A Train Localization Algorithm for Train Protection Systems of the Future

Martin Lauer and Denis Stein

***Abstract*—This paper describes an algorithm that enables a railway vehicle to determine its position in a track network. The system is based solely on onboard sensors such as a velocity sensor and a Global Navigation Satellite System (GNSS) sensor and does not require trackside infrastructure such as axle counters or balises. The paper derives a probabilistic modeling of the localization task and develops a sensor fusion approach to fuse the inputs of the GNSS sensor and the velocity sensor with the digital track map. We describe how we can treat ambiguities and stochastic uncertainty adequately. Moreover, we introduce the concept of virtual balises that can be used to replace balises on the track and evaluate the approach experimentally. This paper focuses on an accurate modeling of sensor and estimation uncertainties, which is relevant for safety critical applications.**

***Index Terms*—Digital track map, eddy current sensor, Global Navigation Satellite System (GNSS), map matching, rail transportation, railway safety, stochastic modeling, train-borne localization, virtual balise.**

I. Introduction

TRADITIONAL train protection systems are based mainly on trackside infrastructure elements like interlockings, signals, axle counters, and balises to ensure the safe operation of railways. This requires large investments. However, many railway lines suffer from low profitability, which makes it difficult to justify high investments into the infrastructure. As a consequence, the safety level of many secondary railway lines remains below standard, i.e., automatic train protection systems are often not used.

One way out of this dilemma is to develop new train protection systems that require less trackside infrastructure. The development of wireless communication devices has paved the way for this idea and allows railway companies to replace balise-based communication between the train and the interlocking by cheaper wireless techniques. The mobile communication standard Global System for Mobile Communications—Railway (GSM-R) [1] is already used in daily operation and offers the required safe wireless communication for railway applications.

However, the determination of the present train position and the decision whether a certain track segment is free or not still require trackside infrastructure like axle counters. An onboard train localization system that is independent of trackside infrastructure would overcome these limitations.

Train protection systems are safety-relevant components since a failure of a train protection system might lead to fatal accidents. Therefore, all components of a train protection system including an onboard localization system have to be safe and reliable. *Safe* means that every error is recognized by the system itself. This establishes an important difference to self-localization approaches used in mobile robotics and intelligent road vehicles [2], [3] since the safety of road traffic is based on the perception of navigable space with onboard sensors rather than on self-localization. Self-localization might assist intelligent road vehicles in driving but never replace obstacle detection with onboard sensors. Therefore, the safety of intelligent road vehicles does not depend on the correctness of self-localization. However, railway vehicles do not use onboard sensors for obstacle detection or free space determination since the range of those sensors is not sufficient to cover the stopping distance of trains. Therefore, they rely fully on the safe operation of all components of the automatic train protection system.

In previous research projects, onboard train localization systems have been investigated considering different aspects and different sensor configurations. The combination of Global Navigation Satellite System (GNSS), inertial sensors, and odometry was examined in [4] and [5]. However, these approaches are not track selective since switches are not modeled. The focus of these approaches was to determine the longitudinal position of a train along a certain track. In [6] and [7] the direction in which switches are passed by the train is estimated using the yaw rate of the train and the global orientation of the rail vehicle so that switches can be treated. Other approaches use cameras to recognize switches [8], [9] or propose the usage of lidars [10] for this purpose. An eddy current sensor was used for switch detection in [11].

The approaches also differ in the way that the information is fused and combined with the track map. The authors of [12] and [13] model the task as a full 2-D- or 3-D-pose estimation task while the authors of [6], [7], [11], and [14] are exploiting the topology of the track map to reduce the estimation task to a 1-D longitudinal train position estimation and a discrete track estimation task. We are following the second idea, which simplifies the estimation procedure. The authors of [11] and [13] derive filter equations for a particle filter.

The results from previous work show that it is possible to build onboard train localization systems. These approaches were focusing on the question of which sensors and which algorithms are necessary to obtain a train localization system.

Manuscript received October 16, 2013; revised May 2, 2014; accepted July 28, 2014. Date of publication September 11, 2014; date of current version March 27, 2015. This work was supported by the European GNSS Agency (GSA) within the FP7 project *GaLoROI*, project No. 277698-2. The Associate Editor for this paper was B. Ning.

The authors are with the Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany (e-mail: martin.lauer@kit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2014.2345498

However, they did not consider the task of onboard train localization from the perspective of safety assessment, i.e., they did not focus on the question of how an onboard train localization system has to be designed so that it can guarantee safe operation in the sense mentioned earlier [15]. Considering the perspective of safety assessment, we have to ask the questions of how we can deal with uncertainties in the sensor signals, which conclusions are allowed from GNSS position measurements [16], and how uncertainties propagate through time [17].

Starting the discussion from this point of view makes it clear that some solutions used in previous work are not suitable. For example, it is very hard to analyze the reliability of a particle filter considering its error probability since it introduces artificial randomness into the localization method. As well, ignoring the uncertainty of the position estimate and relying only on the most probable or most likely position is not sufficient to guarantee safety since those estimates are influenced by sensor noise. Also, unreliable or biased sensors might cause severe problems in the safety case.

Following these ideas, we develop a localization approach in this paper that focuses on the accurate modeling of uncertainties so that the safety of the system can be analyzed easily in a safety case. The approach is generic in the sense that we do not rely on specific sensor devices but on a generic description for these sensors. As long as a sensor meets these assumptions, it can be used together with our localization approach. We build the system on three sources of information, a GNSS receiver, a velocity sensor, and a digital track map. We do not consider a switch detecting sensor since those approaches are not yet reliable enough to be part of a reliable localization system.

The requirements for the sensors and the digital map are described in Section II. Starting from these requirements, we derive in Section III the mathematical basis for the localization algorithm and extend it in Section IV by the concept of virtual balises. Section V analyzes experimental results of the localization approach. This paper is based on a previous conference publication [18] that was restructured and extended by theoretical refinements and additional experiments.

II. Sensor Configuration

We assume in this paper a setup with a digital track map and two onboard sensors, a GNSS receiver, and a velocity sensor as depicted in Fig. 1. The approach is generic so that it works with different GNSS sensors [GPS, Globalnaya navigatsionnaya sputnikovaya sistema (GLONASS), Galileo, and BeiDou] and various velocity sensors (wheel encoders, Doppler radar, and eddy current sensor) as long as they provide unbiased stochastically independent measurements and as long as they are able to provide information on the confidence of their measurements in terms of variance/covariance matrices.

GNSS receivers provide geodetic positions in WGS84 coordinates. To cope with continental drift, we transfer those coordinates to an invariant reference frame. For example, for applications in Europe, we transform WGS84 coordinates to the ETRS89 reference system that is invariant with respect to the continental drift within Europe. All further calculations are done with respect to that reference frame, i.e., we avoid

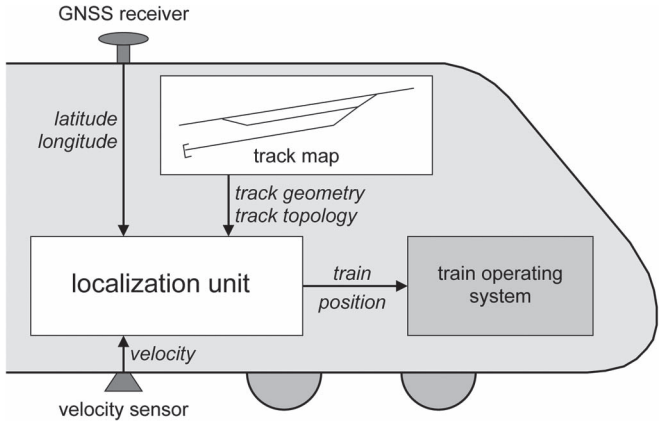

Fig. 1. Four main components of the localization system, the GNSS receiver, the velocity sensor, the digital track map, and the localization unit that fuses the incoming information.

an explicit transformation to Euclidean space. Hence, a GNSS measurement is composed out of a latitude and longitude value (ϕ, λ) and the measurement covariance matrix that describes the uncertainty of the measurement. Hence, we can describe the uncertainty with a bivariate Gaussian in latitude and longitude. The receiver might use satellite-based augmentation systems, if available. This helps to achieve better localization accuracy. However, the approach also works if none of those systems is available.

We assume that the velocity sensor is able to measure the velocity of the train and its driving direction (forward or backward) as a signed velocity measurement v in meters per second. Several velocity sensors are available for railway applications like traditional wheel encoders, Doppler radar [19], and the eddy current sensor [11], [20]. Wheel encoders are cheap and proved themselves in many applications. However, they suffer from slippage, so they do not yield unbiased measurements. Doppler radars overcome this problem; however, they do not work properly in winter on snowy tracks. An eddy current sensor can as well be used. It is drift free and contactless so that it yields unbiased measurements. Furthermore, it is independent of varying weather conditions. For the localization approach, we could use any of these velocity sensors. For the experiments reported in Section V-C, however, we were using an eddy current sensor.

As third source of information for the localization system serves a digital map of the track network. No ubiquitous map format for railway networks exists yet since the requirements depend on the specific application scenario, i.e., whether every sleeper should be modeled or just connecting links between different stations. In previous approaches, maps have been stored in databases [21] or files. The railML file format [22], [23] has been used in the projects *Integrated European Signalling System* (INESS) and in our project *Galileo Localisation for Railway Operation Innovation* (GaLoROI). RailML has been developed by a consortium of researchers and industrial partners so that it has the potential to become a standard in the railway community. Track maps can be derived from construction plans, from aerial images, or even from scratch [14], [24]. The geometry of the tracks can be modeled by straight lines, circular arcs, polygonal models, clothoids, or splines [14].

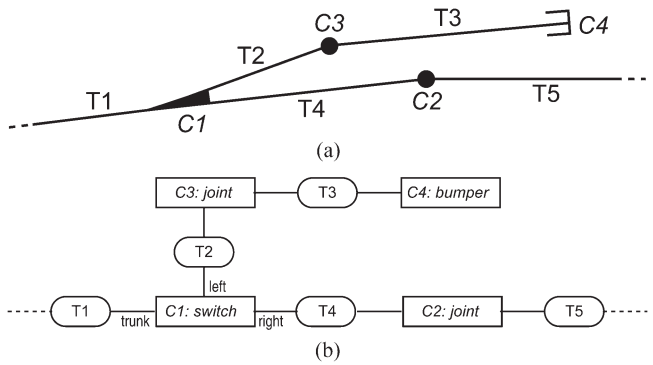

Fig. 2. Track map for a small train station with five track segments T1, . . . , T5, one switch C1, one bumper C4, and two joints C2 and C3. The attributes *left*, *right*, and *trunk* indicate the left branch, right branch, and trunk of the switch. (a) Geographical map. (b) Topological map.

Our map is based on the structure of railML. It contains geographical and geometrical information as well as topological information about the network and is a so-called topographical map. Fig. 2 shows an example of the different aspects. Motivated by [21], we model the topology of a network with a bipartite graph that contains track nodes and connector nodes. Track nodes model straight track segments between two points. By assigning each end point with geodetic coordinates, we obtain the geographical position of the track. Each track has a unique identifier and a track length. Furthermore, we define for each track segment a local 1-D metric coordinate system so that we can refer to each point on the track providing its metric position. We refer to this metric position as the arc length l of a position. Curved tracks are approximated as a sequence of straight track segments.

Track nodes are connected with connector nodes. We distinguish four kind of connectors.

1) *Bumper nodes* that model the end of tracks. Bumper nodes are connected to exactly one track node.
2) *Joint nodes* that model connections between two adjacent track nodes.
3) *Switch nodes* that model typical railway switches that connect three track nodes. The three tracks are attributed the trunk, the left branch, and the right branch. The geodetic reference position of a switch node is the position of the switch blades.
4) *Open ends* that model the boundaries of the map. Open ends are connected to exactly one track node. They indicate that a track continues outside of the given map.

More complex switches like interleaved or slip switches can be modeled as a combination of several switch nodes and short track segments in between. For example, a double slip switch can be represented by four switch nodes (one for each pair of points) and short track segments in between. Hence, we can model all possible railway networks except railway turntables and transfer tables. Since the track map is static, we do not know the position of points at switches, i.e., whether the train passes a turnout facing left or facing right. One exception are spring switches, which guarantee by construction that a train is passing the turnout always facing left or always facing right. In these cases, we can add this information to the map.

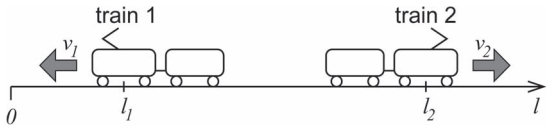

Fig. 3. Illustration of the train orientation o. In this example, train 1 has negative orientation $o_1 = -1$ while train 2 has positive orientation $o_2 = 1$. Both trains are driving forward ($v_1 > 0$ and $v_2 > 0$), which means that the arc length l_1 of the first train decreases over time while the arc length l_2 of the second train increases.

III. Position Filter

A. Stochastic Modeling

Our modeling of the position estimation problem is based on a full Bayesian modeling. Therefore, it is able to cope with sensor and position uncertainties. The description of the train position and movement requires the knowledge of the track on which the train currently is (denoted with t), the position along the track in meters, which we refer to as the arc length of the train position l, the orientation of the train o, and the velocity of the train v (in meters per second). The velocity is a signed variable where the sign indicates whether the train is moving forward (positive velocities) or backward (negative velocities). The train orientation o is either $+1$ or -1. It models whether moving forward means driving in the direction of increasing arc length of the track $(o = +1)$ or in the direction of decreasing arc length $(o = -1)$. Fig. 3 illustrates the train orientation

$$P(t, o, l, v) = P(l, v|t, o) \cdot P(t, o). \tag{1}$$

Now, we can represent $P(t, o)$ as a categorical distribution and store the probabilities for each pair of track and orientation in a table. Conditioned on t and o, we represent $P(l, v|t, o)$ as a bivariate Gaussian. Hence, our knowledge about the present train position can be described by a set of hypotheses. Each hypothesis refers to a certain track and orientation with a certain probability and is assigned with a Gaussian distribution to model the arc length and velocity. The overall distribution turns out to be a mixture distribution [25].

The representation of our target distribution as a mixture of Gaussians enables us to derive a stochastic filter to estimate the distribution incrementally. Such an approach, which is similar to the one presented in [14], is much more efficient than a particle filter approach like in [26], which requires the update of thousands of particles to achieve the same quality of modeling.

B. Incremental Filter

Based on the state space model, we can derive an incremental filter to estimate the state vector. We assume that all measurements of the velocity sensor and of the GNSS receiver are stochastically independent, i.e., that we can model our system as a hidden Markov model [25] with a hybrid state vector. Hence, we can derive an incremental filter that executes alternately prediction and innovation steps. In the prediction step, the variables of the state vector are predicted into the future, while in the innovation steps, new measurements are integrated into the state estimates. Since we are using two unsynchronized

sensors, we have to provide two different innovation steps, one for GNSS measurements and one for velocity measurements.

Let us assume for the moment that we are faced with a network of very long straight tracks that are not connected with each other so that we can ignore the treatment of connectors. For such a simplified system, the system dynamics over time can be described using a Gauss-linear model as

$$\begin{aligned}(t', o') &= (t, o)\\ \begin{pmatrix} l' \\ v' \end{pmatrix} &= \underbrace{\begin{pmatrix} 1 & \delta \cdot o \\ 0 & 1 \end{pmatrix}}_{=A} \cdot \begin{pmatrix} l \\ v \end{pmatrix} + \epsilon_k \end{aligned} \tag{2}$$

where (t, o, l, v) is the present state, (t', o', l', v') is the predicted state δ seconds in future, and ϵ_k models independent unbiased Gaussian noise. We assume that the velocity of the train is almost constant over time. Due to the linearity of (2), we can use the Kalman-filter prediction step with state-transition matrix A to predict the state probability [27].

The innovation step for velocity measurements can be modeled as well with a Gauss-linear model. If we denote with $\tilde{v}$ the velocity sensed by the velocity sensor, we can determine a relationship between measurement and state vector as

$$\tilde{v} = (0, 1) \cdot (l, v)^T + \tilde{\epsilon}_k \tag{3}$$

where $\tilde{\epsilon}_k$ denotes again independent unbiased Gaussian noise. Hence, we can use a standard Kalman-filter innovation step with observation matrix (0, 1) to integrate a velocity measurement into the position estimate.

The integration of GNSS measurements requires two steps. Assume a GNSS measurement $\tilde{g} = (\tilde{\phi}, \tilde{\lambda})$. The posterior distribution can be split into

$$P(t, o, l, v|\tilde{g}) = P(l, v|t, o, \tilde{g}) \cdot P(t, o|\tilde{g}). \tag{4}$$

Let us consider first the term $P(l, v|t, o, \tilde{g})$. Assume that we know the geodetic coordinates of two points on the track, e.g., the two end points of the track, with geodetic coordinates (ϕ_1, λ_1), (ϕ_2, λ_2) and at arc lengths l_1 and l_2. If we approximate the surface of the Earth locally around $\tilde{g}$ by a tangent plane, we can derive a linear dependence between l and $\tilde{g}$ as

$$\tilde{g} = (\phi_1, \lambda_1) + \frac{l - l_1}{l_2 - l_1}(\phi_2 - \phi_1, \lambda_2 - \lambda_1) + \check{\epsilon}_k \tag{5}$$

where $\check{\epsilon}_k$ denotes again independent unbiased Gaussian noise. This allows us again to use a Kalman-filter update to innovate l and v.

$P(t, o|\tilde{g})$ can be transformed using Bayes' formula as

$$\begin{aligned} P(t, o|\tilde{g}) &= \int_{-\infty}^{\infty} P(t, o, l|\tilde{g}) dl \\ &\propto \int_{-\infty}^{\infty} P(\tilde{g}|t, o, l) P(l|t, o) dl \cdot P(t, o). \end{aligned} \tag{6}$$

Hence, the posterior track probability is equal to the prior track probability $P(t, o)$ times an integral term. The integral term can be evaluated analytically to

$$\frac{\sqrt{s_A^2}}{2\pi\sqrt{|\Sigma_{\tilde{g}}|}\delta_l} e^{-\frac{1}{2}\left(r_A - \frac{m_A^2}{s_A^2}\right)} \tag{7}$$

with $s_A^2 = L^2/(\vec{d}\Sigma_{\tilde{g}}^{-1} \cdot \vec{d}^T + (L^2/\sigma_l^2))$, $m_A = L \cdot ((-\vec{d} \cdot \Sigma_{\tilde{g}}^{-1} \cdot \vec{c}^T + L(\mu_l)/(\sigma_l^2)/\vec{d} \cdot \Sigma_{\tilde{g}}^{-1} \cdot \vec{d} + (L^2/\sigma_l^2)))$, and $r_A = \vec{c} \cdot \Sigma_{\tilde{g}}^{-1} \cdot \vec{c}^T + (\mu_l^2/\sigma_l^2)$. $\Sigma_{\tilde{g}}$ denotes the covariance matrix of the GNSS measurement $\tilde{g}$ while L, $\vec{d}$, and $\vec{c}$ are defined as $L = l_2 - l_1$, $\vec{d} = (\phi_2 - \phi_1, \lambda_2 - \lambda_1)$, and $\vec{c} = (\phi_1, \lambda_1) - (l_1/L)\vec{d} - \tilde{g}$.

C. Hop-Along Algorithm

As long as the arc length l estimated with this approach does not exceed the boundaries of the present track t, we can keep t and o unchanged in the prediction and innovation steps. However, as soon as it exceeds the track boundaries, we have to treat the movement from one track to another adequately. Assume for the moment that we are faced with a set of tracks T_1, T_2, T_3, etc., that are connected with joint connectors. An initial arc length l_1 belongs to track T_1. Assume that, after the execution of a prediction or innovation step, the new arc length l_1' exceeds track T_1 at the boundary at which it is connected to T_2. Fig. 4(a) illustrates such a situation where the point A_1 indicates the point with arc length l_1 and B_1 is the new point with arc length l_1'. Then, obviously, the new position cannot be located on track T_1 but on track T_2, and we have to update the track variable t adequately. However, track T_2 has a different geodetic orientation and a different metric coordinate system. Since both tracks share the same corner point, we can transform all arc length coordinates referring to track T_1 into arc length coordinates referring to track T_2 so that l_1 is mapped onto l_2. Eventually, we also have to change the sign of o depending on the orientation of both metric coordinate systems. Plotting the point represented by l_2 into the sketch in Fig. 4(b) yields the point marked with A_2.

In case of a prediction step or an innovation step for velocity measurements, we obtain the new arc length estimate by transforming l_1' into l_2'. However, for an innovation step for a new GNSS measurement, we have to repeat the calculations described in Section III-B to obtain l_2' because we have to consider the geodetic orientation of track T_2 instead of track T_1. As a result, we obtain an estimated point marked with B_2 in Fig. 4(b). If l_2' is in the range of track T_2, we are finished. If not, we have to repeat the process mapping all arc length coordinates into the coordinate system of track T_3 and repeating the calculations for T_3 [cf. points A_3 and B_3 in Fig. 4(c)]. We continue this process until the resulting arc length l_i' is in the range of track T_i or until l_i' exceeds the track on the side of the previous track T_{i-1}. The latter case is illustrated in Fig. 4(d). In this case, the best estimate for the train position is the vertex $V_{i-1,i}$ on the boundary of tracks T_{i-1} and T_i.

As described, the algorithm hops along the track segments until it converges. It can be interpreted as an iterated extended Kalman filter (IEKF) [28] were the linearization of the

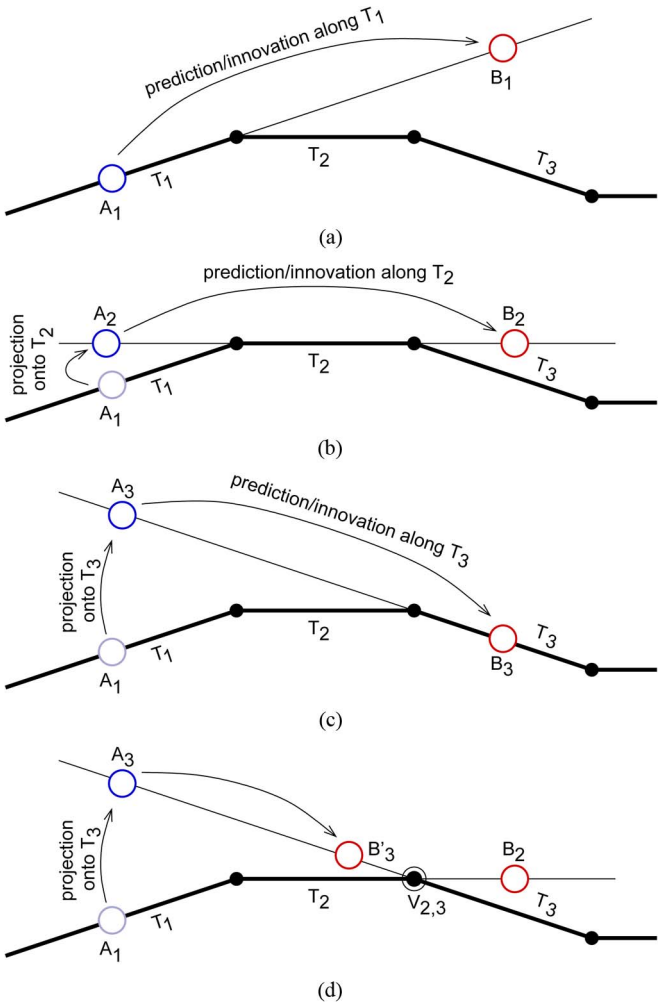

Fig. 4. Example execution of the hop-along algorithm. The thick lines illustrate the tracks, and the thin lines are straight extensions of the tracks. Find a detailed description in the first and second paragraphs of Section III-C. (a) Step I. (b) Step II. (c) Step III. (d) Alternative step III.

estimation task is provided by the straight line segments of the track map.

However, how should we treat switches that are passed in the facing direction if we do not know their position of points? To avoid erroneous decisions, we have to model the uncertainty on which branch the train is on in a stochastic way by splitting the position hypothesis into two hypotheses, one for each branch. To keep the representation consistent, the track probabilities of the two branches have to add up to the track probability of the original hypothesis. A good choice is to set the track probabilities to half of the probability of the original hypothesis.

D. Treatment of Dead Ends and Open Ends

While the hop-along algorithm is adequate for switch and joint connectors, it fails at dead ends since there is no adjacent track segment to hop on. Fig. 5 shows such a situation. However, even if the expected arc length l is beyond a dead end, that does not mean that the position hypothesis is impossible since the hypothesis represents a distribution of arc length, not just a single value. This distribution is plotted in blue in Fig. 5. Obviously, there is some overlap between the support

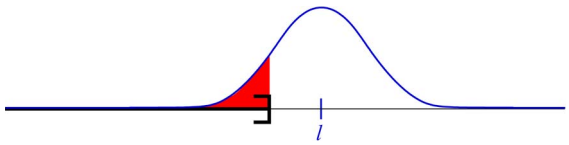

Fig. 5. Position hypothesis beyond a dead end and the probability distribution of this hypothesis (in blue). The red area measures the probability of being in front of the dead end.

of the probability distribution and the physically existing track segment in front of the dead end. The probability of a position in the overlapping area is indicated by the red area.

Our idea is to use this probability to decide whether a hypothesis is impossible or not. Since we are using a Gaussian distribution, the probability will never be zero; however, it will become neglectable at some point. Then, the hypothesis can be removed from the set of position hypotheses. We are using in our implementation a threshold of 10^{-50}. This value is far below the accepted failure rate in railway applications so that our system remains safe in that sense. This approach can be interpreted as a statistical test that checks whether the train position is in front of the dead end. The significance level of the test is provided by the threshold.

While impossible hypotheses behind a bumper can be deleted without risk, this does not apply for hypotheses behind open ends. Since open ends model that a track continues outside of the map, it could be possible that the train is outside of the map that might cause a dangerous situation since we cannot determine its position. If our localization system detects such a situation, it sends a special signal to the train operating system so that a safety-related action can be initiated by the train, e.g., notification of the driver or emergency brake.

E. Treatment of Switches

The position hypotheses are treated at switches in the following way. A position hypothesis that passes a switch in the facing direction is split into a hypothesis on the left branch and a hypothesis on the right branch while a hypothesis that passes a switch in the trailing direction continues on the trunk track. Although this procedure is obvious, it might lead to undesirable artifacts if a train approaches a switch very slowly or if it stops on a switch. In such a case, the position hypothesis might pass the switch several times in both directions due to the randomness in the sensory input. Hence, very many hypotheses are created. It might also happen that physically impossible hypotheses are generated. For example, assume a train that is approaching a switch slowly in the trailing direction on the left branch. The wobbling effect described earlier generates a hypothesis on the right branch every time the position hypothesis passes the points of the switch in the facing direction, although a movement from the left to the right branch is physically impossible.

How can we avoid these kinds of phantom hypotheses? Our idea is to memorize in each hypothesis the tracks that have been passed before. Whenever the position hypothesis tries to pass a switch in the facing direction that has been passed in the trailing direction before, we follow only that branch from which

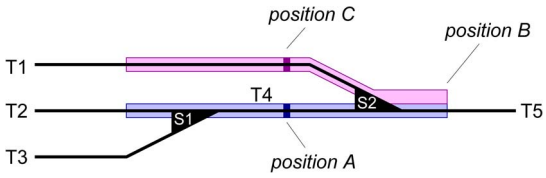

Fig. 6. Example for a reversing train in shunting mode. A detailed description can be found in Section III-E.

the position hypothesis has come before. Hence, the wobbling phenomenon will not cause phantom hypotheses.

However, this procedure fails during the shunting mode whenever a train passes a switch in the trailing mode, reverses, and follows the other branch afterward. An example is illustrated in Fig. 6. We assume that the train enters the illustrated area on track T2 from the left and stops somewhere between switches S1 and S2 on track T4. Our present knowledge on the train position is illustrated by the light blue area that depicts the confidence interval of the position hypothesis and the dark blue bar that indicates the expected position (marked as position A). Track T2 is in the list of memorized previous tracks. Therefore, if the train reverses we would not consider the possibility that the train moves left on switch S1 and enters track T3. Even more, considering the extension of the confidence interval it could also be possible that the train was already on track T5 and moves onto track T1 after reversing.

Therefore, we have to be able to distinguish a wobbling situation and a reversing situation. This can be done by observing whether the sign of the measurements of the velocity sensor has changed, which indicates a reversing movement. In such a case, we clear the whole history of passed tracks. Furthermore, we create new hypotheses for all positions that can be reached by starting at the present expected position (A in Fig. 6), moving half of the confidence interval length into the previous driving direction (reaching position B in Fig. 6), and moving half of the confidence interval length into the new driving direction. In Fig. 6, we would reach in this way the two positions A and C and thus result in two position hypotheses that are illustrated by their confidence intervals in light blue and violet.

F. Merging and Removal of Hypotheses

In a complex railway network, it happens that many hypotheses are generated, which describe a multimodel distribution of the possible train positions. However, the track probability of many of those hypotheses will quickly become very small as soon as the GNSS measurements are far away from the position hypotheses. We decided to remove hypotheses from our hypotheses set as soon as their track probability is significantly small, i.e., below a threshold that we choose again as 10^{-50}. The removal of hypotheses is necessary to avoid an exponential blowup of hypotheses and to keep the approach computationally feasible.

Another phenomenon is that two position hypotheses might become very similar. Then, it is not reasonable to maintain two separate hypotheses, but we perform a merger of both. The similarity of two hypotheses can be measured comparing the track and orientation variables t and o, which have to be identical and using the Mahalanobis distance to determine the similarity of the vectors of arc length and velocity (l, v). Moreover, to cope with the idea of memorized past tracks, we also require that the lists of memorized past tracks are identical.

G. Initialization of the Filter

We assume that we always know at least one position hypothesis. However, this assumption is not met after booting the localization unit. Therefore, we need a special initialization step that is executed as soon as we receive the first GNSS measurement $\tilde{g} = (\tilde{\phi}, \tilde{\lambda})$. The idea is to use a classical map-matching algorithm to determine the initial hypothesis. We assume a uniform prior distribution over (t, o, l). Then

$$P(t, o, l|\tilde{g}) = \frac{P(\tilde{g}|t, o, l) \cdot P(t, o, l)}{P(\tilde{g})} \propto P(\tilde{g}|t, o, l). \quad (8)$$

Let us assume that track t is defined by its front and back vertices with geodetic coordinates (ϕ_1, λ_1), (ϕ_2, λ_2) at arc lengths l_1 and l_2, respectively. Like in Section III-B, we define the variables L, $\vec{d}$, and $\vec{c}$ and denote with $\Sigma_{\tilde{g}}$ the covariance matrix of the GNSS measurement. Then, we can transform $P(\tilde{g}|t, o, l)$ into the following form:

$$P(\tilde{g}|t,o,l) = \underbrace{\frac{\sqrt{s_B^2}}{\sqrt{2\pi \cdot |\Sigma_{\tilde{g}}|}} e^{-\frac{1}{2}\left(r_B - \frac{m_B^2}{s_B^2}\right)}}_{\propto P(t,o)} \cdot \underbrace{\frac{1}{\sqrt{2\pi s_B^2}} e^{-\frac{1}{2}\frac{(l-m_B)^2}{s_B^2}}}_{= P(l|t,o)}$$

with $s_B^2 = L^2/(\vec{d} \cdot \Sigma_{\tilde{g}} \cdot \vec{d}^T)$, $m_B = -L((\vec{d} \cdot \Sigma_{\tilde{g}}^{-1} \cdot \vec{c})/(\vec{d} \cdot \Sigma_{\tilde{g}}^{-1} \cdot \vec{c}^T))$, and $r_B = \vec{c} \cdot \Sigma_{\tilde{g}}^{-1} \cdot \vec{c}^T$.

Note that the first term is independent of l but depends on t. We can interpret it as being proportional to $P(t, o)$. The second term has the form of a Gaussian with mean m_B and standard deviation s_B. Hence, it can be interpreted as $P(l|t, o)$. To initialize our filter, we use the aforementioned formula to generate one hypothesis for each orientation and each track in the vicinity of the GNSS measurement. Afterward, we can normalize the probabilities $P(t, o)$ over all hypotheses.

H. Extension of the Filter to Nonlinear Track Segments

Our derivation of the filter so far models tracks with line segments. This modeling is often sufficient in the railway domain since the radii of most curves are very large so that a piecewise linear approximation of curves is sufficiently good. However, in some cases, e.g., in light rail or tramway systems, heavily curved tracks might occur for which a piecewise linear approximation is not sufficient. We could use arcs, clothoids, or splines instead to model the track geometry. The extension of the position filter to those geometries is straightforward, following the idea of local linearization of the track geometry.

The prediction step and the innovation step for velocity measurements do not change compared to our previous setup. However, the innovation step for GNSS measurements changes. Let us represent the track geometry with two functions $\lambda(l)$ and $\phi(l)$ that model the longitude and latitude of all geometric points on the track depending on their arc length l. We can

easily linearize the track geometry around our present position estimate l_0 by

$$\begin{pmatrix}\lambda(l)\\ \phi(l)\end{pmatrix} \approx \begin{pmatrix}\lambda(l_0)\\ \phi(l_0)\end{pmatrix} + (l - l_0) \cdot \begin{pmatrix}\frac{\partial\lambda}{\partial l}\\ \frac{\partial\phi}{\partial l}\end{pmatrix}. \tag{9}$$

Now, we can apply the arc length of the two end points of the track (l_1 and l_2) to (9) to obtain the virtual end point coordinates (λ_1, ϕ_1) and (λ_2, ϕ_2) of the present linearized track segment that enables us to execute the innovation step derived in (6).

As pointed out in [14], we can repeat the local linearization around our present position estimate until convergence and obtain an IEKF solution. The only difference between our hop-along approach on line segments described in Section III-C and the IEKF on curved track representations is the fact that we use a fixed linearization in the case of line segments while the linearization is calculated on the fly in the IEKF case.

IV. Virtual Balises

One big advantage of train-borne localization is the possibility to replace axle counters and balises by *virtual balises* [4]. A virtual balise is a point specified in the digital track map. Whenever a train passes this point, the train operating system is notified of the virtual balise so that it can react adequately, e.g., adapt the train speed or send a message to the interlocking that a certain block has been entered or left.

How does the concept of virtual balises fit into our stochastic modeling? Since we know the train position only up to a certain probability distribution, we can never be sure that we reached a certain position. However, we can determine when we passed that position with a statistical test that we already introduced in Section III-D to check whether we passed a dead end. Assume a virtual balise at a certain track at arc length l_B and train position estimates, which consist of a single hypothesis. Assume further that the train is moving on the track in the direction of ascending arc length. As soon as the estimated train position l becomes larger than l_B, a routine is started that applies the statistical test in every cycle. As soon as the test returns a significant result, the routine sends a message to the train operating system notifying it that the virtual balise has been passed. Since, in practice, the position uncertainty is small, the delay between really passing the virtual balise and generating the notification is negligible.

The treatment of virtual balises becomes more complicated if our knowledge of the train position is ambiguous, i.e., if we are not perfectly sure about the track that we are currently driving on. Then, it might happen that we are sure that we passed the virtual balise assuming one hypothesis while we would not have passed the virtual balise assuming another hypothesis. The treatment of this case depends on the purpose of the balise. If the balise should be used to notify that the train has left a certain block, we can delay the notification until one of the hypotheses is removed. However, if we want to be notified about a virtual balise when we enter a certain block, we have to report all virtual balises that are potentially passed to obtain a safe behavior of the localization unit.

V. Experimental Results

A. Experimental Setup

The approach described in the sections before was implemented in C++. The software was executed on a 500-MHz Intel processor, and we were able to execute it in real time with ten updates per second. The sensor fusion process required approximately one-third of the computational power while the processing of the eddy current sensor data required two-thirds.

B. Experiments on a Road Network

To obtain first insights into the performance of our localization system, we created a preliminary testbed using a car that is equipped with an inertial measurement unit that provides velocity estimates and a Septentrio AsteRx3 GNSS receiver that is capable of receiving GPS, GLONASS, and Galileo satellite signals and that uses the European Geostationary Navigation Overlay Service (EGNOS). Hence, the only technical difference between our preliminary testbed and the final setup on the train was that we were using the inertial measurement unit instead of the eddy current sensor as velocity sensor.

Since we cannot use our experimental car on a railway network, we were using an urban road network instead and created an artificial railway network for it. The network contains several level junctions, triangular junctions, loops, dead ends, multitrack sections, and a short tunnel so that we could test a large variety of situations. The map was generated from aerial images. Compared to a real railway scenario, there are some differences like smaller radii, smaller distances between parallel tracks, and larger lateral deviation of the vehicle from the track. However, the results can be used to get an assessment of the system performance. We performed several hours of test driving on this network and observed several phenomena that are discussed in the subsequent paragraphs.

a) Multipath effects: Multipath effects disturb the GNSS measurement significantly. Multipath effects occur if the satellite signals are received on a path that is different from the straight line between the satellite and the GNSS receiver, e.g., if the signal is reflected at facades of houses or at leaves of trees. The longer travel distance of the signals on the indirect path causes biased position estimates within the GNSS receiver. This problem is severe especially since it is hard for the GNSS receiver to detect these errors. We observed that those effects occur mainly if the vehicle was standing still or moving at slow speed. Deviations from the true position of more than 10 m occurred which might lead to wrong position estimates in the localization approach. Since the GNSS receiver is unable to detect multipath effects, it underestimates the variance of the position in these situations so that the localization approach also overestimates the position accuracy. To overcome this problem, we changed the update scheme of our localization approach. While we were using all incoming GNSS measurements in our initial approach, we were replacing the update scheme in such a way that the update rate is changed depending on velocity, i.e., the faster the vehicle moves the more often GNSS measurements are used to update the train position. If the vehicle stops, GNSS measurements are suppressed completely.

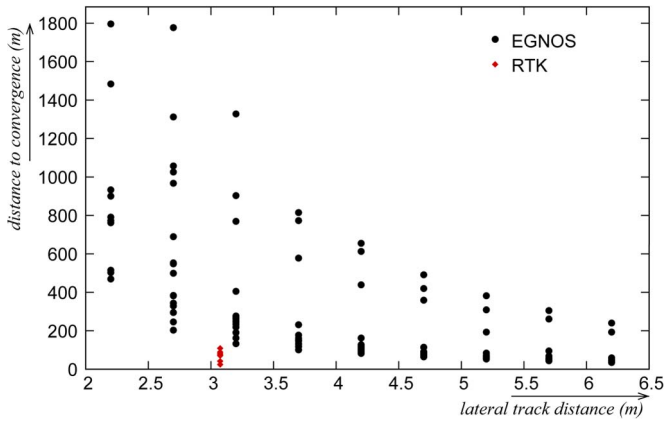

Fig. 7. Scatterplot of the results of the multitrack segment experiments. The horizontal axis describes the lateral distance of neighboring tracks while the vertical axis denotes the distance to convergence (each in meters). Every point marks one run. Black points refer to experiments in which only EGNOS was available, and red diamonds refer to experiments in which RTK was used.

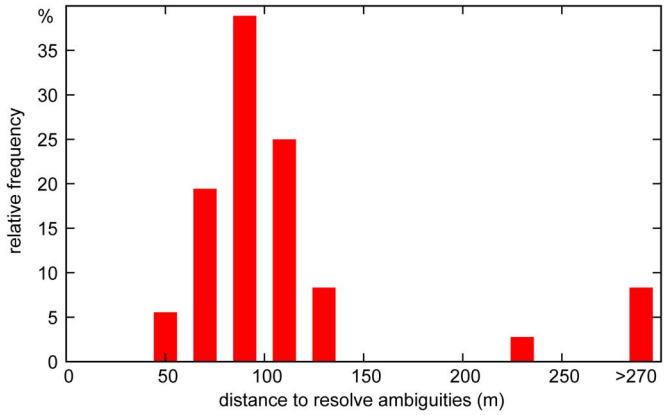

Fig. 8. Histogram of the distance between the points of a facing switch and the position when the localization system has resolved the track ambiguities for parallel tracks of 5-m lateral distance. The statistics was created from 39 trials.

This reduces the influence of disturbed GNSS measurements tremendously.

b) Track selectivity at junctions: The algorithm works as expected at junctions. After a few meters, the approach focuses on the true hypothesis. Typically, ambiguities are resolved completely 10–50 m behind the junction.

c) Track selectivity on multitrack segments: A crucial point of our studies was the question of whether the approach is able to resolve the ambiguity between different parallel tracks. We designed a special experiment in which we were driving on a straight road using a map with three parallel tracks. The middle track was the one that we were driving on. We observed after which distance the hypotheses on the wrong tracks were removed. We repeated the experiment with different maps where we varied the lateral distance between the three parallel tracks between 2.2 and 6.2 m.

For lateral distances of 2.7 m or more, the localization algorithm always converged to the true track, while for a lateral distance of only 2.2 m, the algorithm failed to converge within the maximal track length of 1800 m in 6 runs out of 16. However, 2.2 m is much below the minimal lateral distance of parallel tracks on real railway lines of 3.5 m. Fig. 7 shows that, for reasonable lateral distances, the algorithm converges after, at most, a few hundred meters.

d) GNSS with RTK: In one run of our experiments, we were using a GNSS receiver with real-time kinematics (RTK) instead of the aforementioned system based on EGNOS. The quality of the GNSS measurements was much better, and the standard deviations of the RTK measurements were only between 1 and 1.5 m compared to 1.5 and 5 m for the EGNOS-based measurements. This reduced the estimated longitudinal error in position by 24%. However, the major advantage of an RTK setup for the train localization system was that the system was able to decide earlier on which of several tracks the vehicle actually was. The red diamonds in Fig. 7 illustrate the results.

However, one important disadvantage of RTK is that it relies on base stations along the railway line, which are sending RTK patches to the GNSS receiver via wireless communication. In a practical setup, for a larger railway network, this would require additional effort to maintain this infrastructure. Furthermore, RTK is only available if at least six satellites can be received by the GNSS receiver that is not always possible.

C. Experiments on a Railway Vehicle

In a second set of experiments, we equipped a railway vehicle with our localization system. The vehicle was equipped with two identical units of the system to have the opportunity to cross-check the results of either units. Each unit was composed out of a GNSS receiver, an eddy current sensor, and two processors on which the two programs for the evaluation of the eddy current signals and the localization approach were running. With this vehicle, we were driving on the track network of a small station with three parallel tracks (lateral distance of 5 m), several dead end tracks, one double slip switch, and several standard switches. The velocity of the vehicle was always below 30 km/h, i.e., we were performing shunting movements throughout the whole experiments. The goal of the experiments was to evaluate the quality and reliability of our localization approach with special focus on the question of whether the localization system was able to determine the present track segment correctly. In total, we recorded data of approximately 3 h.

To evaluate whether the localization system is able to determine the correct present track segment, we were driving many times on the parallel tracks of the station and checked whether the localization system was able to determine the correct track and how long it took to resolve the ambiguities. The results are depicted in Fig. 8. Note that, in neither cases, the wrong track was determined, but the system was undecided in three cases since the track length of 270 m was not sufficient to resolve the track ambiguities. In most cases, the correct track was already determined after, at most, 150 m.

As a measure of longitudinal accuracy, we evaluated the estimated standard deviation with respect to the arc length estimation. The results over the whole recorded data set are illustrated in Fig. 9. The standard deviation was below 1 m in 49% of all cases, below 2 m in 84%, and below 3 m in 97%.

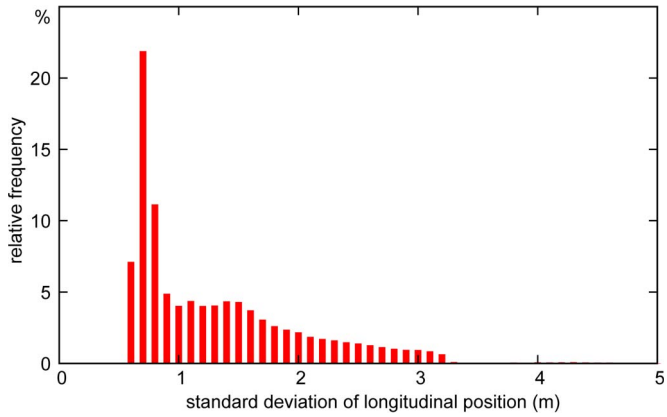

Fig. 9. Histogram of the standard deviation of the longitudinal train position.

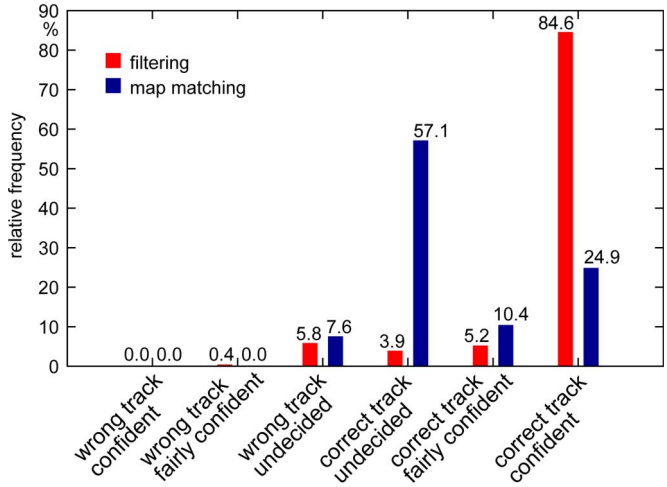

Fig. 10. Comparison of the (blue) map-matching approach and the (red) filtering approach. The histogram shows the frequency of different cases during our experiments. Further explanations can be found in Section V-C.

We compared our localization approach with a purely GNSS-based map-matching approach. The map-matching approach calculates for each GNSS measurement the closest track segments in the vicinity of the GNSS position and determines track probabilities depending on the lateral distance between the track segment and the GNSS position based on (8). We analyzed the decisions of the map-matching approach and the hypothesis-based filtering approach introduced in this paper. Fig. 10 shows the results. We were grouping the situations into six categories as follows.

1) *Correct track*, *very confident*: The localization approach found the correct track, and it was confident about its decision, i.e., the approach was sure that no other track could be the correct one.
2) *Correct track*, *fairly confident*: The localization approach found the correct track, but it was not perfectly sure. It assessed the probability that another track could be the correct one to less than 0.1.
3) *Correct track*, *undecided*: The approach was not sure, but the correct track was assessed as the most probable one.
4) *Wrong track*, *undecided*: The approach was not sure, and a wrong track was assessed as the most probable one.
5) *Wrong track*, *fairly confident*: The approach favored a wrong track, but it assessed the probability of making a wrong decision to a value between 10^{-10} and 0.1.
6) *Wrong track*, *confident*: The approach found a wrong track and was confident about its decision. Situations within this category could lead to fatal errors in safety-relevant applications.

The filtering approach clearly outperformed the map-matching approach. While none of both approaches made fatal errors, the filtering approach was much more often confident about the correct track while the map-matching approach was undecided most of the time, i.e., a track-selective localization was not available with the map-matching approach.

VI. Conclusion

The safe operation of railways depends on the reliability of automatic train protection systems. Future train protection systems will be based on onboard train localization systems. Therefore, train localization systems have to be safe themselves.

In this paper, we have shown that a safe onboard train localization system is possible if we consider all uncertainties of the sensors in an adequate way and if we use a stochastic representation of our knowledge. We were using a Bayesian framework for this purpose, a representation of the train position with mixtures of Gaussians, and a filtering approach to integrate incoming measurements from a GNSS receiver and a velocity sensor. Note that the approach does not rely on a specific sensory device but is generic and can be used with different GNSS receivers and different velocity sensors like wheel encoders, Doppler radars, or eddy current sensors. The approach works with all variants of GNSS measurements like standard GNSS, differential GPS, RTK, or satellite-based augmentation systems. The only difference is the achievable accuracy.

The modeling of uncertainties has been an important point to guarantee the safe operation of the train localization system. Uncertainties are considered by assigning variances/covariances to all measurements and by propagating variances/covariances over time considering the uncertainties in system behavior. We showed how these uncertainty measures can be used to determine confidence levels for position estimates that can themselves be used to implement virtual balises as replacement for infrastructure-based balises.

By experiments we have shown that the localization approach works successfully on a real railway vehicle. We have seen that the most difficult task for the localization approach was to decide on which of several parallel tracks the train is located. The localization approach cannot decide this question immediately but delays its decision until enough GNSS measurement have been collected to be sure. This strategy guarantees safe operation and avoids wrong decisions.

While this paper has introduced a basic algorithm for safety-related train localization based on a GNSS receiver and a velocity filter, our next step is to integrate further sensors into the localization framework like a switch detector device and a landmark detector device. Again, we will consider the

uncertainty of such a sensor and extend the Bayesian framework to cope with it. Additional sensors will not change the overall structure of the localization approach but help the system to reduce uncertainty.

REFERENCES

[1] P. Stanley, *ETCS for Engineers*. Frankfurt, Germany: Eurailpress, 2011.

[2] M. Obst, C. Adam, G. Wanielik, and R. Schubert, "Probabilistic multipath mitigation for GNSS-based vehicle localization in urban areas," in *Proc. 24th Int. Tech. Meet. Satell. Div. Inst. Navigat.*, 2012, pp. 1454–1461.

[3] H. Lategahn and C. Stiller, "Vision-only localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1246–1257, Jun. 2014.

[4] A. Albanese, G. Labbiento, L. Marradi, and G. Venturi, "The RUNE project: The integrity performances of GNSS-based railway user navigation equipment," in *Proc. ASME/IEEE Joint Railway Conf.*, 2005, pp. 211–218.

[5] Z. Jiang, "Digital route model aided integrated satellite navigation and low-cost inertial sensors for high-performance positioning on the railways," Ph.D. dissertation, University College London, London, U.K., 2010.

[6] K. Lüddecke and C. Rahmig, "Evaluating multiple GNSS data in a multi-hypothesis based map-matching algorithm for train positioning," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 1035–1040.

[7] K. Gerlach and C. Rahmig, "Multi-hypothesis based map-matching algorithm for precise train positioning," in *Proc. 12th Int. Conf. Inf. Fusion*, 2009, pp. 1363–1369.

[8] J. Wohlfeil, "Vision based rail track and switch recognition for self-localization of trains in a rail network," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 1023–1028.

[9] R. Ross, "Track and turnout detection in video-signals using probabilistic spline curves," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, 2012, pp. 294–299.

[10] T. Albrecht, K. Lüddecke, and J. Zimmermann, "A precise and reliable train positioning system and its use for automation of train operation," in *Proc. IEEE Int. Conf. Intell. Rail Transp.*, 2013, pp. 134–139.

[11] S. Hensel, C. Hasberg, and C. Stiller, "Probabilistic rail vehicle localization with eddy current sensors in topological maps," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1525–1536, Dec. 2011.

[12] F. Böahringer and A. Geistler, "Comparison between different fusion approaches for train-borne location systems," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst.*, 2006, pp. 267–272.

[13] O. Heirich, P. Robertson, A. C. García, T. Strang, and A. Lehner, "Probabilistic localization method for trains," in *Proc. IEEE Intell. Veh. Symp.*, 2012, pp. 482–487.

[14] C. Hasberg, S. Hensel, and C. Stiller, "Simultaneous localization and mapping for path-constrained motion," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 541–552, Jun. 2012.

[15] H. Manz and E. Schnieder, "Implementation of the normative safety case structure for satellite based railway applications," in *Proc. IEEE Int. Conf. Intell. Rail Transp.*, 2013, pp. 203–208.

[16] D. Lu, F. G. Toro, and E. Schnieder, "RAMS evaluation of GNSS for railway localisation," in *Proc. IEEE Int. Conf. Intell. Rail Transp.*, 2013, pp. 209–214.

[17] J. Beugin and J. Marais, "Simulation-based evaluation of dependability and safety properties of satellite technologies for railway localization," *Transp. Res. C, Emerging Technol.*, vol. 22, pp. 42–57, Jun. 2012.

[18] M. Lauer and D. Stein, "Algorithms and concepts for an onboard train localization system for safety-relevant services," in *Proc. IEEE Int. Conf. Intell. Rail Transp.*, 2013, pp. 65–70.

[19] R. Schubert, P. Heide, and V. Máhgori, "Microwave Doppler sensors measuring vehicle speed and travelled distance: Realistic system tests in railroad environment," in *Proc. MIOP Conf.*, 1995, pp. 365–369.

[20] T. Engelberg, "Design of a correlation system for speed measurement of rail vehicles," *Measurement*, vol. 29, no. 2, pp. 157–164, Mar. 2001.

[21] F. Böhringer and A. Geistler, "Location in railway traffic: Generation of a digital map for secure applications," in *Proc. Comput. Railways X*, 2006, pp. 459–468.

[22] A. Nash, D. Huerlimann, J. Schuette, and V. P. Krauss, "RailML—A standard data interface for railroad applications," in *Proc. Comput. Railways IX*, 2004, pp. 233–240.

[23] K. Gerlach and M. Meyer zu Hörste, "A precise digital map for GALILEO-based train positioning systems," in *Proc. 9th Int. Conf. Intell. Transp. Syst. Telecommun.*, 2009, pp. 343–347.

[24] O. Heirich, P. Robertson, and T. Strang, "RailSLAM—Localization of rail vehicles and mapping of geometric railway tracks," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 5192–5199.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer-Verlag, 2006.

[26] O. Heirich, A. Lehner, P. Robertson, and T. Strang, "Measurement and analysis of train motion and railway track characteristics with inertial sensors," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst.*, 2011, pp. 1995–2000.

[27] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA, USA: MIT Press, 2006.

[28] W. F. Denham and S. Pines, "Sequential estimation when measurement function nonlinearity is comparable to measurement error," *J. Amer. Inst. Aeronaut. Astronaut.*, vol. 4, no. 6, pp. 1071–1076, Jun. 1966.

Martin Lauer received the diploma degree in computer science from Karlsruhe University, Karlsruhe, Germany, and the Ph.D. degree in computer science from Osnabrück University, Osnabrück, Germany, in 2004.

He was a Postdoctoral Researcher with Osnabrück University. He currently is a Research Group Leader with Karlsruhe Institute of Technology, Karlsruhe, Germany, in the fields of machine vision, autonomous mobile robotics, and machine learning.

Denis Stein received the diploma degree in computer science from Technische Universität Dresden, Dresden, Germany, in 2008.

He is a Researcher with the Karlsruhe Institute of Technology, Karlsruhe, Germany. His research interests include train-borne localization, lidar sensors, and signal interpretation.