this thesis and an extended map geometry for an integration of IMU data. The track geometry interpolation of the present thesis results in more dense sample points with more geometry values. Nevertheless, the amount of storage for large track maps with dense samples is technically unproblematic as shown in Chapter 3.

The magnetic and vibration signatures in Section 4.3.1 show promising characteristics for an along-track estimation and a switch-way estimation. An extension with magnetic and vibration signatures for localization and mapping may improve also the track map accuracy with a more accurate data association in along-track. Further, a correct switch detection and switch-way estimation is vital for the map quality.

Finally, it can be questioned whether the simultaneous property of the SLAM approach is needed. A mapping result may be needed immediately or after short time for the use cases of an online map verification or map-change detection. For a general mapping, the sensor data can be recorded and the track map generation may be shifted to a later post-processing. One example has been given with the mean track map that was used for comparison reasons. A post-processing approach can further exclude or weight any measurements according to their residuals to an estimated track map. This post-processing approach is considered as an alternative to the SLAM approach with immediate processing.

Chapter 9

Conclusion

Summary

This thesis presents and evaluates methods for a train localization with exclusive onboard sensors without the need for extra infrastructure. This onboard train localization requires a track map. Therefore, an iterative generation of the track map is presented with the RailSLAM method. The thesis comprises a theoretic part with Bayesian estimation (Chapter 2) and definitions for the map based train localization (Chapter 3). A sensor analysis is found in Chapter 4 with global navigation satellite system (GNSS) measurements, track geometry measurements with inertial measurement unit (IMU) as well as first analysis of magnetic and vibration signatures. Chapter 5 defines the probabilistic posterior distributions for train localization and track mapping. These posterior distributions can be seen as general estimation problem definitions and the factorization divides the estimation problem into smaller and manageable parts. The further probabilistic implementations are based on these factorized posterior distributions. Three map based train localization methods are defined in theory (Chapter 5), implemented (Chapter 6) and evaluated in Chapter 8: a simple map-match method, a particle filter method and a multi hypothesis tracker (MHT) method. These implementations are evaluated with reference data and over different track maps. A simultaneous localization and mapping (SLAM) method, called RailSLAM, is defined (Chapter 5), implemented (Chapter 6) and evaluated in Chapter 8. Real measurement data was recorded on a regional train during regular transport service for realistic scenarios. These measurements are used for the sensor data analysis and also for the evaluations of the proposed algorithms. Evaluation methods with performance figures are defined in Chapter 7.

Conclusions of the Research

The map based train localization requires low deviations between measurements and the track map values, especially for the switch-way estimation. The repeatability of measurements over different train runs is a good indicator, because the map itself is created from measurements. The measurement analysis with the IMU (inertial measurement unit) showed a high repeatability of the curvature signals that are measured with gyroscopes in a non-integrative way. The differences in terms of repeatability and measured curvatures between a low-cost MEMS (micro electro-mechanical system) IMU and high-end FOG (fiber optical gyroscope) IMU turned out to be marginal. The measurement analysis of the magnetic field and the vibration signals showed promising results along-track and the identification of the switch way. The GNSS position analysis with a reference track map showed an insufficient accuracy for a track-selective train localization with a simple map-match approach for a GPS L1 receiver with a patch antenna. The combination of GNSS and IMU with an integrated navigation system (INS/GNSS) could increase the position accuracy in terms of repeatability and deviation to a ground truth reference. However, the position accuracy was still not suitable for a reliable track-selective localization based on simple map-matching.

The conclusions from the localization evaluation of the three different methods are: The simple map-match approach is the method with the lowest complexity. This method depends directly on the instant accuracy of the position measurements but shows also the lowest performance. In contrast, the probabilistic estimation methods of particle filter and MHT regard prior estimates and showed better results for track selectivity. Further, the use of the geometry features and appropriate measurements are very beneficial for the switch-way estimation. The particle filter combines the sensor fusion, along-track estimation and track identification internally in the filter. One outcome of the particle filter evaluation is the advantage of a combination of measurements and the need for a feature rich and precise track map. The evaluation results of the particle filter method with a coarse track map showed complex parameter adjustments and difficulties in along-track estimation with divergence as well as wrong track identifications. The particle filter could not show its full capabilities because of the low non-linearities of the train localization estimation problem and the measurement likelihoods that are presented and used in this thesis. The MHT method follows a divide-and-conquer strategy with the INS/GNSS as a pre-filter stage and the separation of along-track estimation and track identification. This strategy has practical implementation advantages as each part can be improved, adjusted, tested, and optimized individually. The evaluation results of the MHT method with a coarse track map quality showed a similar track selectivity as for the particle filter with fixed random seeds and optimized parameters. Nevertheless, the results of the MHT methods are repeatable in contrast to the particle filter results, because the MHT method does not use random generators. MHT method and particle filter method showed both perfect results in terms of track-selective accuracy with a RailSLAM track map. An accurate track map is vital and critical for a successful onboard train localization.

An iterative mapping approach has been presented with the simultaneous localization and mapping for railways, called RailSLAM. Two different track maps over 230 km with ten switches have been created an analyzed. The analysis of the cross-track error evolution over iterations showed similar results of the RailSLAM method and an averaging method, called the mean track map. The accuracy of the track map geometry depends on the accuracy of the measurements and of unresolved and systematic errors over different train runs. The correct weighting of the measurements is difficult due to unknown biases and inadequate estimated measurement covariances. The cross-track error of the RailSLAM track map after ten iterations is in the range of the accuracy of the reference.

All three train localization methods performed best with the RailSLAM track map compared to the track map based on Open Street Map (OSM) data and the reference track map based on data of the official land survey office. The RailSLAM track map has the richest set of track features and the lowest deviations between measurements and track map, compared to the other track maps. The RailSLAM track map geometry comprises geographic positions, height, the attitude angles of bank, slope, and heading, as well as bank curvature, slope curvature and heading curvature.

Final Remarks

A reliable and track-selective train localization is vital for collision avoidance systems and automated train driving. The onboard train localization is independent from other railway signaling, uses cost-effective sensors and there is no need for additional infrastructure elements. Future railway applications benefit directly from an absence of costly infrastructure based sensor systems. The automated mapping saves surveying effort as the track map is created during regular train operations. With a continuous and automatic mapping, the track map is always up-to-date.

Future onboard train localization may be completely independent of GNSS for a use-case in long tunnels, underground and for subway trains.

This thesis contributes to the GNSS independent train localization with three different measurements of track features: the kinematic measurements of the track geometry, the passive magnetic track features and the vibration track features.

Finally, the thesis contributions in terms of measurement analysis, localization methods, mapping methods, and evaluations, bring the development of future railway applications one step forward towards a safer and more efficient railway traffic.

Appendix A
Probabilistic Toolbox

The basic rules for of probabilistic calculus are given in basic and complex forms with additional joint and conditional variables. Random variables can be multi dimensional and grouped or split in variables:

$$a = a_{1:n} = \{a_1, a_2, a_{3:n}\} = \{a_{1:n-1}, a_n\}. \tag{A.1}$$

Product Rule

The basic product rule, product rule with additional conditional c and product rule with additional estimate j:

$$\begin{aligned} \mathrm{p}(a,b) &= \mathrm{p}(a|b)\cdot\mathrm{p}(b), & \text{(A.2)}\\ \mathrm{p}(a,b|c) &= \mathrm{p}(a|b,c)\cdot\mathrm{p}(b|c), & \text{(A.3)}\\ \mathrm{p}(a,b,j|c) &= \mathrm{p}(a,j|b,c)\cdot\mathrm{p}(b|c). & \text{(A.4)} \end{aligned}$$

Chain Rule

The generic chain rule, the chain rule with three joint variables a, b, j, and the chain rule with additional conditional c:

$$\begin{aligned} \mathrm{p}(a_{1:n}) &= \mathrm{p}(a_n|a_{n-1},...a_1)\cdot\mathrm{p}(a_{n-1}|a_{n-2},...a_1)\cdot...\cdot\mathrm{p}(a_1) & \text{(A.5)}\\ \mathrm{p}(a,b,j) &= \mathrm{p}(a|b,j)\cdot\mathrm{p}(b|j)\cdot\mathrm{p}(j), & \text{(A.6)}\\ \mathrm{p}(a,b,j|c) &= \mathrm{p}(a|b,j,c)\cdot\mathrm{p}(b|j,c)\cdot\mathrm{p}(j|c). & \text{(A.7)} \end{aligned}$$

Bayes' Rule

The basic Bayes' rules are given with and without expanded normalization constant:

$$\begin{aligned} \mathrm{p}(a|b) &= \frac{\mathrm{p}(b|a) \cdot \mathrm{p}(a)}{\mathrm{p}(b)}, && \text{(A.8)} \\ &= \frac{\mathrm{p}(b|a) \cdot \mathrm{p}(a)}{\int \mathrm{p}(b|a) \cdot \mathrm{p}(a)\, da} && \text{(A.9)} \end{aligned}$$

Bayes' rule with an additional conditional c:

$$\mathrm{p}(a|b,c) = \frac{\mathrm{p}(b|a,c) \cdot \mathrm{p}(a|c)}{\mathrm{p}(b|c)} \tag{A.10}$$

Bayes' rule with an additional conditional c and estimate j:

$$\mathrm{p}(a,j|b,c) = \frac{\mathrm{p}(b|a,j,c) \cdot \mathrm{p}(a,j|c)}{\mathrm{p}(b|c)} \tag{A.11}$$

Marginalization, Theorem of Total Probability

Marginalization over b is:

$$\begin{aligned} \mathrm{p}(a) &= \int \mathrm{p}(a,b)\, db, && \text{(A.12)} \\ &= \int \mathrm{p}(a|b) \cdot \mathrm{p}(b)\, db\,. && \text{(A.13)} \end{aligned}$$

Marginalization over b with additional conditional c is:

$$\begin{aligned} \mathrm{p}(a|c) &= \int \mathrm{p}(a,b|c)\, db, && \text{(A.14)} \\ &= \int \mathrm{p}(a|b,c) \cdot \mathrm{p}(b|c)\, db\,. && \text{(A.15)} \end{aligned}$$

Appendix B

Posterior Factorizations

A factorization splits a posterior probability function into smaller factors in order to compute and estimate the posterior. The factorization uses Bayesian rule, product rule, conditional independence, marginalization, normalization and the Markov assumption. The posterior can be proportional ($\propto$) to the factors, or alternatively, a normalization factor η can be used. A general estimation state is denoted with X. For the use-case of train localization, T is the train state and B stands for biases and systematic sensor errors. Measurements are denoted by Z, control input by U and the map by M. The current time step is denoted with k and the previous with $k-1$. A series of all time steps is indexed with $0:k$ and all past time steps are $0:k-1$. The initial state is indexed with zero (X_0) and the first measurement with one (Z_1). The goal is a factorization with a prior posterior, which is the estimation result of the last step. The following general factorizations use the definitions of random variables and causal dependencies on the dynamic Bayesian network (DBN) in Fig. 2.2 (Section 2.1), the train localization factorizations are based on the DBN in Fig. 5.2 (Section 5.1).

B.1 Bayesian Filter Example: Full Joint Posterior

The full posterior $\mathrm{p}(X_{0:k}|Z_{1:k})$ is factorized according to the DBN in Fig. 2.2. The first step of the factorization uses the Bayes' rule Eq. (A.10) with $a = X_{0:k}$, $b = Z_k$ and $c = Z_{1:k-1}$ as additional conditional:

$$
\begin{aligned}
\mathrm{p}(X_{0:k}|Z_{1:k}) &\overset{\substack{\text{Bayes}\\ \text{cond.indep.}}}{=} \frac{\mathrm{p}(Z_k|X_{\cancel{0:}k}, \cancel{Z_{1:k\text{-}1}}) \cdot \mathrm{p}(X_{0:k}|Z_{1:k\text{-}1})}{\mathrm{p}(Z_k|Z_{1:k\text{-}1})}, \\
&\overset{\substack{\text{prod.rule}\\ \text{cond.indep.}}}{=} \frac{1}{\mathrm{p}(Z_k|Z_{1:k\text{-}1})} \cdot \mathrm{p}(Z_k|X_k) \cdot \mathrm{p}(X_k|X_{\cancel{0:}k\text{-}1}, \cancel{Z_{1:k\text{-}1}}) \cdot \mathrm{p}(X_{0:k\text{-}1}|Z_{1:k\text{-}1}), \\
&\overset{\substack{\text{cond.indep.}\\ \text{Markov}}}{=} \underbrace{\eta}_{\text{normalization}} \cdot \underbrace{\mathrm{p}(Z_k|X_k)}_{\text{meas. likelihood}} \cdot \underbrace{\mathrm{p}(X_k|X_{k\text{-}1})}_{\text{transition}} \cdot \underbrace{\mathrm{p}(X_{0:k\text{-}1}|Z_{1:k\text{-}1})}_{\text{prior}}.
\end{aligned}
\tag{B.1}
$$

The final factorization of the full posterior consists of measurement likelihood, state transition, prior posterior and a normalization constant.

B.2 Bayesian Filter Example: Filter Posterior

The Bayesian filter posterior is factorized according to the DBN in Fig. 2.2 as well. The filter posterior $\mathrm{p}(X_k|Z_{1:k})$ is factorized with Bayes' rule Eq. (A.10) with $a = X_k$, $b = Z_k$ and $c = Z_{1:k-1}$ as additional conditional:

$$
\begin{aligned}
\mathrm{p}(X_k|Z_{1:k}) &\overset{\text{Bayes}}{=} \frac{\mathrm{p}(Z_k|X_k, \cancel{Z_{1:k-1}}) \cdot \mathrm{p}(X_k|Z_{1:k-1})}{\mathrm{p}(Z_k|Z_{1:k-1})}, \\
&\overset{\substack{\text{cond.}\\ \text{indep.}}}{=} \eta \cdot \underbrace{\mathrm{p}(Z_k|X_k)}_{\text{meas. likelihood}} \cdot \mathrm{p}(X_k|Z_{1:k-1}).
\end{aligned}
\tag{B.2}
$$

The current hidden state depends on the last state X_{k-1} and the last factor is expanded by 2.13 and the last state X_{k-1}:

$$
\mathrm{p}(X_k|Z_{1:k-1}) = \int \underbrace{\mathrm{p}(X_k|X_{k-1}, \cancel{Z_{1:k-1}})}_{\text{transition}} \cdot \underbrace{\mathrm{p}(X_{k-1}|Z_{1:k-1})}_{\text{prior}} dX_{k-1}. \tag{B.3}
$$

The normalization η is given by:

$$
\eta^{-1} = \mathrm{p}(Z_k|Z_{1:k-1}) = \int \mathrm{p}(Z_k|X_{k-1}, \cancel{Z_{1:k-1}}) \cdot \mathrm{p}(X_{k-1}|Z_{1:k-1}) dX_{k-1}. \tag{B.4}
$$

B.3 Factorization of the Localization Full Posterior

The localization full posterior is split into factors for a recursive factorization. The DBN in Fig. 5.2 shows the train localization problem with random variables and causal dependencies. The first step of the factorization is again with the Bayes' rule:

$$\begin{aligned}
&\mathrm{p}(T_{0:k}, B_{0:k} | Z_{1:k}, U_{1:k}, M) = \\
&\overset{\text{Bayes}}{=} \frac{\mathrm{p}(Z_k | Z_{1:k\text{-}1}, T_{0:k}, B_{0:k}, U_{1:k}, M) \cdot \mathrm{p}(T_{0:k}, B_{0:k} | Z_{1:k\text{-}1}, U_{1:k}, M)}{\mathrm{p}(Z_k | Z_{1:k\text{-}1}, U_{1:k}, M)}, \\
&\overset{\text{norm.}}{\propto} \mathrm{p}(Z_k | Z_{1:k\text{-}1}, T_{0:k}, B_{0:k}, U_{1:k}, M) \cdot \mathrm{p}(T_{0:k}, B_{0:k} | Z_{1:k\text{-}1}, U_{1:k}, M), \\
&\overset{\substack{\text{cond.indep.}\\ \text{Markov}}}{\propto} \mathrm{p}(Z_k | T_k, B_k, M) \cdot \mathrm{p}(T_{0:k}, B_{0:k} | Z_{1:k\text{-}1}, U_{1:k}, M).
\end{aligned} \tag{B.5}$$

The first factor of Eq. (B.5) splits in the two types of Z:

$$\begin{aligned}
\mathrm{p}(Z_k | T_k, B_k, M) = {} & \mathrm{p}(Z_k^{\mathrm{IMU}}, Z_k^{\mathrm{GNSS}} | T_k, B_k^{\mathrm{IMU}}, B_k^{\mathrm{GNSS}}, M) \\
\overset{\substack{\text{prod. rule}\\ \text{cond.indep.}}}{=} {} & \mathrm{p}(Z_k^{\mathrm{IMU}} | T_k, B_k^{\mathrm{IMU}}) \cdot \mathrm{p}(Z_k^{\mathrm{GNSS}} | T_k, B_k^{\mathrm{GNSS}}).
\end{aligned} \tag{B.6}$$

The second factor of Eq. (B.5) is factored in a recursive form:

$$\begin{aligned}
&\mathrm{p}(T_{0:k}, B_{0:k} | Z_{1:k\text{-}1}, U_{1:k}, M) = \\
&\overset{\text{prod. rule}}{=} \mathrm{p}(T_k, B_k | T_{0:k\text{-}1}, B_{0:k\text{-}1}, Z_{1:k\text{-}1}, U_{1:k}, M) \cdot \mathrm{p}(T_{0:k\text{-}1}, B_{0:k\text{-}1} | Z_{1:k\text{-}1}, U_{1:k}, M), \\
&\overset{\substack{\text{cond.indep.}\\ \text{Markov}}}{=} \mathrm{p}(T_k, B_k | T_{0:k\text{-}1}, B_{0:k\text{-}1}, U_k, M) \cdot \mathrm{p}(T_{0:k\text{-}1}, B_{0:k\text{-}1} | Z_{1:k\text{-}1}, U_{1:k\text{-}1}, M).
\end{aligned} \tag{B.7}$$

The first factor of Eq. (B.7) is factorized again:

$$\begin{aligned}
&\mathrm{p}(T_k, B_k | T_{0:k\text{-}1}, B_{0:k\text{-}1}, U_k, M) = \\
&\overset{\text{prod. rule}}{=} \mathrm{p}(T_k | T_{0:k\text{-}1}, B_{0:k\text{-}1}, B_k, U_k, M) \cdot \mathrm{p}(B_k | T_{0:k\text{-}1}, B_{0:k\text{-}1}, U_k, M) \qquad \text{(B.8)} \\
&\overset{\text{cond.indep.}}{=} \mathrm{p}(T_k | T_{k\text{-}1}, U_k, M) \cdot \mathrm{p}(B_k | B_{k\text{-}1})
\end{aligned}$$

The train state transition of Eq. (B.8) is factorized by a linear and an non-linear part of the train state ($T = \{T^l, T^n\}$) with product rule and a conditional independence of the linear part from the map:

$$\mathrm{p}(T_k | T_{k\text{-}1}, U_k, M) = \mathrm{p}(T_k^n | T_k^l, T_{k\text{-}1}^n, U_k, M) \cdot \mathrm{p}(T_k^l | T_{k\text{-}1}^l, U_k) \tag{B.9}$$

Inserting Eq. (B.9), Eq. (B.8) in Eq. (B.7) together with Eq. (B.6) in Eq. (B.5) results in the factorized localization posterior:

$$\begin{aligned}\underbrace{\mathrm{p}(T_{0:k}, B_{0:k}|Z_{1:k}, U_{1:k}, M)}_{\text{posterior}} \propto\ &\underbrace{\mathrm{p}(Z_k^{\mathrm{GNSS}}|T_k, B_k^{\mathrm{GNSS}}) \cdot \mathrm{p}(Z_k^{\mathrm{IMU}}|T_k, B_k^{\mathrm{IMU}})}_{\text{meas. likelihoods}} \cdot \\ &\cdot \underbrace{\mathrm{p}(B_k|B_{k-1}) \cdot \mathrm{p}(T_k^n|T_k^l, T_{k\text{-}1}^n, U_k, M) \cdot \mathrm{p}(T_k^l|T_{k\text{-}1}^l, U_k)}_{\text{transitions}} \cdot \\ &\cdot \underbrace{\mathrm{p}(T_{0:k\text{-}1}, B_{0:k\text{-}1}|Z_{1:k\text{-}1}, U_{1:k\text{-}1}, M)}_{\text{prior: posterior of k-1}}.\end{aligned} \tag{B.10}$$

B.4 Factorization of the Localization Filter Posterior

The train localization filter posterior is derived in a recursive factorization:

$$\begin{aligned}&\mathrm{p}(T_k, B_k|Z_{1:k}, U_{1:k}, M) = \\ &\overset{\text{Bayes}}{=} \frac{\mathrm{p}(Z_k|Z_{1:k\text{-}1}, T_k, B_k, U_{1:k}, M) \cdot \mathrm{p}(T_k, B_k|Z_{1:k\text{-}1}, U_{1:k}, M)}{\mathrm{p}(Z_k|Z_{1:k\text{-}1}, U_{1:k}, M)}, \\ &\overset{\substack{\text{cond.indep.}\\ \text{Markov}}}{\propto} \quad \mathrm{p}(Z_k|T_k, B_k, M) \cdot \mathrm{p}(T_k, B_k|Z_{1:k\text{-}1}, U_{1:k}, M).\end{aligned} \tag{B.11}$$

The first factor of Eq. (B.11) splits in the different sensor types of Z. In the following, only intrinsic measurements are used and the factor gets conditionally independent of M. The second factor of Eq. (B.11) is extended and marginalized over the previous estimates of $T_{k\text{-}1}$ and $B_{k\text{-}1}$. The equation uses the substitution of $X = \{T, B\}$ for the marginalization over $X_{k\text{-}1}$:

$$\begin{aligned}&\mathrm{p}(X_k|Z_{1:k\text{-}1}, U_{1:k}, M) = \\ &\overset{\substack{\text{margin.}\\ \text{over } X_{k\text{-}1}}}{=} \int \mathrm{p}(X_k|X_{k\text{-}1}, Z_{1:k\text{-}1}, U_{1:k}, M) \cdot \mathrm{p}(X_{k\text{-}1}|Z_{1:k\text{-}1}, U_{1:k\text{-}1}, M) dX_{k\text{-}1}.\end{aligned} \tag{B.12}$$

The first factor of the integral is split after reversing the substitution:

$$\mathrm{p}(T_k, B_k|T_{k\text{-}1}, B_{k\text{-}1}, Z_{1:k\text{-}1}, U_{1:k}, M) \overset{\substack{\text{prod.rule}\\ \text{cond.indep.}}}{=} \mathrm{p}(T_k|T_{k\text{-}1}, U_k, M) \cdot \mathrm{p}(B_k|B_{k\text{-}1}). \tag{B.13}$$

The following factorization results with Eq. (B.13) inserted in Eq. (B.12) again inserted in Eq. (B.11), a conditionally independent likelihood of M due to DBN in Fig. 5.2, and a reversed substitution $\{T, B\} = X$:

$$\underbrace{\mathrm{p}(T_k, B_k | Z_{1:k}, U_{1:k}, M)}_{\text{posterior}} \propto \underbrace{\mathrm{p}(Z_k | T_k, B_k)}_{\substack{\text{measurement}\\ \text{likelihood}}} \cdot$$

$$\int \underbrace{\mathrm{p}(T_k | T_{k\text{-}1}, U_k, M) \cdot p(B_k | B_{k\text{-}1})}_{\text{transitions}} \cdot \underbrace{\mathrm{p}(T_{k\text{-}1}, B_{k\text{-}1} | Z_{1:k\text{-}1}, U_{1:k\text{-}1}, M)}_{\text{prior}} \; d(T_{k\text{-}1}, B_{k\text{-}1}). \tag{B.14}$$

The train state transitions can be further split in linear and non-linear values with Eq. (B.9).

B.5 Factorization of the Pre-processed Measurements

The factorization of the pre-processed measurements is similar to the generic Bayesian filter posterior. The IMU measurements are used here as an input. The pre-processed measurements posterior estimates the pre-processed measurements Z^{P} and the IMU biases B^{IMU} from IMU and GNSS measurements. This posterior is factorized with Bayes' rule Eq. (A.10) with $a = \{Z^{\mathrm{P}}_k, B_k\}$, $b = Z^{\mathrm{GNSS}}_k$ and $c = \{Z^{\mathrm{GNSS}}_{1:k\text{-}1}, Z^{\mathrm{IMU}}_{1:k}\}$:

$$\begin{aligned}\mathrm{p}(Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k | Z^{\mathrm{GNSS}}_{1:k}, Z^{\mathrm{IMU}}_{1:k}) &= \\ &\overset{\text{Bayes}}{=} \frac{\mathrm{p}(Z^{\mathrm{GNSS}}_k | Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k, \cancel{Z^{\mathrm{GNSS}}_{1:k-1}}, \cancel{Z^{\mathrm{IMU}}_{1:k}}) \cdot \mathrm{p}(Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k | Z^{\mathrm{GNSS}}_{1:k-1,} Z^{\mathrm{IMU}}_{1:k})}{\mathrm{p}(Z^{\mathrm{GNSS}}_k | Z^{\mathrm{GNSS}}_{1:k-1}, Z^{\mathrm{IMU}}_{1:k})}, \\ &\overset{\substack{\text{cond. indep.}\\ \text{norm.}}}{\propto} \quad \mathrm{p}(Z^{\mathrm{GNSS}}_k | Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k) \cdot \mathrm{p}(Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k | Z^{\mathrm{GNSS}}_{1:k-1}, Z^{\mathrm{IMU}}_{1:k}).\end{aligned} \tag{B.15}$$

The last factor is extended and marginalized with Eq. (A.15), and $a = \{Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k\}$, the previous estimate $b = \{Z^{\mathrm{P}}_{k\text{-}1}, B^{\mathrm{IMU}}_{k\text{-}1}\}$ and $c = \{Z^{\mathrm{GNSS}}_{1:k\text{-}1}, Z^{\mathrm{IMU}}_{1:k\text{-}1}\}$:

$$\mathrm{p}(Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k | Z^{\mathrm{GNSS}}_{1:k-1}, Z^{\mathrm{IMU}}_{1:k}) \overset{\substack{\text{extended,}\\ \text{marginalized}}}{=}$$

$$\int \mathrm{p}(Z^{\mathrm{P}}_k, B^{\mathrm{IMU}}_k | Z^{\mathrm{P}}_{k\text{-}1}, B^{\mathrm{IMU}}_{k\text{-}1}, \cancel{Z^{\mathrm{GNSS}}_{1:k\text{-}1}}, Z^{\mathrm{IMU}}_{\cancel{1}:k}) \cdot \mathrm{p}(Z^{\mathrm{P}}_{k\text{-}1}, B^{\mathrm{IMU}}_{k\text{-}1} | Z^{\mathrm{GNSS}}_{1:k\text{-}1}, Z^{\mathrm{IMU}}_{1:k\text{-}1}) \; d(Z^{\mathrm{P}}_{k\text{-}1}, B^{\mathrm{IMU}}_{k\text{-}1}). \tag{B.16}$$

Finally, the posterior factorization for the pre-processed measurements is:

$$\mathrm{p}(Z_k^{\mathrm{P}}, B_k^{\mathrm{IMU}} | Z_{1:k}^{\mathrm{GNSS}}, Z_{1:k}^{\mathrm{IMU}}) \propto \underbrace{\mathrm{p}(Z_k^{\mathrm{GNSS}} | Z_k^{\mathrm{P}}, B_k^{\mathrm{IMU}})}_{\text{GNSS measurement update}} \cdot \\ \int \underbrace{\mathrm{p}(Z_k^{\mathrm{P}}, B_k^{\mathrm{IMU}} | Z_{k\text{-}1}^{\mathrm{P}}, B_{k\text{-}1}^{\mathrm{IMU}}, Z_k^{\mathrm{IMU}})}_{\text{estimate with IMU input}} \cdot \underbrace{\mathrm{p}(Z_{k\text{-}1}^{\mathrm{P}}, B_{k\text{-}1}^{\mathrm{IMU}} | Z_{1:k\text{-}1}^{\mathrm{GNSS}}, Z_{1:k\text{-}1}^{\mathrm{IMU}})}_{\text{prior}} \; d(Z_{k\text{-}1}^{\mathrm{P}}, B_{k\text{-}1}^{\mathrm{IMU}}). \tag{B.17}$$

Appendix C

Coordinate Frames

C.1 Frame Rotations

This section contains the rotation matrix definition with Euler angles. A rotation matrix is needed for a conversion of measurements from one frame into another frame. A frame here is a Cartesian coordinate system. The turn directions and the Cartesian axes definitions follow the right-hand rule according to [66]. The angles are defined with roll ϕ about the Cartesian x- axis, pitch angle θ about the y-axis and yaw angle ψ about the z-axis. The yaw rotation matrix rotates the coordinate axes and is [66]:

$$\mathbf{C}_{\psi} = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{C.1}$$

The pitch rotation matrix for the coordinate axes rotation is [66]:

$$\mathbf{C}_{\theta} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}. \tag{C.2}$$

The roll rotation matrix for the coordinate axes rotation is [66]:

$$\mathbf{C}_{\phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}. \tag{C.3}$$

The full rotation from the frame b to the frame n is denoted with $\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}$, and computed from the three axes rotations in the following order (see [66]):

$$\mathbf{C}_{\mathrm{n}}^{\mathrm{b}} = \mathbf{C}_{\phi} \cdot \mathbf{C}_{\theta} \cdot \mathbf{C}_{\psi}. \tag{C.4}$$

The transverse rotation from frame n to the frame b is the transposed matrix:

$$\mathbf{C}_{\mathrm{b}}^{\mathrm{n}} = \mathbf{C}_{\mathrm{n}}^{\mathrm{b}\,\mathrm{T}}. \tag{C.5}$$

The full direction cosine matrix (DCM) $\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}$ is [66]:

$$\mathbf{C}_{\mathrm{b}}^{\mathrm{n}} = \begin{pmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{pmatrix}. \tag{C.6}$$

The rotation between a horizontal leveled frame and a tilted track frame t is defined with the roll and pitch rotations Eq. (C.3), Eq. (C.2), or with the transposed DCM of Eq. (C.6) and $\psi = 0$:

$$\begin{aligned} \mathbf{C}_{\mathrm{hor.}}^{\mathrm{t}} &= \mathbf{C}_{\mathrm{n}}^{\mathrm{t}}(\psi = 0) \\ &= \mathbf{C}_{\phi}\mathbf{C}_{\theta} \\ &= \mathbf{C}_{\mathrm{t}}^{\mathrm{hor.}\ \mathrm{T}} \\ &= \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ \sin\phi\sin\theta & \cos\phi & \sin\phi\cos\theta \\ \cos\phi\sin\theta & -\sin\phi & \cos\phi\cos\theta \end{pmatrix}. \end{aligned} \tag{C.7}$$

The turn rates $\vec{\omega}$ $(\omega_x, \omega_y, \omega_z)$ are measured with the gyroscopes. The relation between measurements and the turn rates of the coordinate axes $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ are given by [66]:

$$\begin{aligned} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} &= \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \mathbf{C}_{\phi} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \mathbf{C}_{\phi}\mathbf{C}_{\theta} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\ &= \begin{pmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta \\ \dot{\psi}\cos\phi\cos\theta - \dot{\theta}\sin\phi \end{pmatrix}. \end{aligned} \tag{C.8}$$

C.2 Earth Frame

The world geodetic system of 1984 (WGS84) is defined with a semi-major axis of the ellipsoid of $a = 6\,378\,137\,\mathrm{m}$ and a flattening of $f = \frac{1}{298.257223563}$ [47].

The rotation matrix from the ECEF coordinates of the Earth-frame e to NED coordinates in the navigation frame n is defined with the latitude φ and longitude λ of the Earth ellipsoid:

$$\mathbf{C}_{\mathrm{e}}^{\mathrm{n}} = \begin{pmatrix} -\sin\varphi\cos\lambda & \sin\varphi\sin\lambda & \cos\varphi \\ -\sin\lambda & \cos\varphi & 0 \\ -\cos\varphi\cos\lambda & -\cos\varphi\sin\lambda & \sin\varphi \end{pmatrix}. \tag{C.9}$$

The inverse transformation from navigation frame to ECEF frame is again the transposed DCM: $\mathbf{C}_{\mathrm{n}}^{\mathrm{e}} = \mathbf{C}_{\mathrm{e}}^{\mathrm{n}T}$.

The meridian radius in north direction and the transverse Earth radius in east direction depend on the latitude φ and are defined in [66, 47] with:

$$R_n(\varphi) = \frac{a \cdot (1 - e^2)}{(1 - e^2 \sin^2 \varphi)^{\frac{3}{2}}}, \tag{C.10}$$

$$R_e(\varphi) = \frac{a}{(1 - e^2 \sin^2 \varphi)^{\frac{1}{2}}}. \tag{C.11}$$

The major eccentricity squared e^2 is defined with the flattening f:

$$e^2 = f(2 - f). \tag{C.12}$$

Appendix D

Strapdown Method for an Inertial Navigation System

This section describes the differential equations that are needed for the computation of attitude, velocity and position from acceleration and turn rate measurements. An acceleration measurement $\vec{a}_{\mathrm{ib}}^{\mathrm{b}}$ is also called specific force measurement, because the acceleration of the train and the gravity is measured in combination. The turn rate measurements are $\vec{\omega}_{\mathrm{ib}}^{\mathrm{b}}$. Specific force and turn rates are defined in body frame (b) and measured between inertial frame and body frame (ib).

Attitude

The attitude is computed from the turn rate measurements of the gyroscopes. The differential equation for attitude with the direction cosine matrix (DCM) $\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}$ from body frame (b) to navigation frame (n) is defined with [66]:

$$\dot{\mathbf{C}}_{\mathrm{b}}^{\mathrm{n}} = \mathbf{C}_{\mathrm{b}}^{\mathrm{n}} \mathbf{\Omega}_{\mathrm{nb}}^{\mathrm{b}}. \tag{D.1}$$

This equation shows how the attitude, represented with the DCM, changes over time with the turn rate measurements contained in $\mathbf{\Omega}_{\mathrm{nb}}^{\mathrm{b}}$. This turn rate matrix are defined in body frame (b) and measured between navigation frame and body frame (nb) A DCM can be defined with the Euler angles roll (ϕ), pitch (θ) and yaw (ψ) between body and navigation frame, see Eq. (C.6). Alternatively, the attitude and the differential equation can be represented by quaternions or Euler angles. Algorithmic realizations for this differential equation with discrete turn rate data updates and coning correction can be found in [91, 66, 47]. The matrix $\Omega^{\mathrm{b}}_{\mathrm{nb}}$ is the skew symmetric matrix of the components of $\vec{\omega}^{\mathrm{b}}_{\mathrm{nb}}$ [66]:

$$\mathbf{\Omega}^{\mathrm{b}}_{\mathrm{nb}} = \begin{pmatrix} 0 & -\omega^{\mathrm{b}}_{\mathrm{nb,z}} & \omega^{\mathrm{b}}_{\mathrm{nb,y}} \\ \omega^{\mathrm{b}}_{\mathrm{nb,z}} & 0 & -\omega^{\mathrm{b}}_{\mathrm{nb,x}} \\ -\omega^{\mathrm{b}}_{\mathrm{nb,y}} & \omega^{\mathrm{b}}_{\mathrm{nb,x}} & 0 \end{pmatrix}. \tag{D.2}$$

The body turn rates $\vec{\omega}^{\mathrm{b}}_{\mathrm{nb}}$ are defined between navigation and body frame and contain Earth rate, transport rate and turn rate measurements [66]:

$$\vec{\omega}^{\mathrm{b}}_{\mathrm{nb}} = \underbrace{\vec{\omega}^{\mathrm{b}}_{\mathrm{ib}}}_{\substack{\text{turn rate}\\ \text{measurements}}} - \mathbf{C}^{\mathrm{b}}_{\mathrm{n}}(\underbrace{\vec{\omega}^{\mathrm{n}}_{\mathrm{ie}}}_{\substack{\text{Earth}\\ \text{rate}}} + \underbrace{\vec{\omega}^{\mathrm{n}}_{\mathrm{en}}}_{\substack{\text{transport}\\ \text{rate}}}). \tag{D.3}$$

The Earth turn rate Ω is defined in Earth frame between inertial and Earth frame and is further converted to the navigation frame with the latitude φ [66]:

$$\vec{\omega}^{\mathrm{n}}_{\mathrm{ie}} = \begin{pmatrix} \Omega\cos\varphi & 0 & -\Omega\sin\varphi \end{pmatrix}^T. \tag{D.4}$$

The navigation frame of a vehicle with a horizontal motion over the Earth shape requires to rotate according to the Earth curvature. The transport rate considers this effect and is defined in the navigation frame between Earth frame and navigation frame [66]:

$$\vec{\omega}^{\mathrm{n}}_{\mathrm{en}} = \begin{pmatrix} \frac{v_e}{R_e+h} & \frac{-v_n}{R_n+h} & \frac{-v_e\tan\varphi}{R_e+h} \end{pmatrix}^T, \tag{D.5}$$

with the local Earth radius in north (R_n) and east (R_e) directions, and the velocity components the in navigation frame in north (v_n), east (v_e) and down (v_d) direction. A common simplification for low velocities and for low-cost IMUs is to neglected the Earth rate and the transport rate.

Velocity and position

The differential equation for the velocity $\vec{v}^{\mathrm{n}}_{\mathrm{eb}} = \begin{pmatrix} v_n & v_e & v_d \end{pmatrix}^T$ in navigation frame between Earth and body frame (eb) is given by [66, 47] with adjusted notation:

$$\dot{\vec{v}}^{\mathrm{n}}_{\mathrm{eb}} = \mathbf{C}^{\mathrm{n}}_{\mathrm{b}} \underbrace{\vec{a}^{\mathrm{b}}_{\mathrm{ib}}}_{\substack{\text{accel.}\\ \text{meas.}}} - \overbrace{(2\underbrace{\vec{\omega}^{\mathrm{n}}_{\mathrm{ie}}}_{\substack{\text{Earth}\\ \text{rate}}} + \underbrace{\vec{\omega}^{\mathrm{n}}_{\mathrm{en}}}_{\substack{\text{transport}\\ \text{rate}}}) \times \vec{v}^{\mathrm{n}}_{\mathrm{eb}}}^{\text{Coriolis correction}} + \underbrace{\vec{g}^{\mathrm{n}}_{\mathrm{l}}}_{\substack{\text{local}\\ \text{gravity}}}. \tag{D.6}$$

This equation shows how the train velocity changes from acceleration measurements that are corrected from Coriolis effects and the gravity. Algorithmic realizations for the differential equation of velocity Eq. (D.6) with discrete acceleration data updates and sculling correction can be found in [92, 66, 47]. The Coriolis correction term considers the Coriolis acceleration from the vehicle motion and a combined turn rate from Earth and transport rate. The gravity part $\vec{g}_{\mathrm{l}}^{\mathrm{n}}$ of Eq. (D.6) respects the local gravity vector. There are various realizations in the literature [93, 66] and the local gravity models depend usually on position and height. Finally, the differential equation for the position propagation over time is given by [66]:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{pmatrix} = \begin{pmatrix} v_n \frac{1}{(R_n+h)} \\ v_e \frac{1}{(R_e+h)\cos\varphi} \\ -v_d \end{pmatrix}. \tag{D.7}$$

A common discretization of Eq. (D.6) and Eq. (D.7) is a first order approximation with an integration over the time step period Δt and a high update rate compared to the dynamics.

Appendix E

Train Experiment Schedule

Run	**Departure station**	**Arrival station**	**Motion direction**	**Branching switches**	**Time**	**Distance** [km]
1	Augsburg	Friedberg	backward	8	10 min	8
2	Friedberg	Augsburg	forward	7	10 min	8
3	Augsburg	Ingolstadt	backward	22	1 h	66
4	Ingolstadt	Augsburg	forward	20	1 h	66
5	Augsburg	Friedberg	backward	8	10 min	8
6	Friedberg	Augsburg	forward	7	10 min	8
7	Augsburg	Aichach	backward	11	30 min	25
8	Aichach	Augsburg	forward	9	30 min	25
9	Augsburg	Friedberg	backward	8	10 min	8
10	Friedberg	Augsburg	forward	7	10 min	8
				107	4 h	230

Table E.1: Train routes (data set 1).

Run	**Departure station**	**Arrival station**	**Motion direction**	**Branching switches**	**Time**	**Distance** [km]
101	Augsburg	Ingolstadt	backward	21	1 h	66
102	Ingolstadt	Augsburg	forward	20	1 h	66
103	Augsburg	Friedberg	backward	8	10 min	8
104	Friedberg	Augsburg	forward	7	10 min	8
105	Augsburg	Ingolstadt	backward	22	1 h	66
106	Ingolstadt	Augsburg	forward	20	1 h	66
107	Augsburg	Friedberg	backward	8	10 min	8
108	Friedberg	Augsburg	forward	7	10 min	8
109	Augsburg	Aichach	backward	11	30 min	25
110	Aichach	Augsburg	forward	9	30 min	25
111	Augsburg	Friedberg	backward	8	10 min	8
				141	5:50 h	354

Table E.2: Train routes (data set 2).

Nomenclature

Navigation, Inertial Navigation

φ	°	latitude
λ	°	longitude
h	m	height
$\vec{p}$	°, °, m	position vector: latitude, longitude, height (WGS84)
$x_\mathrm{n}, x_\mathrm{e}, x_\mathrm{d}$	m	position: north, east, down (navigation frame)
$\vec{v}$	m/s	speed vector (navigation frame)
$v_\mathrm{n}, v_\mathrm{e}, v_\mathrm{d}$	m/s	speed: north, east, down (navigation frame)
t	s	time
ϕ	°	roll
θ	°	pitch
ψ	°	yaw
$\dot{\phi}, \dot{\theta}, \dot{\psi}$	rad/s	turn rates in roll, pitch, yaw direction
a_x, a_y, a_z	m/s^2	acceleration x, y, z in train frame
$\vec{a}$	m/s^2	acceleration vector (x,y,z) in train frame
$\omega_x, \omega_y, \omega_z$	rad/s	turn rate x, y, z in train frame
$\vec{\omega}$	rad/s	turn rate vector (x,y,z) in train frame
$b^\mathrm{ax}, b^\mathrm{ay}, b^\mathrm{az}$	m/s^2	acceleration sensor bias
$b^{\omega\mathrm{x}}, b^{\omega\mathrm{y}}, b^{\omega\mathrm{z}}$	rad/s	gyroscope sensor bias
g, $\vec{g}$	m/s^2	gravity, gravity vector
$\mathbf{C}_\mathrm{s}^\mathrm{b}$	3×3	direction cosine matrix, from sensor to body frame
$\mathbf{C}_\mathrm{b}^\mathrm{n}$	3×3	direction cosine matrix, from body to navigation frame
R_e	m	local transverse Earth radius, east
R_n	m	local meridian Earth radius, north

Train Trajectory and Railway Track Geometry

φ	°	latitude
λ	°	longitude
h	m	height
ϕ	°	roll (train), bank angle (track)
θ	°	pitch (train), slope angle (track)
ψ	°	yaw (train), heading angle to north (track)
c^{ϕ}	$1/\,\mathrm{km}$	bank curvature
c^{θ}	$1/\,\mathrm{km}$	slope curvature
c^{ψ}	$1/\,\mathrm{km}$	heading curvature

Measurements, Track Geometry Computations

a_{lat}	$\mathrm{m}/\,\mathrm{s}^2$	acceleration in lateral direction
ψ_{m}	°	motion heading
$\vec{B}$	T	magnetic field, flux density
$\vec{B}_N$		magnetic field, flux density normalized to Earth field
f	Hz	frequency
n		harmonic
f_n	Hz	n^{th} harmonic frequency
d	m	wheel diameter
r	m	track radius
δ_{AT}	m	along-track distance on tracks
δ_{CT}	m	cross-track distance between a position and track
d_{AT}	m	along-track distance (length of track tangent vector)
d_{CT}	m	cross-track distance (length of track normal vector)

Railway Navigation

id	$\in \mathbb{N}$	discrete railway track ID
s	m	track location
Δs	m	displacement (track frame)
o^{t}	$\pm$	direction of the train frame in track frame
o^{v}	$+, 0, -$	direction of the train speed in train frame
o^{m}	$+, 0, -$	direction of the train motion in track frame
$\{id, s\}$	$\mathbb{N}, \mathrm{m}$	topological location
T^{topo}	$\mathbb{N}, \mathrm{m}, \pm$	topological pose: $\{id, s, o^{\mathrm{t}}\}$

U^{acc}	$\mathrm{m/\,s^2}$	train control: acceleration/brake
U^{m}	$+,0,-$	train control: forward/stop/backward motion
U^{sw}	left/right	train control: switch way (alternatively straight/turn)
$\dot{s}$	$\mathrm{m/\,s}$	train speed (track frame)
$\ddot{s}$	$\mathrm{m/\,s^2}$	train acceleration (track frame)
v	$\mathrm{m/\,s}$	train speed (train frame)
a_{tb}	$\mathrm{m/\,s^2}$	train acceleration/deceleration (train frame)
$\mathbf{C}(o^{\mathrm{t}})$	3×3	train-to-track frame and track-to-train frame rotation

Random Variables

X_k	state or random variable X at time step k
$X_{k\text{-}1}$	prior, X of last time step
$X_{1:k}$	history of X from first to current time step k
$X_{k_0:k}$	sequence of X from k_0 to k
X^l, X^n	linear states, non-linear states
T	train state
$T^{\mathrm{topo}}, T^v, T^{\mathrm{traj}}$	train state: topological pose, train velocity, trajectory
T^l	linear train states (velocity, displacement)
T^n	non-linear train states (topological pose, trajectory)
M	map of railway tracks
U	train control input
V	network control input
Z	measurement(s)
$Z^{\mathrm{IN}}, Z^{\mathrm{EX}}$	intrinsic measurement, extrinsic measurement
$Z^{\mathrm{GNSS}}, Z^{\mathrm{IMU}}, Z^{\mathrm{P}}$	GNSS, IMU, pre-processed measurements
B	bias
$B^{\mathrm{GNSS}}, B^{\mathrm{IMU}}$	GNSS bias, IMU bias
E^{INS}	INS error state

Probabilistic Theory, Estimation Filters

$\mathrm{p}(x)$	probability density function of x
$\mu(x)$	point mass function for discrete random variables
$\delta()$	Dirac delta function
$\mathrm{p}(x\|y)$	conditional probability density function of x given y
$\mathrm{E}(x)$	expected value of x
$\mathrm{Cov}(x,y)$	covariance of x and y

$\hat{x}$	estimate of quantity x
n	noise
η	normalization factor
$\mathcal{N}\left(\mu, \sigma^2\right)$	Normal distribution function (mean μ and variance σ^2)
$\boldsymbol{x}_k$	mean vector of estimation state X at time k
$\boldsymbol{\Sigma}, \boldsymbol{P}$	covariance matrix
$\boldsymbol{F}$	state-transition or system matrix
$\mathrm{f}(x)$	non-linear state-transition or system model
$\boldsymbol{Q}$	process noise covariance matrix
$\boldsymbol{z}$	measurement vector
$\boldsymbol{H}$	measurement model matrix
$\mathrm{h}(x)$	non-linear measurement model
$\boldsymbol{R}, \boldsymbol{\Sigma}_z$	measurement noise covariance matrix
$\boldsymbol{S}$	innovation covariance matrix
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{I}$	Identity matrix
$\mathcal{H}$	set of all hypotheses
$\mathcal{H}^j$	single hypothesis with index j
w^j	weight of j'th hypothesis
N_p	number of particles
x^i	i-'th particle state
$\mathrm{q}()$	proposal function, importance density
w^i	weight of the i'th particle
$\hat{N}_{\mathrm{eff}}$	effective number of particles
N_{th}	threshold for resampling
$\hat{\boldsymbol{x}}$	mean estimate vector
$\hat{\boldsymbol{P}}$	covariance estimate matrix

Implementations

$\boldsymbol{\mu}^{\mathrm{m}}, \boldsymbol{\Sigma}^{\mathrm{m}}$	track geometry: mean vector (track frame), covariance matrix
$\boldsymbol{\mu}^{\mathrm{traj}}, \boldsymbol{\Sigma}^{\mathrm{traj}}$	train trajectory: mean vector (train frame), covariance matrix
$\boldsymbol{z}^{\mathrm{P}}, \boldsymbol{\Sigma}^{\mathrm{P}}$	pre-processed measurements: mean (train frame), cov. matrix
$\boldsymbol{z}^{\varphi,\lambda}, \boldsymbol{\Sigma}^{\varphi,\lambda}$	position measurements: mean, covariance matrix
$\boldsymbol{z}^{\mathrm{traj}}, \boldsymbol{\Sigma}^{\mathrm{z,traj}}$	pre-processed trajectory meas. mean (train frame), cov. matrix
$\boldsymbol{z}^{\mathrm{IMU}}, \boldsymbol{z}^{\mathrm{GNSS}}$	IMU measurement vector, GNSS measurement vector

$\boldsymbol{x}^{\mathrm{odo}}$	odometry state vector: displacement, speed, acceleration, bias
Δs^{odo}	displacement, computed from the odometry
Δs^{meas}	measured along-track displacement
σ_s^2	along-track variance, track location variance
$\sigma_{\Delta s}^2$	variance of displacement
$\sigma_{\mathrm{AT}}^2, \sigma_{\mathrm{CT}}^2$	along-track distance variance, cross-track distance variance
σ_ψ^2	heading variance
$\sigma_{c^\psi}^2$	heading curvature variance
$\boldsymbol{H}, \boldsymbol{H}^{\varphi,\lambda}, \boldsymbol{H}^{\mathrm{TI}}$	measurement matrix: position, track identification
$\boldsymbol{z}^{\mathrm{TI}}, \boldsymbol{\Sigma}^{\mathrm{TI}}$	track identification meas. mean (train frame), cov. matrix
$\boldsymbol{\Delta}^{\mathrm{TI}}, \boldsymbol{\Sigma}^{\Delta\mathrm{TI}}$	track identification innovation, innovation covariance
ξ	SLAM mapping phase (white-space phase or prior-map phase)
$s_{k':k''}$	1-D track location sequence over time between k' and k''
$s_{s':s''}^{\mathrm{m}}$	constant sample grid: 1-D track location between s' and s''
$\boldsymbol{\mu}_{k':k''}^{\mathrm{m}}$	track geometry sequence over time between k' and k''
$\boldsymbol{\mu}_{s':s''}^{\mathrm{m}}$	track geometry sequence over 1-D locations between s' and s''
$\boldsymbol{x}^{\mathrm{INS}}$	INS state vector (position, velocity, attitude, acc. & gyr. biases)
$\Delta\boldsymbol{x}, \boldsymbol{\Sigma}^{\Delta}$	ESKF error state vector, covariance matrix
$\Delta x_{\mathrm{n}}, \Delta x_{\mathrm{e}}, \Delta x_{\mathrm{d}}$	error state: correction of north, east, down position
$\Delta v_{\mathrm{n}}, \Delta v_{\mathrm{e}}, \Delta v_{\mathrm{d}}$	error state: correction of north, east, down velocity
$\Delta\alpha, \Delta\beta, \Delta\gamma$	error state: Euler angle increments
$\Delta b^{\mathrm{ax}}, \Delta b^{\mathrm{ay}}, \Delta b^{\mathrm{az}}$	error state: correction of acceleration biases
$\Delta b^{\omega\mathrm{x}}, \Delta b^{\omega\mathrm{y}}, \Delta b^{\omega\mathrm{z}}$	error state: correction of gyroscope biases

Evaluation

ϵ	track-selective evaluation result
ϵ_k^{ok}	correct track estimate
$\epsilon_k^{\mathrm{tol}}$	wrong track estimate but tolerated within switch sections
$\epsilon_k^{\mathrm{err}}$	wrong track estimate
A_{TS}	track-selective accuracy

E_{TS}	track-selective error ratio
A_{SW}	switch-way accuracy
$\vec{p}_{\mathrm{ref}}$	map-matched position on reference track map
e	cross-track position error: measurement to reference
E_{meas}	RMSE of cross-track position error
$e_{id,s}$	cross-track map error: estimated map to reference map
E_{map}	RMSE of cross-track map error
$F_e(\hat{e})$	CDF of cross-track position error
N	sum of all samples for the error statistics

Acronyms

ATO	automatic train operation
ATP	automatic train protection
CDF	cumulative distribution function
COTS	commercial-off-the-shelf
DBN	dynamic Bayesian network
DCM	direction cosine matrix
ECEF	Earth centered, Earth fixed
EGNOS	European geostationary navigation overlay service
EKF	extended Kalman filter
ESKF	error-state Kalman filter
ETCS	European train control system
FOG	fiber optical gyroscope
GNSS	global navigation satellite system
GPS	global positioning system
HMM	hidden Markov model
IMU	inertial measurement unit
INS	inertial navigation system
LLH	latitude, longitude, height
MEMS	micro electro mechanical system
MHT	multi hypotheses tracker
MMSE	minimum mean square estimate
NED	north-east-down
PDF	probability density function
PF	particle filter

PTC	positive train control
PVT	position, velocity, time
RCAS	railway collision avoidance system
RMSE	root mean squared error
RTK	real time kinematic
SAR	synthetic aperture radar
SBAS	satellite based augmentation system
SLAM	simultaneous localization and mapping
UKF	unscented Kalman filter
UTO	unattended train operation
WAAS	wide area augmentation system
WGS84	world geodetic system of 1984

List of Figures

List of Tables

List of Publications

List of peer-reviewed publications in chronological order.

Journals

[J1] O. Heirich, “Bayesian train localization with particle filter, loosely coupled GNSS, IMU, and a track map,” *Journal of Sensors*, 2016.

[J2] O. Heirich, B. Siebler, and E. Hedberg, “Study of train-side passive magnetic measurements with applications to train localization,” *Journal of Sensors*, 2017.

Conference Papers

[P1] O. Heirich, A. Lehner, P. Robertson, and T. Strang, “Measurement and analysis of train motion and railway track characteristics with inertial sensors,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct. 2011, pp. 1995 –2000.

[P2] O. Heirich, P. Robertson, A. Cardalda Garcia, T. Strang, and A. Lehner, “Probabilistic localization method for trains,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012.

[P3] O. Heirich, P. Robertson, A. Cardalda Garcia, and T. Strang, “Bayesian train localization method extended by 3d geometric railway track observations from inertial sensors,” in *ISIF Information Fusion Conference*, July 2012.

[P4] O. Heirich, P. Robertson, and T. Strang, “RailSLAM - localization of rail vehicles and mapping of geometric railway tracks,” in *International Conference on Robotics and Automation, IEEE ICRA*, Mai 2013.

[P5] O. Heirich, A. Steingass, A. Lehner, and T. Strang, “Velocity and location information from onboard vibration measurements of rail vehicles,” in *Information Fusion (FUSION), 2013 16th International Conference on*, July 2013.

[P6] O. Heirich and B. Siebler, “Train-side passive magnetic measurements,” in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Pisa, Italy, May 2015.

[P7] ——, “Onboard train localization with track signatures: Towards GNSS redundancy,” in *ION GNSS+ Conference*, 2017.

Coauthored Papers

[C1] A. Lehner, C. R. García, T. Strang, and O. Heirich, “Measurement and analysis of the direct train to train propagation channel in the 70 cm UHF-band,” in *Communication Technologies for Vehicles*, ser. Lecture Notes in Computer Science, T. Strang, A. Festag, A. Vinel, R. Mehmood, C. R. Garcia, and M. Röckl, Eds., vol. 6596. Springer, 2011, pp. 45–57.

[C2] T. Strang, A. Lehner, C. Rico García, O. Heirich, and A. Grosch, “Cooperative situation awareness for a railway collision avoidance system (RCAS),” in *9th Int. Conference on Pervasive Computing (Pervasive)*, 2011.

[C3] B. Belabbas, A. Grosch, O. Heirich, A. Lehner, and T. Strang, “Curvature classification for trains using along-track and cross-track accelerometer and a heading rate gyroscope,” in *European Navigation Conference, ENC GNSS*, 2013.

[C4] O. Crespillo, O. Heirich, and A. Lehner, “Bayesian GNSS/IMU tight integration for precise railway navigation on track map,” in *Position, Location and Navigation Symposium - IEEE/ION PLANS*, May 2014.

[C5] 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 Location and GNSS (ICL-GNSS)*, 2017.

[C6] B. Siebler, O. Heirich, and S. Sand, “Bounding INS positioning errors with magnetic-field-signatures in railway environments,” in *ION GNSS+ Conference*, 2017.

Supervised Master Theses

[M1] A. Cardalda Garcia, “RailSLAM: Simultaneous localization and mapping for railways,” Master’s thesis, University of Oviedo, Spain, 2012.

[M2] O. G. Crespillo, “Map based multisensor railway localization enhanced by raw GNSS data,” Master’s thesis, University of Málaga, Spain, 2013.

[M3] D. Vicente-Diaz, “RailSLAM: Localizacion Multi - Sensor y Mapeado Probabilistico para Ferrocarriles,” Master’s thesis, Universidad de Oviedo, 2015.

[M4] E. Hedberg and M. Hammar, “Train localization and speed estimation using on-board inertial and magnetic sensors,” Master’s thesis, Linköping University, Sweden., 2015.

Bibliography

[1] P. Winter, Ed., *Compendium on ERTMS: European Rail Traffic Management System.* UIC International Union of Railways, 2009.

[2] Positive train control overview. US Department of Transportation, Federal Railroad Administration. [Online]. Available: https://www.fra.dot.gov/Page/P0152

[3] Positive train control. Association of American Railroads. [Online]. Available: https://www.aar.org/ptc

[4] T. Strang, M. Meyer zu Hoerste, and X. Gu, "A Railway Collision Avoidance System Exploiting Ad-hoc Inter-vehicle Communications and Galileo," in *13th World Congress on Intelligent Transportation Systems, London, UK*, 2006.

[5] ERA, "Railway safety performance in the European Union, 2014," European Railway Agency (ERA), Tech. Rep., 2014.

[6] J. Winter, A. Lehner, and E. Polisky, "Electronic coupling of next generation trains," in *Third International Conference on Railway Technology: Research, Development and Maintenance*, Cagliari, Italien, 2016.

[7] A. Doherty, "Stakeholder meeting on the shift^{2}rail strategic master plan - improving the network: advanced infrastructure and traffic management systems for more efficient railways," June 2014. [Online]. Available: http://ec.europa.eu/transport/media/events/doc/2014-06-24-s2r-ip2-ip3.pdf

[8] U. Bock and G. Bicker, "Design and development of a future train concept- virtually coupled train formations," in *9th IFAC Symposium on Transportation Systems*, 2000.

[9] E. Fischer, "Justifying automation," *Railway-Technology.com*, 23.8.2011.

[10] S. Cappaert-Blondelle. (2013) A global bid for automation: UITP observatory of automated metros confirms sustained growth rates for the coming years. http://www.uitp.org/. The International Association of Public Transport (UITP).

[11] R. Müller, "RUBIN moves into testing phase," *Railway Gazette*, 1.2.2005.

[12] "Bahn plant Züge ohne Lokführer," Frankfurter Allgemeine Zeitung, FAZ.NET, 9.6.2016.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics.* The MIT Press, 2006.

[14] P. Stanley, Ed., *ETCS for Engineers.* Institution of Railway Signal Engineers, 2011.

[15] H. Jochim and F. Lademann, *Planung von Bahnanlagen.* Carl Hanser Verlag München, Germany, 2009.

[16] F. Böhringer, "Gleisselektive Ortung von Schienenfahrzeugen mit bord-autonomer Sensorik," Ph.D. dissertation, Universität Karlsruhe (TH), Germany, 2008.

[17] C. Hasberg, S. Hensel, and C. Stiller, "Simultaneous localization and mapping for path-constrained motion," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, June 2012.

[18] O. Plan, "GIS-gestützte Verfolgung von Lokomotiven im Werkbahnverkehr," Ph.D. dissertation, University of German Federal Armed Forces Munich, Germany, 2003.

[19] K. Gerlach and C. Rahmig, "Multi-hypothesis based map-matching algorithm for precise train positioning," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, July 2009.

[20] A. Broquetas, A. Comerón, A. Gelonch, J. M. Fuertes, J. Antonio Castro, D. Felip, M. a. López, and J. a. Pulido, "Track detection in railway sidings based on MEMS gyroscope sensors," *Sensors (Switzerland)*, vol. 12, no. 12, Jan. 2012.

[21] S. Hensel, C. Hasberg, and C. Stiller, “Probabilistic rail vehicle localization with eddy current sensors in topological maps,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, Dec. 2011.

[22] M. Lauer and D. Stein, “A train localization algorithm for train protection systems of the future,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 2, pp. 970–979, April 2015.

[23] J. Wohlfeil, “Vision based rail track and switch recognition for self-localization of trains in a rail network,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011.

[24] R. Ross, “Track and turnout detection in video-signals using probabilistic spline curves,” in *Intelligent Transportation Systems Conference (ITSC), 2012*, Sept 2012.

[25] D. Stein, M. Lauer, and M. Spindler, “An analysis of different sensors for turnout detection for train-borne localization systems,” *Computers in Railways XIV, WIT press*, vol. 135, 2014.

[26] C. Rahmig, L. Johannes, and K. Lüddecke, “Detecting track events with a laser scanner for using within a modified multi-hypothesis based map-matching algorithm for train positioning,” in *Proc. European Navigation Conf. (ENC)*, April 2013.

[27] C. Fouque and P. Bonnifait, “Matching raw GPS measurements on a navigable map without computing a global position,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, June 2012.

[28] 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*, 2017.

[29] S. Saab, “A map matching approach for train positioning. I. Development and analysis,” *Vehicular Technology, IEEE Transactions on*, vol. 49, no. 2, pp. 467 –475, Mar 2000.

[30] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, “Particle filters for positioning, navigation, and tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002.

[31] (2016) DBOpenData: Streckennetz. DeutscheBahn (DB). [Online]. Available: http://data.deutschebahn.com/datasets/streckennetz/

[32] H. Kahmen, *Angewandte Geodäsie: Vermessungskunde*, ser. De-Gruyter-Lehrbuch. de Gruyter, 2005.

[33] N. Heuwold, "Parametrisierung der Gleisgeometrie aus Befahrungsdaten von Gleismesswagen," Ph.D. dissertation, Technische Universität Berlin - Institut für Geodäsie und Geoinformationstechnik, 2001.

[34] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier, "Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments," *Robotics Automation Magazine, IEEE*, vol. 21, no. 3, pp. 26–40, Sept 2014.

[35] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping: Part I the essential algorithms," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, 2006.

[36] 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 National Conference on Artificial Intelligence.* Edmonton, Canada: AAAI, 2002.

[37] ——, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI).* Acapulco, Mexico: IJCAI, 2003.

[38] M. Angermann and P. Robertson, "FootSLAM: Pedestrian simultaneous localization and mapping without exteroceptive sensors - hitchhiking on human perception and cognition," *Proceedings of the IEEE*, May 2012.

[39] T. Bayes and R. Price, "An essay towards solving a problem in the doctrine of chances," *Philosophical Transactions of the Royal Society of London*, vol. 53, 1763.

[40] F. Gustafsson, *Statistical Sensor Fusion.* Studentlitteratur Lund, 2012.

[41] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques and Software.* Artech House, 1993.

[42] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.

[43] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[44] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, 2001.

[45] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[46] E. Wan, R. Van Der Merwe *et al.*, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000.* IEEE, 2000, pp. 153–158.

[47] J. Wendel, *Integrierte Navigationssysteme.* Oldenbourg Verlag München, 2011.

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

[49] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *Aerospace Conference, IEEE*, vol. 4, March 2003.

[50] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence.* Morgan Kaufmann Publishers Inc., 2000.

[51] S. Huang and G. Dissanayake, "A critique of current developments in simultaneous localization and mapping," *International Journal of Advanced Robotic Systems*, 2016.

[52] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Proc. Ubicomp*, 2009.

[53] M. J. Garcia Puyol, “Crowdsourcing motion maps based on FootSLAM for reliable indoor pedestrian navigation in multistory environments.” Ph.D. dissertation, Technische Universität München (TUM), 2017.

[54] Bundesministerium der Justiz und für Verbraucherschutz, “Eisenbahn-Bau- und Betriebsordnung (EBO),” 19th July 2016.

[55] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications.* Artech house, 2005.

[56] Railway markup language railML. railML.org. [Online]. Available: https://www.railml.org

[57] (2015) Open Street Map (OSM). [Online]. Available: http://www.openstreetmap.org

[58] Google Earth. [Online]. Available: https://www.google.com/intl/de/earth/

[59] NASA WorldWind. [Online]. Available: https://worldwind.arc.nasa.gov/

[60] Bayersische Vermessungsverwaltung. [Online]. Available: https://geoportal.bayern.de/geodatenonline

[61] *Railway safety performance in the European Union 2012.* European Railway Agency, 2011. [Online]. Available: http://www.era.europa.eu

[62] C. Hasberg, “Simultane Lokalisierung und Kartierung spurgeführter Systeme,” Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2011.

[63] B. Parkinson and J. Spilker, *Global Positioning System: Theory and Applications*, ser. Progress in Astronautics and Aeronautics. American Institute of Aeronautics and Astronautics, 1996.

[64] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance.* Ganga-Jamuna Press, 2006.

[65] “EGNOS Monthly Performance Report, August 2016,” ESSP (European Satellite Service Provider), Tech. Rep., 2016. [Online]. Available: https://egnos-user-support.essp-sas.eu/new_egnos_ops/?q=content/monthly-performance-reports

[66] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology, 2nd Edition.* The Institution of Electrical Engineers (IEE), 2004.

[67] (2010, Oct) MTi and MTx user manual and technical documentation. Xsens Technologies B.V.

[68] (2013) KVH 1750 inertial measurement unit (IMU), technical manual. KVH Industries, Inc.

[69] R. Czommer, "Leistungsfähigkeit fahrzeugautonomer Ortungsverfahren auf der Basis von Map-Matching-Techniken," Ph.D. dissertation, Universität Stuttgart, Fakultät für Bauingenieur- und Vermessungswesen, 2000.

[70] J. Trehag, P. Händel, and M. Ögren, "Onboard estimation and classification of a railroad curvature," *IEEE Transactions on Instrumentation and Measurement*, 2010.

[71] N. Heuwold, "Parametrisierung der Gleisgeometrie aus Befahrungsdaten von Gleismesswagen," Ph.D. dissertation, TU Berlin, Bauingenieur- und Vermessungswesen, 2001.

[72] M. Malvezzi, B. Allotta, and M. Rinchi, "Odometric estimation for automatic train protection and control systems," *Vehicle System Dynamics*, vol. 49, no. 5, pp. 723–739, 2011.

[73] A. Mirabadi, N. Mort, and F. Schmid, "Application of sensor fusion to railway systems," in *Multisensor Fusion and Integration for Intelligent Systems, 1996. IEEE/SICE/RSJ International Conference on.* IEEE, 1996.

[74] J. Kehrbeck and W. Wiesbeck, "Dual channel microwave front-end at 24 GHz for true ground speed measurement," in *IEEE Vehicular Technology Conference (VTC)*, Jun 1994.

[75] D. Stein, M. Spindler, J. Kuper, and M. Lauer, "Rail detection using lidar sensors," *International Journal of Sustainable Development and Planning*, 2016.

[76] H. Iqbal, M. B. Sajjad, M. Mueller, and C. Waldschmidt, "SAR imaging in an automotive scenario," in *IEEE Mediterranean Microwave Symposium (MMS) 2015*, Nov 2015.

[77] B. Stanley, “Localizing ground-penetrating radar,” MIT Lincoln Laboratory, Tech. Rep., 2016.

[78] T. Engelberg and F. Mesch, “Eddy current sensor system for non-contact speed and distance measurement of rail vehicles.” *Computers in Railways VII, WIT Press: Southampton*, 2000.

[79] A. Geistler, “Bordautonome Ortung von Schienenfahrzeugen mit Wirbelstrom-Sensoren,” Ph.D. dissertation, Universität Karlsruhe (TH), Germany, 2007.

[80] M. Spindler, D. Stein, and M. Lauer, “Low power and low cost sensor for train velocity estimation,” in *IEEE International Conference on Intelligent Rail Transportation (ICIRT) ,Birmingham, UK*, 2016.

[81] International Union of Railways, “RMTS - European Rail Management Traffic System,” http://www.uic.org/spip.php?rubrique847.

[82] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[83] “Coradia Lint 41/H für die Veolia Verkehr Regio GmbH,” ALSTOM Transport, 11 2009.

[84] H. Rinne, *Taschenbuch der Statistik*. Harry Deutsch Verlag, 2008.

[85] A. Iliopoulos, C. Enneking, O. G. Crespillo, T. Jost, S. Thoelert, and F. Antreich, “Multicorrelator signal tracking and signal quality monitoring for GNSS with extended Kalman filter,” in *2017 IEEE Aerospace Conference*, März 2017.

[86] A. Grosch, O. G. Crespillo, I. Martini, and C. Günther, “Snapshot residual and Kalman filter based fault detection and exclusion schemes for robust railway navigation,” in *2017 European Navigation Conference (ENC)*, May 2017.

[87] O. G. Crespillo, A. Grosch, J. Skaloud, and M. Meurer, “Innovation vs residual KF based GNSS/INS autonomous integrity monitoring in single fault scenario,” in *ION GNSS+ 2017*, 2017.

[88] “RHINOS - Railway High Integrity Navigation Overlay System will define a GNSS-based system to support the localization of trains respecting the challenging requirements of the railway safety standards.” European Commission: CORDIS, Projects and Results,

Tech. Rep., 2017. [Online]. Available: https://cordis.europa.eu/project/rcn/199612_en.html

[89] O. G. Crespillo, D. A. Medina, A. Grosch, J. Skaloud, and M. Meurer, "Robust tightly coupled GNSS/INS estimation for navigation in challenging scenarios," *GPS World*, vol. 28, no. 8, August 2017.

[90] O. G. Crespillo, D. Medina, J. Skaloud, and M. Meurer, "Tightly coupled GNSS/INS integration based on robust m-estimators," in *ION PLANS 2018*, 2018.

[91] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms," *Journal of Guidance, Control and Dynamics*, vol. 21, no. 1, 1998.

[92] ——, "Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms," *Journal of Guidance, Control and Dynamics*, vol. 21, no. 2, 1998.

[93] K. Britting, *Inertial navigation systems analysis.* Wiley-Interscience, 1971.