Constrained Gaussian-Process State-Space Models for Online Magnetic-Field Estimation

Berntorp, Karl; Menner, Marcel

TR2023-075 July 04, 2023

Abstract

We address the magnetic-field simulteanous localization and mapping (SLAM) problem for global positioning. We leverage a previously-developed particle filter (PF)-based frame- work for online Bayesian inference and learning of Gaussian Process state-space models (GP-SSMs). We extend the framework to directly incorporate physical properties of the magnetic field in the GP formulation, by leveraging that magnetic fields under the absence of free currents are curl free. Because of its flexibility, the method can include any motion model that can be expressed by a general nonlinear function, with potential applications to, e.g., mobile robotics and pedestrian localization. Simulation results indicate that our method performs similar to recent batch methods for magnetic-field slam, while the computation times are feasible for online implementations.

International Conference on Information Fusion (FUSION) 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Constrained Gaussian-Process State-Space Models for Online Magnetic-Field Estimation

Karl Berntorp and Marcel Menner

***Abstract*—We address the magnetic-field simulteanous localization and mapping (SLAM) problem for global positioning. We leverage a previously-developed particle filter (PF)-based framework for online Bayesian inference and learning of Gaussian Process state-space models (GP-SSMs). We extend the framework to directly incorporate physical properties of the magnetic field in the GP formulation, by leveraging that magnetic fields under the absence of free currents are curl free. Because of its flexibility, the method can include any motion model that can be expressed by a general nonlinear function, with potential applications to, e.g., mobile robotics and pedestrian localization. Simulation results indicate that our method performs similar to recent batch methods for magnetic-field slam, while the computation times are feasible for online implementations.**

I. Introduction

The magnetic materials present in buildings cause anomalies in the ambient magnetic field. These anomalies are dominated by spatial variations, although temporal variations exist, for example, in the vicinty of mobile magnetic structures, such as elevators or robots [1], [2]. Because the spatial variations dominate over the temporal variations, using the ambient magnetic field for indoor positioning and navigation has been increasingly studied over the last decade, often, but not alywas, in a simultaneous localization and mapping (SLAM) setting [3]–[8]. Relying on the magnetic field for localization is convenient for several reasons. First, it can be measured by a magnetometer that is more or less present in any inertial measurement unit (IMU). Second, it does not need to rely on additional infrastructure or line-of-sight measurements [4].

Localization and SLAM using magnetic fields have been tackled using several different types of methods and using different sensor setups. It is common to complement the magnetometer measurements with odometry readings—for example, acceleration and gyroscope measurements from an IMU. Different examples of combining magnetometer readings with odometry are [6], which uses an odometry readings of the relative pose, [9], which uses odometry readings for pedestrian localization, and [10], which uses a foot-mounted IMU. Gaussian processes (GPs) have been proven potent for magnetic-field mapping. GPs are nonparametric modeling tools, which imply flexibility in the ability to model general nonlinear functions without a priori enforcing an explicit parametric structure [11]. The paper [12] models the magnetic field in a GP by incorporating physical knowledge from Maxwell's equations into the GP prior. Later, [4] extended [12] by rewriting the GP model in terms of a reduced-rank approximation [13] and performed subsequent sequential update of the magnetic field as new measurements arrive. Another follow-up work is [5], which combined the physical knowledge about the magnetic field in the GP prior and the reduced-rank formulation in a Rao-Blackwellized particle filter (RBPF) [14] for SLAM. The approach in [5] relies on an RBPF and is computationally tractable, as it represents the magnetic-field map as a linear combination of basis functions on hexagonal domains. Another related approach is presented in [6], which implements an extended Kalman filter (EKF) for slightly less accurate, but substantially faster, magnetic-field SLAM.

In previous work, we developed a flexible framework for online learning of GP state-space models (GP-SSMs) [15]. The framework is based on a variant of the particle filter (PF) that leverages a reduced-rank formulation [13], [16] of the GP-SSM, in which connections between GPs and a finite basis-function expansion of the unknown system dynamics is made by introducing certain priors on the basis-function coefficients. The method in [15] is highly accurate and shown to provide computation times suitable for real-time execution of low-dimensional problems such as tire-friction estimation for vehicle-control applications. In subsequent work, we showed that our framework can be used for online learning where the system to be learned has intrinsic constraints. In [17] we showed that we could handle symmetry, antisymmetry, and various types of boundary conditions.

In this paper, we further extend our method in [15], [17] and show that it is well suited to solve the magnetic-field SLAM problem. In current-free environments, the magnetic field is curl-free, which can be formulated as a linear operator constraint on the magnetic field. We show that our method readily handles such constraint by a transformation of the basis functions.

A. Related Work

Exploiting that magnetic fields are curl-free has been done before, also with GPs (e.g., [4]–[6], [12]). While [4] uses sequential updating of the magnetic field and a reduced-rank approximation for computational efficiency, it does not perform localization. The work [5] uses an IMU-based RBPF for SLAM and similar to us leverages the reduced-rank formulation [13] leading to a weighted basis-function expression of the magnetic field map. However, the RBPF in [5] is based on a state augmentation of the IMU state with the map state and tailored to the particular setting where the map does not evolve in time, whereas we leverage the general framework in [15], [17] where marginalization leads to analytic map updates

Mitsubishi Electric Research Laboratories (MERL), 02139 Cambridge, MA, USA. Email: `karl.o.berntorp@ieee.org`

that are different from the RBPF, and we introduce the notion of a forgetting factor to handle magnetic fields that have a slowly varying time component.

B. Notation

For a vector $\boldsymbol{x}$, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates that $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ and x_n denotes the nth component of $\boldsymbol{x}$. Matrices are indicated in capital bold font as $\boldsymbol{X}$ and the element on row i, column j of $\boldsymbol{X}$ is denoted with x_{ij}. The determinant of $\boldsymbol{X}$ is $|\boldsymbol{X}|$ and the trace of $\boldsymbol{X}$ is $\text{Tr}(\boldsymbol{X})$. With $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$, we mean the posterior density function of the state trajectory $\boldsymbol{x}_{0:k}$ from time step 0 to time step k given the measurement sequence $\boldsymbol{y}_{0:k} := \{\boldsymbol{y}_0, \dots, \boldsymbol{y}_k\}$, and $\boldsymbol{x}^i_{0:k}$ is the ith realization of $\boldsymbol{x}_{0:k}$. The notation $f \sim \mathcal{GP}(0, \kappa(x, x'))$ means that the function $f(x)$ is a realization from a GP prior with a given covariance function $\kappa(x, x')$ subject to some hyperparameters $\boldsymbol{\vartheta}$, and ∇ is the standard vector differential operator.

II. Modeling and Problem Formulation

We are interested in jointly estimating online the state $\boldsymbol{x} \in \mathbb{R}^{n_x}$ of a magnetometer and the magnetic field $\boldsymbol{h}(\boldsymbol{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ measured by the magnetometer, for example, included as part of an IMU, where the position $\boldsymbol{p}$ is included in $\boldsymbol{x}$. We consider the estimation model

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{w}_k, \tag{1a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{e}_k, \tag{1b}$$

where the (latent) state $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ at each time step k is implicitly observed through the measurement $\boldsymbol{y}_k \in \mathbb{R}^3$. The nonlinear function $\boldsymbol{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ describes the motion of the magnetometer mounting point and is a description capturing a wide range of scenarios, for example, robotics, pedestrian localization, and vehicles, both airborne and wheeled. For simplicity we have omitted an external (control) input $\boldsymbol{u}_k$ in (1a). The process noise $\boldsymbol{w}_k$ and measurement noise $\boldsymbol{e}_k$ are Gaussian distributed with covariance $\boldsymbol{Q}$ and $\boldsymbol{R} = \sigma^2 \boldsymbol{I}$ according to $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and $\boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$, respectively. Note that we can extend (1b) to also include nonmagnetic measurements. Similar to previous work [4]–[6], [12], we model the magnetic field $\boldsymbol{h}$ by a GP with prior $\boldsymbol{h}(\boldsymbol{p}) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\kappa}(\boldsymbol{p}, \boldsymbol{p}'))$. See [12] for a motivation of the choice of GPs. Furthermore, from Maxwell's equations, by assuming no free currents, the magnetic field $\boldsymbol{h}$ is curl-free [18], implying

$$\nabla \times \boldsymbol{h} = \boldsymbol{0}. \tag{2}$$

According to [4], (2) is valid in most indoor environments.

We assume that the state-transition model $\boldsymbol{f}$ is known, that is, we have a model of the motion of the magnetometer. This is reasonable and a standard assumption, since we have control of where the magnetometer is mounted. However, the magnetic field $\boldsymbol{h}$ is unknown, and the state $\boldsymbol{x}$ is only implicitly measured through (1b). We solve the joint state and magnetic-field estimation problem in a Bayesian setting, where the goal is to estimate the joint posterior

$$p(\boldsymbol{x}_{0:k}, \boldsymbol{h}|\boldsymbol{y}_{0:k}) \tag{3}$$

at each time step k under the constraint (2).

A. Approximate Formulation of a GP

The learning method in [15] expresses an unknown function $\boldsymbol{g} = [g_1 \ \cdots \ g_{n_g}]^\top$ by a basis-function expansion according to

$$\hat{g}_i(\boldsymbol{x}) = \sum_{j=1}^{M} \gamma_{ij} \phi_j(\boldsymbol{x}) \tag{4}$$

for each $i = 1, 2, \dots, n_g$, where the weights γ_{ij} are to be determined. The basis-function expansion (4) originally proposed in [13] leads to a joint inference and learning approach that is linear in the weights γ_{ij}. The eigenfunctions with associated eigenvalues λ_j are solutions to the Laplace operator over a domain. For our setting, over the rectangular domain $[-L_1, L_1] \times [-L_2, L_2] \times [-L_3, L_3] \in \mathbb{R}^3$, the solution is

$$\phi_{j_1,\dots,j_3} = \prod_{n=1}^{3} \frac{1}{\sqrt{L_n}} \sin\left(\frac{\pi j_n (x_n + L_n)}{2L_n}\right), \tag{5a}$$

$$\lambda_{j_1,\dots,j_3} = \sum_{n=1}^{n_x} \left(\frac{\pi j_n}{2L_n}\right)^2. \tag{5b}$$

The connection between a basis-function expansion (4) with basis functions (5a) and a GP is

$$\boldsymbol{g}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{x}')) \Leftrightarrow \boldsymbol{g}(\boldsymbol{x}) \approx \sum_{j=1}^{M} \boldsymbol{\gamma}_j \phi_j(\boldsymbol{x}), \tag{6}$$

for a function $\boldsymbol{g}$ with $\boldsymbol{\gamma}_j = [\gamma_{1j} \ \cdots \ \gamma_{3j}]^\top$ and

$$\gamma_{ij} \sim \mathcal{N}(0, \mathcal{S}(\lambda_j)), \tag{7}$$

where $\mathcal{S}$ is the spectral density of the GP covariance function $\boldsymbol{\kappa}$ [13].

The Earth's underlying magnetic field depends on the location on the Earth but can locally be modeled as constant, which also deviate from the Earth's magnetic field in indoor environments due to magnetic material in the structure of the building. The unknown mean can be modeled as one part of the covariance function $\boldsymbol{\kappa}$, $\kappa_\text{C} = \sigma_\text{C}^2 \boldsymbol{p}^\top \boldsymbol{p}'$ for each dimension. Because the magnetic field is smooth, we use the squared exponential covariance function to model the magnetic field variations,

$$\kappa_\text{SE}(\boldsymbol{p}, \boldsymbol{p}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{p} - \boldsymbol{p}'\|_2^2}{2\ell^2}\right), \tag{8}$$

with hyperparameters $\boldsymbol{\vartheta} = \{\sigma, \ell\}$, where for simplicity we assume the same hyperparameters for each dimension. Using the basis-function expansion, the total covariance function is approximated as

$$\kappa \approx \kappa_\text{C} + \sum_{j=1.}^{M} \mathcal{S}(\lambda_j) \phi_j(\boldsymbol{p}) \phi_j(\boldsymbol{p}'), \tag{9}$$

where the covariance (8) has the spectral density

$$\mathcal{S}(\omega) = \sigma^2 \sqrt{2\pi\ell^2} \exp\left(-\frac{\pi^2 \ell^2 \omega^2}{2}\right). \tag{10}$$

While setting $\boldsymbol{h} = \boldsymbol{g}$ would be possible, it would violate (2). With a matrix-multiplication interpretation, we can rewrite (2) as

$$\mathcal{H}\boldsymbol{h} = \boldsymbol{0}, \tag{11}$$

where

$$\mathcal{H} = \begin{bmatrix} 0 & -\frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & -\frac{\partial}{\partial x_1} \\ -\frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \end{bmatrix}.$$

Hence, if we choose $\boldsymbol{h} = \nabla g$ for a scalar basis-function expansion g,

$$\mathcal{H}\nabla g = \boldsymbol{0}, \tag{12}$$

that is, by the transformation $\boldsymbol{h} = \nabla g$, $\boldsymbol{h}$ is ensured to be curl free. In this paper, we choose g as in (4) with ϕ_j according to (5a) and with the connection (7), (10) being used to assign informed priors on the weights. With the basis-function expansion (4) and basis functions (5), a reduced-rank model of (1b) is

$$\boldsymbol{y}_k = \underbrace{\begin{bmatrix} \boldsymbol{c}^\top & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{c}^\top & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{c}^\top \end{bmatrix}}_{\boldsymbol{C}} \underbrace{\mathrm{vec}(\nabla\phi_1, \cdots, \nabla\phi_M)}_{\boldsymbol{\varphi}(\boldsymbol{x}_k)} + \boldsymbol{e}_k, \tag{13}$$

where $\boldsymbol{c} = [\gamma_{11}, \cdots, \gamma_{1m}]^\top$, $\boldsymbol{c}$ is Gaussian distributed with prior

$$\boldsymbol{c} \sim \mathcal{N}(\boldsymbol{c}|\boldsymbol{0}, \sigma^2\boldsymbol{V}) \tag{14}$$

and $\boldsymbol{V} \in \mathbb{R}^{M\times M}$ is a diagonal matrix with entries $\mathcal{S}(\lambda_j)$ [16]. Hence, the problem is to estimate $\boldsymbol{x}$ and $\boldsymbol{c}$ at each time step k using the estimation model

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{w}_k, \tag{15a}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{\varphi}(\boldsymbol{x}_k) + \boldsymbol{e}_k. \tag{15b}$$

III. Online Magnetic-Field SLAM

Our magnetic-field SLAM method considers approximate joint state inference and learning of the GP-SSM (1a) with unknown measurement equation (1b). Instead of targeting (1), the method learns (15), where we leverage the block-diagonal structure in $\boldsymbol{C}$, allowing a dimension reduction and instead estimating $\boldsymbol{c}$. This implies estimating the posterior distributions of $\boldsymbol{x}$ and $\boldsymbol{c}$ at each time step k. The problem cannot be resolved analytically, and the estimation of $\boldsymbol{x}$ and $\boldsymbol{c}$ is therefore based on a tailored PF for approximating $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ and $p(\boldsymbol{c}_k|\boldsymbol{y}_{0:k})$ recursively. Specifically, the method approximates the joint posterior density $p(\boldsymbol{x}_{0:k}, \boldsymbol{c}_k|\boldsymbol{y}_{0:k})$, from which marginal densities $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ and $p(\boldsymbol{c}_k|\boldsymbol{y}_{0:k})$ can be computed. We decompose the joint posterior as

$$p(\boldsymbol{x}_{0:k}, \boldsymbol{c}_k|\boldsymbol{y}_{0:k}) = p(\boldsymbol{c}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k})p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}). \tag{16}$$

The two densities on the right-hand side of (16) are estimated recursively. Given the state trajectory, we update the sufficient statistics of the unknown parameters. The second distribution on the right-hand side of (16) is approximated with the PF, while the first distribution on the right-hand side is updated analytically, for each particle.

A. *Estimating the Parameter Posterior*

The distribution of the basis-function weights $\boldsymbol{c}_k$ are computed conditioned on the realization of the state and measurement trajectories for each particle. Using Bayes' rule, $p(\boldsymbol{c}_k|\boldsymbol{x}_{0:k+1}, \boldsymbol{y}_{0:k})$ can be decomposed into a likelihood and prior as

$$p(\boldsymbol{C}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{c}_k, \boldsymbol{x}_{0:k})p(\boldsymbol{c}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k-1}). \tag{17}$$

The first term on the right-hand side of (17) is (13),

$$p(\boldsymbol{y}_k|\boldsymbol{c}_k, \boldsymbol{x}_{0:k}) = \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{C}_k\boldsymbol{\varphi}(\boldsymbol{x}_k), \boldsymbol{R}_k). \tag{18}$$

The magnetometer measurements are assumed mutually independent, and we proceed by updating the parameters of $\boldsymbol{c}$ iteratively, leading to updates of (17).

Theorem 1: Suppose that the initial prior $p(\boldsymbol{c}_0)$ is distributed according to $p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{c}_0|\boldsymbol{0}, \boldsymbol{V})$. Define $\boldsymbol{\Xi} := \boldsymbol{\Sigma}_{k|k} + \boldsymbol{V}^{-1}$. Then, for realizations $\boldsymbol{x}_{0:k}$ and $\boldsymbol{y}_{0:k}$, (17) can be computed at each time step k as

$$p(\boldsymbol{c}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k}) = \mathcal{N}(\boldsymbol{c}_k|\boldsymbol{\Psi}_{k|k}\boldsymbol{\Xi}^{-1}, \sigma^2\boldsymbol{\Xi}^{-1}), \tag{19}$$

where $\boldsymbol{\Psi}_{k|k}$ and $\boldsymbol{\Sigma}_{k|k}$ can be recursively updated as

$$\boldsymbol{\Psi}_{k|k} = \boldsymbol{\Psi}_{k|k-1} + \sum_{j=1}^{3} y_{j,k}\boldsymbol{\varphi}(\boldsymbol{x}_k)^\top, \tag{20a}$$

$$\boldsymbol{\Sigma}_{k|k} = \boldsymbol{\Sigma}_{k|k-1} + \boldsymbol{\varphi}(\boldsymbol{x}_k)\boldsymbol{\varphi}(\boldsymbol{x}_k)^\top. \tag{20b}$$

Proof 1: See [15] for the matrix-Normal case. Eqs. (19), (20) is achieved by collapsing the matrix-Normal case to a standard Gaussian distribution.

For constant parameters, the time-update step is $(*)_{k|k-1} = (*)_{k-1|k-1}$ for both quantities in (20). However, for PFs it can be problematic to estimate static parameters. Using the principle of exponential forgetting [19], for slowly time-varying parameters an approach is to write the time-update step of the predictive statistics as

$$\boldsymbol{\Psi}_{k|k-1} = \lambda\boldsymbol{\Psi}_{k-1|k-1}, \tag{21a}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \lambda\boldsymbol{\Sigma}_{k-1|k-1}. \tag{21b}$$

The forgetting factor $0 \leq \lambda \leq 1$ helps in the estimation of dynamic variables by forgetting older data as new data are accumulated in time. As a rule of thumb, for a forgetting factor λ the update of the statistics will rely on the last $1/(1-\lambda)$ time instants.

To find the posterior distribution of $\boldsymbol{c}_k$, we marginalize out the state trajectory according to

$$\begin{aligned} p(\boldsymbol{c}_k|\boldsymbol{y}_{0:k}) &= \int p(\boldsymbol{c}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k})p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})\,\mathrm{d}\boldsymbol{x}_{0:k} \\ &\approx \sum_{i=1}^{N} q_k^i p(\boldsymbol{c}_k|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k}), \end{aligned} \tag{22}$$

where $p(\boldsymbol{c}_k|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k})$ is given by (19) and q_k^i is the particle weight, see Sec. III-B.

Remark 1: The scalar real-valued number $\lambda \in [0, 1]$ provides exponential forgetting in $p(\boldsymbol{c}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k})$ that allows the algorithm to adapt to (slowly time-varying) changes in $\boldsymbol{c}$ over time. That is, our method handles magnetic fields that also have a slow temporal dependence. For $\lambda = 1$ we recover the case of static parameters, that is, a static magnetic field.

B. Particle Filtering for State Inference

PFs approximate the posterior density $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$ by a set of N weighted trajectories,

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}) \approx \sum_{i=1}^{N} q_k^i \delta_{\boldsymbol{x}_{0:k}^i}(\boldsymbol{x}_{0:k}), \tag{23}$$

where q_k^i is the importance weight of the ith state trajectory $\boldsymbol{x}_{0:k}^i$ and $\delta(\cdot)$ is the Dirac delta mass. The PF algorithm iterates between prediction and weight update, combined with a resampling step that removes particles with low weights and replaces them with more likely particles. The particle weights are typically updated as

$$q_k^i \propto q_{k-1}^i p(\boldsymbol{y}_k|\boldsymbol{x}_k^i). \tag{24}$$

C. Determining the GP Hyperparameters

In this section we discuss how to adjust the GP hyperparameters. For simplicity, we assume one set of range and hyperparameters for all three dimensions. Determining online all parameters associated with GPs is usually prohibitive. If there are data available a priori, it is possible to set the parameters based on offline learning methods—for example, using GP hyperparameter optimization [12] or offline particle Markov chain Monte-Carlo methods [16]. One approach is then to collect data, run an offline method to get a sensible hyperparameter set, and then execute the proposed online magnetic-field SLAM method based on the offline estimated parameters. Another possibility is to occasionally update the parameters online using a data batch. While fully recursive adaptation of the basis-function range L may be infeasible, it is possible to adapt the GP the length ℓ and scale σ online when L has been set. Intuitively, the GP hyperparameters adjust to the prior choice L, as long as it covers the feasible space.

To this end, let the GP hyperparameters evolve according to a random walk and assume independence between the different dimensions,

$$\boldsymbol{\vartheta}_k = \boldsymbol{\vartheta}_{k-1} + w_{\vartheta,k-1}, \tag{25}$$

where $w_{\vartheta,k} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_\vartheta)$, $\boldsymbol{Q}_\vartheta$ is a diagonal matrix. Instead of having $\boldsymbol{V}$ in (14) fixed, we now let it be updated according to the sampled GP hyperparameters. The additional steps involved when including GP hyperparameter estimation amount to; (i) sample $\{\boldsymbol{\vartheta}^i\}_{i=1}^N$ from the prior (25); (ii) update the covariance prior $\{\boldsymbol{V}_k^i\}_{i=1}^N$ in (14) according to (10). The updated $\{\boldsymbol{V}_k^i\}_{i=1}^N$ affect the update of $\boldsymbol{c}$ by Theorem 1. The additional steps for the GP hyperparameter updates imply an estimation of the joint posterior density $p(\boldsymbol{x}_{0:k+1}, \boldsymbol{\vartheta}_{0:k}, \boldsymbol{c}_k|\boldsymbol{y}_{0:k})$. However, in the end we are interested in the marginal densities $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ and $p(\boldsymbol{c}_k|\boldsymbol{y}_{0:k})$. The standard PF implementation marginalizes out $\boldsymbol{x}_{0:k-1}$ by discarding it, which leads to an $\mathcal{O}(N)$ implementation. This approximation relies on sufficient mixing properties in the dynamic model to avoid the depletion problem, essentially meaning that errors in the state are forgotten as time progresses, which is the reason why the $\mathcal{O}(N)$ implementation of the PF works in many realistic scenarios. In addition, the use of exponential forgetting by introducing λ helps in this regard [20]. However, when adapting the GP hyperparameters the model used to generate the state samples will differ at each time step, which increases the risk of particle depletion. Especially in high signal-to-noise ratios, it may be important to account for different paths, which leads to the modified weight update

$$q_k^i \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k^i) \sum_{j=1}^{N} q_{k-1}^j p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^j). \tag{26}$$

Eq. (26) results in an $\mathcal{O}(N^2)$ method, but it is possible to implement (26) with $\mathcal{O}(N \log N)$ complexity and even linearly (e.g., using accept-reject sampling [21]).

D. Algorithm Summary

Algorithm 1 summarizes the proposed SLAM method. In the proposed method, each particle i retains its own set of $\boldsymbol{\Psi}^i, \boldsymbol{\Sigma}^i$, which forms the sufficient statistics to describe the magnetic field.

Algorithm 1 Pseudo-code Magnetic-Field SLAM

```
Initialize: Set $\{\boldsymbol{x}_0^i\}_{i=1}^N \sim p_0(\boldsymbol{x}_0)$, $\{q_{-1}^i\}_{i=1}^N = 1/N$,
    $\{\boldsymbol{\Psi}^i, \boldsymbol{\Sigma}^i\}_{i=1}^N = \{\boldsymbol{0}, \boldsymbol{0}\}$.
for $k = 0, 1, \ldots$ do
    for $i \in \{1, \ldots, N\}$ do
        if GP hyperparameter estimation then
            Update weight $\bar{q}_k^i$ using (24).
        else
            Update weight $\bar{q}_k^i$ using (26).
        end if
        Determine $\boldsymbol{\Psi}_{k|k-1}^i, \boldsymbol{\Sigma}_{k|k-1}^i$ using (21).
    end for
    Normalize weights as $q_k^i = \bar{q}_k^i / (\sum_{i=1}^N \bar{q}_k^i)$.
    Compute $N_{\text{eff}} = 1/(\sum_{i=1}^N (q_k^i)^2)$.
    if $N_{\text{eff}} \leq N_{\text{thr}}$ then
        Resample particles and copy the corresponding
        statistics. Set $\{q_k^i\}_{i=1}^N = 1/N$.
    end if
    for $i \in \{1, \ldots, N\}$ do
        if GP hyperparameter estimation then
            Sample $\boldsymbol{\vartheta}_k$ from (25).
        end if
        Sample $\boldsymbol{x}_{k+1}^i$ from a proposal distribution.
        Determine $\boldsymbol{\Psi}_{k|k}^i, \boldsymbol{\Sigma}_{k|k}^i$ using (20).
    end for
end for
```

Remark 2: In this work we use the basis functions (5b), which are coupled in all three dimensions. However, for many positioning applications the movement in the horizontal plane dominates over the movement in the vertical plane, which enables a decoupling between the different dimensions. This can be done in numerous ways, for example, according to the decoupling in [5]. We have not considered decoupling in this work but it is conceptually straightforward to include.

IV. Results

In this section we evaluate our magnetic-field SLAM method using synthetic data and compare with our base method proposed in [15], as well as comparing with a GP formulation. The scenario we consider is the example used in [12] tailored to the recursive setting. Consider a sphere centered at the origin with a radius of $r_0 = 3$m having a uniform magnetization of $\boldsymbol{m} = [0, 1, 0]^\top$ A/m. Then, the magnetic field $\boldsymbol{h}$ in (15b) is described by

$$\boldsymbol{h} = \begin{cases} -\boldsymbol{m}/3 & \text{if } r < r_0 \\ \frac{m_0}{4\pi}\left(\boldsymbol{m}/r^3 + 3/r^5(\boldsymbol{m}^\top \boldsymbol{p})\boldsymbol{p}\right) & \text{if } r \geq r_0 \end{cases} \tag{27}$$

where $m_0 = 4/3\pi r_0^3$ and $r = \|\boldsymbol{p}\|$. We consider a robot described by a nearly-constant velocity model moving in three-dimensional space

$$\dot{\boldsymbol{p}} = \boldsymbol{v} + \boldsymbol{w}, \tag{28}$$

where the velocity $\boldsymbol{v}$ is controlled by a state-feedback controller

$$\boldsymbol{v} = -\boldsymbol{L}(\boldsymbol{p}_r - \boldsymbol{p}), \tag{29}$$

where $\boldsymbol{L}$ is the feedback gain matrix and $\boldsymbol{p}_r$ is the reference path. The process noise describing model uncertainties is assumed zero-mean Gaussian distributed with covariance matrix $\boldsymbol{Q}_w$ according to $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_w)$. We discretize the system (28), (29) with a sampling rate $T_s = 1$s and execute the system for 40s. We set $L = 20$ in (5), use $M = 8^3 = 512$ basis functions (i.e., 8 in each dimension), and set $\ell = 5$, $\sigma_f = 1$ in (10). The particle filter uses $N = 50$ particles and we set the forgetting factor to $\lambda = 0.99$. We initialize Algorithm 1 at each of the $N_{\text{MC}} = 100$ Monte-Carlo runs around the true intial state with 1m standard deviation in each dimension. We use the root mean-square error (RMSE) at each time step as the measure of performance. The position RMSE at time step k is computed as

$$\text{RMSE}_k = \sqrt{\frac{1}{N_{\text{MC}}} \sum_{j=1}^{N_{\text{MC}}} \|\boldsymbol{p}_{k,j} - \hat{\boldsymbol{p}}_{k,j}\|^2}. \tag{30}$$

To get a sense of the generated trajectory used in the SLAM problem, Fig. 1 shows the true path and generated particle cloud for one of the realizations. The robot moves outside the sphere with radius r_0 at all times, with most of the movement in the XY-plane, which is consistent with many indoor positioning applications, for example, for mobile robotics and pedestrians. Except for the initial time step, the estimates are close to the true values. To get a measure of the performance of Algorithm 1, Fig. 2 displays the state RMSE for Algorithm 1 and for the unconstrained case, that is, applying the method in [15] directly. The proposed method suppresses the initial transient faster, but in steady state the difference in RMSE is small.

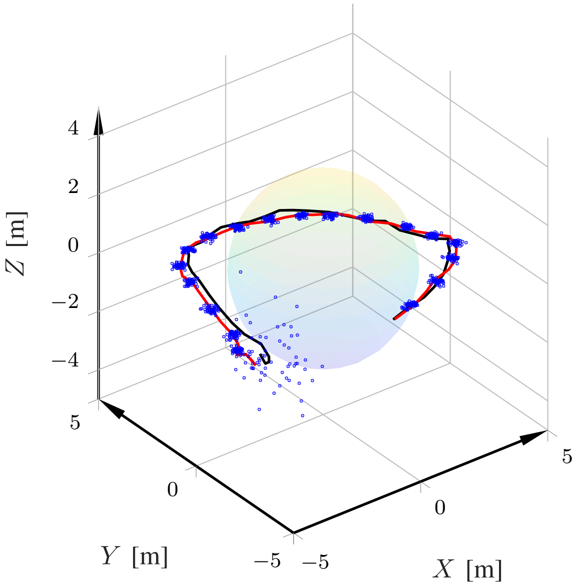

Fig. 1. True path (black) and estimated path (red) for one realization, as well as the particle cloud (blue) at every other time step.

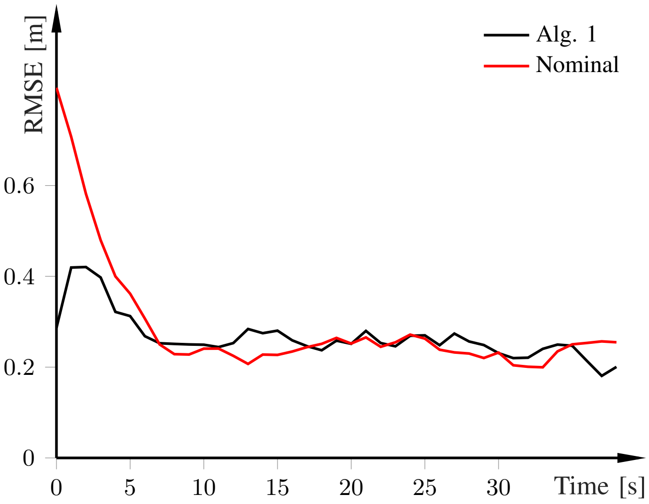

Fig. 2. State RMSE computed as (30) of proposed (black) and unconstrained case (red) for 100 Monte-Carlo runs.

The benefits with Algorithm 1 become more clear when investigating Fig. 3, which shows the true magnetic field overlaid with the estimated field and the training points. While the estimates for the unconstrained case seem reasonable when studying them very close to the observed points, the results do not generalize. For instance, inside the sphere or around any of the corners, which the robot has not traveled, the proposed method provides estimates that are of similar direction and magnitude as the true field. In contrast, not including the physical properties yields decent results in the close vicinity of the traversed points, but the magnetic-field estimates are wrong in both magnitude and direction for most of the unobserved points. As a further illustration, Fig. 4 shows the error fields for the two cases for the same realization as in Fig. 3. The proposed method (left plot) displays smaller error

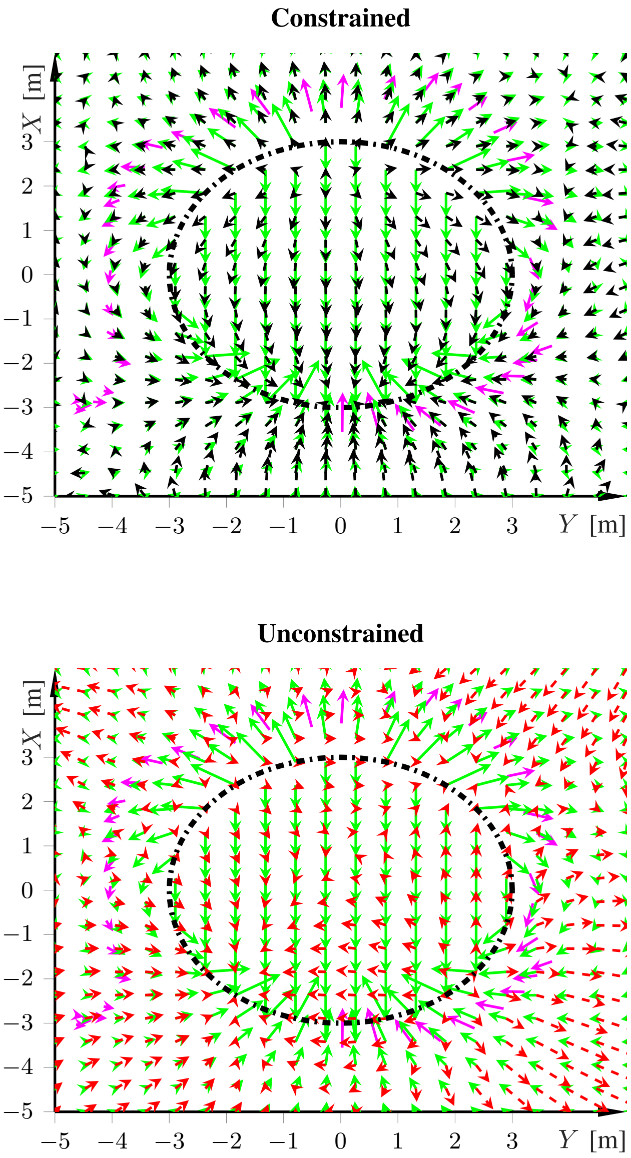

Fig. 3. True magnetic field in green, our proposed Algorithm 1 in black (top plot), without including constraints in red (lower plot), and the observed points in magenta. Note that the observed points are increased in magnitude for clarity. The results are for one typical realization.

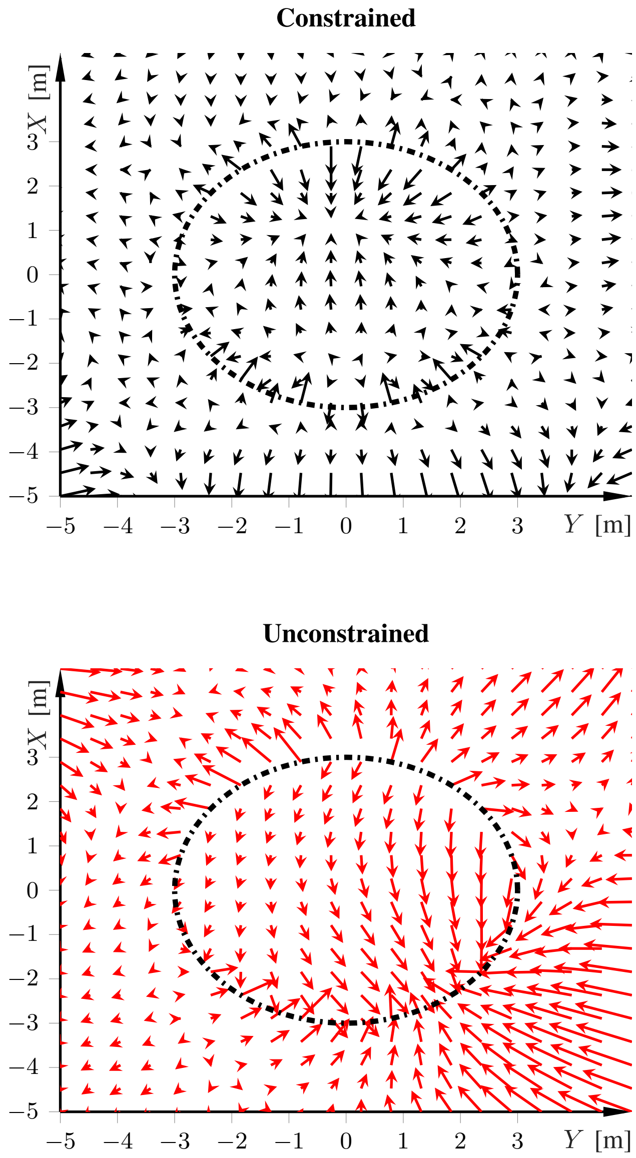

Fig. 4. Error fields of our proposed Algorithm 1 in black (top plot), unconstrained learning in red (lower plot). The results are for the same realization as in Fig. 3.

fields throughout and particularly in unexplored areas.

To quantify the performance increase by the proposed approach and see how the method estimates the magnetic field in areas the robot has not traveled, Fig. 5 shows the RMSE of the magnetic field as a function of time over an equidistant grid in the state space. The RMSE is computed by gridding the XY-plane equidistantly with an interval of 0.5m, resulting in $N_t = 400$ evaluation points, and evaluating

$$\text{RMSE}_k = \frac{1}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} \sqrt{\frac{1}{N_t} \sum_{j=1}^{N_t} \|\boldsymbol{h}(\boldsymbol{p}_j) - \hat{\boldsymbol{h}}_{k,i}(\boldsymbol{p}_j)\|^2}. \quad (31)$$

The proposed method in black enjoys a reduction in RMSE with roughly 50%. The performance after 1 time step is significantly better than the end point when not incorporating the constraint. Table I summarizes the results as time-averaged RMSEs. The conclusion is that while our base formulation [15] exhibits good positioning performance and good (local) mapping, Algorithm 1 improves the map also in locations it has not observed, implying it needs much fewer data points to create a sensible magnetic-field map. The RMSE values are very similar to the results reported in [12] where a batch method was used on the same magnetic field, although there are some differences in the setup so the results are not exactly comparable.

V. Conclusion

In this paper we extended our recently developed recursive Bayesian inference and learning method (see [15]) to

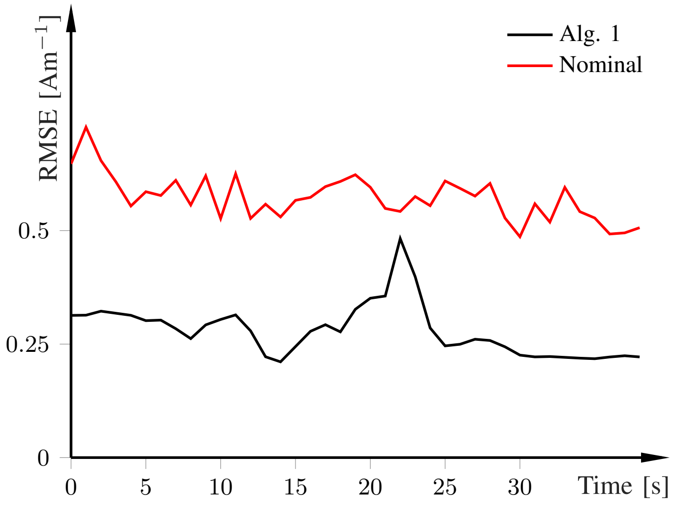

Fig. 5. RMSE computed as (31) of the magnetic field as a function of time for the proposed (black) and unconstrained case (red) over 100 Monte-Carlo Runs.

TABLE I
TIME-AVERAGED RMSES FOR ALGORITHM 1 AND WHEN NOT INCLUDING THE CURL-FREE PROPERTY INTO THE BASIS FUNCTIONS.

Measure	Algorithm 1	[15]
State RMSE (30)	0.26	0.29
Map RMSE (31)	0.31	0.59

magnetic-field SLAM. The method is based on PFs and a reduced-rank formulation of GPs, in which we leveraged the curl-free property of magnetic fields to determine basis functions such that the constraints are inherently satisfied in the estimation formulation. In the formulation, a finite basis-function expansion of the GP is used and by leveraging the marginalization concept in PFs, the method is well suited for real-time estimation. While similar methods for magnetic-field mapping have been presented before, our proposed method is part of a general framework which, together with the conclusions in [17], can handle a large number of system constraints applicable to a wide range of applications . In a simulation study we showed that while the positioning performance is not notably increased compared to the nominal formulation in [15], its ability to learn maps in areas it has not already observed makes a large difference on mapping performance.

There are several ways to extend the method. The method currently relies on modeling the magnetic field using a single rectangular grid. While this is sensible for single rooms, it will become intractable for larger buildings. Here, various ways to combine segments of grids is something that has been explored (e.g., [5]) but further work is possible. In the future we plan to apply the method to mobile robotics applications related to legged robots and drones for indoor localization and control (see, e.g., [22]).

REFERENCES

[1] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos, “How feasible is the use of magnetic field alone for indoor positioning?” in *Int. Conf. Indoor Positioning and Indoor Navigation*, 2012.

[2] M. Angermann, M. Frassl, M. Doniec, B. J. Julian, and P. Robertson, “Characterization of the indoor magnetic field for applications in localization and mapping,” in *Int. Conf. Indoor Positioning and Indoor Navigation*, 2012.

[3] J. Haverinen and A. Kemppainen, “Global indoor self-localization based on the ambient magnetic field,” *Robotics Auton. Syst.*, vol. 57, pp. 1028–1035, 2009.

[4] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Sarkka, “Modeling and interpolation of the ambient magnetic field by Gaussian processes,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1112–1127, Aug. 2018.

[5] M. Kok and A. Solin, “Scalable magnetic field SLAM in 3D using Gaussian process maps,” in *Int. Conf. Information Fusion*, Cambrdige, UK, Jul. 2018.

[6] F. Viset, R. Helmons, and M. Kok, “An extended Kalman filter for magnetic field SLAM using Gaussian process regression,” *Sensors*, vol. 22, no. 8, 2022.

[7] A. Solin, S. Särkkä, J. Kannala, and E. Rahtu, “Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning,” in *Eur. navigation Conf.*, 2016.

[8] E. Le Grand and S. Thrun, “3-axis magnetic field mapping and fusion for indoor localization,” in *Int. Conf. Multisensor Fusion and Integration for Intell. Syst.*, 2012.

[9] 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 *Int. Conf. Indoor Positioning and Navigation*, 2013.

[10] M. Osman, F. Viset, and M. Kok, “Indoor SLAM using a foot-mounted IMU and the local magnetic field,” in *Int. Conf. Information Fusion*, 2022.

[11] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[12] N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson, “Modeling magnetic fields using Gaussian processes,” in *Int. Conf. Acoustics, Speech, and Signal Process.*, 2013.

[13] A. Solin and S. Särkkä, “Hilbert space methods for reduced-rank Gaussian process regression,” *Statistics and Computing*, vol. 30, no. 2, pp. 419–446, 2020.

[14] T. B. Schön, F. Gustafsson, and P.-J. Nordlund, “Marginalized particle filters for mixed linear nonlinear state-space models,” *IEEE Trans. Signal Process.*, vol. 53, pp. 2279–2289, 2005.

[15] K. Berntorp, “Online Bayesian inference and learning of Gaussian-process state-space models,” *Automatica*, vol. 129, p. 109613, 2021.

[16] A. Svensson and T. B. Schön, “A flexible state-space model for learning nonlinear dynamical systems,” *Automatica*, vol. 80, pp. 189–199, 2017.

[17] K. Berntorp and M. Menner, “Online constrained Bayesian inference and learning of Gussian-process state-space models,” in *Amer. Control Conf.*, Atlanta, GA, Jun. 2022.

[18] D. J. Griffiths, *Introduction to Electrodynamics*. Cambridge University Press, 2017.

[19] K. J. Åström and B. Wittenmark, *Adaptive Control (2 rev. Dover ed.)*. Dover Publications, 2008.

[20] E. Özkan, V. Šmídl, S. Saha, C. Lundquist, and F. Gustafsson, “Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters,” *Automatica*, vol. 49, no. 6, pp. 1566–1575, 2013.

[21] F. Gustafsson, “On marginal particle filters with linear complexity,” in *Computational Adv. Multi-Sensor Adaptive Process.*, 2013.

[22] A. Schperberg, S. Di Cairano, and M. Menner, “Auto-tuning of controller and online trajectory planner for legged robots,” *IEEE Robot. Automation Letters*, vol. 7, no. 3, pp. 7802–7809, 2022.