Magnetic Field-based Indoor Localization of a Tracked Robot with Simultaneous Calibration

Benjamin Siebler*, Tim Gerstewitz*†, Stephan Sand*, and Uwe D. Hanebeck‡
*German Aerospace Center (DLR), Institute of Communications and Navigation
†Technical University of Munich (TUM)
‡Karlsruhe Institute of Technology (KIT), Intelligent Sensor-Actuator-Systems Laboratory
benjamin.siebler@dlr.de

***Abstract*—Magnetic-field based indoor localization uses distortions of the Earth magnetic field caused by magnetic material in the surrounding of the platform that is to be localized. Existing magnetic localization methods compare measurements from a magnetometer with a magnetic field map to estimate the position of the platform, on which the sensor is mounted. For the comparison it is typically assumed that the magnetometer is calibrated. Unfortunately, for some platforms calibration is not straightforward, particularly when they are large or heavy. In this paper we therefore propose a simultaneous localization and calibration approach. The approach is based on a Rao-Blackwellized particle filter that makes use of the conditional linearity of magnetometer measurements and calibration parameters. The feasibility of the proposed particle filter is evaluated with three data sets recorded with a differential drive robot in an indoor environment.**

I. Introduction

Localization is still a challenge in environments where global navigation satellite system (GNSS) signals are strongly degraded or completely blocked. Examples of such environments are the interior of buildings, tunnels, and urban canyons, where high-rise buildings block some of the satellite signals, resulting in multipath effects, a poor geometry, and eventually large position errors. For localization in GNSS-denied areas a great variety of approaches can be found in the literature. The proposed approaches range from deploying dedicated infrastructure for localization, e.g. ultra-wideband devices [1], and the use of signals of opportunity (SoO) [2], to inertial sensor-based solutions [3]. In this paper the focus is on the use of the Earth magnetic field as a SoO. The idea is based on the observation that magnetic material in the environment, e.g. the steel in the concrete of a building, distorts the Earth's magnetic field which leads to a position dependent variation of the magnetic field. The feasibility of magnetic field-based localization was already shown for interiors [4], [5], [6], roads [7], railways [8] and the airspace [9]. Typically, the proposed algorithms for magnetic localization are based on a comparison of magnetometer measurements with a map. This comparison requires the magnetometer to be calibrated. Unfortunately, common calibration methods require a rotation of the sensor and the platform it is mounted on in a homogenous field [10]. For heavy or large platforms this might be difficult to achieve. We encountered this problem in the context of magnetic localization of trains, which have very limited degrees of freedom in their movement and are very large and heavy. To enable magnetic localization also for trains we therefore proposed in [11] and [12] a simultaneous localization and calibration (SLAC) algorithm. The algorithm utilizes a Rao-Blackwellized particle filter that exploits that conditioned on the position, the calibration is a linear estimation problem. In contrast to our previous work, in this paper we propose a SLAC algorithm for the localization of a differential drive robot in an indoor environment. The presented algorithm is evaluated with three different measurements in an indoor laboratory environment where the ground-truth is provided by an optical tracking system.

II. Simultaneous Localization and Calibration

A. Magnetometer Sensor Model

To understand the proposed algorithm, it is crucial to define the magnetometer model that will be used in the filter. Here a common linear calibration model is considered, see e.g. [13]. With the linear model the measurement of the magnetometer $\tilde{\mathbf{z}}^b$ at time step k can be written as

$$\tilde{\mathbf{z}}_k^b = \mathbf{C}\mathbf{R}_{n,k}^b \mathbf{z}_k^n + \mathbf{b} + \mathbf{n}_k \ , \tag{1}$$

where $\mathbf{C}$ is a $3{\times}3$ calibration matrix, $\mathbf{R}_{n,k}^b$ is the rotation matrix between the sensor's body frame b and the navigation frame n, $\mathbf{z}_k^n$ is the three-dimensional true magnetic vector field given in the navigation frame, and $\mathbf{b} \in \mathbb{R}^3$ is the sensor bias. Additionally, additive white Gaussian measurement noise $\mathbf{n}_k \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I})$ is considered. The calibration matrix $\mathbf{C}$ accounts for soft magnetic material in the sensor platform that interacts with the true external magnetic field $\mathbf{z}_k^n$. In contrast, the bias vector $\mathbf{b}$ is a result of hard magnetic material that has a permanent magnetization.

For the case that $\mathbf{z}_k^n$ and $\mathbf{R}_{n,k}^b$ are known for each measurement $\tilde{\mathbf{z}}_k^b$ the estimation of the calibration parameters in (1) is a linear problem. This becomes clearer by rearranging (1) to

$$\begin{aligned}\tilde{\mathbf{z}}_k^b &= \begin{bmatrix} \mathrm{h}^T(\mathbf{R}_{n,k}^b \mathbf{z}_k^n) & \mathbf{0}_{1\times 4} & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{1\times 4} & \mathrm{h}^T(\mathbf{R}_{n,k}^b \mathbf{z}_k^n) & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{1\times 4} & \mathbf{0}_{1\times 4} & \mathrm{h}^T(\mathbf{R}_{n,k}^b \mathbf{z}_k^n) \end{bmatrix} \boldsymbol{\theta} + \mathbf{n}_k \\ &= \mathbf{H}(\mathbf{R}_{n,k}^b \mathbf{z}_k^n)\boldsymbol{\theta} + \mathbf{n}_k, \end{aligned} \tag{2}$$

with the function

$$\mathrm{h}^T(\mathbf{R}_n^b(k)\mathbf{z}_k^n) = \begin{bmatrix} (\mathbf{R}_{n,k}^b \mathbf{z}_k^n)^T & 1 \end{bmatrix}. \tag{3}$$

The parameter vector $\boldsymbol{\theta}$ contains the nine elements of matrix $\mathbf{C}$ and the three elements of $\mathbf{b}$

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{c}_1^T & b_1 & \mathbf{c}_2^T & b_2 & \mathbf{c}_3^T & b_3 \end{bmatrix}^T, \tag{4}$$

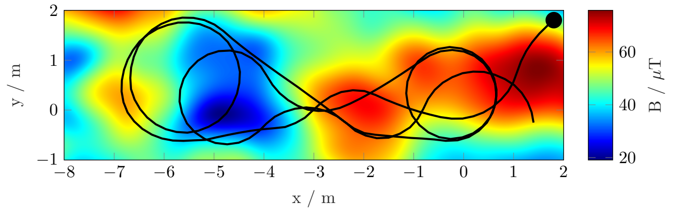

Fig. 1. Magnitude of the magnetic flux density in DLR's laboratory and the trajectory of the first measurement run. The black dot marks the start position of the robot.

where b_i is the i-th element of $\mathbf{b}$ and $\mathbf{c}_i = \begin{bmatrix} c_{i1} & c_{i2} & c_{i3} \end{bmatrix}^T$ is the transposed i-th row of $\mathbf{C}$. With (2) and the values of $\mathbf{R}^b_{n,k}$ and $\mathbf{z}^n_k$ it is then straightforward to implement a least squares estimator or a Kalman filter to estimate $\boldsymbol{\theta}$ from a sequence of magnetometer measurements. For the estimation at least four measurements are required, otherwise the system of equations is underdetermined. Furthermore, the measurements must be taken at different attitudes represented by $\mathbf{R}^b_{n,k}$ or different values of the external field $\mathbf{z}^n_k$ to render the parameters observable. This is also why calibration algorithms require a rotation of the magnetometer in a homogeneous field. As will become clear later in this paper, for SLAC we make use of both, changes in the field and in the attitude to estimate the parameters $\boldsymbol{\theta}$.

B. Likelihood of Magnetometer Measurements

For magnetic localization a map of the magnetic field is required that serves as a kind of magnetic "fingerprint" data base. A comparison of a calibrated magnetometer measurement with the map then enables to estimate the most likely positions at which the measurement was performed. Here the magnetic map is simply a function $\mathbf{m}^n(\mathbf{p})$ that for a position $\mathbf{p}$ returns the magnetic vector at that position given in the navigation frame. The measurement model (2) thereof becomes

$$\tilde{\mathbf{z}}^b_k = \mathbf{H}(\mathbf{R}^b_{n,k}\,\mathbf{m}^n(\mathbf{p}_k))\boldsymbol{\theta} + \mathbf{n}_k, \tag{5}$$

where $\mathbf{z}^n_k$ is replaced by the map at position $\mathbf{p}_k$. Let us assume for now that the sensor is calibrated and that $\boldsymbol{\theta}$ is known. For this case the likelihood $p(\tilde{\mathbf{z}}^b|\mathbf{p})$ of a magnetometer measurement at position $\mathbf{p}$ can be calculated from the distribution of the sensor noise

$$p(\tilde{\mathbf{z}}^b_k|\mathbf{p}_k) = \mathcal{N}(\tilde{\mathbf{z}}^b; \mathbf{H}(\mathbf{R}^b_{n,k}\,\mathbf{m}^n(\mathbf{p}_k))\boldsymbol{\theta}, \sigma^2_n\mathbf{I}). \tag{6}$$

Note that the map is always assumed to be known, thus the conditioning on the map is neglected for brevity in this paper.

In general, the magnetic field and therefore the map is a nonlinear function of the position. This can be easily seen from the example in Fig. 1 that shows the magnitude of the magnetic field measured in DLR's Holodeck laboratory. The non-linearity of the map leads to a nonlinear measurement model. In the remainder of the paper we therefore propose the use of a particle filter for localization.

C. Particle Filter for Localization

In order to implement the particle filter the state space model of the platform that should be localized has to be defined. The platform used in this paper is the tracked robot shown in the left part of Fig. 2. The state vector contains the robot's pose consisting of the x- and y-position and the heading

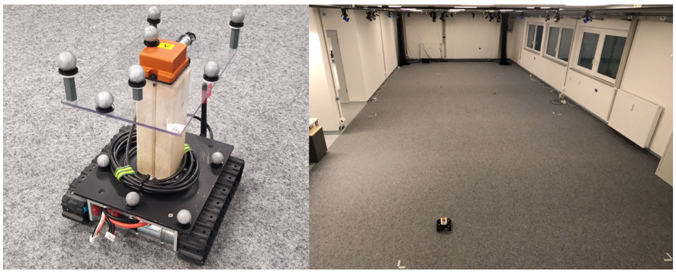

Fig. 2. (left) Differential drive robot. On top of the transparent plastic platform, the IMU and the infrared markers for the Vicon tracking system can be seen. (right) Differential drive robot in the Holodeck laboratory with Vicon cameras on the ceiling.

$$\mathbf{x}_k = \begin{bmatrix} p_{\mathrm{x},k} & p_{\mathrm{y},k} & \varphi_k \end{bmatrix}^T = \begin{bmatrix} \mathbf{p}^T_k & \varphi_k \end{bmatrix}^T. \tag{7}$$

The motion model is set to

$$\begin{aligned} p_{\mathrm{x},k} &= p_{\mathrm{x},k-1} + \cos(\varphi_{k-1})(v_{k-1} + w_{v,k})T \\ p_{\mathrm{y},k} &= p_{\mathrm{y},k-1} + \sin(\varphi_{k-1})(v_{k-1} + w_{v,k})T \\ \varphi_k &= \varphi_{k-1} + (\omega_{\varphi,k-1} + w_{\omega,k-1})T, \end{aligned} \tag{8}$$

where v is the linear speed of the robot, ω its turn rate, T the sampling period between two time steps, and w_v and w_ω is the speed and turn rate process noise which is assumed to be Gaussian. The turn rate is measured with a gyroscope and the speed is calculated from the two wheel encoders of the differential drive. The speed is obtained from the equation

$$v_k = r_{\text{wheel}}\frac{\omega_{\mathrm{R},k} + \omega_{\mathrm{L},k}}{2}, \tag{9}$$

with the turn rates of the left $\omega_{\mathrm{L},k}$ and right track $\omega_{\mathrm{R},k}$, and the wheel radius r_{wheel}.

The particle filter approximates the posterior probability density function (pdf) $p(\mathbf{x}_{0:k}|\tilde{\mathbf{z}}^b_{1:k})$ of the state sequence $\mathbf{x}_{0:k}$ from time step 0 to k conditioned on the sequence $\tilde{\mathbf{z}}^b_{1:k}$ of all measurements available up to that point. The approximation is given by a set of particles $\{\mathbf{x}^{(i)}_{0:k}\}^{N_p}_{i=1}$ with associated weights $\{w^{(i)}_k\}^{N_p}_{i=1}$. With the weighted set we can write the posterior in the form of a Dirac mixture density

$$p(\mathbf{x}_{0:k}|\tilde{\mathbf{z}}^b_{1:k}) \approx \sum_{i=1}^{N_p} w^{(i)}_k \delta_{\mathbf{x}^{(i)}_{0:k}}(\mathbf{x}_{0:k}), \tag{10}$$

where $\delta_{\mathbf{x}^{(i)}_{0:k}}(\mathbf{x}_{0:k})$ is the Dirac distribution that vanishes everywhere in the state space except at the value of the i-th particle $\mathbf{x}^{(i)}_{0:k}$. In the following we use a simple particle filter with resampling when the effective number of particles falls below a threshold. As importance density, from which the particles are drawn, the one-step prediction pdf $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ obtained from the motion model (8) is used. For this particular choice of importance density the weight update is the product of the likelihood with the previous weight

$$w^{(i)}_k = w^{(i)}_{k-1} \cdot p(\tilde{\mathbf{z}}^b_k|\mathbf{x}^{(i)}_k). \tag{11}$$

When the calibration parameters are known, the likelihood required in the weight update is given by (6) and the chosen state vector (7)

$$p(\tilde{\mathbf{z}}_k^b|\mathbf{x}_k) = \mathcal{N}(\tilde{\mathbf{z}}^b; \mathbf{H}(\mathbf{R}_n^b(\varphi_k)\,\mathbf{m}^n(\mathbf{p}_k))\boldsymbol{\theta}, \sigma_n^2\mathbf{I}), \tag{12}$$

where the rotation matrix from the navigation into the body frame is a function of the robot's heading

$$\mathbf{R}_n^b(\varphi_k) = \begin{bmatrix} \cos(\varphi_k) & \sin(\varphi_k) & 0 \\ -\sin(\varphi_k) & \cos(\varphi_k) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{13}$$

D. Joint Estimation of Position and Calibration Parameters

In the previous section the particle filter for magnetic localization with known calibration parameters was presented. Unfortunately, performing calibration for heavy and large platforms is a complicated and laborious task or it is even impossible. Therefore, we propose to jointly estimate the calibration parameters during localization in a SLAC algorithm. From a theoretic point of view, the task is to find the joint posterior pdf $p(\mathbf{x}_{0:k}, \boldsymbol{\theta}|\tilde{\mathbf{z}}_{1:k}^b)$ of the state and the calibration parameters. In principle this task can be directly solved with a standard particle filter but the particle filter suffers from the curse of dimensionality, which means that when the dimension of the state space increases the number of required particles to densely sample the space increases quickly. For SLAC the state dimension is at least 15 (2D position, heading and the calibration parameters) which is already high for a particle filter. Fortunately, the estimation of the calibration parameters is a linear problem when conditioned on the position, which allows for Rao-Blackwellization of the filter. Instead of estimating the joint density in a single particle filter, the posterior is decomposed into a non-linear and a linear part

$$p(\mathbf{x}_{0:k}, \boldsymbol{\theta}|\tilde{\mathbf{z}}_{1:k}^b) = \underbrace{p(\boldsymbol{\theta}|\tilde{\mathbf{z}}_{1:k}^b, \mathbf{x}_{0:k})}_{\text{Kalman filter}}\,\underbrace{p(\mathbf{x}_{0:k}|\tilde{\mathbf{z}}_{1:k}^b)}_{\text{particle filter}}. \tag{14}$$

The decomposition of the posterior in (14) is straightforward but the implementation requires some thought. One way to see how the filter can be implemented is by looking on the weight update of the particle filter. From (11) it is clear that for the weight update the likelihood $p(\tilde{\mathbf{z}}_k^b|\mathbf{x}_k^{(i)})$ has to be evaluated but how do we do this for the case when we do not know the calibration parameters? The trick that allows us to evaluate the likelihood for unknown calibration parameters is to introduce the calibration parameters into the likelihood and then marginalize over them. With marginalization the likelihood becomes

$$\begin{aligned} p(\tilde{\mathbf{z}}_k^b|\mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b) &= \int_{-\infty}^{\infty} p(\tilde{\mathbf{z}}_k^b, \boldsymbol{\theta}_k|\mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b)\,d\boldsymbol{\theta}_k \\ &= \int_{-\infty}^{\infty} p(\tilde{\mathbf{z}}_k^b|\boldsymbol{\theta}_k, \mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b)\,p(\boldsymbol{\theta}_k|\mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b)\,d\boldsymbol{\theta}_k \\ &= \int_{-\infty}^{\infty} p(\tilde{\mathbf{z}}_k^b|\boldsymbol{\theta}_k, \mathbf{x}_k)\,p(\boldsymbol{\theta}_k|\mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b)\,d\boldsymbol{\theta}_k. \end{aligned} \tag{15}$$

Note, in (11) the Markov property $p(\tilde{\mathbf{z}}_k^b|\mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b) = p(\tilde{\mathbf{z}}_k^b|\mathbf{x}_k)$ was assumed to hold but for uncertain calibration parameters this is no longer the case. Only in the last line of (15) the Markov property for $p(\tilde{\mathbf{z}}_k^b|\boldsymbol{\theta}_k, \mathbf{x}_k)$ was used since the parameters are considered completely known in this case. Intuitively this makes sense, the belief in the parameters $p(\boldsymbol{\theta}_k|\mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b)$, over which the marginalization is performed, depends on the complete history of the observations. From (15) now the likelihood can be obtained by recognizing that the posterior pdf $p(\boldsymbol{\theta}_k|\mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b)$ is Gaussian and can be obtained from a Kalman filter due to the linearity of the calibration parameters when conditioned on the robot pose. Since the likelihood $p(\tilde{\mathbf{z}}_k^b|\boldsymbol{\theta}_k, \mathbf{x}_k)$ of the magnetometer is also considered to be Gaussian the integral in (15) has the closed form solution

$$\begin{aligned} p(\tilde{\mathbf{z}}_k^b|\mathbf{x}_{0:k}, \tilde{\mathbf{z}}_{1:k-1}^b) &= \mathcal{N}\left(\tilde{\mathbf{z}}_k^b; \hat{\mathbf{z}}_k^b, \sigma_n^2\mathbf{I} + \mathbf{H}^{\mathrm{d}}(\mathbf{x}_k)\boldsymbol{\Sigma}_{k,\boldsymbol{\theta}}^-\,\mathbf{H}^{\mathrm{d}}(\mathbf{x}_k)^T\right) \\ &= \mathcal{N}\left(\tilde{\mathbf{z}}_k^b; \hat{\mathbf{z}}_k^b, \sigma_n^2\mathbf{I} + \mathbf{S}_k\right) \end{aligned} \tag{16}$$

where $\mathbf{H}^{\mathrm{d}}(\mathbf{x}_k)$ is the shorthand for $\mathbf{H}(\mathbf{R}_n^b(\varphi_k)\,\mathbf{m}^n(\mathbf{p}_k))$, $\hat{\mathbf{z}}_k^b$ is the expected measurement

$$\hat{\mathbf{z}}_k^b = \mathbf{H}^{\mathrm{d}}(\mathbf{x}_k)\hat{\boldsymbol{\theta}}_k^-, \tag{17}$$

and $\hat{\boldsymbol{\theta}}_k^-$ and $\boldsymbol{\Sigma}_{k,\boldsymbol{\theta}}^-$ are the Kalman filter state and covariance at time step k before the measurement update. From (16) we see that the marginalization adds the uncertainty of the calibration parameters $\boldsymbol{\Sigma}_{k,\boldsymbol{\theta}}^-$ projected into the measurement space to the pure noise covariance. Thus, a high uncertainty in the calibration parameters results in an high uncertainty of the likelihood and vice versa. To see how the particle filter and the Kalman filter interact likelihood (16) is plugged into the weight update (11) of the particle filter. Here it becomes clear that due to the conditioning of the likelihood on the state of the individual particles each particle must have a individual Kalman filter attached to it.

For SLAC the prediction and update step of the standard resampling particle filter is modified. The first modification is that when the robot does not move or turn the filter just keeps its current weights and particles, this means we do not perform any prediction or update. The second modification affects the weight and Kalman filter update. When the robot is moving or turning the Kalman filters are updated every time a new magnetometer measurement is available. In contrast, the particle weights are updated only when the robot has driven a predefined distance w.r.t. the last weight update. This modification is done to avoid rapid particle impoverishment. Imagine the following situation, the robot drives slowly and has not yet estimated its calibration parameters properly. Due to the slow movement the magnetic field does not change significantly and hence the filter gets multiple similar measurements. If now the positions and calibration parameters of a few particles fit particularly well to that similar measurements it can happen that only these few particles will remain after the resampling even though the particles are not following the correct trajectory. Therefore, the weights are only updated when the robot has traveled a distance after which one can expect to measure a considerable change in the magnetic field, in our case the update is performed every $20\,\mathrm{cm}$. When an update of the weights is performed not only the likelihood of the current measurement is considered. Instead the geometric mean of the likelihoods of all N_{mes} measurements received since the last update is used. The weight update therefore becomes

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot \prod_{j=k-N_{\text{mes}}+1}^{k} p(\tilde{\mathbf{z}}_j^b | \mathbf{x}_{0:j}^{(i)}, \tilde{\mathbf{z}}_{1:j-1}^b)^{\frac{1}{N_{\text{mes}}}} \quad (18)$$

when the robot has traveled the required distance. Between two updates the weights are kept constant $w_k^{(i)} = w_{k-1}^{(i)}$. The geometric mean is used here because the likelihoods are multiplied with the weight from the previous step and by using the mean of the likelihoods we avoid that the filter is over sensitive to noise and other errors in a single measurement. A similar idea was also utilized in [5]. To further reduce the sensitivity of the filter to errors in the magnetic map and the measurements, the Gaussian likelihood in (16) is replaced with a Gaussian mixture with two components

$$p(\tilde{\mathbf{z}}_k^b | \mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1}^b) = 0.3 \cdot \mathcal{N}\left(\tilde{\mathbf{z}}^b; \hat{\mathbf{z}}_k^b, \sigma_n^2 \mathbf{I} + \mathbf{S}_k\right) + 0.7 \cdot \mathcal{N}\left(\tilde{\mathbf{z}}^b; \hat{\mathbf{z}}_k^b, \sigma_{\text{GMM}}^2 \mathbf{I}\right). \quad (19)$$

The standard deviation σ_{GMM} should be chosen larger than the noise standard deviation σ_n, e.g., in this paper the values are set to $\sigma_n = 2.5\,\mu\text{T}$ and $\sigma_{\text{GMM}} = 5\,\mu\text{T}$. This reduces particle impoverishment in the presence of deviations in the data that are not explained by noise. For our experiments with a tracked robot described in the next section we observed for example deviations caused by motor currents and non-zero roll and pitch angles.

III. Evaluation

The feasibility of the proposed approach is shown on three complex trajectories driven with a tracked differential drive robot in DLR's Holodeck laboratory. The used robot and the laboratory are shown in Fig. 2.

A. Experimental Setup

For logging the magnetic field components and angular velocities, we placed an Xsens-MTi-G700 IMU on top of the robot, at a height of approximately 23 cm above ground. The IMU features both a three-axis magnetometer and a three-axis gyroscope and was placed on a transparent plastic platform 10 cm above the robot to mitigate the effects of the motors' magnetic fields. With this setup, we recorded the magnetic field components and angular velocities at a rate of 100 Hz. Additionally, we measured the robot's linear velocity at 20 Hz with encoders placed at the motors on both sides of the robot. For obtaining the ground truth pose, we recorded the IMU's 6D pose with an optical Vicon tracking system with sub-centimeter accuracy at a rate of 100 Hz. All signals were logged on a laptop connected wirelessly to the robot.

For building the magnetic map used for localization the robot drove a meander course through the laboratory using a path planner for control. The obtained data was then calibrated via ellipsoid fitting [10] with a previously recorded data set, measured on a meadow at the DLR site without any buildings nearby. Based on the calibrated data, we interpolated each component of the magnetic field with an individual Gaussian process (GP) on a 5 mm regularly spaced grid using a squared exponential kernel. To do so, the Matlab GPML Toolbox [14] was employed.

For evaluating the SLAC algorithm, the robot drove three complex trajectories in an area of 10 m by 3 m through the laboratory. The speed during the measurements was varying between 0.3 m/s and 0.7 m/s. The resulting trajectories have a duration of 135 s-186 s. In Fig. 1 the trajectory of the first measurement is shown on top of the magnitude of the magnetic field obtained from the GP regression.

TABLE I
Position and Heading Error Statistics

	SLAC				Odometry			
Traj.	RMSE_p [m]	RMSE_φ [°]	MaxAE_p [m]	MaxAE_φ [°]	RMSE_p [m]	RMSE_φ [°]	MaxAE_p [m]	MaxAE_φ [°]
1	0.090	1.829	0.241	6.216	1.616	32.147	4.293	59.478
2	0.094	1.695	0.297	6.881	2.556	45.863	5.786	79.976
3	0.072	1.381	0.161	3.668	2.390	52.502	5.187	91.431

B. Filter Setup

In the evaluation the filter uses 3000 particles and the Gaussian mixture model described in (19). With this amount of particles and an update rate of 10 Hz the processing time of the filter in Matlab was roughly one third of the data set length showing that the filter can be implemented in real time. The states of the Kalman filters are initialized randomly from a Gaussian distribution. The biases are drawn independently from $\mathcal{N}(0, (5\,\mu\text{T})^2)$. For $\mathbf{C}$ the diagonal elements are drawn from $\mathcal{N}(1, 1^2)$ and the off-diagonal elements from $\mathcal{N}(0, 1^2)$. The mean values of the Gaussian distributions are therefore chosen such as for a perfectly calibrated sensor, for which the biases and off-diagonal elements of $\mathbf{C}$ are all zero and the diagonal elements are one. The positions and headings of the initial particles are drawn from a normal distribution centered at the ground truth. The standard deviation of the Gaussians is set to 10 cm for the position states and 10° for the heading.

C. Results

To evaluate the performance of the proposed SLAC algorithm 100 Monte Carlo runs were performed for each of the three trajectories to see if the filter works also for different realizations of the process noise. For the evaluation we then selected the Monte Carlo run that led to the highest position root-mean-square-error (RMSE) although no run deviated considerably from the others. In Tab. I the RMSE and the maximal absolute error (MaxAE) for the 2D position and the heading is shown for SLAC and the unaided odometry (obtained from the linear speed and gyro turn rate). From Tab. I it is clearly visible that the pure odometry is worse than the proposed SLAC algorithm showing that the magnetic field significantly improves the odometry performance. Overall, the SLAC algorithm achieved an RMSE in the position below 10 cm and an absolute error below 30 cm. For the heading the RMSE is below 2° and the MaxAE below 7°. We also evaluated a particle filter that used the magnetometer data without calibration. The performance of this filter w.r.t. the

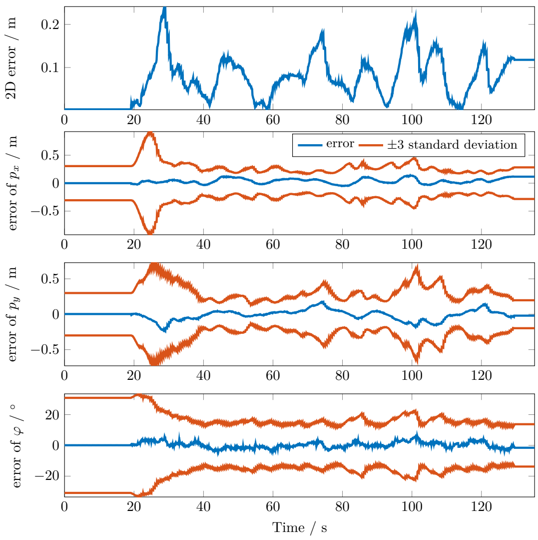

Fig. 3. Position and heading errors over time for trajectory one. The errors are indicated by a blue line and for the lower three plots the red lines show three times the standard deviation of this errors estimated by the Rao-Blackwellized particle filter.

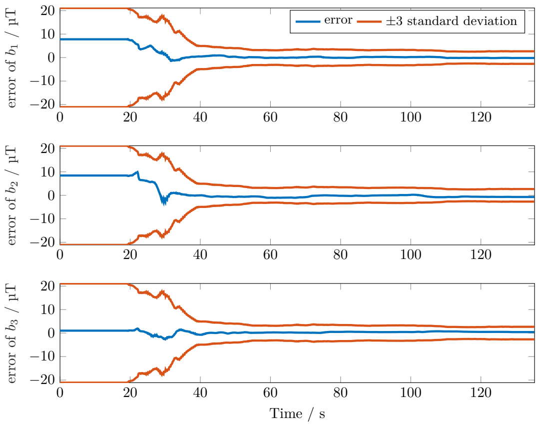

Fig. 4. Estimation error of the bias vector $\mathbf{b} = [b_1 \quad b_2 \quad b_3]^T$. The errors are shown in blue and in red three times the standard deviation estimated by SLAC is shown.

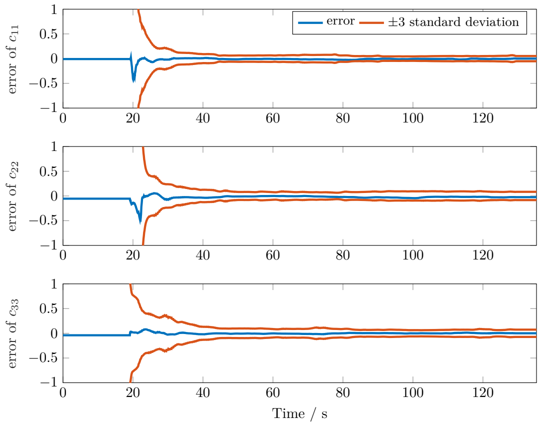

Fig. 5. Estimation errors of the diagonal elements of calibration matrix $\mathbf{C}$. The errors are shown in blue and in red three times the standard deviation estimated by SLAC is shown.

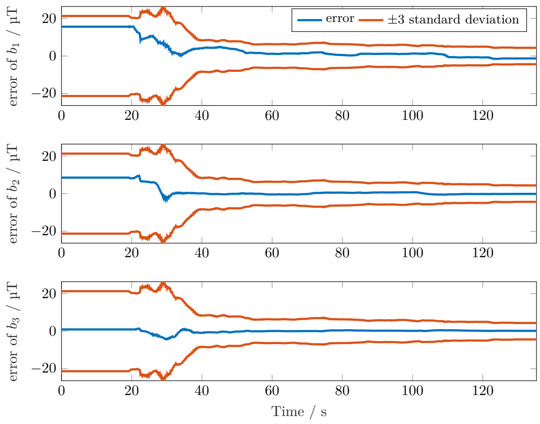

Fig. 6. Estimation error of the bias vector when the data of the first magnetometer axis is scaled by a factor of two. The errors are shown in blue and in red three times the standard deviation estimated by SLAC is shown.

position RMSE is in the range of 0.42 to $6.43\,\mathrm{m}$ and varies considerably not only between different trajectories but also between different Monte Carlo runs for the same trajectory. From what we saw in the results we would even argue that the particle filter without calibration does not track the robot position and the performance is somewhat random and sometimes even worse than the odometry.

In Fig. 3 the estimation errors of SLAC for the first trajectory from Fig. 1 are shown. For the heading and the position error in the x- and y- direction in addition three times the standard deviation estimated by the Rao-Blackwellized particle filter is shown by red lines. To obtain the errors the minimum-mean-square-error-estimate (MMSEE) is calculated by taking the weighted mean over all particles. The resulting MMSEE is then compared to the ground truth pose of the robot from the Vicon system. As seen from Fig. 3 the errors in the beginning are close to zero, this is because the particle cloud is scattered around the true position and heading as mentioned above. But that does not mean that the filter exactly knows the starting position and heading since the initial particles are randomly drawn from a Gaussian distribution and all have the same weight. At the beginning of Fig. 3 the robot is at standstill and hence also the filter is not updated leading to a constant error (besides some noise in the Vicon ground truth). After roughly $20\,\mathrm{s}$ the robot starts driving and the filter begins updating the weights. As a result, the error and the size of the particle cloud increases, noticeable by the increasing standard deviation. But as can be seen from Fig. 3 SLAC is able to bound the errors by incorporating information from the magnetic field to aid the odometry. The standard deviation estimated by the filter is well above the true errors which shows the filter is able to produce conservative estimates and does not underestimate its own uncertainty.

In Fig. 4 and Fig. 5 the estimation errors of the bias vector $\mathbf{b}$ and the diagonal elements of the calibration matrix $\mathbf{C}$ are shown with three times the corresponding standard deviation estimated by SLAC. The errors are calculated by comparing the MMSEE of SLAC, i.e., the weighted mean over all Kalman

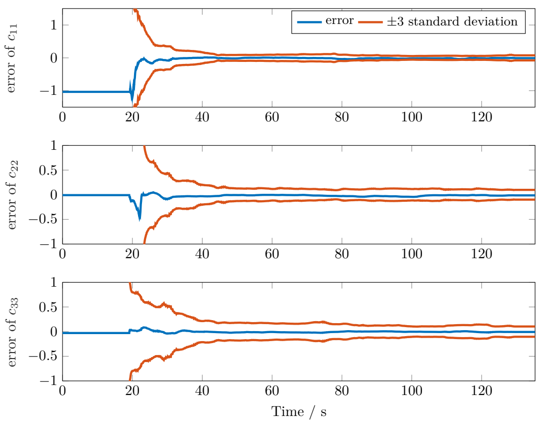

Fig. 7. Estimation errors of the diagonal elements of calibration matrix $\mathbf{C}$ when the data of the first magnetometer axis is scaled by a factor of two. The errors are shown in blue and in red three times the standard deviation estimated by SLAC is shown.

filter states, with the parameters obtained from ellipsoid fitting with the calibration data set. As for the position and heading, the parameter estimates are conservative. For b_1 and b_2 the initial error and standard deviation is reduced after the robot starts driving, showing that the biases are actually observable. Bias b_3 is relatively close to zero and since all biases are initialized from a zero mean Gaussian pdf the error is pretty small at the beginning and already within the accuracy SLAC can achieve. For the diagonal elements of $\mathbf{C}$, called also scale factors here, a similar behavior as for b_3 can be observed in Fig. 5. Since the true scale factors are close to one and we initialized all scale factors by sampling from a Gaussian with mean one, the errors are from the beginning on relatively small and within the achievable accuracy indicated by the standard deviation after convergence of the filter. For the off-diagonal elements of $\mathbf{C}$ a similar behavior is observed and hence the results are not shown here for brevity.

In order to test if the proposed SLAC algorithm can also cope with large deviations in the scale factors, another evaluation was performed in which the magnetometer measurements of the first trajectory were scaled. To achieve this, we simply multiplied the measurements of the first magnetometer axis by a factor of two before providing them to the SLAC algorithm. After multiplying the data, the scale factor is close to two and the bias is doubled. As seen in Fig. 6 and Fig. 7 the initial error of bias b_1 and scale factor c_{11} is now considerable larger than in Fig. 4 and Fig. 5. Fortunately, SLAC can also cope with this larger initial deviations in the parameters and can reduce the errors as soon the robot starts driving. With respect to the accuracy only a small degradation is observed, the RMSE for the position is $0.096\,\mathrm{m}$ and for the heading $2.107°$. The MaxAE has increased to $0.251\,\mathrm{m}$ for the position and slightly decreased to $5.681°$ for the heading.

IV. Conclusion

In this paper we showed how a tracked robot can be localized in an indoor environment with the distortions of the Earth magnetic field and the measurements of an uncalibrated magnetometer. In order to achieve this, a real time capable simultaneous localization and calibration (SLAC) algorithm was proposed. In the SLAC algorithm a Rao-Blackwellized particle is used to jointly estimate the robot pose and the calibration parameters.

The feasibility of the algorithm was shown is an evaluation using three data sets recorded in DLR's Holodeck laboratory. From the evaluation we could conclude that the proposed SLAC algorithm works and can achieve a position RMSE in the range of $10\,\mathrm{cm}$ and a heading RMSE around $2°$. Furthermore, the results showed that the calibration parameters are observable and that the algorithm can also cope with large initialization errors in the parameters.

Acknowledgment

This work is carried out in the Ubiquitous Spatio-Temporal Learning for Future Mobility (ULearn4Mobility) project funded by the Helmholtz AI Cooperation Unit under the grant ZT-I-PF-5-49.

References

[1] C. Gentner and M. Schmidhammer, "Ranging and multipath-enhanced device-free localisation with densely-meshed ultra-wideband devices," *IET Microwaves, Antennas & Propagation*, 2023.

[2] C. Gentner, T. Jost, W. Wang, S. Zhang, A. Dammann, and U.-C. Fiebig, "Multipath assisted positioning with simultaneous localization and mapping," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6104–6117, 2016.

[3] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Proceedings of the 11th International Conference on Ubiquitous Computing*, 2009, p. 93–96.

[4] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning, "Simultaneous localization and mapping using ambient magnetic field," in *2010 IEEE Conference on Multisensor Fusion and Integration*, Sep. 2010, pp. 14–19.

[5] P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. J. Julian, M. Garcia Puyol, M. Khider, M. Lichtenstern, and L. Bruno, "Simultaneous Localization and Mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments," in *International Conference on Indoor Positioning and Indoor Navigation*, Oct. 2013, pp. 1–10.

[6] M. Kok and A. Solin, "Scalable Magnetic Field SLAM in 3D Using Gaussian Process Maps," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 1353–1360.

[7] J. A. Shockley and J. F. Raquet, "Navigation of Ground Vehicles Using Magnetic Field Variations," *Navigation*, vol. 61, no. 4, pp. 237–252, 2014.

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

[9] A. Canciani and J. Raquet, "Airborne Magnetic Anomaly Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 67–80, Feb. 2017.

[10] J. Fang, H. Sun, J. Cao, X. Zhang, and Y. Tao, "A novel calibration method of magnetic compass based on ellipsoid fitting," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 6, pp. 2053–2061, 2011.

[11] B. Siebler, S. Sand, and U. D. Hanebeck, "Localization With Magnetic Field Distortions and Simultaneous Magnetometer Calibration," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3388–3397, Feb. 2021.

[12] B. Siebler, A. Lehner, S. Sand, and U. D. Hanebeck, "Simultaneous Localization and Calibration (SLAC) Methods for a Train-Mounted Magnetometer," *NAVIGATION: Journal of the Institute of Navigation*, vol. 70, no. 1, 2023.

[13] M. Kok and T. B. Schön, "Magnetometer Calibration Using Inertial Sensors," *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5679–5689, 2016.

[14] C. E. Rasmussen and H. Nickisch, "Gaussian Processes for Machine Learning (GPML) Toolbox," *Journal of Machine Learning Research*, vol. 11, no. 100, pp. 3011–3015, 2010.