Robust Particle Filter for Magnetic field-based Train Localization

Benjamin Siebler, Oliver Heirich, Andreas Lehner, Stephan Sand, *German Aerospace Center (DLR)*
Uwe D. Hanebeck, *Karlsruhe Institute of Technology (KIT)*

BIOGRAPHY

Benjamin Siebler received his M.Sc. in electrical engineering and information technology from the KIT in 2014 and is currently pursuing his PhD at the Intelligent Sensor-Actuator-Systems Laboratory of the KIT. Since 2014 he is a researcher at the DLR with focus on multisensor localization with inertial sensors and magnetic field distortions.

Oliver Heirich received his Dipl.-Ing. in electrical engineering from University Ulm in 2008 and his Ph.D. in 2020 from TU Munich. Since 2010 he is a researcher at the German Aerospace Center (DLR) with focus on multisensor train localization with digital maps, GNSS, inertial sensors and magnetic signatures.

Andreas Lehner received his Ph.D. from University Erlangen-Nuremberg in 2008. Since 2002 he is researcher at DLR focusing on GNSS multipath characterization, robust multi-sensor localization and decentralized wireless communication for land-mobile applications. Andreas has authored and co-authored more than 100 publications and co-founded Intelligence on Wheels GmbH in 2012.

Stephan Sand received his Ph.D. from ETH Zurich, Switzerland in 2010. Since 2002 Stephan has researched wireless communications andmulti-sensor navigation at DLR. Since 2014 he is leading the Vehicular Applications Group researching robust navigation and wireless communications technologies for road users and railways. Stephan has authored and co-authored more than 100 publications.

Uwe D. Hanebeck is full professor at the KIT and director of the ISAS Laboratory. He obtained his Ph.D. degree and his habilitation degree in Electrical Engineering from the Technical University in Munich, in 1997 and 2003, respectively. His research interests are information fusion, nonlinear state estimation, stochastic modeling, system identification, and control. He is author and coauthor of more than 500 publications and an IEEE Fellow.

ABSTRACT

For the automation of railway systems, a crucial requirement is a reliable localization system that is able to localize all trains in the track network. While in most parts of the track network, GNSS signals are available and provide a satisfying navigation solution, there are also parts where shadowing and multipath renders GNSS signals unavailable. In such scenarios, magnetic field-based localization can complement GNSS. Magnetic field localization is based on the observation that ferromagnetic material in the vicinity of a railway track introduces distortions in the Earth magnetic field. These distortions are persistent over time and therefore can be used for localization when stored in a map. In our a prior work we showed that particle filters can be used to perform magnetic localization. In this paper, we extend the particle filter with fault detection and exclusion capabilities to make it more robust and accurate in the presence of sensor faults and measurement errors. The advantage of the proposed particle filter w.r.t. accuracy and robustness is shown in an evaluation with measurement data.

I. INTRODUCTION

Increasing urbanization and climate change require transportation systems that have a small carbon footprint and a large transport capacity. Both requirements can be addressed with a highly automated railway system that uses trains powered by renewable energy sources. For the automation of a railway system with a large capacity, one crucial requirement is a reliable localization system that is able to localize all trains in the track network. As the braking distances of trains are beyond the measurement range of most sensors, the train positions in a large area have to be determined and distributed over a communication channel in real time to enable safe operation. Based on this real-time information, it is then possible to automate train control and reduce the headways from the absolute to the relative braking distance. Headway reduction increases the capacity of existing track networks without building new tracks. This is particularly important in urban areas, where space is scarce and expensive. The challenge in the development of an appropriate localization system is to provide reliable position information in the whole track network independent of the environment. While in most parts of the track network global navigation satellite system (GNSS) signals are available and provide a satisfying navigation solution, there are also parts where shadowing and multipath renders GNSS signals unavailable or inaccurate, e.g., in tunnels and urban canyons. In our research, we therefore propose the use of magnetic

Table 1: Overview on the considered models. For each model the sensor axes without measurement errors are indicated by a "-"and erroneous axes by a "x".

	Sensor Axis		
Model	1	2	3
$M^{(1)}$	-	-	-
$M^{(2)}$	-	-	x
$M^{(3)}$	-	x	-
$M^{(4)}$	x	-	-
$M^{(5)}$	x	x	-
$M^{(6)}$	-	x	x
$M^{(7)}$	x	-	x
$M^{(8)}$	x	x	x

field-based localization to complement GNSS in difficult environments. Magnetic field-based localization uses distortions in the Earth magnetic field caused by ferromagnetic material next to the railway track. When the ferromagnetic material is at a static position also the distortions are static and a magnetic map can be used for localization. In our prior work, we proposed multiple approaches for magnetic localization of trains and showed their feasibility based on measurements collected with different types of trains (Siebler et al., 2018, 2020). Furthermore, we already addressed practical issues such as magnetometer calibration (Siebler et al., 2021), which enables the use of the same magnetic map for different trains and magnetometers. Until now, research was mainly concerned with the development of position estimation methods when the magnetometer measurements are affected only by sensor noise or small noise-like errors. Unfortunately, in practice this assumption is often violated and the measurements contain large correlated errors. This type of error is caused by different events like other trains in the vicinity of the magnetometer, changes in the magnetic landscape due construction or the use of magnetic brakes. First attempts to handle such errors can be found in (Siebler et al., 2020), where multiple noise models and a particle filter were used to reduce the effect of measurement errors on the position estimation. In this paper, we now propose an alternative approach. Instead of using different noise models, a fault detection and exclusion (FDE) method is developed that is tightly integrated into a particle filter that estimates the position.

In (Siebler et al., 2020), for each noise model weights and a model probability were calculated. With the probabilities and the weights, a combined weight over all models was calculated to represent the posterior density estimated by the particle filer. This is approach is similar to (Menke and Maybeck, 1995; Hashimoto et al., 2001; Zhang and Li, 1998) where the results of multiple Kalman filters are combined based on estimated model probabilities. In this paper, also different models are considered but the different models assume that the measurements of either none, one, two or all three axes of the magnetometer are erroneous. Erroneous here means that either the sensor itself is faulty or that the measurements are affected by distortions that are not consistent with the magnetic map. Based on the model with the highest probability then none, one, two or three of the sensor axes are excluded from the weight update. The exclusion avoids updating the filter with erroneous measurements that carry no information about the position and might potentially lead to a divergence of the filter.

II. ROBUST PARTICLE FILTER

1. Model Probabilities and Fault Detection

Magnetic train localization uses distortions of the Earth magnetic field that are fixed to a certain position. With a map of the distorted magnetic field it is possible to estimate the position of a train by comparing a series of magnetometer measurements against it. In our prior work (Siebler et al., 2018) we showed that a particle filer can be used to sequentially compare the data to the map and to track the train position. Furthermore, in (Siebler et al., 2020) we showed that in addition to the along-track position also the correct switch way can be identified and that multiple noise models can be used to account for erroneous measurements. The weights of the filter are obtained by combining the different noise models. In contrast to this, in this paper we propose the use of a fault detection scheme that avoids using any information from erroneous measurements instead of combing information from multiple models. Erroneous measurement here means that the difference between the measurement and the magnetic field in the map cannot be explained by sensor noise. The difference between measurement and map can be caused by actual sensor faults or by changes in the environment. One example when such a change occurs is when another train is passing by the magnetometer or when constructions are performed on the track. In the proposed fault detection, the different models account for the case that either one, two or all three magnetometer axes are faulty. For a three axes magnetometer therefore in total the eight different models in Table 1 are considered to cover all fault combinations.

The output of the proposed particle filter is a Dirac mixture approximation of the posterior probability density function (pdf) $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$ of the current state $\mathbf{x}_k$ at time step k conditioned on the measurement history $\mathbf{Z}_{1:k}$

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k}) = \sum_{i=1}^{N_p} w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k), \tag{1}$$

where the weights $w_k^{(i)}$ are calculated only from non-erroneous measurements. To achieve this, in each time step the probabilities of all eight models are calculated. Based on these probabilities the most likely model is selected and the weights are updated only with the sensors not detected as faulty. Therefore, for the fault detection the model probabilities have to be updated in each time step. From a formal point of view the model probabilities $\Pr(M_k^{(j)}|\mathbf{Z}_{1:k})$ are the probability that the j-th model $M_k^{(j)}$ is correct given the measurements up to time step k. With Bayes' theorem and the law of total probability the model probabilities $\Pr(M_k^{(j)}|\mathbf{Z}_{1:k})$ can be calculated based on the probabilities of the previous time step and the likelihood of the current measurement

$$\begin{aligned}
\Pr(M_k^{(j)}|\mathbf{Z}_{1:k}) &= \frac{p(\mathbf{z}_k|M_k^{(j)}, \mathbf{Z}_{1:k-1}) \Pr(M_k^{(j)}|\mathbf{Z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{1:k-1})} \\
&\propto \underbrace{p(\mathbf{z}_k|M_k^{(j)}, \mathbf{Z}_{1:k-1})}_{\text{Likelihood } \lambda_k^{(j)}} \Pr(M_k^{(j)}|\mathbf{Z}_{1:k-1}) \\
&\propto \lambda_k^{(j)} \sum_{i=1}^{8} \Pr(M_k^{(j)}|M_{k-1}^{(i)}) \Pr(M_{k-1}^{(i)}|\mathbf{Z}_{1:k-1}).
\end{aligned} \tag{2}$$

The denominator in the first line of (2) is only a proportionality constant that normalizes the sum over all models to one. This normalization can also be ensured without $p(\mathbf{z}_k|\mathbf{Z}_{1:k-1})$ by

$$\Pr(M_k^{(j)}|\mathbf{Z}_{1:k}) = \frac{\lambda_k^{(j)} \sum\limits_{i=1}^{8} \Pr(M_k^{(j)}|M_{k-1}^{(i)}) \Pr(M_{k-1}^{(i)}|\mathbf{Z}_{1:k-1})}{\sum\limits_{j=1}^{8} \lambda_k^{(j)} \sum\limits_{i=1}^{8} \Pr(M_k^{(j)}|M_{k-1}^{(i)}) \Pr(M_{k-1}^{(i)}|\mathbf{Z}_{1:k-1})}. \tag{3}$$

From now on it is assumed that the model transition probabilities are time invariant and we can write $\Pr(M_k^{(j)}|M_{k-1}^{(i)}) = p^{(ji)}$ which simplifies (3) to

$$\Pr(M_k^{(j)}|\mathbf{Z}_{1:k}) = \frac{\lambda_k^{(j)} \sum\limits_{i=1}^{8} p^{(ji)} \Pr(M_{k-1}^{(i)}|\mathbf{Z}_{1:k-1})}{\sum\limits_{j=1}^{8} \lambda_k^{(j)} \sum\limits_{i=1}^{8} p^{(ji)} \Pr(M_{k-1}^{(i)}|\mathbf{Z}_{1:k-1})}. \tag{4}$$

Before the model probabilities can be calculated, we have to take a closer look on the likelihood $\lambda_k^{(j)}$. The likelihood in its form stated in the second line of (2) cannot be evaluated directly. Instead, one has to introduce the state $\mathbf{x}$ into the equation by marginalization

$$\begin{aligned}
\lambda_k^{(j)} = p(\mathbf{z}_k|M_k^{(j)}, \mathbf{Z}_{1:k-1}) &= \int_{-\infty}^{\infty} p(\mathbf{z}_k, \mathbf{x}_k|M_k^{(j)}, \mathbf{Z}_{1:k-1}) d\mathbf{x}_k \\
&= \int_{-\infty}^{\infty} p(\mathbf{z}_k|\mathbf{x}_k, M_k^{(j)}) p(\mathbf{x}_k|M_k^{(j)}, \mathbf{Z}_{1:k-1}) d\mathbf{x}_k.
\end{aligned} \tag{5}$$

In (5), the one step prediction pdfs $p(\mathbf{x}_k|M_k^{(j)}, \mathbf{Z}_{1:k-1})$ for all models are required. These pdfs could be obtained by running a bank of particle filters. Unfortunately, the number of required filters will increase with each time step because not only the

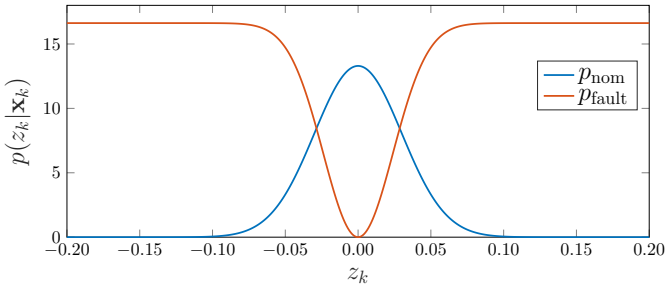

Figure 1: Likelihood for the nominal and the fault case. For both likelihoods here the parameter $\mathrm{m}^{(l)}(s_k)$ is set to zero.

current model is of importance but also the sequence of models before. Running many filters in parallel not only leads to a high computational complexity but in any case, some of this filters are prone to divergence. The divergence comes either from the fact that a model labels some sensor measurements as erroneous while in fact no errors are present or the other way around. The idea in this paper is that instead of running parallel filters, only one filter that uses the correct model is used, where the correct model is determined by the model probabilities. Hence, all models use the same predictive pdf and we drop the dependency on the model and write $p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$. Because a particle filter is used, the predictive pdf is a Dirac mixture density for which the closed-form solution of the likelihood in (5) is given by

$$\begin{aligned}\lambda_k^{(j)} &= \int_{-\infty}^{\infty} p(\mathbf{z}_k|\mathbf{x}_k, M_k^{(j)}) p(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) d\mathbf{x}_k \\ &= \sum_{i=1}^{N_p} w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)}, M_k^{(j)}).\end{aligned} \tag{6}$$

2. Model Likelihoods

In (6), the likelihood $p(\mathbf{z}_k|\mathbf{x}_k^{(i)}, M_k^{(j)})$ plays a crucial role since this is the part where the different model assumptions are introduced. In order to derive these likelihoods, it is useful to first introduce the measurement model of the magnetometer in the fault-free case. If no faults are present, the model for the magnetometer measurement $\mathbf{z}_k$ at a specific along-track train position s_k is

$$\mathbf{z}_k = \mathbf{m}(s_k) + \mathbf{n}_k, \tag{7}$$

where $\mathbf{m}(s_k)$ is the magnetic value in the map at the current train position and $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$ is white Gaussian noise. Note, the along-track position is the one-dimensional position of the train on the track and is part of the estimated state vector $\mathbf{x}_k$ that will be defined in detail in the next section. With (7) the likelihood for a fault free, or nominal, measurement is defined by $p(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; \mathbf{m}(s_k), \sigma_n^2 \mathbf{I})$. The magnetometer measurement $\mathbf{z}_k$ is a vector and by assuming independence between the different sensor axes the likelihood can be written as the product of the likelihoods for the individual axes

$$\begin{aligned}p(\mathbf{z}_k|\mathbf{x}_k) &= \prod_{l=1}^{3} \mathcal{N}(z_k^{(l)}; \mathrm{m}^{(l)}(s_k), \sigma_n^2) \\ &= \prod_{l=1}^{3} p_{\text{nom}}^{(l)}(z_k^{(l)}|\mathbf{x}_k),\end{aligned} \tag{8}$$

with $z_k^{(l)}$ and $\mathrm{m}^{(l)}$ being the l-th element of $\mathbf{z}_k$ and $\mathbf{m}(s_k)$. Hence, for model $M^{(1)}$, where all sensor axes are assumed to be fault-free, the likelihood $p(\mathbf{z}_k|\mathbf{x}_k, M_k^{(1)})$ in (6) is equal to (8). In contrast to the fault free-case, where the likelihood is defined by the map and the sensor noise, the likelihood for the faulty case can be chosen more freely. Loosely speaking, the likelihood for the faulty case should behave in an opposite way than the nominal likelihood. In the nominal case the likelihood is high when the measurement is close to the expected value from the map, hence for the faulty case the likelihood should decrease when the measurement is close to the map value. With this is mind many different likelihoods are possible. On possible function that

fulfills the desired property is given by

$$p_{\text{fault}}^{(l)}(z_k^{(l)}|\mathbf{x}_k) = c_{\text{fault}}\left[1 - \exp\left(-\frac{1}{2\sigma_{\text{fault}}^2}(z_k^{(l)} - \mathrm{m}^{(l)}(s_k))^2\right)\right]. \tag{9}$$

The value of c_{fault} defines the maximum value the likelihood can take and σ_{fault} how fast the likelihood converges towards this value. If c_{fault} is set to $c_{\text{nom}} = 1/\sqrt{2\pi\sigma_n^2}$, the maximum value of the nominal likelihood, both likelihoods are balanced in the sense that their maximum value is equal. In this paper $c_{\text{fault}} = 0.8c_{\text{nom}}$ and $\sigma_{\text{fault}} = 0.8\sigma_n$ was chosen. The resulting likelihoods for this parameters are shown in Fig. 1. The likelihood p_{fault} is higher than the nominal likelihood which should speed up the switching to a model in which a sensor fault is considered. It should be pointed out that (9) is not a proper pdf since its integral is not equal to one. Theoretically (9) could be normalized, e.g., by restricting the domain on which it takes values $\neq 0$ but in the context the likelihood is used in this paper we don't see an advantage by doing so.

With $p_{\text{nom}}^{(l)}(z_k^{(l)}|\mathbf{x}_k)$ and $p_{\text{fault}}^{(l)}(z_k^{(l)}|\mathbf{x}_k)$ now the pdfs $p(\mathbf{z}_k|\mathbf{x}_k^{(i)}, M_k^{(j)})$ for the different models required in equation (6) can be expressed

$$p(\mathbf{z}_k|\mathbf{x}_k^{(i)}, M_k^{(j)}) = \left(\prod_{l\in L^{(j)}} p_{\text{nom}}^{(l)}(z_k^{(l)}|\mathbf{x}_k^{(i)})\right) \prod_{l\in\{1,2,3\}\setminus L^{(j)}} p_{\text{fault}}^{(l)}(z_k^{(l)}|\mathbf{x}_k^{(i)}),$$

where the index set $L^{(j)}$ contains the indices of the axes that are assumed to work normally. For example for $M^{(3)}$ the set would be $L^{(3)} = \{1, 3\}$ and for $M^{(7)}$ the set is $L^{(7)} = \{2\}$.

3. Filter Algorithm

Now that the fault detection scheme is introduced and the likelihoods are defined, the filter can be implemented. The algorithm is mainly a sampling importance resampling (SIR) particle filter (Arulampalam et al., 2002) with a prediction step based on an motion model and an update step based on the magnetometer measurements and a single axis accelerometer.

The integration of the accelerometer is done in a straightforward way. The sensor is pointing into driving direction and is leveled with the track and hence as an approximation we assume it measures only the train acceleration. The measurements are then accounted for in the update step of the filter. In the update step, the weight of each particle is multiplied with the likelihood of the current accelerometer measurement. The likelihood of the acceleration measurements a_k is set to a Gaussian pdf $p(a_k|\mathbf{x}_k^{(i)}) = \mathcal{N}(a_k; \ddot{s}_k^{(i)}, \sigma_a^2)$ where the mean is the acceleration of the particle corresponding to the updated weight. The accelerometer measurements are extremely noisy due to vibrations of the train and hence should not be strongly weighted in the update step. Here a value of $\sigma_a = 1\,\mathrm{m\,s^{-2}}$ was chosen, which is close to the maximum acceleration of the train. The accelerometer measurements therefore only have a considerable effect on the resulting weights when there is big difference between a particle's acceleration and the measured one, e.g., when the train accelerates from standstill. Due to the large σ_a value and the strong noise the pitch angle and the bias of the sensor can be neglected. The sensor bias must be only estimated when it is in the order of magnitude than σ_a but even then, it should be sufficient to estimate it once when the localization system is booted.

Based on the above stated definitions and equations one iteration of the filter algorithm can be described in the following steps:

1. **Prediction:** Take the particle cloud form the previous time step $\{\mathbf{x}_{k-1}\}^{1:N_p}$ and apply the motion model to each particle $\mathbf{x}_{k-1}^{(i)}$ to obtain $\{\mathbf{x}_k\}^{1:N_p}$. Together with the set of weights from the previous time step $\{w_{k-1}\}^{1:N_p}$ this gives the predictive pdf for (6).

2. **Calculation of likelihood:** Calculate $\lambda_k^{(j)}$ for each model based on (6) and the predicted particles.

3. **Update the model probabilities:** Use the likelihoods $\lambda_k^{(j)}$ to update the model probabilities from the previous time step $\Pr(M_{k-1}^{(j)}|\mathbf{Z}_{1:k-1})$ according to (4).

4. **Decide for a model and update weights:** Select the model index $\hat{j} = \arg\max_j \Pr(M_k^{(j)}|\mathbf{Z}_{1:k})$ with the highest probability and then update the weights. In the weight update the chosen model decides which sensor axes are used. For a SIR particle filter (Arulampalam et al., 2002) the update simply becomes $w_k^{(i)} \propto w_{k-1}^{(i)} \prod_{l\in L^{(\hat{j})}} p_{\text{nom}}^{(l)}(z_k^{(l)}|\mathbf{x}_k^{(i)})$ where the index set L is the one for model $\hat{j}$.

5. **Output result and resample:** Compute a point estimate from the updated Dirac mixture density for the filter output and perform resampling if required, e.g., when the effective number of samples is below a threshold.

For the prediction step an motion model of the train is required. Here a simple piecewise constant Wiener process acceleration model according to (Bar-Shalom et al., 2002) is used

$$\mathbf{x}_k = \begin{bmatrix} s_k \\ \dot{s}_k \\ \ddot{s}_k \end{bmatrix} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{v}_k, \tag{10}$$

where the process noise $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is Gaussian distributed. The covariance matrix of the process noise is

$$\mathbf{Q} = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & \frac{T^2}{2} \\ \frac{T^3}{2} & T^2 & T \\ \frac{T^2}{2} & T & 1 \end{bmatrix} \sigma_a^2. \tag{11}$$

The parameter σ_a^2 should be set roughly in the order of magnitude of the maximum change in acceleration within on sampling period T.

In addition to the five steps described above, after resampling additional noise is added to the resampled particles. This procedure is sometimes called roughening and was proposed in (Gordon et al., 1993). At the first glance, adding more noise to the particles seems to be a bad idea because this increases the volume of the particle cloud and reduces its density in the relevant parts of the state space. Nevertheless, from our experience adding additional noise helps to keeps the particle cloud "wide" enough to avoid that particles lose track of the true position in the presence of small inaccuracies in the magnetic map and measurements.

4. Removal of Alternating Magnetic Fields

The error detection is supposed to detect and exclude errors in the measurements caused by passing trains, outdated maps, and sensor faults. In addition to such errors, the measurements are also disturbed by magnetic fields caused from the currents in the overhead line. These disturbances can be also observed when the train on which the magnetometer is mounted on itself has no electrical engine. Fortunately, the frequency of the overhead line current is known and fixed to $16.67\,\mathrm{Hz}$. The current introduces a sinusoidal signal into the measurements with a changing unknown amplitude and phase. This sinusoidal signal can be filtered out with a band-stop filter but care must be taken to keep the filter lag small. Here we propose an alternative to a band-stop filter by estimating the amplitude and the phase of the sinusoidal disturbance. To do this, the following model is assumed for the measurements of the l-th magnetometer axis

$$z_k^{(l)} = \mathrm{m}^{(l)}(s_k) + \alpha_k^{(l)} \sin(2\pi f_0 k T + \varphi_k^{(l)}) + n_k^{(l)}. \tag{12}$$

The phase appears in a nonlinear fashion in the above equation which would complicate the estimation. Fortunately, by applying the addition theorem the above equation can be reformulated to

$$\begin{aligned} z_k^{(l)} &= \mathrm{m}^{(l)}(s_k) + \alpha_k^{(l)} \Big(\sin(2\pi f_0 k T) \cos(\varphi^{(l)}) + \cos(2\pi f_0 k T) \sin(\varphi^{(l)}) \Big) + n_k^{(l)} \\ &= \mathrm{m}^{(l)}(s_k) + \beta_k^{(l)} \sin(2\pi f_0 k T) + \gamma_k^{(l)} \cos(2\pi f_0 k T) + n_k^{(l)}, \end{aligned} \tag{13}$$

where the phase and amplitude information is combined in the parameters $\beta_k^{(l)}$ and $\gamma_k^{(l)}$ that are linearly connected to the measurements. For the estimation of the sine parameters we assume that for a few samples of the magnetometer the amplitude and phase stay constant and that the train movement during this few samples does not lead to fast changes in the magnetic field. Under this assumptions, first the mean of the samples is removed from the data and second a linear least squares estimator is applied to the mean-free data to estimate $\beta_k^{(l)}$ and $\gamma_k^{(l)}$. Once the parameters are estimated the sine is reconstructed and removed from the measured data. Since the estimator is applied only to a few samples, the computational complexity is low and hence the estimation can be extended to other known frequencies that might disturb the magnetometer measurement. In this paper in addition to $16.67\,\mathrm{Hz}$ of the overhead line also the parameters of a $50\,\mathrm{Hz}$ sine corresponding to the German low-voltage network is estimated.

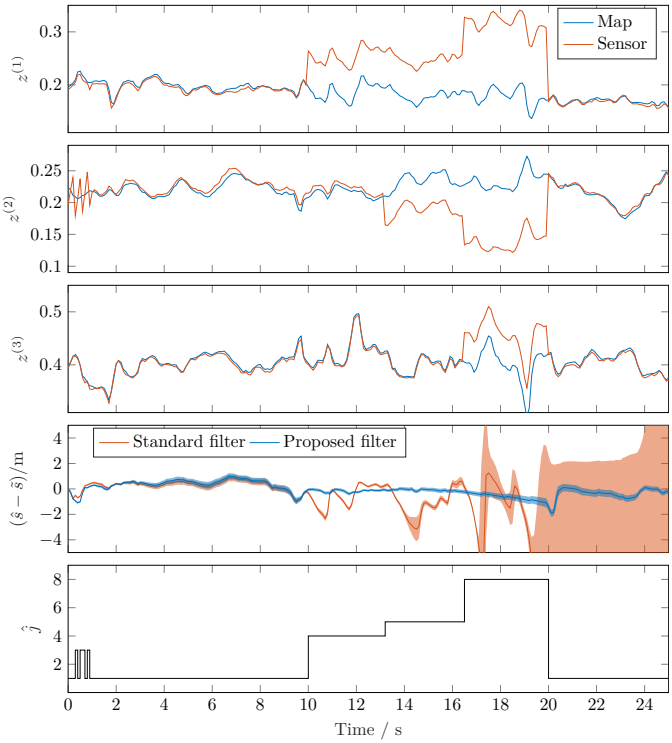

Figure 2: Result for simulated bias errors on data set 1. The top three plots show in blue the magnetic value stored in the map for the three sensor axes at the ground-truth position. The red lines are the measurements from the sensor used for localization. The fourth plot shows the position error of the proposed filter and a SIR particle filter without fault detection. The solid line is the mean and the transparent area around it is three times the standard deviation of the error over all Monte Carlo runs. In the bottom plot the model index $\hat{j}$ selected by the proposed filter is shown. The shown model index is the result of one Monte Carlo run but for the other runs the result looks very similar.

III. EVALUATION

1. Measurement and Evaluation Setup

In this section, the above presented algorithms are applied to data recorded with an Alstom Coradia Lint41 diesel train running on a track section between Augsburg central station and Friedberg station. The magnetometer and accelerometer data were recorded with an Xsens MTi inertial measurement unit (IMU). For mapping of the magnetic field and as ground-truth for the particle filter evaluation, a u-blox LEA 6T single frequency GNSS receiver was used. The IMU data was recorded with a rate of 200 Hz and the GNSS data with 1 Hz. For the evaluation, five data sets recorded on the same track section are considered. One data set is used for creating the map and the other four to test the proposed filter. For the map we selected the data set that was not affected by disturbances other than the ones caused by the static infrastructure.

The track section where the data is recorded starts close to Augsburg central station and runs below multiple bridges with some streets on it and passes the Lech river at the end. Overall the environment is a mixture of urban and suburban areas. The track has an overhead line but the train itself has a diesel engine, hence the currents disturbing the magnetic measurements are caused by other train with an electrical engine in the vicinity. The different data sets have a duration of $\approx 300 - 390\,\mathrm{s}$ during which the train travels roughly $4.4\,\mathrm{km}$ with an top speed of $105\,\mathrm{km\,h^{-1}}$.

In the evaluation 100 Monte Carlo runs were performed with different realizations of the process noise to see of the filter results are reproducible. Throughout the evaluation the filter uses 7500 particles and is updated at a rate of 10 Hz. With this setting the run time of the filter on the workstation used in the Monte Carlo simulation was roughly five times shorter than the duration of the processed data set, indicating that the filter can be run in real time.

2. Proof of Concept with Simulated Errors

Before realistic faults and disturbances are considered, a short test was performed in which errors are added to the measurements of data set one. In the error simulation, a bias is added to one, then two and finally all three sensor axes. Figure 2 shows the data with the simulated errors, the magnetic map and the position error of the proposed filter in comparison to a particle filter

Table 2: Mean and standard deviation of the RMSE and the maximum absolute position error. The values are calculated across all Monte Carlo runs to show the repeatability of the filter results.

	Proposed Filter				SIR Filter			
data set	$\mu_{\text{RMSE}}/\text{m}$	$\sigma_{\text{RMSE}}/\text{m}$	$\mu_{\text{max}}/\text{m}$	$\sigma_{\text{max}}/\text{m}$	$\mu_{\text{RMSE}}/\text{m}$	$\sigma_{\text{RMSE}}/\text{m}$	$\mu_{\text{max}}/\text{m}$	$\sigma_{\text{max}}/\text{m}$
1	1.78	0.01	5.77	0.04	1.78	0.01	5.77	0.03
2	1.78	0.01	7.54	0.04	1.86	0.01	7.61	0.30
3	1.56	0.00	5.44	0.03	1.56	0.01	5.45	0.03
4	2.22	0.00	7.97	0.03	2.27	0.07	15.84	6.46

without fault detection. In the top three plots the data of the map at the ground truth position is compared to the value measured by the three magnetometer axes. Note, the used sensor performs an internal normalization w.r.t. the Earth's magnetic field strength hence a value of one translates roughly to $40\,\mu\text{T}$. From the sensor data in Fig. 2 it is clearly visible that after $10\,\text{s}$, $13.2\,\text{s}$ and $16.5\,\text{s}$ the bias errors are added first to sensor one then sensor two and finally to sensor three. The errors have the values 0.07 , -0.05 and 0.06 and are relatively large compared to the overall variability of the magnetic field to see if the filter can handle large errors without divergence. The fourth plot from the top shows the estimation error of the proposed and the standard SIR particle filer. The solid lines in the plot are the average position error calculated over all Monte Carlo runs. To indicate the spread between the different Monte Carlo runs the area around the mean with same color indicates three times the standard deviation. From the error it is clearly visible that the proposed filter keeps tracking the train position quite well during the whole duration in which the errors are present whereas the filter without fault detection shows an increasing position error and a high variance between the different Monte Carlo runs. From the bottom plot in Fig 2 and Table 1 it can be seen that for all three errors the correct model index $\hat{j}$ is selected. Furthermore, the correct model was selected in the same time step the error occurred and hence no faulty measurements was passed to the filter weight update. It should be noted, that at the beginning of the data set sensor two has sawtooth like disturbances on it. Those are caused by currents in the overhead line which are not filtered out because the buffer of the sine estimator was not filled jet. As can been seen in the bottom plot those distortions were also detected by the filter and the selected model switched to $M^{(3)}$ which excludes sensor two from the weight update. From this example we see that the proposed filter can detected faults correctly and decrease the error is the estimated position. Since this was a rather artificial example we no will have a look how the filter performs in a real scenario.

3. Evaluation with Measurement Data

For the evaluation of the filter under realistic conditions, the proposed filter and a standard SIR filter were applied to all four data sets. In these four data sets at different parts of the track disturbances in the measurements are present. In Table 2 some statistics of the error over the 100 Monte Carlo runs for both filters are shown. For the results in the table the root-mean-squared error (RMSE) for each Monte Carlo run was calculated and then the mean μ_{RMSE} and standard deviation σ_{RMSE} over all runs was determined. In a similar way the values μ_{max} and σ_{max} of the maximum absolute error across all Monte Calro runs was calculated.

From the table it can be seen that in terms of the RMSE both filters perform almost identical for data set 1 and 3. For run 2 and 4 the differences in the RMSE values are still small but more noticeable. Considering that the RMSE averages the errors over the duration of the complete data set and that disturbances in the measurements, e.g., due to passing trains, are present only for a couple of seconds it was to be expected that the RMSE is quite close. The differences between the filter algorithms become more pronounced when the absolute maximum error is considered. For run 2 and 4 the spread of the maximum error between the different Monte Carlo runs for the SIR filter is noticeable higher than for the proposed filter. This shows that the standard filter has difficulties to track the train position at some point of the data set and that the obtained results are not that repeatable. To illustrate this, in Fig 3 Fig 4 a short section of the output of both filters is shown alongside with the map and the measured data. As before in Fig. 2, the figures show the mean and three times the standard deviation of the position error over all Monte Carlo runs, the measurements and the magnetic field values at the ground truth position obtained from the map. In both examples it can be seen that for a couple of seconds the measurements do not fit to the expected value from the map. During the presence of this erroneous measurements, the proposed filter switches to a faulty model and the estimation errors are considerable smaller. Also, its standard deviation across multiple Monte Carlo runs is reduced leading to repeatable and consistent results. For the SIR filter the standard deviation, especially for data set 4, is considerable higher which in the worst case could lead to a divergence of the particle cloud.

In contrast to the simulated errors, the true model index is not known for the real data and hence the estimated model index $\hat{j}$ can be evaluated only qualitatively based on the position error. Furthermore, care must be taken when comparing the expected magnetic field value from the map to the measured data because the expected value is obtained from the ground-truth GNSS position. For example, in Fig 4 the train is at standstill and there is a bias between the measurements and the map. This bias

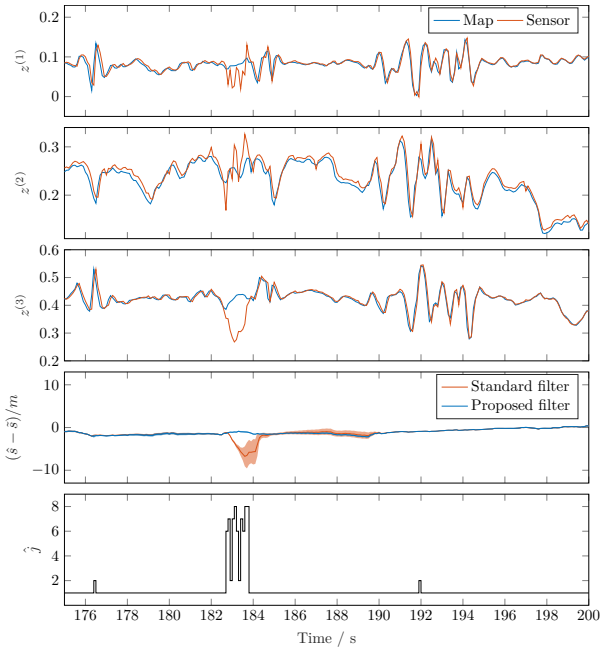

Figure 3: Zoomed in result of data set 2 with larger estimation errors due to disturbances in the measurements.

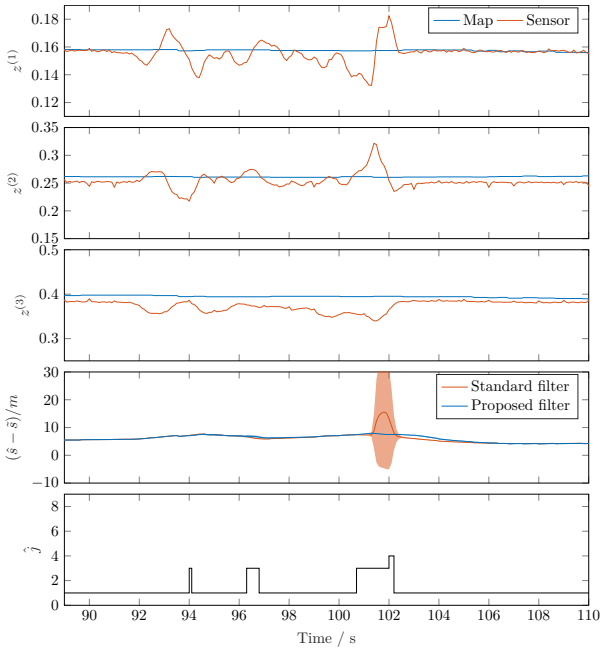

Figure 4: Zoomed in result of data set 4 with larger estimation errors due to disturbances in the measurements.

could be either a real bias and hence should be detected by the proposed filter when it large enough or it is simply caused by an offset in the GNSS ground truth. If the bias is caused by GNSS errors the bias might not be present for filter and hence cannot be detected.

IV. CONCLUSION

In this paper, a novel robust particle filter for magnetic localization was proposed and evaluated. The proposed filter is based on a multiple model fault detection and exclusion scheme that detects if one, two or all three magnetometer axes are affected by external disturbances that are not consistent with the magnetic map. When a fault is detected, the faulty sensor axis is excluded from the weight update of the particle filter. To show the performance of the proposed filter, it was compared to a standard SIR particle filter in an evaluation based on real train data. The evaluation showed that the proposed fault detection method works and that the exclusion of erroneous measurements from the filter update step can decrease the position error considerably compared to a filter without fault detection.

REFERENCES

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T. (2002). *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, Inc., Hoboken.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113(6).

Hashimoto, M., Kawashima, H., Nakagami, T., and Oba, F. (2001). Sensor fault detection and identification in dead-reckoning system of mobile robot: interacting multiple model approach. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326.

Menke, T. and Maybeck, P. (1995). Sensor/actuator failure detection in the vista f-16 by multiple model adaptive estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 31(4):1218–1229.

Siebler, B., Heirich, O., and Sand, S. (2018). Train Localization with Particle Filter and Magnetic Field Measurements. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–5.

Siebler, B., Heirich, O., Sand, S., and Hanebeck, U. D. (2020). Joint train localization and track identification based on earth magnetic field distortions. In *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 941–948.

Siebler, B., Lehner, A., Sand, S., and Hanebeck, U. D. (2021). Evaluation of Simultaneous Localization and Calibration of a Train Mounted Magnetometer. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, pages 2285–2293.

Zhang, Y. and Li, X. (1998). Detection and diagnosis of sensor and actuator failures using imm estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4):1293–1313.