Schlupkothen *et al. EURASIP Journal on Advances in Signal Processing* (2020) 2020:5
https://doi.org/10.1186/s13634-019-0657-x

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Random field-aided tracking of autonomous kinetically passive wireless agents

Stephan Schlupkothen*, Tim Heidenblut and Gerd Ascheid

Abstract

Continuous miniaturization of circuitry has open the door for various novel application scenarios of millimeter-sized wireless agents such as for the exploration of difficult-to-access fluid environments. In this context, agents are envisioned to be employed, e.g., for pipeline inspection or groundwater analysis. In either case, the demand for miniature sensors is incompatible with propulsion capabilities. Consequently, the agents are condemned to be kinetically passive and are, thus, subject to the fluid dynamics present in the environment. In these situations, the localization is complicated by the fact that unknown external forces (e.g., from the fluid) govern the motion of the agents. In this work, a comprehensive framework is presented that targets the simultaneous estimation of the external forces stemming from the fluid and the agents' positions which are traversing the environment. More precisely, a Bayesian hierarchical model is proposed that models' relevant characteristics of the fluid via a spatial random field and incorporates this as control input into the motion model. The random field model facilitates the consideration of spatial correlation among the agents' trajectories and, thereby, improves the localization significantly. Additionally, this is combined with multiple particle filtering to account for the fact that within such underground fluid environments, only a localization based on distance and/or bearing measurements is feasible. In the results provided in this work, which are based on realistic computational fluid dynamics simulations, it is shown that—via the proposed spatial model—significant improvements in terms of localization accuracy can be achieved.

Keywords: Wireless sensor networks, Random field, Tracking, Multiple particle filtering

1 Introduction

Technological advances played a pivotal role in leveraging the use of miniature wireless agents for novel application cases. Among these are for example scenarios where millimeter-sized agents are employed for pipeline inspection [1–3] or the exploration of difficult-to-access environments [4]. In the former case, agents are deployed to monitor the physical state of the piping system, i.e., to analyze the pipes for physical damages. Moreover, the agents facilitate the inspection for fluid residuals which eventually could lead to reduced throughput. Additional application cases include for example the use in underground scenario such as sewage networks or groundwater systems. In the latter scenario, agents could play a key role in the analysis of water pollution and, hence, in water safety. In either of the scenarios, it is assumed that due to energy limitations resulting from constraints on the agents' physical size, only very limited communication among the agents is possible. More precisely, it is assumed that only pair-wise distance and/or bearing measurements are feasible, necessitating the use of centralized localization schemes in a fusion center (FC).

However, the use of miniature agents in these situations is complicated by the following facts. First, as mentioned above, the agents need to be small to ensure that they spread sufficiently and to avoid that they get stuck in the environment. Second, these size-constraints impose energy and processing limitations on the agents that, among other effects, condemn the agents to be kinetically passive. Consequently, the agents' motion is fully governed by the fluid. Third, the system under investigation may be partially or fully unknown. This

*Correspondence: schlupkothen@ice.rwth-aachen.de
Chair for Integrated Signal Processing Systems, RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany

is particularly likely for underground systems. Consequently, the effect of the fluid on the agents' kinetics is either difficult to predict or unknown. Fourth, due to the deployment in underground or isolated environments, a localization[1], e.g., via global positioning system (GPS) is not possible. For this reason, the agents are equipped with ultrasonic transceivers that facilitate agent-to-agent measurements (AAMs) and agent-to-beacon measurements (ABMs). Fifth and finally, in the mentioned scenarios, the deployment of beacons[2] is costly and/or difficult, as the environment is unknown and, thus, proper locations for the beacons cannot be determined a priori.

In this work, attention is drawn to issues that result from the abovementioned constraints. More precisely, a framework is presented that addresses these challenges as follows: A statistical model is used to describe parameters of the environment that are pivotal to the tracking of the agents. The model comprises a spatial time-invariant random field (RF) which is approximated as a Gaussian Markov random field (GMRF) with generally non-zero-mean and covariance components. This model is used to enhance the motion model of the agents by considering the RF as artificial control input (ACI). More specifically, the classical motion models employed (cf. [5]), e.g., for kinetically autonomous agent $\mathbb{i}$, $\boldsymbol{x}_{\mathbb{i},k+1} = \boldsymbol{f}(\boldsymbol{x}_{\mathbb{i},k}, \boldsymbol{\theta}, \boldsymbol{v})$, where $\boldsymbol{v}$ is the process noise, is augmented as $\boldsymbol{x}_{\mathbb{i},k+1} = \boldsymbol{f}(\boldsymbol{x}_{\mathbb{i},k}, u_{\mathbb{i},k}, \boldsymbol{\theta}, \boldsymbol{v})$. The ACI $u_{\mathbb{i},k}$ takes the role of the control input (CI) in the case of kinetically active agents, with the only difference that the former is unknown and, hence, needs to be estimated. This modeling is additionally complicated by the fact that the agents are operating in spatially confined areas where boundary effects are relevant, considering these effects are pivotal for improving the localization accuracy through the abovementioned RF model. To this end, corresponding adaptations to classical GMRF models are adopted. Moreover, to facilitate efficient tracking of multiple agents, multiple particle filtering is adopted, which employs a separate particle filter (PF) for each agent. In summary, a comprehensive framework is proposed in this work that reduces the GMRF estimation to a parameter estimation problem via efficient parametrization. Additionally, it utilizes novel multiple particle filter (MPF) (cf. [6]) schemes for state estimation. Consequently, the resulting problem tackled in this work is a joint state and parameter estimation problem.

1.1 Related works

Several application cases have been presented where uncertainty regarding the environment and the locations of the agents need to be addressed. For example, in simultaneous localization and mapping (SLAM) context, in [7], a spatial GMRF is estimated that is simplified through the availability of direct field measurements. The objective is to simultaneously localize a robot and estimate the RF. Compared to our scenario, a simpler scenario is considered as noisy localization data and direct field measurements are assumed to be readily available. Moreover, no coupling between the motion of the robot and the field needs to be considered as the robot is assumed to be kinetically active. Similarly, in [8], a wireless sensor network (WSN) with known positions is considered which aim to estimate the parameters of a GMRF via direct field measurements. In [9–11], a stationary WSN is considered which estimates a RF using Gaussian process regression and direct field measurements. Noisy positions of the agents are assumed to be available. In [12], an underwater robot is considered whose motion is affected by the water. The localization is based on a PF, where direct available flow measurements are compared to an a priori available velocity field model. Due to the fact that the robot is kinetically active, a field model is available prior to deployment and because also field measurements are readily available, several assumptions of our scenario are defied. Additionally, in our previous conference work [13], a first step towards the consideration of the fluid's effects on the agents' motion has been taken. More precisely, a multivariate Gaussian ACI model is used to describe additional changes regarding the agents' speed and turn rate. The model is fixed and chosen prior to localization. Since this previous work targets the very same application case, also distance and/or bearing based localization is considered.

In [14], a cooperative scheme for decentralized localization is presented. It is assumed that the agents can communicate additional information besides, e.g., distance measurements. This additional information is inevitable because of the decentralized localization procedure. Consequently, due to the decentralized localization scheme, agents with significantly higher battery capacity are required which are unavailable for the application scenario considered in this work due to the hardware constraints mentioned in Section 1. Moreover, in [14], no effects of the environment on the agents are considered which presents the main contribution of this work.

In summary, none of the available schemes considers all requirements set by our application case. Most importantly, in all but [12]'s scenario and our previous conference work, coupling between the field and the motion is neglected. Moreover, most of the works assume direct field measurements which significantly simplifies the field inference. For a brief summary of the most related works, see Table 1.

1.2 Contribution

Motivated by the performance achieved through abstract and position-independent models shown in our previous

[1] In this work, we may use the words localization and tracking interchangeably.
[2] In this work, beacons denote stationary agents whose absolute position is known a priori and which aid in localizing the ordinary mobile agents.

Table 1 Overview of important related works

Ref.	Estimates field	Requires field measurements	Estimates positions	Means of localization	Kinetic control	Considers barriers	Kinetic coupling[‖]
[7]	✓ (GMRF)	✓	✓[‡]	—[‡]	✓ / active[§]	✗	✗
[8]	✓ (GMRF)	✓	✗	Positions are known	✓ / active	✗	✗
[12]	✗	✓	✓	Fluid flow measurements[¶]	✓ / active	✗	✓
[14]	✗	✗	✓[††]	Distance/bearing AAMs and ABMs	✓ / active	✗	✗
[13]	✗[*]	✗	✓	Distance/bearing AAMs and ABMs	✗ / passive	✗	✓
This work	✓ (GMRF)	✗[†]	✓	Distance/bearing AAMs and ABMs	✗ / passive	✓	✓

[¶] Magnitude and angle of the flow is directly measured and compared to an a priori known oceanographic flow model

[†] No direct field measurements required. Inference indirectly via localization, using distance and/or bearing measurements

[††] Distributed localization is considered which is considered sub-optimal in the scenario considered in this work due to the availability of all data in the FC

[‡] AWGN corrupted positions estimates are assumed

[§] Affine dynamics are considered

[*] Position-independent RVs whose parameters are fixed a priori are used instead of spatial field

[‖] Refers to the case where the environment's effect on the agents' motion is considered work [13], in this work, new extensions to the input modeling are proposed. More precisely, a RF model is used to directly model the ACI density on a spatial level and under consideration of boundary effects which are present at the border of the environment. The RF is estimated without additional measurements, i.e., is inferred indirectly via distance and/or bearing measurements and the localization which is performed using these measurements. Consequently, no additional complexity is added to the resource-limited agents as all processing takes place in the FC (centralized localization).

To facilitate the inference of the RF, a computationally efficient GMRF model with few hyper-parameters is used to model the spatial correlation. Moreover, to account for the fact that no direct field measurements are assumed, the standard GMRF model is extended by a mean component which is estimated simultaneously.

In summary, a joint framework is presented in which the parameters of the GMRF as well as the positions of the agents are estimated. This is achieved also through the use sequential Monte Carlo squared (SMC^2) that has been extended to operate with novel MPF schemes. Thereby, a spatial model is built which naturally extends the abstract model proposed in [13] and considers spatial correlation in the environment and, thus, the agents' trajectories to improve the localization. Consequently, the following list of contributions of this work can be given.

1) Extension of the ACI scheme presented in [13] via a GMRF for spatial modeling. Through this, also the spatial correlation among agents in the environment can be exploited efficiently for improved localization accuracy.
2) Derivation of a corresponding Bayesian hierarchical model (BHM) that describes the coupling of the modeled field with the motion of the agents
3) Formalization of a joint state and parameter estimation problem that comprehensively couples the localization with the field inference problem.
4) Adaptions to the SMC^2 framework, which is used to solve the joint state and parameter estimation problem, to leverage MPF and through this tackles the high-dimensional state space and alleviates the "curse of dimensionality."
5) Efficacy improvements for low-complexity time-updating which is relevant for the sequential Monte Carlo (SMC) steps.

1.3 Organization

This work is organized as follows. In Section 2, a summary of relevant background information is given. This covers for example multiple particle filtering which is employed for state estimation. Moreover, methods for combined state and parameter estimation are discussed, and special attention is drawn to a method which is known as SMC^2. Additionally, RFs in general and GMRFs in particular are introduced and corresponding parameterizations are discussed. In Section 3, the modeling approach developed in this work is presented which builds upon the GMRFs. Section 5 outlines the link between this model and the ACI. In Section 6, the time update model for the SMC-based estimation and inference is derived which is followed by the final proposed algorithm in Section 7. Sections 8 and 9 introduce the simulation setup as well as discusses numerical results. Final conclusions are drawn in Section 10.

2 Background

As briefly outlined in Section 1, this paper is concerned with the localization of wireless agents that is improved through the use of an environment model and its direct embedding into the localization framework. This model is used to represent relevant kinetic quantities that directly affect each agent's motion. Moreover, distance and/or bearing measurement needs to be considered because of the operation in GPS-denied areas. Since both types of measurements are inherently nonlinear and because realistic motion in such environments is best described using curvilinear models, particle filtering is applied in this work for state estimation. Subsequently, a brief introduction to particle filtering, multiple particle filtering, and combined state and parameter estimation, to the extent required for this work, is given.

2.1 State space models

Considered henceforth is the following state space model (SSM)

$$\boldsymbol{x}_{\mathbb{i},k+1} = \boldsymbol{f}(\boldsymbol{x}_{\mathbb{i},k}, u_{k,\mathbb{i}}, \boldsymbol{\theta}, \boldsymbol{v}_k), \tag{1a}$$

$$\boldsymbol{y}_{\mathbb{i},k} = \boldsymbol{h}(\boldsymbol{x}_{\mathbb{i},k}, \mathcal{Z}_{-\mathbb{i},k}, \boldsymbol{\eta}_k), \tag{1b}$$

where $\boldsymbol{x}_{\mathbb{i},k} \in \mathbb{R}^{n_x}$ represents the state of agent $\mathbb{i}$ at discrete time k and $\boldsymbol{y}_{\mathbb{i},k} \in \mathbb{R}^{n_y}$ represents the corresponding measurements of this agent, each of which, by the nature of AAMs, also depends on the state vectors of the other agents involved in these measurements, $\mathcal{Z}_{-\mathbb{i},k}$. More precisely, $\mathcal{Z}_{-\mathbb{i},k}$ denotes the collection of state vectors of all agents but $\mathbb{i}$. The state evolution parameters $u_{k,\mathbb{i}}, \boldsymbol{\theta}$ denote respectively the control input and a set of deterministic parameters. Moreover, $\boldsymbol{v}_k$ and $\boldsymbol{\eta}_k$ denote the process noise and measurement noise, respectively. The generally nonlinear functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ denote the state evolution and measurement models, respectively, with noise included.

Equivalently, (1) can be written as

$$\boldsymbol{x}_{\mathbb{i},k} \sim p(\boldsymbol{x}_{\mathbb{i},k} | \boldsymbol{x}_{\mathbb{i},k-1}, u_{k,\mathbb{i}}, \boldsymbol{\theta}), \tag{2a}$$

$$\boldsymbol{y}_{\mathbb{i},k} \sim p(\boldsymbol{y}_{\mathbb{i},k} | \boldsymbol{x}_{\mathbb{i},k}, \mathcal{Z}_{-\mathbb{i},k}). \tag{2b}$$

where we used the short-hand notation $p(a|b)$ to denote the probability density function (PDF) $p_{A|B}(A = a|B = b)$. Under the assumption that $u_{k,\mathbb{i}}$ and $\boldsymbol{\theta}$ are known or not relevant, classical particle filtering can be employed since tracking the agents through time is a state estimation problem, cf. Section 2.2 and Appendix A.1.

Remark 1 *Note that, as mentioned in Section 1, kinetically passive agents are considered, i.e., $u_{k,\mathbb{i}} \equiv 0$ strictly speaking. However, in the course of this work, a scheme is presented which aims to resemble external forces originating from the fluid that are modeled as ACI $u_{k,\mathbb{i}}$.*

The more general case where the parameters $\boldsymbol{\theta}$ are unknown and, hence, need to be estimated is addressed in Appendix A.2. In this work, $\boldsymbol{\theta}$ describes parameters of the GMRF which, in turn, is used to model the external forces. Background information on RFs in general and GMRFs in particular, as well as its underlying finite element method (FEM) description is given in Appendix A.4.

2.2 Multiple particle filtering

Classical (single) PFs (cf. Appendix A.1) are known to suffer from the "curse of dimensionality," i.e., the fact that an exponentially increasing number of particles is required to accurately capture the posterior distribution as more agents need to be tracked [15]. This is because the dimensionality of the state space grows in proportion to the number of agents. To this end, the concept of multiple particle filtering, in which one PF is employed for each agent individually, has been proposed. However, its application to our setup gives rise to a problem denoted as likelihood approximation problem (LAP). The problem is due to the fact that the utilization of AAMs is required for accurate localization and the fact that in MPF, the individual particles of each agent are processed by their individual PF. This introduces dependencies among the PFs since for each PF, such as the one for agents $\mathbb{i}$, the likelihood $p(\boldsymbol{y}_{\mathbb{i},k}|\boldsymbol{x}_{\mathbb{i},k})$ is required (cf. (50)), while only $p(y_{\mathbb{i},k}|\boldsymbol{x}_{\mathbb{i},k}, \boldsymbol{x}_{\mathbb{j},k})$ is available. This is due to the fact that distance and bearing measurements are relative measurements between two agents such as $\mathbb{i}$ and $\mathbb{j}$ and, thus, the measurements are dependent on the state of both agents. To nevertheless employ MPF, the likelihood approximation (LA) proposed in [16] is used. This approximation relies on intermediary (after time update but before the measurement update) particles from all near-by agents. Due to the fact that also the proposed LA scheme is employed in a FC, no additional communication cost is incurred despite the assumed knowledge on the intermediary particles. With this approximation, a separate PF can be employed for each agent to reduce the computational complexity compared to single particle filtering which would demand significantly more particles for comparable localization accuracy.

2.3 Gaussian Markov random fields

A zero-mean GMRF can also be understood as a discrete approximation through FEMs of the continuously indexed Gaussian RF (cf. Appendix: A.4) $z(\boldsymbol{s})$ using a set of N_α basis functions $\left\{\psi_{n_\alpha}\right\}$

$$z(\boldsymbol{s}) = \sum_{n_\alpha=1}^{N_\alpha} \psi_{n_\alpha}(\boldsymbol{s})\alpha_{n_\alpha}, \tag{3}$$

where $\psi_{n_\alpha} \;:\; \Omega \rightarrow [0,1]$ and $\psi_{n_\alpha}(\boldsymbol{s})$ is the n_αth basis function evaluated at position $\boldsymbol{s} \in \Omega$ and $\{\alpha_{n_\alpha}\}$ is a set of weights for the basis functions. The random vector $\boldsymbol{\alpha} = [\alpha_1 \ldots \alpha_{N_\alpha}]^\mathsf{T} \in \mathbb{R}^{N_\alpha}$ in conjunction with $\mathcal{G}$ describes a GMRF as per Definition 7 [17], where the positions of the vertices $\mathcal{V}$ of $\mathcal{G}$ are equal to the locations where the corresponding basis function achieve their maximum value of 1 and all other basis functions are 0.

The GMRF with Matérn covariance function (cf. Appendix A.5) is used subsequently to model the effect of the fluid on the agents' motion. In this context, special attention is drawn on the modeling through basis functions as presented above.

3 Proposed Gaussian Markov random field model

This work targets the estimation and modeling of external (i.e., driving) forces driving the agents' motion to improve the localization. These forces are relevant particularly for the localization of kinetically passive agents, where the fluid governs the agents' motion and where a localization solely based on distance and/or bearing measurements is inaccurate or only feasible with huge computational complexity.

To some extent, the work presented herein can be understood as a generalization of the concept proposed in our previous conference work [13], where an abstract statistical model for the external forces has been used in combination with particle filtering. In [13], as well as in this work, the objective is to describe these forces, for example by means of additional changes in the agents' speed or heading direction. The extension proposed herein targets the estimation and modeling of the driving forces by means of a RF and, thus, aims to improve the localization accuracy through the consideration of spatial correlation of these forces across the environment. Considering the correlation is important, not only because it is more realistic due the common fluid of all agents, but also because limited information in the measurement update phase can be compensated to some extent by knowledge of the underlying RF that models the driving forces. The resulting algorithm is henceforth denoted as random field-aided tracking (RFaT) algorithm. To account for the fact that in most of the considered application scenarios, agents are deployed in confined areas, also boundary effects are considered. As will be shown in more detail later, this is achieved by introducing two spatial domains with different correlation properties which respectively correspond to the fluid- and the non-fluid-carrying parts of the environment.

For the modeling, a GMRF is considered which presents a computationally efficient FEM approximation of a Gaussian RF (cf. Appendix A.4). Via the modeling as a RF, a single position-dependent model is obtained that is of use for all agents traversing the environment. This is in contrast to the procedure devised in [13], where only a fixed, position-independent statistical model is used. The GMRF is modeled using only a few hyper-parameters and reduces the problem of jointly localizing the agents and estimating the field to a joint parameter and state estimation problem.

The estimation of the field parameters is performed only through the distance and/or bearing measurements between agents and beacons. Consequently, no direct inference of the field parameters is possible. To nevertheless infer information about the field, a complex BHM is proposed, which is tackled through a combination of SMC and particle Markov Monte Carlo chain (PMCMC) methods. As mentioned in Section 2.2, the scheme is combined with the MPF framework in general and, in particular, the Monte Carlo approximation (MCA)-based LA proposed in [16] to ensure convergence of the high-dimensional state estimation problem.

4 Environment modeling via Gaussian Markov random fields

In this work, a scalar GMRF is used to model the effects of the fluid on the agents' motion. The assumption of a scalar field reduces computational complexity while, yet, offering sufficient modeling flexibility for the considered application case. The general concepts presented herein are, however, extensible to multivariate fields. The RF is used to model, for example, additional changes in the agents' heading direction, which has already been shown to be effective in improving the localization accuracy in [13]. Moreover, this is motivated by the fact that for example in piping systems, the variation of the speed (via tangential acceleration) is usually small, such that a major impact on the agents' motion is due to normal acceleration which can be captured through changes in the heading direction.

As detailed in Definition 7, a scalar GMRF is a spatial random process defined on $\Omega \subset \mathbb{R}^2$ and is henceforth denoted as $\{z(\boldsymbol{s})|\boldsymbol{s} \in \Omega\}$. The GMRF is fully described by its mean and covariance, which promises computationally efficient modeling and estimation. As mentioned in Section 2.3, the GMRF can be regarded as a FEM approximation to a continuous RF. The GMRF can, consequently, be described using basis functions. In the course of this work, only linear basis functions are considered for the sake of computational simplicity. Such basis functions have been reported in [17] to provide reasonable results.

In the following, the GMRF is parameterized to facilitate joint state and parameter estimation using the SMC^2 framework (cf. Appendix A.3). The SMC^2 framework leverages SMC and PMCMC methods to sequentially estimate the joint parameter and state posterior $p(\boldsymbol{x}_{k,\mathbb{i}}, \boldsymbol{\theta}|\boldsymbol{y}_{1:k})$ for every agent $\mathbb{i} \in \mathcal{A}$, where $\boldsymbol{\theta}$ denotes the field parameters, which are common to all agents. To account for the field parameterization, the GMRF evaluated at $\boldsymbol{s} \in \Omega$ is henceforth denoted by $z_{\boldsymbol{\theta}}(\boldsymbol{s})$ to make the dependence on the parameters $\boldsymbol{\theta}$ explicit.

In this work, the GMRF is modeled as a non-zero-mean field, which results in the following description, using the basis functions ψ for covariance modeling and the basis functions φ for mean modeling:

$$\begin{aligned} z_{\boldsymbol{\theta}}(\boldsymbol{s}_{k,\mathbb{i}}) &= \sum_{n_\alpha=1}^{N_\alpha} \psi_{n_\alpha}(\boldsymbol{s}_{k,\mathbb{i}})\alpha_{n_\alpha} + \sum_{n_\beta=1}^{N_\beta} \varphi_{n_\beta}(\boldsymbol{s}_{k,\mathbb{i}})\beta_{n_\beta} \\ &= \psi(\boldsymbol{s}_{k,\mathbb{i}})^\top \boldsymbol{\alpha} + \varphi(\boldsymbol{s}_{k,\mathbb{i}})^\top \boldsymbol{\beta}, \end{aligned} \tag{4a}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{N_\alpha}$ is the zero-mean random GMRF vector with precision matrix $\boldsymbol{Q}(\boldsymbol{\theta}_{\boldsymbol{Q}})$. Moreover, the weights $\boldsymbol{\beta} \in \mathbb{R}^{N_\beta}$ for the mean field are deterministic but unknown and, consequently, need to be estimated as well. With this description, the full set of parameters that are sought is given by

$$\boldsymbol{\theta} \equiv [\,\boldsymbol{\theta_Q}^\top, \boldsymbol{\beta}^\top]^\top\,. \tag{5}$$

Remark 2 *Importantly, in contrast to most cases in which RFs are employed for environment modeling, no direct field measurements are available or required in this work. In scenarios in which direct field measurements are available, the mean of the field can be directly inferred from the field measurements, thus avoiding the estimation of the mean component $\varphi(\boldsymbol{s}_{k,\mathbb{i}})^\top\boldsymbol{\beta}$ and the associated parameters $\boldsymbol{\beta}$. Consequently, the case considered in this work is more complex because the number of parameters to be estimated is significantly increased. For example, in the model considered above, at most three parameters are required to model the covariance (as in the Matérn model; cf. Appendix A.5) of the GMRF, whereas $N_\beta \gg 3$ mean parameters $\boldsymbol{\beta}$ are needed to obtain reasonable results, even for relatively small environments.*

The details regarding the FEM approach, i.e., the form of the precision matrix $\boldsymbol{Q}$ and its parameters $\boldsymbol{\theta_Q}$, are detailed below. Subsequently, an approach is presented that is based on [18] and is used to approximate the boundary effects that occur at the borderline between the fluid-carrying and non-fluid-carrying domains.

4.1 Barrier/Matérn environment model

Special handling of the GMRF model is required in cases in which physical boundaries are considered within the field's domain. This is because boundary effects, such as those described by Dirichlet or Neumann conditions, are generally incompatible with an isotropic GMRF (cf. Definition 5) because the field is no longer solely dependent on the distance between two points. To nevertheless describe the fluid properties in a sufficiently accurate manner using a GMRF, the original domain of the field Ω is subdivided into two disjoint domains, each of which describes different properties via the underlying covariance function. More precisely, the domain is split such that $\Omega = \Omega_n \cup \Omega_b$, where Ω_n denotes the normal (i.e., fluid-carrying) domain and Ω_b denotes the barrier (i.e., non-fluid-carrying) domain.

An example is given in Fig. 1b, which shows the mesh used in the simulations. Note that the covariance grid is present in both domains, i.e., in Ω_n and Ω_b, whereas the mean grid is defined only within Ω_n. This is because, in the barrier domain, no fluid and thus no external forces are present. The covariance grid on the other hand is defined in both domains to model said boundary effects, as detailed below.

The objective of separating the domains is to introduce different covariance properties in each domain such that the spatial correlations within and across the barrier domain are significantly lower than those in the normal domain, thus eventually approximating the boundary conditions mentioned above. A corresponding procedure is presented in the following. To simplify the derivation, Bakka et al. [18] proposed the following reparameterization of the Matérn covariance function for isotropic fields (cf. Definition 6):

$$\mathrm{Cov}_z[\, d = \|\boldsymbol{s}_1 - \boldsymbol{s}_2\|_2] \equiv \frac{\sigma_\alpha^2 d\sqrt{8}}{r} K_1\left(\frac{d\sqrt{8}}{r}\right), \tag{6}$$

where $\sigma_{\boldsymbol{\alpha}} \geq 0$ is the marginal standard deviation of the GMRF basis function weights $\boldsymbol{\alpha}$ and $r > 0$ is a scaled version of the original Matérn range ρ such that $r = \rho/\sqrt{8}$. Despite the reparameterization, the general interpretation of the range does not change: the smaller the range is, the faster the correlation between two points in the domain, say $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, decays. In this model, the ν parameter of the original Matérn covariance function is set to $\nu = 1$ because of the difficulty of inferring its value as part of the general estimation process [19]. Thus, the parameters to be estimated for the precision matrix are given as follows:

$$\boldsymbol{\theta}_{\boldsymbol{Q}} \equiv [\, r, \sigma_\alpha]^\top \in \mathbb{R}^2_{\geq 0}. \tag{7}$$

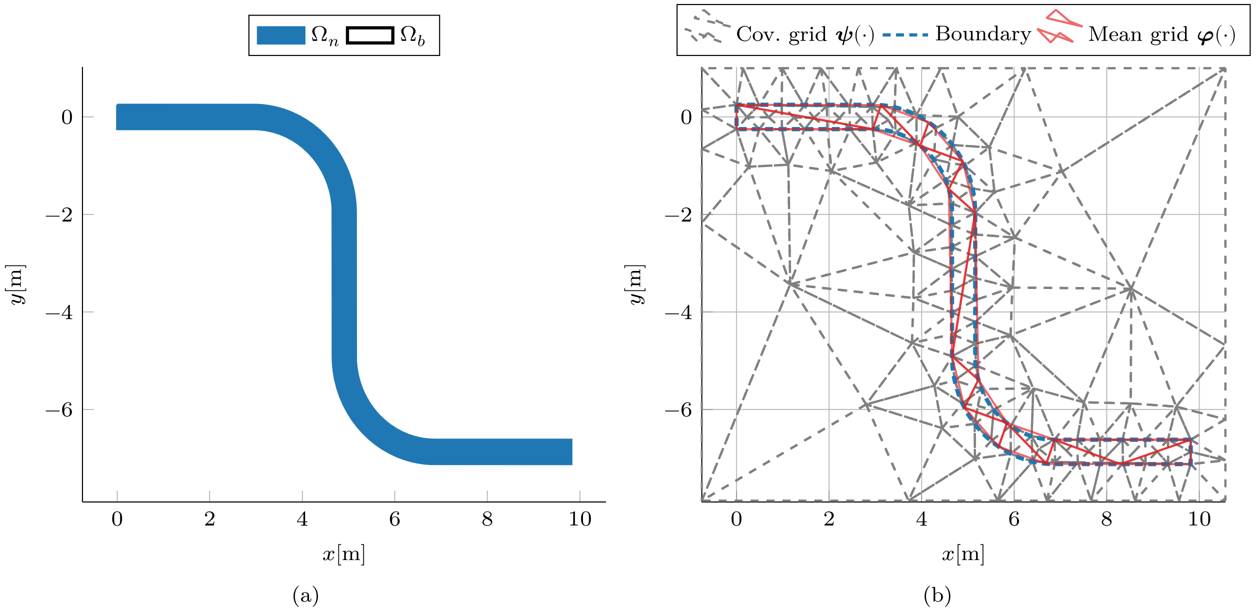


Fig. 1 **a** shows the simulation setup and the differentiation between the two domains: the normal domain Ω_n (blue), which is equivalent to the fluid-carrying part, and the barrier domain Ω_b (white), where no fluid is present. **b** visualizes both the mean and covariance parts of the FEM mesh. In total, $N_\alpha = 127$ grid vertices and ψ basis functions are used for the covariance field in this example, and $N_\beta = 19$ grid vertices and φ basis functions are used for the mean field

Using the FEM description given in (4), the precision matrix $\boldsymbol{Q}$ can be calculated as follows [18]:

$$\boldsymbol{Q}(r, \sigma_{\boldsymbol{\alpha}}) = \sigma_{\boldsymbol{\alpha}}^{-2} \boldsymbol{A}(r) \widetilde{\boldsymbol{C}}(r) \boldsymbol{A}(r), \tag{8a}$$

$$\boldsymbol{A}(r) = \boldsymbol{J} - \frac{1}{8}\left(r^2 \boldsymbol{D}_n + \frac{r^2}{100} \boldsymbol{D}_b\right), \tag{8b}$$

$$\widetilde{\boldsymbol{C}}(r) = \frac{\pi}{2}\left(r^2 \widetilde{\boldsymbol{C}}_n + \frac{r^2}{100} \widetilde{\boldsymbol{C}}_b\right), \tag{8c}$$

where $\boldsymbol{J}, \boldsymbol{D}_q, \widetilde{\boldsymbol{C}}_q \in \mathbb{R}^{N_\alpha \times N_\alpha}$ are computed as follows from the basis functions [18]:

$$[\boldsymbol{J}]_{i,j} \equiv \int \psi_i(\boldsymbol{s}) \psi_j(\boldsymbol{s}) d\boldsymbol{s} \qquad \forall i,j = 1, \ldots, N_\alpha \tag{9a}$$

$$\left[\boldsymbol{D}_q\right]_{i,j} \equiv \int_{\Omega_q} \nabla(\psi_i(\boldsymbol{s})) \nabla(\psi_j(\boldsymbol{s})) d\boldsymbol{s}, \qquad \forall i,j = 1, \ldots, N_\alpha, \ \forall q \in \{n, b\} \tag{9b}$$

$$\left[\widetilde{\boldsymbol{C}}_q\right]_{i,i} \equiv \int_{\Omega_q} \psi_i(\boldsymbol{s}) d\boldsymbol{s}, \qquad \forall i,j = 1, \ldots, N_\alpha, \ \forall q \in \{n, b\} \tag{9c}$$

Moreover, in (8), the range in the barrier domain Ω_b has been set to one-tenth of the range in the normal domain Ω_n, i.e., $r_b = r/10$. Although the hyper-parameter r_b could also theoretically be estimated, using a fixed factor of one-tenth has been empirically shown to yield reasonable results, with the additional benefit of reducing the parameter space and, thus, reducing the computational complexity.

Remark 3 *Note that $\boldsymbol{J}$, $\boldsymbol{D}_n$, $\boldsymbol{D}_b$, $\widetilde{\boldsymbol{C}}_n$, and $\widetilde{\boldsymbol{C}}_b$ are independent of r and $\sigma_{\boldsymbol{\alpha}}$ and thus can be computed offline before the estimation process.*

4.2 Finite element method

As mentioned before, effectively, two FEM approximations are used: one for the covariance field and one for the mean field. While the covariance grid covers the complete domain Ω, i.e., both the normal domain Ω_n and the barrier domain Ω_b, the mean grid needs to cover only the normal domain Ω_n. In the normal domain, a high mesh resolution is needed to enable accurate modeling of the fluid dynamics, whereas in the barrier domain, a coarse grid is sufficient.

During the creation of the covariance mesh, it is important to note that irregular grid boundaries are known to cause numerical artifacts and unrealistic behavior. For this reason, it is advised to extend the domain and instead consider, e.g., its convex hull [17, 18]. Thereby, the impact of the outer domain boundary conditions on model fitting is reduced. The quantity that represents the distance by which the domain is extended is denoted by d_e and is set as described below.

To ensure accurate generation of the mesh, a resolution formula is defined that controls the target distance between two vertices of the mesh (i.e., the desired edge length). For the covariance grid, the following formula is used:

$$l_{\boldsymbol{\psi}}(\boldsymbol{s}) \equiv \begin{cases} l_{\boldsymbol{\psi}}, & \text{if } \boldsymbol{s} \in \Omega_n \\ l_{\boldsymbol{\psi}}\left(1 + s_{\Omega_b} \cdot d_{\Omega_n}(\boldsymbol{s})\right), & \text{if } \boldsymbol{s} \in \Omega_b \end{cases}, \tag{10}$$

$$d_{\Omega_n}(\boldsymbol{s}) \equiv \min_{\boldsymbol{s}' \in \Omega_n} \|\boldsymbol{s} - \boldsymbol{s}'\|_2, \tag{11}$$

where $d_{\Omega_n}(\boldsymbol{s})$ is the smallest distance between $\boldsymbol{s} \in \Omega_b$ and the domain Ω_n. In this work, the value of the increase factor s_{Ω_b} is fixed to two. For the mean grid, which is present only in Ω_n, the desired edge length is $l_\varphi(s) \equiv l_\varphi$. In this work, the parameters d_e, l_ψ, and l_φ are defined relative to the *typical length*[3] L_{typ} as follows:

$$d_e = 0.075\, L_{\text{typ}}, \quad l_\psi = 0.05\, L_{\text{typ}}, \quad l_\varphi = 0.35\, L_{\text{typ}}. \tag{12}$$

4.3 Contributions of this section

The key contributions of this section are as follows: first, modeling the RF by a computational efficient GMRF with piece-wise linear basis functions; second, modeling of pipe systems as barrier-RF to address boundary effects; third, enabling inference even when no direct field measurements are available via modeling the mean of the RF by β parameters and via piece-wise linear basis functions; and fourth and finally, modeling of the GMRF by means of parameters to obtain a joint state and parameter estimation problem.

5 Gaussian Markov random field as control input

The objective of the GMRF model is to improve the localization performance by embedding the external forces directly into the motion model. To this end, the GMRF is used as an ACI for, e.g., the autonomous motion models discussed in [5]. Note that neither the ACI nor the field is directly observable.

Subsequently, the following dynamics are considered, which are obtained via the field parameterization given in Sections 4 and 4.1.

$$p(\boldsymbol{x}_{k+1,\mathbb{i}} | \boldsymbol{x}_{k,\mathbb{i}}, u_{k,\mathbb{i}}, \boldsymbol{\theta}) = p(\boldsymbol{x}_{k+1,\mathbb{i}} | \boldsymbol{x}_{k,\mathbb{i}}, u_{k,\mathbb{i}}(\boldsymbol{x}_{k,\mathbb{i}}, \boldsymbol{\theta})), \tag{13}$$

where $u_{k,\mathbb{i}}$ is the ACI modeled by means of $z_{\boldsymbol{\theta}}(\boldsymbol{s}_{k,\mathbb{i}})$ in combination with the zero-mean Gaussian noise ϵ:

$$u_{k,\mathbb{i}} = z_{\boldsymbol{\theta}}(\boldsymbol{s}_{k,\mathbb{i}}) + \epsilon, \tag{14}$$

$$p(u_{k,\mathbb{i}} | z, \boldsymbol{s}_{k,\mathbb{i}}, \boldsymbol{\theta}) = \mathcal{N}\left(z_{\boldsymbol{\theta}}(\boldsymbol{s}_{k,\mathbb{i}}), \sigma_\epsilon^2\right). \tag{15}$$

Henceforth, we may refer to $u_{k,\mathbb{i}}$ and ϵ as *virtual* measurements and measurement noise, respectively, because, in fact, no field measurements are collected and the model above is merely used in the proposed BHM. Moreover,

[3] In the literature, the typical length is not a properly defined quantity. In this work, it is interpreted as the width of the pipe system.

ϵ plays an important role in modeling local variations of the field.

Remark 4 *Although the field depends only on the positions encoded in the state vector $\boldsymbol{x}_{k,\mathbb{i}}$, $z_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,\mathbb{i}})$ may be used instead of $z_{\boldsymbol{\theta}}(\boldsymbol{s}_{k,\mathbb{i}})$ henceforth to simplify the notation and avoid confusion.*

The field's conditional density is given by

$$\begin{aligned} &z_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,\mathbb{i}}) \sim p(z|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta}) = \\ &\mathcal{N}(\varphi(\boldsymbol{x}_{k,\mathbb{i}})^{\mathsf{T}}\boldsymbol{\beta}, \psi(\boldsymbol{x}_{k,\mathbb{i}})^{\mathsf{T}}\boldsymbol{Q}^{-1}(r,\sigma_{\boldsymbol{\alpha}})\psi(\boldsymbol{x}_{k,\mathbb{i}})). \end{aligned} \tag{16}$$

Hence, given the state and the parameters, the input distribution of $u_{k,\mathbb{i}}$ can be written as

$$p(u_{k,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta}) = \int p(u_{k,\mathbb{i}}|z,\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta})p(z|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta})dz, \tag{17}$$

where the closed form can be easily deduced due to the linearity of (15). To this end, through the use of (15) and (16), a Gaussian distribution is obtained for the sought density:

$$\begin{aligned} &p(u_{k,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta}) = \\ &\mathcal{N}\left(\varphi(\boldsymbol{x}_{k,\mathbb{i}})^{\mathsf{T}}\boldsymbol{\beta}, \psi(\boldsymbol{x}_{k,\mathbb{i}})^{\mathsf{T}}\boldsymbol{Q}^{-1}(r,\sigma_{\boldsymbol{\alpha}})\psi(\boldsymbol{x}_{k,\mathbb{i}}) + \sigma_{\epsilon}^{2}\right). \end{aligned} \tag{18}$$

Remark 5 *Note that in comparison to the scheme devised in our previous conference work* [13], *for which the particular choice of $p(u_{k,\mathbb{i}}) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ is made in the simulations, the GMRF input model defined in* (18) *facilitates the consideration of spatial dependencies through position-dependent mean and covariance terms.*

In summary, the contribution of this section is the formalization of the closed form, i.e., non-integral, description of the ACI's conditional distribution. As will be shown in the subsequent section, this presents a major advantage for effective and efficient time-updating.

6 Time update model

Because the agents' motion is now modeled by means of an unobservable process, the new effective transition PDF, which is needed for the state and parameter inference process, needs to be derived. This derivation is achieved through marginalization and with the use of (13) and (18). More precisely, using (13) under the assumption of zero-mean process noise with a covariance of Σ_{ν}, the new state evolution model is obtained by averaging over the unobservable field such that

$$\begin{aligned} p(\boldsymbol{x}_{k+1,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta}) = \int & p(\boldsymbol{x}_{k+1,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},u_{k,\mathbb{i}}) \\ & \cdot p(u_{k,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta})du_{k,\mathbb{i}}. \end{aligned} \tag{19}$$

Note that this integral does not have an analytical solution in general. Therefore, approximations are needed when considering, for example, nonlinear motion models. For this purpose, two different procedures are presented in the following. The first approximation method uses a standard procedure based on linearization to solve the integral, while the second method exploits the fact that a PF-based approach is taken to solve the joint state and parameter inference problem, in which samples from the sought density (in the subsequently proposed Algorithm 1, see Line 9) are required for the time update.

6.1 Approximation by linearization

A standard approach to approximating (19) for a nonlinear transition function $\boldsymbol{f}$ with additive Gaussian process noise is based on the linearization thereof around a development point of the integration variable, which is henceforth denoted by $u^{0}_{k,\mathbb{i}}$. The approximated state evolution function $\widetilde{\boldsymbol{f}}(\cdot)$ is obtained as

$$\boldsymbol{f}(\boldsymbol{x}_{k,\mathbb{i}},u_{k,\mathbb{i}}) \approx \widetilde{\boldsymbol{f}}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}},u_{k,\mathbb{i}}) \tag{20}$$

$$\begin{aligned} &\equiv \boldsymbol{f}\left(\boldsymbol{x}_{k,\mathbb{i}},u^{0}_{k,\mathbb{i}}\right) \\ &\quad + \nabla(\boldsymbol{f}(\boldsymbol{x}_{k,\mathbb{i}},u_{k,\mathbb{i}}))\big|_{u^{0}_{k,\mathbb{i}}}\left(u_{k,\mathbb{i}} - u^{0}_{k,\mathbb{i}}\right), \end{aligned} \tag{21}$$

which can be divided into a constant term and a slope term as follows:

$$\widetilde{\boldsymbol{f}}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}},u_{k,\mathbb{i}}) = \boldsymbol{c}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}}) + \boldsymbol{s}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}})u_{k,\mathbb{i}} \tag{22}$$

with

$$\begin{aligned} \boldsymbol{c}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}}) &\equiv \boldsymbol{f}\left(\boldsymbol{x}_{k,\mathbb{i}},u^{0}_{k,\mathbb{i}}\right) - \nabla_{u_{k,\mathbb{i}}}\left(\boldsymbol{f}\left(\boldsymbol{x}_{k,\mathbb{i}},u_{k,\mathbb{i}}\right)\right)\big|_{u^{0}_{k,\mathbb{i}}} u^{0}_{k,\mathbb{i}} \\ \boldsymbol{s}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}}) &\equiv \nabla_{u_{k,\mathbb{i}}}\left(\boldsymbol{f}\left(\boldsymbol{x}_{k,\mathbb{i}},u_{k,\mathbb{i}}\right)\right)\big|_{u^{0}_{k,\mathbb{i}}}. \end{aligned}$$

With this linearization, the following holds for the nonmarginalized (i.e., input-conditioned) transition PDF under the assumption of additive Gaussian process noise:

$$\widehat{p}(\boldsymbol{x}_{k+1,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},u_{k,\mathbb{i}}) = \mathcal{N}\left(\widetilde{\boldsymbol{f}}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{u}_{k,\mathbb{i}}),\Sigma_{\nu}\right). \tag{23}$$

Using this affine Gaussian transition PDF, (19) can be approximated as follows using (18):

$$\begin{aligned} \widehat{p}(\boldsymbol{x}_{k+1,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta}) = \int & \mathcal{N}\left(\widetilde{\boldsymbol{f}}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{u}_{k,\mathbb{i}}),\Sigma_{\nu}\right) \\ & \cdot \mathcal{N}\left(\varphi(\boldsymbol{x}_{k,\mathbb{i}})^{\mathsf{T}}\boldsymbol{\beta},\ \psi(\boldsymbol{x}_{k,\mathbb{i}})\boldsymbol{Q}^{-1}(r,\sigma_{\boldsymbol{\alpha}}) \right. \\ & \left. \cdot\ \psi(\boldsymbol{x}_{k,\mathbb{i}}) + \sigma_{\epsilon}^{2}\right)\ du_{k,\mathbb{i}} \end{aligned} \tag{24}$$

which is multivariate Gaussian with mean and covariance matrix

$$\mathbb{E}[\boldsymbol{x}_{k+1,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta}] = \boldsymbol{c}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{c}_{k,\mathbb{i}}) + \boldsymbol{s}_{u^{0}_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}})(\varphi(\boldsymbol{x}_{k,\mathbb{i}})^{\mathsf{T}}\boldsymbol{\beta}) \tag{25a}$$

$$\begin{aligned}\mathrm{Cov}[\boldsymbol{x}_{k+1,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta}] = \boldsymbol{s}_{u^0_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}}) \\ \cdot \left(\psi(\boldsymbol{x}_{k,\mathbb{i}})^\mathsf{T}\boldsymbol{Q}^{-1}(r,\sigma_\alpha)\psi(\boldsymbol{x}_{k,\mathbb{i}}) + \sigma_\epsilon^2\right) \\ \cdot \boldsymbol{s}_{u^0_{k,\mathbb{i}}}(\boldsymbol{x}_{k,\mathbb{i}})^\mathsf{T} + \Sigma_\nu. \end{aligned} \tag{25b}$$

This approximation, in turn, facilitates efficient sampling for the state evolution within a PF.

To obtain reasonably accurate approximations, the development point $u^0_{k,\mathbb{i}}$ must be properly chosen, i.e., sufficiently close to the actual evaluation points for the density. To this end, the conditional mean is used, which can be obtained as follows:

$$u^0_{k,\mathbb{i}} \equiv \mathbb{E}[\,u_{k,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta}] = \varphi(\boldsymbol{x}_{k,\mathbb{i}})^\mathsf{T}\boldsymbol{\beta}. \tag{26}$$

6.2 Sequential sampling

In this subsection, a procedure is presented that aims not to approximate the integral in (19), per se, but rather to address the higher-level goal of sampling $p(\boldsymbol{x}_{k+1,\mathbb{i}}|\boldsymbol{x}_{k,\mathbb{i}},\boldsymbol{\theta})$, which is needed due to the adopted PF approach.

First, the rationale for the procedure is illustrated. Let $p(x)$ denote the density that is sought, which is given by

$$p_x(x) = \int p_{x|y}(x|y)p_y(y)\,dy. \tag{27}$$

Then, samples $\left\{y^{(\ell)}\right\}_\ell$ are drawn from $p_y(y)$, followed by samples $x^{(\ell)} \sim p_{x|y}(x|y = y^{(\ell)})\forall\ell = 1,\ldots,L$, which are obtained through conditioning on the previous samples of y. From the obtained set of samples $\left\{\left\langle x^{(\ell)}, y^{(\ell)}\right\rangle\right\}_\ell$, $\left\{x^{(\ell)}\right\}_\ell$ can be regarded as samples from $p_x(x)$ because averaging over y can be interpreted as ignoring the samples $\left\{y^{(\ell)}\right\}_\ell$ from the tuple.

Applying this principle to the problem at hand, i.e., to sampling from the transition density (19), yields the following procedure. First, sample from the control input distribution (18):

$$u^{(\ell)}_{k,\mathbb{i}} \sim p\left(u_{k,\mathbb{i}}|\boldsymbol{x}^{(\ell)}_{k,\mathbb{i}},\boldsymbol{\theta}\right). \tag{28}$$

Second, obtain a sample from the process noise density

$$\nu^{(\ell)}_{k,\mathbb{i}} \sim p(\boldsymbol{\nu}). \tag{29}$$

Finally, obtain the sought sample as

$$\boldsymbol{x}^{(\ell)}_{k+1,\mathbb{i}} = \boldsymbol{f}\left(\boldsymbol{x}^{(\ell)}_{k,\mathbb{i}}, u^{(\ell)}_{k,\mathbb{i}}\right) + \boldsymbol{\nu}^{(\ell)}_{k,\mathbb{i}}. \tag{30}$$

This procedure is repeated until the required quantity of particles is obtained, i.e., for $\ell = 1,\ldots,L$. Note that this procedure does not introduce function approximation errors but does incur a slight increase in computational complexity due to the additional sampling step.

6.3 Control input-driven motion model

The motion model used in this work shows some similarities to the model used in our previous conference work [13] because it also assumes that the input represents an additional change in the agent's heading direction, i.e., the input is similar to a turn rate. However, unlike in [13], in this work, the turn rate itself is not part of the state vector; thus, the computational complexity is reduced, and numerical issues are avoided in the case that the turn rate is close to zero. This modification is motivated by the fact that the joint parameter and state estimation is significantly more complex, and reducing the number of state dimensions is one way to reduce the computational load.

Consequently, the state vector used in this work is given by $\boldsymbol{x}_{k,\mathbb{i}} = \begin{bmatrix} x_{k,\mathbb{i}} & y_{k,\mathbb{i}} & \upsilon_{k,\mathbb{i}} & \phi_{k,\mathbb{i}}\end{bmatrix}$ which leads with the definition of the input $u_{k,\mathbb{i}}$ as the turn rate to the following transition density:

$$\boldsymbol{x}_{k+1,\mathbb{i}} \sim \mathcal{N}(\boldsymbol{f}(\boldsymbol{x}_{k,\mathbb{i}}, u_{k,\mathbb{i}}), \Sigma_\nu), \tag{31}$$

$$\boldsymbol{f}(\boldsymbol{x}_{k,\mathbb{i}}, u_{k,\mathbb{i}}) = \begin{bmatrix} x_{k,\mathbb{i}} + T\cdot\upsilon_{k,\mathbb{i}}\cos(\phi_{k,\mathbb{i}} + T\cdot u_{k,\mathbb{i}}) \\ y_{k,\mathbb{i}} + T\cdot\upsilon_{k,\mathbb{i}}\sin(\phi_{k,\mathbb{i}} + T\cdot u_{k,\mathbb{i}}) \\ \upsilon_{k,\mathbb{i}} \\ \phi_{k,\mathbb{i}} + T\cdot u_{k,\mathbb{i}} \end{bmatrix}, \tag{32}$$

where the process noise covariance Σ_ν is defined as

$$\Sigma_\nu \equiv \mathrm{diag}\left[\sigma_x{}^2 \quad \sigma_y{}^2 \quad T^2\sigma_{\dot{\upsilon}}{}^2 \quad T^2\sigma_{\dot{\phi}}{}^2\right]. \tag{33}$$

The function $\boldsymbol{f}(\cdot)$ as well as the lower-right part of the covariance matrix, which belongs to the linear part of the motion model, is derived based on the method outlined in [20]. For the upper-left part of the covariance matrix, which belongs to the nonlinear part of the motion model, no such method exists. For this reason, a diagonal part has been assumed that is parameterized through σ_x and σ_y. The covariance matrix parameters used in the simulation of this work are provided in Table 3.

6.4 Contributions of this section

The contributions of this section are twofold: First, two approximations of the transition distribution to incorporate uncertainties of the control input were deduced, of which the second is later shown to be computationally efficient and effective. Second, a well-performing motion model for the agents that incorporates the ACI through modeling the field as incremental change of the heading was obtained. As will be shown in the simulations presented in Section 9, this model is shown to be effective for the pipe-based application scenarios.

7 Proposed random field-aided tracking algorithm

This section introduces to the framework used to facilitate joint parameter and state estimation as well as to the required prerequisites thereof. Finally, a pseudo code description of the procedure is presented.

7.1 Estimation framework

Although several parameter estimation methods have been presented in the literature, this work builds upon the SMC^2 framework (cf. Appendix A.5). The SMC^2 framework efficiently combines SMC methods for the state space with PMCMC methods for the parameters. Informally, the SMC^2 framework operates one PF for every parameter particle.

The main motivation for adopting the SMC^2 framework is provided by its online capabilities, i.e., the fact that the sought posteriors $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:k})$ and $p(\boldsymbol{x}_{k,1:|\mathcal{A}|}|\boldsymbol{y}_{1:k})$ are sequentially estimated using all previous measurements $\boldsymbol{y}_{1:k}$[4]. This is in contrast to pure PMCMC methods and expectation maximization (ExpMax) methods such as those presented in [21, 22]. Both types of methods can be regarded as offline methods because the inference over the parameters is not performed sequentially, i.e., the parameter candidates are updated only after all measurements $\boldsymbol{y}_{1:K}$ have already been processed. The consequence of this is that poor parameter candidates are not discarded in each iteration but rather are discarded only at the end of one round of inference over all measurements.

Conversely, the SMC^2 framework facilitates online evaluation of the performance of the parameter particles and includes a rejuvenation step that is performed on an on-demand basis. Moreover, based on the acceptance ratio (AR) for the PMCMC step of SMC^2, the number of state particles L is adapted accordingly. This adaptation process simultaneously targets the reduction of the high computational complexity of the scheme, which generally increases over time, and the efficient exploration of the parameter space. Regarding the latter, the notion is to allow several PMCMC steps (if needed) in the early time steps to discard uninteresting parts of the parameter search space to avoid costly PMCMC steps later on. Recall that PMCMC steps require a complete reevaluation of the past history, denoted by 1:k. Costly PMCMC steps can be avoided in later steps because an increased number of particles is generally associated with a higher AR [23]. Moreover, the ability to discard parameter candidates early on is particularly advantageous if the a priori information is less informative in the sense that the prior particles provide an insufficient description of the posterior [24].

7.1.1 Alternative online estimation methods

In [25], a procedure is presented that is similar to SMC^2 in the sense that two layers of Monte Carlo (MC) methods are adopted, where the *inner* layer also includes a PF technique. However, unlike SMC^2, a fixed parameter particle set is used; because the particle set is not updated over time, this method does not take advantage of information gathered from later measurements [26]. Hence, SMC^2 is a more flexible framework because it considers this information.

More alternative schemes for combined state and parameter estimation exist. Most importantly, this is [27], in which the state vector is augmented by the parameters and in which a kernel density is used to update the parameter particles. In this way, the time-invariant parameters are treated as time-varying parameters. Similar to the PMCMC kernel used in SMC^2, the kernel density is assumed to be Gaussian [28]. The important differences relative to SMC^2 are that the joint state and parameter posterior is approximated as a Gaussian mixture and that this approach facilitates neither adaptation of the number of state particles nor a recovery procedure (particle rejuvenation) in the case that the current parameter particles do not aid in estimating the sought posterior (parameter particle degeneracy).

Remark 6 *Although the estimation framework proposed herein is based on SMC^2, our modeling contributions are not specific to SMC^2. Therefore, the work presented herein can be easily adapted to other frameworks.*

7.2 Parameter priors

Because a Bayesian approach to parameter inference is taken, prior models for the individual components of $\boldsymbol{\theta}$ are required. To this end, recall that three types of parameters are to be estimated:

$$\boldsymbol{\theta} = [r, \sigma_{\boldsymbol{\alpha}}, \boldsymbol{\beta}]^\intercal \in \mathbb{R}^2_{\geq 0} \times \mathbb{R}^{N_\beta}. \tag{34}$$

Regarding the first two parameters, i.e., the range and the marginal standard deviation of the random vector $\boldsymbol{\alpha}$ of the GMRF, a reasoning similar to that of [18] is applied: the randomness of the parameters should be independent of how much randomness has already been considered through other components in the models. This characteristic is modeled through the *memoryless* property of the exponential distribution, which means that for $X \sim \mathcal{E}(\lambda)$, it holds that $p(X > s + t|X > s) = p(X > t), \forall s, t \geq 0$. A similar reasoning for this choice can be found in [29, 30], where the authors argue with a “constant rate penalization” property that is achieved through the choice of exponential priors and which is considered paramount to avoid overfitting.

Moreover, it is not known a priori whether the independent and identically distributed (i.i.d.) effect modeled through ϵ (i.e., virtual measurement noise), with a standard deviation of σ_ϵ (cf. (15)), or the spatial effects modeled through $\boldsymbol{\alpha}$, with a marginal standard deviation of $\sigma_{\boldsymbol{\alpha}}$, should be dominant. In other words, the local effects from the virtual measurement noise should not, a priori, be

[4] The term *online* is not to imply any real-time capabilities of the procedure. In fact, it will later become obvious that real-time processing might be infeasible. More precisely, an algorithm is denoted as an *online* method if it estimates the state and/or parameters of a system when new data becomes available during the operation of the physical system.

favored over the spatial correlation effects. For this reason, the prior for $\sigma_{\boldsymbol{\alpha}}$ is set to

$$\sigma_{\boldsymbol{\alpha}} \sim \mathcal{E}(\lambda_{\sigma_\alpha}), \quad \lambda_{\sigma_\alpha}^{-1} \equiv \mathbb{E}[\sigma_{\boldsymbol{\alpha}}] \Leftrightarrow \lambda_{\sigma_\alpha} = 1/\sigma_\epsilon. \tag{35}$$

On the other hand, for the range parameter r, which is coupled through the factor of one-tenth to the range of the barrier domain Ω_b (cf. Section 4.1), there is an a priori preference for large values. The reason is that the range parameter, which controls the spatial correlations in the sense that larger values ensure that farther apart locations in the domain are correlated, is used to model the field that is designed to capture the nonlocal variability of the environment. For this reason, the prior for the range is set to

$$\frac{1}{r} \sim \mathcal{E}(\lambda_r), \qquad \lambda_r = \frac{\ln 2}{2} L_{\text{typ}}, \tag{36}$$

where L_{typ} is the typical length of the domain.

An important factor determining the estimation performance is modeled through the mean basis function weights $\boldsymbol{\beta}$. Although the spatial correlation is modeled through the basis functions ψ with random weights $\boldsymbol{\alpha}$, the a priori mean components should also be subject to spatial correlations. To this end, the following procedure is adopted to obtain a priori for $\boldsymbol{\beta}$: First, the median of the range r and the inverse mean of $\sigma_{\boldsymbol{\alpha}}$ priors are obtained, i.e., λ_r and λ_{σ_α}. Second, these values, which are originally associated with the covariance field, are used for the mean field to obtain $\boldsymbol{Q}_{\boldsymbol{\beta}}(r = \lambda_r, \sigma_\beta = \lambda_{\sigma_\alpha})$ via (8), with the basis functions φ. Finally, the prior is set to

$$p(\boldsymbol{\beta}) = \mathcal{N}\left(\mathbf{0}, (\boldsymbol{Q}_{\boldsymbol{\beta}})^{-1}\right). \tag{37}$$

Note that the choice of a Gaussian prior is motivated by the fact that the turn rate, which is modeled through the field (cf. Section 6.3), has a prevalence of zero in many scenarios. This is because almost straight-ahead motion is more common than strong left or right turns[5].

7.3 The algorithm

With the abovementioned models and the MCA-based LA for the MPF framework as proposed in [16], the procedure summarized in Algorithm 1 is obtained. Note that in Lines 8 and 9, the sequential sampling approach (cf. Section 6.2) is used, which can alternatively be replaced with linearization (cf. Section 6.1). Figure 2 visualizes the RFaT algorithm, including the hierarchical models and the two resampling stages. Whereas the state particles are resampled in every time step in the first stage, the parameter particles are resampled only if the effective sample size (ESS) drops below the set threshold.

Algorithm 1: SMC^2-based Random Field-aided Tracking algorithm with sequential sampling, cf. Sec. 6.2

```
input  : The measurement data $\mathbf{y}_{1:K}$
output : The posterior approx. $\hat{p}(\mathbf{x}_{k,\mathbb{i}}, \boldsymbol{\theta} \,|\, \mathbf{y}_{1:k})$,
         $\forall \mathbb{i} \in \mathcal{A}, k = 1, \dots, K$
1  foreach $\ell_\theta \in \{1, \dots, L_\theta\}$ do                      /* Initialization */
2      Sample param. $\boldsymbol{\theta}_0^{(\ell_\theta)} \sim p(\boldsymbol{\theta}_0)$ and set $w_{\theta,0}^{(\ell_\theta)} = \frac{1}{L_\theta}$;
3      foreach $\langle \mathbb{i}, \ell \rangle \in \mathcal{A} \times \{1, \dots, L\}$ do $\mathbf{x}_{0,\mathbb{i}}^{(\ell,\ell_\theta)} \sim p(\mathbf{x}_{0,\mathbb{i}})$ ;
4  end
5  foreach $k \in \{1, \dots, K\}$ do
6      foreach $\ell_\theta \in \{1, \dots, L_\theta\}$ do
7          foreach $\langle \mathbb{i}, \ell \rangle \in \mathcal{A} \times \{1, \dots, L\}$ do  // cf. Sec. 6.2
8              $u_{k-1,\mathbb{i}}^{(\ell,\ell_\theta)} \sim p\left(u_{k-1,\mathbb{i}}^{(\ell,\ell_\theta)} \,\middle|\, \mathbf{x}_{k-1,\mathbb{i}}^{(\ell,\ell_\theta)}, \boldsymbol{\theta}_k^{(\ell_\theta)}\right)$   // cf. (18);
9              $\mathbf{x}_{k,\mathbb{i}}^{(\ell,\ell_\theta)} \sim p\left(\mathbf{x}_{k,\mathbb{i}}^{(\ell,\ell_\theta)} \,\middle|\, \mathbf{x}_{k-1,\mathbb{i}}^{(\ell,\ell_\theta)}, u_{k-1,\mathbb{i}}^{(\ell,\ell_\theta)}\right)$   // cf. (2a);
10         end
11         foreach $\mathbb{i} \in \mathcal{A}$ do
             $w_{k,\mathbb{i}}^{(\ell,\ell_\theta)} \propto \hat{p}\left(\mathbf{y}_{k,\mathbb{i}} \,\middle|\, \mathbf{x}_{k,\mathbb{i}}^{(\ell,\ell_\theta)}\right), \quad \forall \ell = 1, \dots, L;$
             /* MCA LA, cf. [16] */
12         $w_{\theta,k}^{(\ell_\theta)} = w_{\theta,k-1}^{(\ell_\theta)} \frac{1}{|\mathcal{A}|L} \sum_{\mathbb{i} \in \mathcal{A}} \sum_{\ell=1}^{L} w_{k,\mathbb{i}}^{(\ell,\ell_\theta)}$;
13     end
14     foreach $\mathbb{i} \in \mathcal{A}$ do resample($w_{k,\mathbb{i}}^{(1:L,1:L_\theta)}$, $\mathbf{x}_{k,\mathbb{i}}^{(1:L,1:L_\theta)}$) ;
15     ESS $\leftarrow \left(\sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)}\right)^2 \Big/ \sum_{\ell_\theta=1}^{L_\theta} \left(w_{\theta,k}^{(\ell_\theta)}\right)^2$ ;
16     if $ESS < L_{\theta_{th}}$ then                          /* cf. Appendix A.3 */
17         calculate proposal parameters $\boldsymbol{\mu}_{\pi(\boldsymbol{\theta}_k)}$ and $\boldsymbol{\Sigma}_{\pi(\boldsymbol{\theta}_k)}$ ;
18         resample($\boldsymbol{\theta}_k^{(1:L_\theta)}$) ;
19         AR $\leftarrow$ performPMCMCstep($\boldsymbol{\mu}_{\pi(\boldsymbol{\theta}_k)}$, $\boldsymbol{\Sigma}_{\pi(\boldsymbol{\theta}_k)}$) ;
20         if AR $<$ accept-ratio-threshold then increase $L$ ;
21     end
22     foreach $\mathbb{i} \in \mathcal{A}$ do $\hat{p}(\mathbf{x}_{k,\mathbb{i}}, \boldsymbol{\theta} \,|\, \mathbf{y}_{1:k}) =$
         $\sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)} \sum_{\ell=1}^{L} w_{k,\mathbb{i}}^{(\ell,\ell_\theta)} \delta\left([\mathbf{x}_{k,\mathbb{i}}, \boldsymbol{\theta}] - \left[\mathbf{x}_{k,\mathbb{i}}^{(\ell,\ell_\theta)}, \boldsymbol{\theta}_k^{(\ell_\theta)}\right]\right)$ ;
23 end
```

The algorithm obtains the sought posterior via

$$\widehat{p}(\boldsymbol{x}_{k,\mathbb{i}}, \boldsymbol{\theta} | \boldsymbol{y}_{1:k}) = \sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)} \cdot \sum_{\ell=1}^{L} w_{k,\mathbb{i}}^{(\ell,\ell_\theta)} \delta\left([\boldsymbol{x}_k, \boldsymbol{\theta}] - \left[\boldsymbol{x}_{k,\mathbb{i}}^{(\ell,\ell_\theta)}, \boldsymbol{\theta}_{k,\mathbb{i}}^{(\ell_\theta)}\right]\right). \tag{38}$$

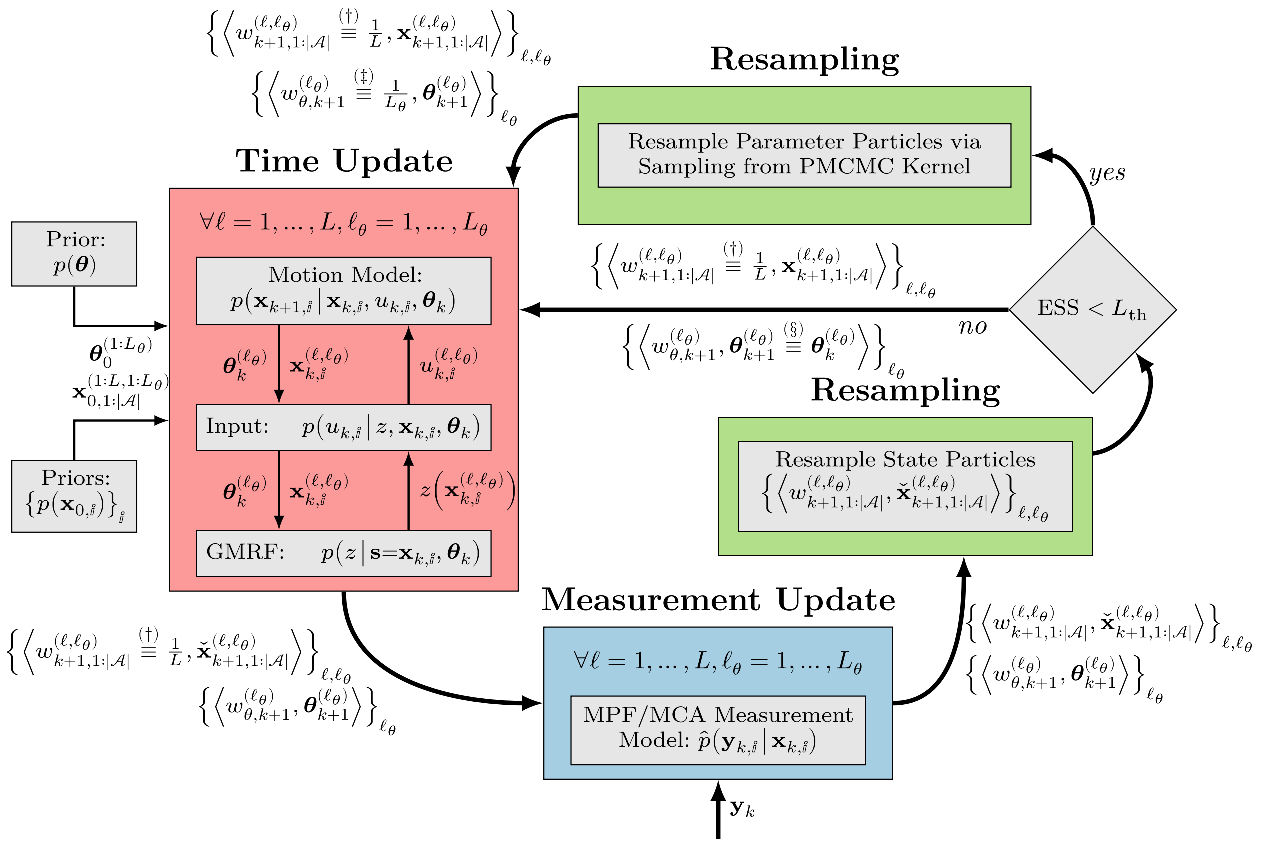
Based on the particle description, the state and parameter estimates are obtained as follows:

$$\hat{\boldsymbol{x}}_{k,\mathbb{i}} = \sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)} \sum_{\ell=1}^{L} w_{k,\mathbb{i}}^{(\ell,\ell_\theta)} \boldsymbol{x}_{k,\mathbb{i}}^{(\ell,\ell_\theta)}, \quad \forall \mathbb{i} \in \mathcal{A} \tag{39}$$

$$\widehat{\boldsymbol{\theta}}_k = \sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)} \boldsymbol{\theta}_k^{(\ell_\theta)}. \tag{40}$$

These estimates build the basis for the results presented in Section 8.

[5] By the same reasoning, a Gaussian distribution has been used for the input in [13].


$\left\{\left\langle w^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|} \overset{(\dagger)}{\equiv} \frac{1}{L}, \mathbf{x}^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|}\right\rangle\right\}_{\ell,\ell_\theta}$
$\left\{\left\langle w^{(\ell_\theta)}_{\theta,k+1} \overset{(\ddagger)}{\equiv} \frac{1}{L_\theta}, \boldsymbol{\theta}^{(\ell_\theta)}_{k+1}\right\rangle\right\}_{\ell_\theta}$
Resampling
Resample Parameter Particles via Sampling from PMCMC Kernel
Time Update
$\forall \ell = 1, \dots, L, \ell_\theta = 1, \dots, L_\theta$
Prior: $p(\boldsymbol{\theta})$
Motion Model: $p(\mathbf{x}_{k+1,\jmath} \,|\, \mathbf{x}_{k,\jmath}, u_{k,\jmath}, \boldsymbol{\theta}_k)$
$\boldsymbol{\theta}^{(1:L_\theta)}_0$
$\mathbf{x}^{(1:L,1:L_\theta)}_{0,1:|\mathcal{A}|}$
$\boldsymbol{\theta}^{(\ell_\theta)}_k$
$\mathbf{x}^{(\ell,\ell_\theta)}_{k,\jmath}$
$u^{(\ell,\ell_\theta)}_{k,\jmath}$
Input: $p(u_{k,\jmath} \,|\, z, \mathbf{x}_{k,\jmath}, \boldsymbol{\theta}_k)$
Priors: $\{p(\mathbf{x}_{0,\jmath})\}_\jmath$
$\boldsymbol{\theta}^{(\ell_\theta)}_k$
$\mathbf{x}^{(\ell,\ell_\theta)}_{k,\jmath}$
$z\left(\mathbf{x}^{(\ell,\ell_\theta)}_{k,\jmath}\right)$
GMRF: $p(z \,|\, \mathbf{s}{=}\mathbf{x}_{k,\jmath}, \boldsymbol{\theta}_k)$
$\left\{\left\langle w^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|} \overset{(\dagger)}{\equiv} \frac{1}{L}, \mathbf{x}^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|}\right\rangle\right\}_{\ell,\ell_\theta}$
yes
ESS $< L_{\text{th}}$
no
$\left\{\left\langle w^{(\ell_\theta)}_{\theta,k+1}, \boldsymbol{\theta}^{(\ell_\theta)}_{k+1} \overset{(\S)}{\equiv} \boldsymbol{\theta}^{(\ell_\theta)}_{k}\right\rangle\right\}_{\ell_\theta}$
Resampling
Resample State Particles
$\left\{\left\langle w^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|}, \check{\mathbf{x}}^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|}\right\rangle\right\}_{\ell,\ell_\theta}$
$\left\{\left\langle w^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|} \overset{(\dagger)}{\equiv} \frac{1}{L}, \check{\mathbf{x}}^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|}\right\rangle\right\}_{\ell,\ell_\theta}$
$\left\{\left\langle w^{(\ell_\theta)}_{\theta,k+1}, \boldsymbol{\theta}^{(\ell_\theta)}_{k+1}\right\rangle\right\}_{\ell_\theta}$
Measurement Update
$\forall \ell = 1, \dots, L, \ell_\theta = 1, \dots, L_\theta$
MPF/MCA Measurement Model: $\hat{p}(\mathbf{y}_{k,\jmath} \,|\, \mathbf{x}_{k,\jmath})$
$\left\{\left\langle w^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|}, \check{\mathbf{x}}^{(\ell,\ell_\theta)}_{k+1,1:|\mathcal{A}|}\right\rangle\right\}_{\ell,\ell_\theta}$
$\left\{\left\langle w^{(\ell_\theta)}_{\theta,k+1}, \boldsymbol{\theta}^{(\ell_\theta)}_{k+1}\right\rangle\right\}_{\ell_\theta}$
$\mathbf{y}_k$

†) After resampling and prior to the measurement update, the state particle weights are uniform
‡) After parameter particle resampling, the parameter particles weights are uniform
§) If no parameter particle resampling takes place, all $\boldsymbol{\theta}^{(\ell_\theta)}_k$ are denoted as $\boldsymbol{\theta}^{(\ell_\theta)}_{k+1}$

Fig. 2 Illustration of the proposed RFaT scheme

7.4 Contributions of this section

The key contributions of this chapter are as follows: first, applying SMC^2 for the resulting joint state and parameter estimation problem; second, adapting SMC^2 to handle high-dimensional state estimation problems by incorporating MPF; third, obtaining reasonable GMRF parameter priors, particularly for the beta parameters; and fourth and finally, obtaining optimal SMC^2 hyper-parameters for the specific application scenario.

8 Simulation setup and method

The presented simulations are based on the environment depicted in Fig. 3, for which the typical length is defined as $L_{\text{typ}} = 10\,\text{m}$. In conjunction with $\sigma_\epsilon = 0.2\,\text{rad}\,\text{s}^{-1}$, all prior parameters are defined (cf. Section 7.2). For the ESS threshold in the parameter domain, a value of $L_{\theta\text{th}} = 0.3 L_\theta$ is chosen.

The computational fluid dynamics (CFD) simulations were performed to provide realistic trajectories and spatial coupling between the trajectories of different agents. The CFD results were obtained by simulating the pipe

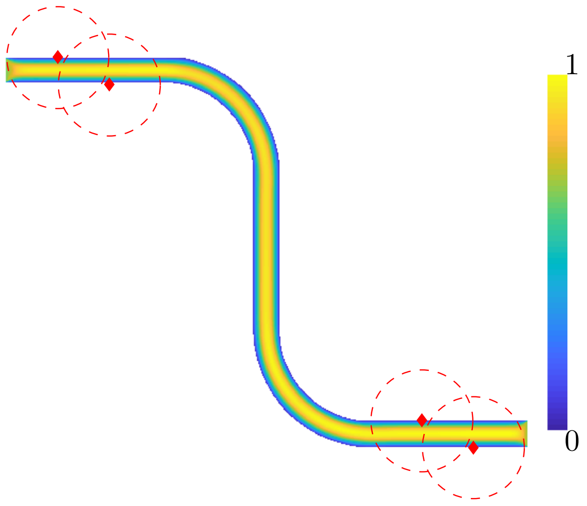


Fig. 3 Simulation environment used in this work. A color scale ranging from blue to yellow is used to show the normalized velocity. Additionally, the locations of the four beacons and their communication range are indicated as red diamonds and dashed circles, respectively

model shown in Fig. 3 using a D2Q9 Lattice Boltzmann method (LBM). The pipe in this model is filled with water at 25 °C and has a length of 10 m and a diameter of 0.50 m. The LBM parameters used for the CFD simulation are listed in Table 2. Details such as the boundary method implemented for the D2Q9 LBM are given in [31], where this method is reported to be accurate to approximately second order.

In the simulation, bearing and distance measurements between agents or beacons $\mathbb{i}$ and $\mathbb{j}$ of the form

$$\boldsymbol{y}_{\mathbb{i},\mathbb{j},k} = \begin{bmatrix} \sqrt{(\mathrm{x}_{\mathbb{i},k} - \mathrm{x}_{\mathbb{j},k})^2 + (\mathrm{y}_{\mathbb{i},k} - \mathrm{y}_{\mathbb{j},k})^2} \cdot (1+\eta_{d,k}) \\ \operatorname{atan2}(\mathrm{y}_{\mathbb{i},k} - \mathrm{y}_{\mathbb{j},k},\ \mathrm{x}_{\mathbb{i},k} - \mathrm{x}_{\mathbb{j},k}) + \eta_{b,k} \end{bmatrix} \tag{41}$$

are considered, where $\boldsymbol{\eta}_k = [\eta_{d,k}, \eta_{b,k}]^\mathsf{T} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$, with $\boldsymbol{\Gamma} = \operatorname{diag}\left(\sigma_d^2, \sigma_b^2\right)$, where σ_d and σ_b are the standard deviations of the noise in the distance and the bearing measurements, respectively. The multiplicative model for the distance measurements accommodates the observation that distance measurements made with respect to farther agents are less accurate.

The full measurement vector for agent $\mathbb{i}$ is then obtained as follows:

$$\boldsymbol{y}_{\mathbb{i},k} = [\ldots \quad \boldsymbol{y}_{\mathbb{i},\mathbb{j},k}{}^\mathsf{T} \quad \ldots]^\mathsf{T}. \tag{42}$$

The simulation results presented below were obtained using the following procedure and settings: During the first 20 time steps of the simulation, one agent was inserted per time step, such that after 20 time steps, 20 spatially distributed agents were present. The location at which the agents were inserted along the cross-section of the pipe was chosen randomly for each simulation. In total, 45 time steps were simulated, and the sampling period was set to $T = 1$ s. In total four beacons were present in the environment as illustrated in Fig. 3. Distance and bearing measurements are obtained with near-by agents and beacons only. For simplicity, a circular communication range is assumed that is described though the sensing range that is set to $R_s = 1.5$ m. The following two measurement noise scenarios (MNSs) are considered:

- *MNS 1:* Multiplicative distance measurement noise with a standard deviation of $\sigma_d = 0.04$ and bearing measurement noise with a standard deviation of $\sigma_b = 5°$
- *MNS 2:* Multiplicative distance measurement noise with a standard deviation of $\sigma_d = 0.06$ and bearing measurement noise with a standard deviation of $\sigma_b = 10°$

Table 2 D2Q9 LBM simulation parameters

Parameter	Value
LBM relaxation time	0.515
LBM speed	0.05
Max. LBM iterations	1x10^4
LBM convergence threshold	1x10^{-7}
Viscosity [m^2/s]	0.89×10^{-6}
Density [kg/m^3]	997.05

8.1 Environment uncertainty

In contrast to the other tracking algorithms (cf. Table 3), which make no assumptions regarding the environment to be explored, RFaT requires some environmental information. Theoretically, RFaT uses a priori knowledge of the environment solely to reduce the computational complexity of the estimation and FEM meshing procedures. In other words, a priori knowledge of the environment is exploited to provide fine-grained FEM meshes only where needed. To evaluate the performance and run-time effects of this in case of limited a priori knowledge, an artificial environment uncertainty (EU) is used.

To simulate EU, the procedure presented in Algorithm 2, where a polyline representation of the borderline is deformed, is employed. The reason is that the actual environment is also stored as a polyline, and precise control over the intensity of the deformation is required while ensuring that a similar pipe course as that of the original environment is still present in the deformed environment. The parameter that controls the intensity is denoted by σ_{EU}.

Figure 4 illustrates the effects of the EU on the FEM mesh as well as on the general shape of the assumed environment. The effects of the EU are twofold: First, the dynamics are changed, mostly at the turning points in the S-shaped environment. Second, the grid vertex distribution, which is computed in this work using the method of [32], also changes. For the mean grid $\boldsymbol{\varphi}$, in particular, this change in the vertex distribution results in more vertices being placed closer to the turning points due to the irregular deformation there. In turn, fewer vertices, and thus lower accuracy, are available for the other parts of the environment (e.g., the outlet portion of the pipe). Consequently, the mean estimation will be less accurate in these other locations unless the grid resolution is increased.

8.2 Performance metric: random field error

In addition to the root-mean-squared error (RMSE), an average field estimate is evaluated. To this end, the random field error (RFE) is defined as

$$\mathrm{RFE} = \frac{1}{|\Omega_n|} \int_{\Omega_n} \| z(\boldsymbol{s}) - \widehat{z}(\boldsymbol{s}) \| d\boldsymbol{s}. \tag{43}$$

Because the field models the turn rate in this work, the field $z(\boldsymbol{s})$ visualized in Fig. 5b serves as the ground truth.

Table 3 Algorithm overview

Abbreviation	Proposed in	MPF LA	Time update (TUD)	Extra TUD parameters/ RF variable	Motion model	State variables
MPF	[16]	MCA [16]	$\boldsymbol{f}(\boldsymbol{x}_{\mathbb{i},k}, \boldsymbol{u}_{k,\mathbb{i}} = 0) + \boldsymbol{v}_{\text{MPF}}$ [†]	—	(31)	$[x, y, \upsilon, \phi]$
IPF, $\sum_{\text{MPF}}$	[13]	MCA [16]	$\boldsymbol{f}(\boldsymbol{x}_{\mathbb{i},k}, \boldsymbol{u}_{k,\mathbb{i}}) + \boldsymbol{v}_{\text{MPF}}$ [†]	$\boldsymbol{u}_{k,\mathbb{i}} = \omega_{k,\mathbb{i}}$	(31)	$[x, y, \upsilon, \phi]$
IPF, $\sum_{\text{RF}}$			$\boldsymbol{f}(\boldsymbol{x}_{\mathbb{i},k}, \boldsymbol{u}_{k,\mathbb{i}}) + \boldsymbol{v}_{\text{RF}}$ [‡]			
RFaT	This work	MCA [16]	$\boldsymbol{f}(\boldsymbol{x}_{\mathbb{i},k}, \boldsymbol{u}_{k,\mathbb{i}}) + \boldsymbol{v}_{\text{RF}}$ [†]	$\boldsymbol{u}_{k,\mathbb{i}} = z_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,\mathbb{i}})$	(31)	$[x, y, \upsilon, \phi]$

[†] $\sum_{\text{MPF}} = \text{diag}[0.05\ \text{m}, 0.05\ \text{m}, T^2 0.01\ \text{m/s}^2, T^2 0.7\ \text{rad/s}^2]$ has been found to be the optimal covariance matrix of the zero-mean process noise for the MPF algorithm in results not reported in this work

[‡] $\sum_{\text{RF}} = \text{diag}[0.02\ \text{m}, 0.02\ \text{m}, T^2 0.002\ \text{m/s}^2, T^2 0.01\ \text{rad/s}^2]$ has been found to be the optimal covariance matrix of the zero-mean process noise for the RFaT algorithm in results not reported in this work

8.3 Boxplots

Some of the results presented in the next section are illustrated using boxplots. Boxplots (cf. Fig. 5c) visualize statistical properties using four components: First, a box is drawn that spans the interquartile range (IQR), i.e., extends from the 25th to the 75th percentile. Second, the median is visualized as a red line within the IQR box. Third, whiskers (black) extend to the most extreme points that are not classified as outliers. Fourth and finally, red crosses indicate the outliers, which are those samples that do not lie within $[q_1 - w(q_3 - q_1), q_3 + w(q_3 - q_1)]$, where q_1 and q_3 are the 25th and 75th percentiles, respectively, and w denotes the whisker length, which is set to $w = 1.5$ in this work. The span of the whiskers corresponds to the $\pm\, 2.7\sigma$ range and, thus, to 99.3% coverage in the case of Gaussian data.

8.4 Algorithms

Based on the results presented in the following section, the algorithms and configurations listed in Table 3 are evaluated. In total, two process noise configurations are evaluated which are denoted with Σ_{MPF} and Σ_{RF}. These configurations have been found optimal for MPF and RFaT, respectively, in the used environment based on a parameter sweep whose results are not provided in this work. Since the input-aided particle filter (IPF) algorithm can be understood as a middle ground between the MPF and RFaT algorithms, the IPF algorithm is evaluated using both configurations to assess the precise performance differences with respect to (w.r.t.) both.

9 Numerical results and discussion

All results presented herein were obtained using a Bullx Blade B500 system with an Intel Westmere X5675 CPU running CentOS 7. The simulations are performed using MATLAB. The simulations show averages over 50 simulations.

9.1 MCA-based multiple particle filtering vs. input-aided particle filtering

The first set of results is presented in Fig. 6 and compares the MPF and IPF algorithms for different ACI density variances. This comparison serves two purposes. First, its findings in terms of adequate parameters are used in subsequent simulations. Second, it motivates the need for ACI embeddings, e.g., as per IPF or the proposed RFaT.

The MPF results are shown as dotted lines and do not vary along the x-axis because no input is considered by this method. In the left and right panels, the performance gains due to the use of the IPF algorithm for MNS 1 and MNS 2, respectively, are annotated. Specifically, improvements of up to 69% are achieved with respect to MPF. Notably, different optimal ACI density variances are found depending on the measurement noise intensity, as illustrated by the different MNSs. Because $\sigma_{u,\omega} = 1\text{rad s}^{-1}$ is found to be optimal for MNS 2 and because the

Algorithm 2: Boundary Polyline Deformation

```
input  : Non-closed polyline describing the assumed boundary $\wp \in \mathbb{R}^{N\times 2}$, $[\wp]_i \in \mathbb{R}^2$ being the vertices of the polyline
input  : EU intensity measure $\sigma_{\text{EU}}$
output : Deformed boundary $\tilde{\wp}$
Compute distance matrix $\boldsymbol{\Delta}$, $[\boldsymbol{\Delta}]_{i,j} = \left\| [\wp]_i - [\wp]_j \right\|_2$;
$d_{\text{max}} \leftarrow \max_{i,j\in 1,\ldots,N} [\boldsymbol{\Delta}]_{i,j}$;
$\boldsymbol{\Sigma} \leftarrow \sigma_{\text{EU}}^2 (1 - \boldsymbol{\Delta}/(2d_{\text{max}}))$;
$\widetilde{\boldsymbol{\Sigma}} \leftarrow [\boldsymbol{\Sigma}]_{2:N,2:N} - [\boldsymbol{\Sigma}]_{2:N,1}[\boldsymbol{\Sigma}]_{1,2:N} \Big/ [\boldsymbol{\Sigma}]_{1,1}$        // Condition on first vertex of polyline;
$[\tilde{\wp}]_1 \leftarrow [\wp]_1$;
$[\tilde{\wp}]_{2:N} \sim \mathcal{N}\left([\wp]_{2:N}, \widetilde{\boldsymbol{\Sigma}}\right)$;
```

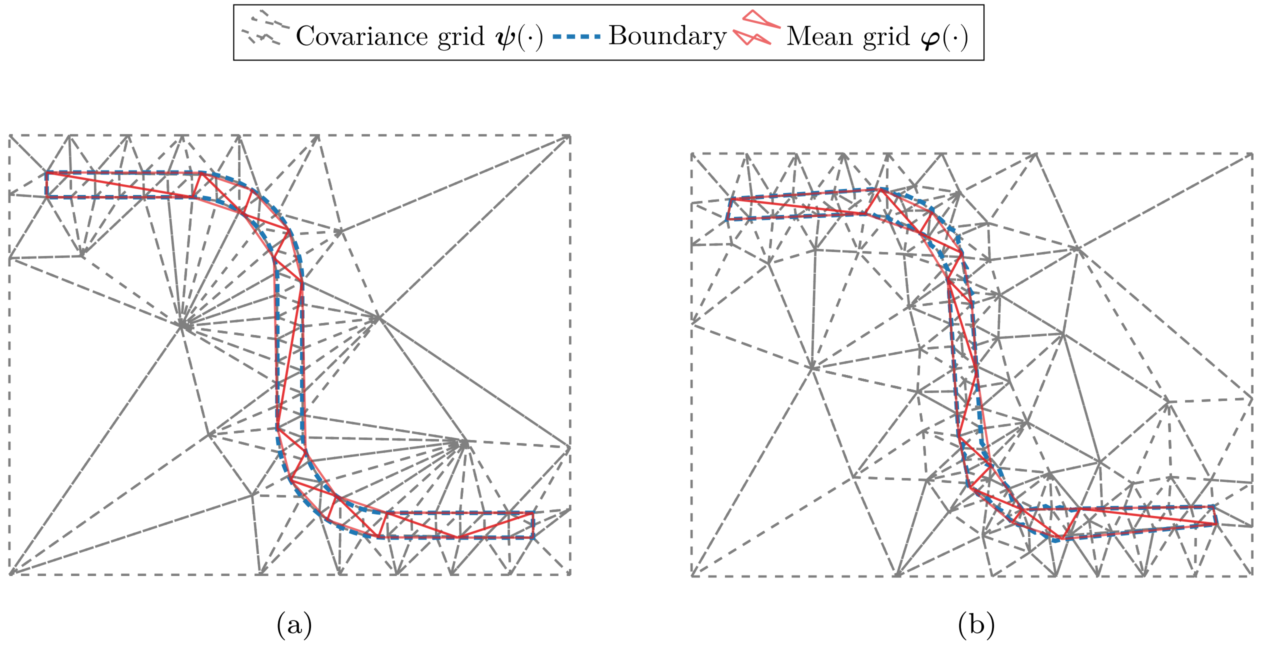


Fig. 4 Meshing examples (grids): **a** shows the actual environment (no EU) and **b** shows the case of $\sigma_{\text{EU}} = 0.5$

difference in performance seen when employing $\sigma_{u,\omega} = 1\,\text{rad}\,\text{s}^{-1}$ for MNS 1 is only −6%, $\sigma_{u,\omega} = 1\,\text{rad}\,\text{s}^{-1}$ is used for the subsequent IPF simulations.

The second set of results is presented in Fig. 7 and compares both methods as a function of the number of state particles per agent (PPA) used, where $\sigma_{u,\omega} = 1\,\text{rad}\,\text{s}^{-1}$ is used for the IPF algorithm. With an increasing PPA value, the MPF algorithm achieves the same performance as the IPF algorithm. However, the IPF algorithm achieves this performance with a significantly lower PPA value and, thus, lower computational complexity. The performance gains when the IPF algorithm is employed are 64% and 57% for MNS 1 and MNS 2, respectively.

A direct run-time comparison between the methods for a fixed computational complexity is presented in Fig. 8, which shows that the IPF algorithm can achieve the same localization error of RMSE = 2 m within approximately $\frac{1}{16}$th and $\frac{1}{9}$th of the time required by the MPF algorithm for MNS 1 and within approximately $\frac{1}{5}$th and $\frac{1}{3}$rd of the time for MNS 2.

In summary, the results presented above motivate the use of ACI as insufficient localization accuracy is reported for MPF. It has been found that already the simplistic ACI scheme used in IPF achieves significant performance gains which are henceforth compared to the scheme proposed in this work.

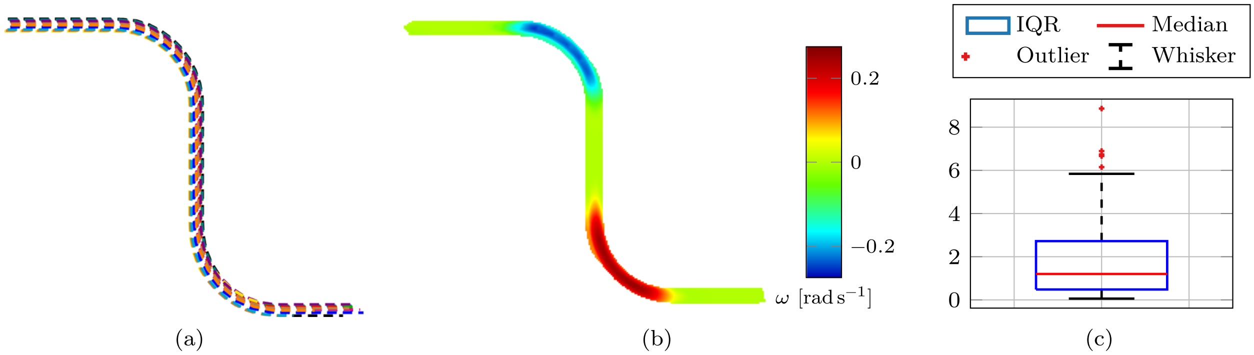


Fig. 5 **a** True example trajectories. **b** Actual turn rates in the environment used in the simulations presented in the following section. At the top turning point of the environment, the turn rate is negative, while it is positive at the bottom turning point. In between, i.e., in the nearly straight section, the turn rate is almost zero. **c** Illustrative boxplot of the exponential distribution $\mathcal{E}(2)$

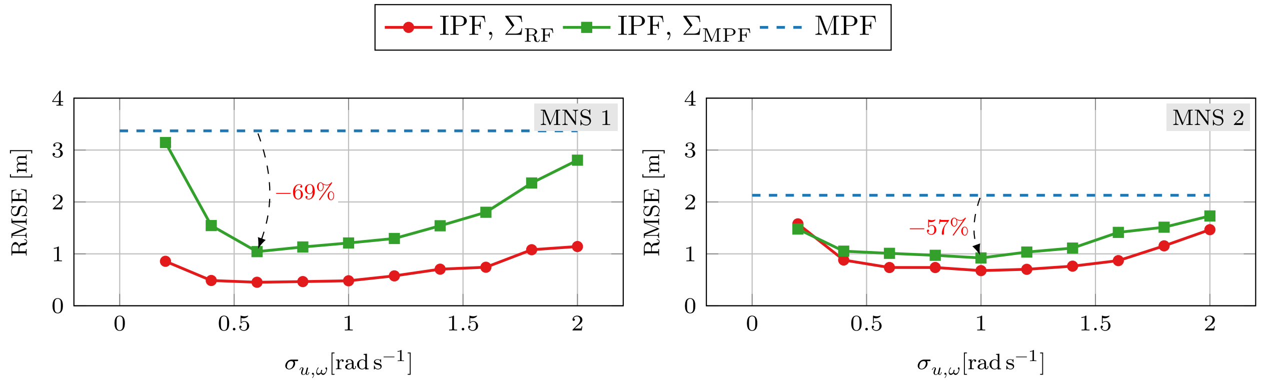


Fig. 6 Comparison between the MPF and IPF algorithms for $L = 100$ and varying input variances. The left panel shows the results for MNS 1, and the right panel shows the results for MNS 2

9.2 Input-aided particle filtering vs. random field-aided tracking

In this subsection, the IPF algorithm, which has been found to yield the better performance in the results presented above, is compared to the RFaT algorithm. For this purpose, the two methods proposed for the time update derivation are first compared (cf. Section 6), of which the better performing is then considered henceforth. The results of the former comparison for MNS 1 are presented in Fig. 9, which shows that in addition to run-time improvements, the sequential sampling method also achieves RMSE reductions of − 46% ($L_\theta = 250$) and − 34% ($L_\theta = 750$) in terms of median performance. Consequently, all subsequently reported results for RFaT are based on sequential sampling.

Recall that RFaT is equipped with a state PPA adaptation scheme (cf. Algorithm 1), thus making a comparison with an algorithm with a fixed state PPA value with a complicated prospect. To overcome this problem, the average PPA value for RFaT is plotted instead. A consequence of the adaption scheme when the initial state PPA value is small can be observed in Fig. 10, which shows that for MNS 2, the state PPA value of RFaT increases such that this value is no smaller than 100 on average.

Of particular interest in all subsequently presented results is the performance of the RFaT algorithm with $L_\theta = 1$. In this case, no PMCMC rejuvenation steps are performed, and thus, only the prior parameters are used. Consequently, this case can be used to assess whether even despite rather inaccurate a priori information, the spatial coupling of the field is able to offer an improvement on the position-independent statistical model used in the IPF algorithm. To this end, it can be noted in Fig. 10 that even without optimization of the sought parameters ($L_\theta = 1$), the RFaT algorithm is able to achieve RMSE values that are 32% and 21% lower for MNS 1 and MNS 2, respectively, compared to those of the IPF algorithm for an average state PPA value of $L = 100$.

Additional improvements through actual optimization of the $\boldsymbol{\theta}$ parameter are possible, as indicated by

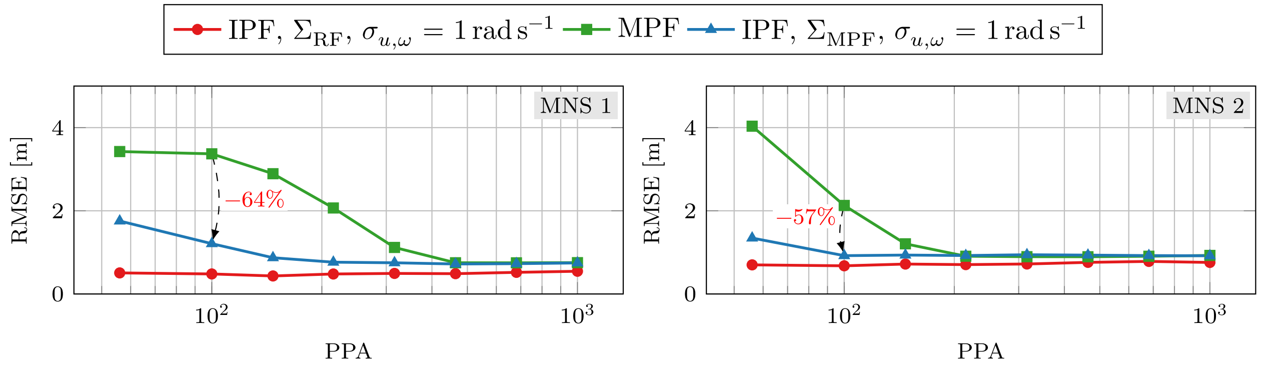


Fig. 7 Comparison between the MPF and IPF algorithms for $\sigma_{u,\omega} = 1\mathrm{rad\ s}^{-1}$ and varying state PPA values. The left panel shows the results for MNS 1, and the right panel shows the results for MNS 2

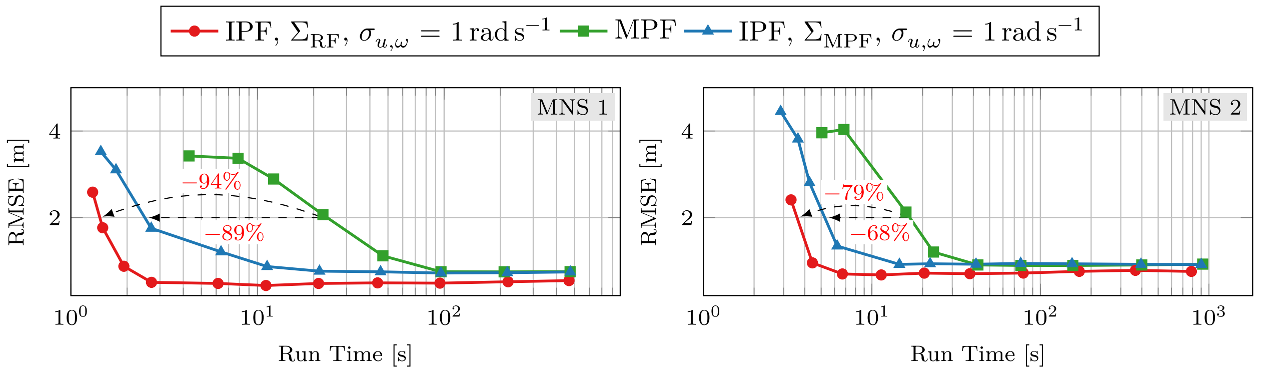


Fig. 8 Run-time comparison between the MPF and IPF algorithms for varying state PPA values. The left panel shows the results for MNS 1, and the right panel shows the results for MNS 2

considering larger L_θ values. The corresponding results are presented in Fig. 11, which compares the best-performing IPF configuration with a series of RFaT configurations. Both panels show that RFaT offers significantly improved localization accuracy at the cost of a higher computational complexity. For example, for MNS 2 (right panel), the RMSE can be reduced by 33% and 50% by employing $L_\theta = 10$ and $L_\theta = 100$ parameter particles, respectively. However, this would require an additional 1100 s and 8100 s, respectively, of computing time.

9.3 RFaT: parameter particle set size and environment uncertainty analysis

In this subsection, the performance of RFaT is analyzed w.r.t. its dependence on the state and parameter particle set sizes. Moreover, the effects of the EU are investigated.

In Fig. 12, a statistical analysis of the impact of the EU on the estimation performance is presented for two main configurations: $L_\theta = 100$ (left half) and $L_\theta = 750$ (right half). The boxplots on white backgrounds represent L_θ configurations without any EU. The boxplots on light gray backgrounds show the results for $\sigma_{\mathrm{EU}} = 0.5$

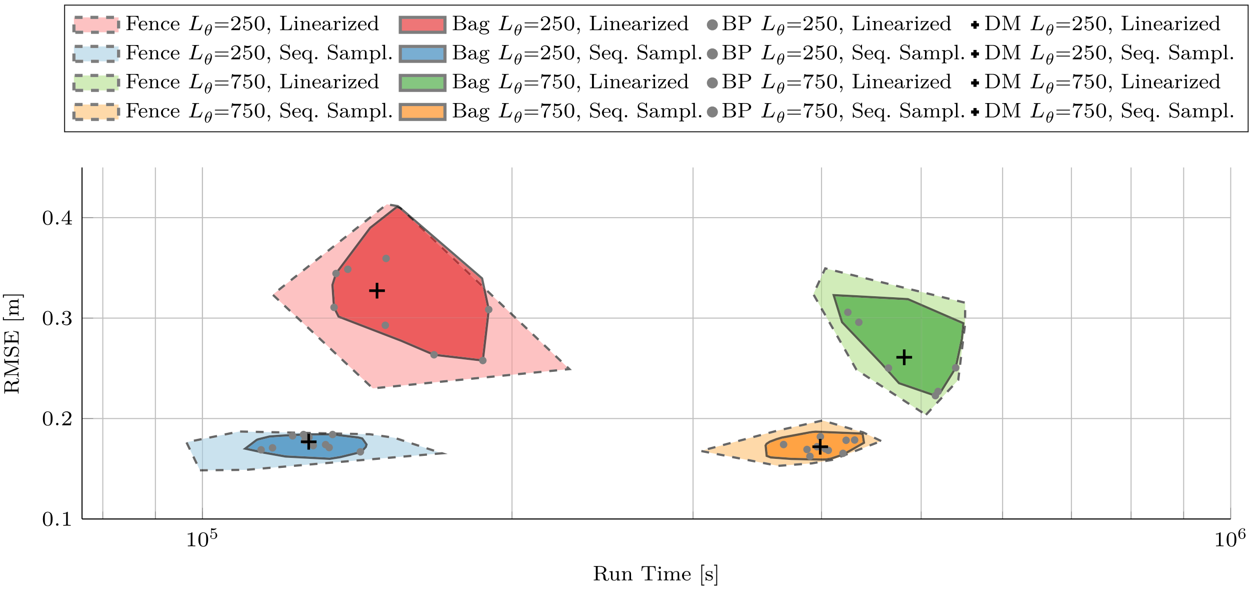


Fig. 9 Run-time and performance comparison between sequential sampling (cf. Section 6.2) and linearization (cf. Section 6.1) for $L = 316$ and MNS 1 as bagplots. Bagplots, as proposed in [33], are a generalization of boxplots for two objectives and consist of the following parts. The bag that contains at most 50% of the points in the dataset. The fence that is an inflated bag by a factor of three. All points outside the fence are considered outliers. The depth median (DM), which is a generalization of median, is the point with the largest Tukey depth. Here, only those data points (DPs) within the bag are visualized for clearness

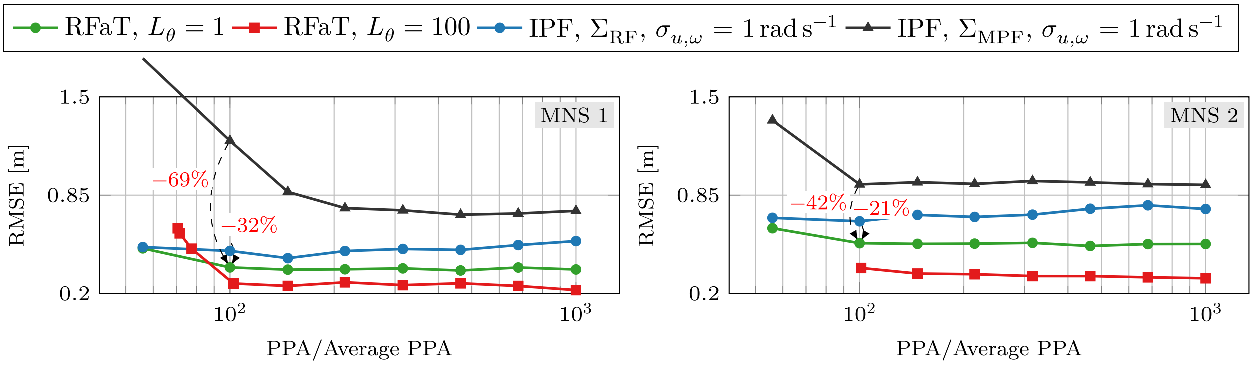


Fig. 10 Comparison between the RFaT and IPF algorithms for varying state PPA values. The left panel shows the results for MNS 1, and the right panel shows the results for MNS 2

but an otherwise identical configuration. The boxplots on darker gray backgrounds are based on a reduced target edge length l_φ for the mean mesh, where a smaller value of l_φ indicates that the number of mean-modeling vertices used, N_β, is larger. In this case, the number of vertices is increased from $N_\beta = 19$ ($l_\varphi = 3.5$) to $N_\beta = 29$ ($l_\varphi = 2.5$). In Fig. 12a, the RMSE performance is depicted whereas Fig. 12b shows the RFE performance.

The results support the notion that the RFaT's demand for a priori information on the environment arises solely from the demand for computational efficiency, which can be achieved through the use of a coarser grid. For $l_\varphi = 2.5$ and $\sigma_{\mathrm{EU}} = 0.5$, the median performance losses (RMSE increases) due to the EU are 0.010 m ($L_\theta = 100$) and 0.014 m ($L_\theta = 750$). The performance loss could likely be further reduced by means of an even finer mesh. Similar observation can be made for the RFE performance which generally increases through EU which, however, approaches the non-EU ($\sigma_{\mathrm{EU}} = 0$) performance through increased mesh resolution. In this regard, the performance loss (RFE increase) through EU ($\sigma_{\mathrm{EU}} = 0.5$) is 0.004 rad s^{-1} ($L_\theta = 100$) and 0.006 rad s^{-1} ($L_\theta = 750$).

Figures 13 and 14 show respectively averaged estimated trajectories and non-averaged RF estimates for in total three different EU and l_φ configurations: In the left panel, the results without EU are given. In the right panel, results for EU with intensity $\sigma_{\mathrm{EU}} = 0.5$ and the same mesh resolution as for the left panel is given. In the center panel, the mesh resolution is increased to compensate for the EU.

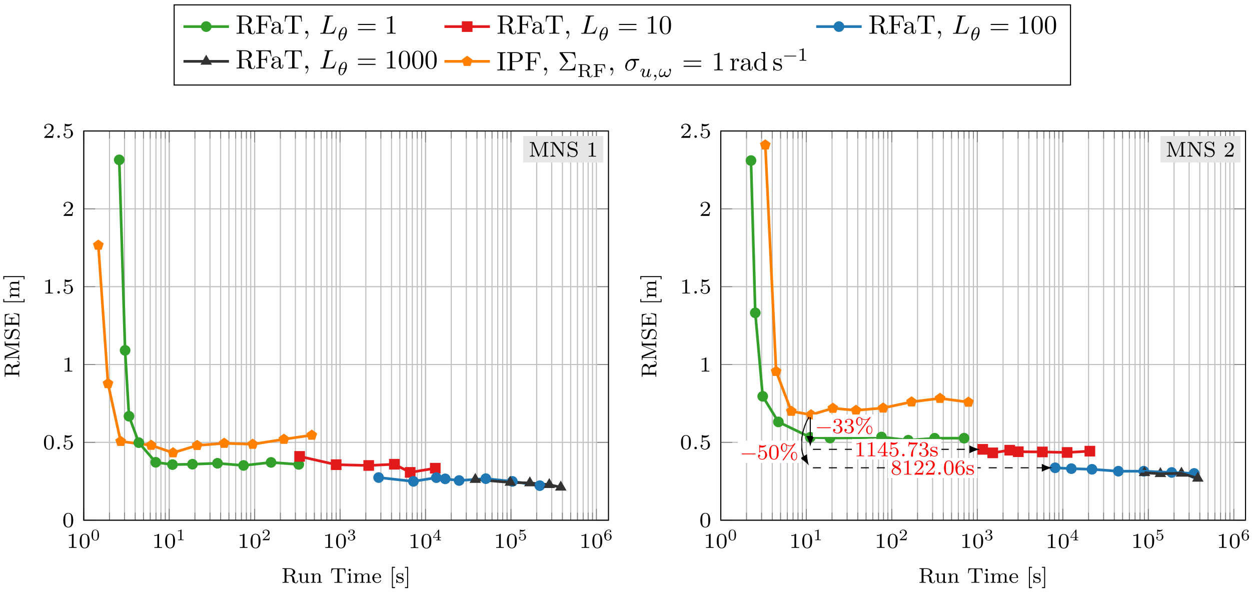


Fig. 11 Run-time comparison between the RFaT and IPF algorithms for various L_θ configurations. The left panel shows the results for MNS 1, and the right panel shows the results for MNS 2

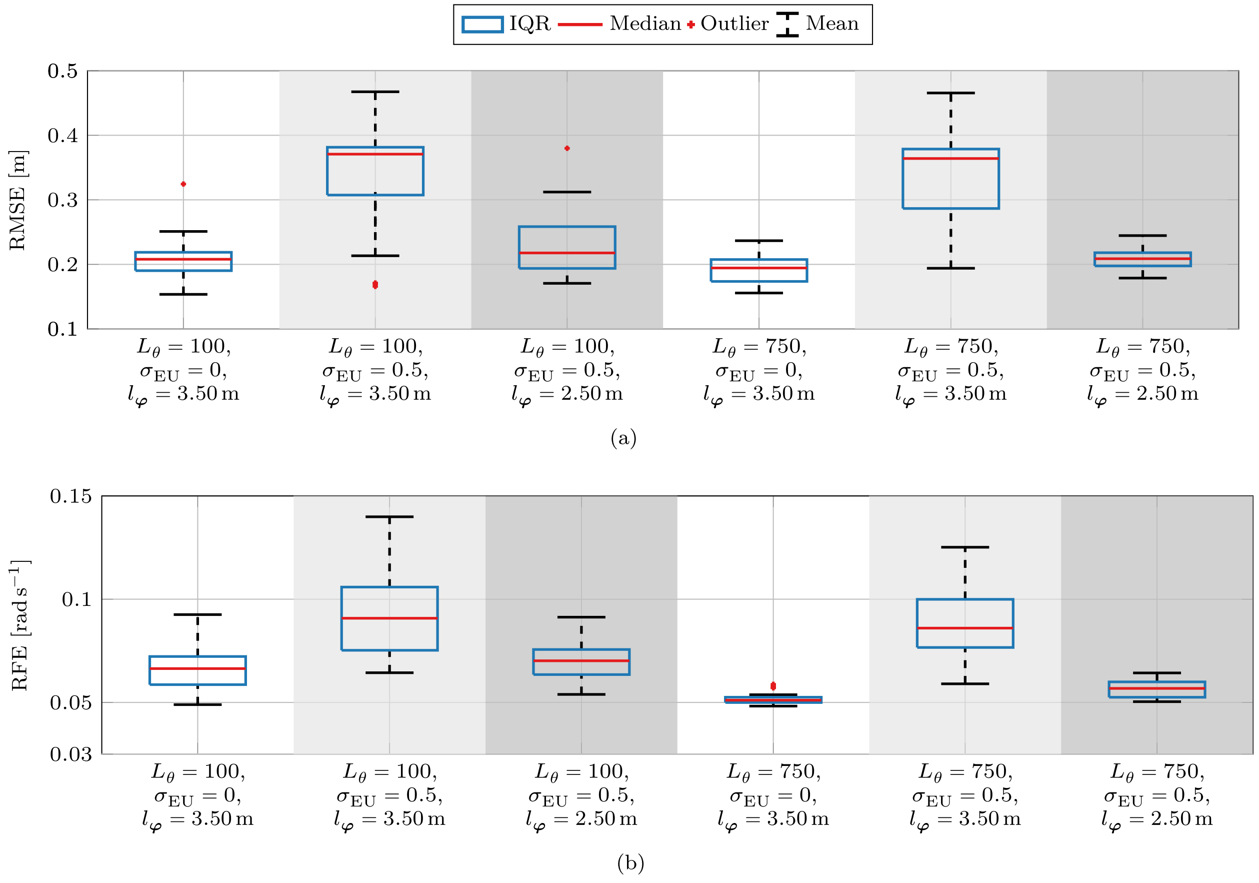


Fig. 12 a, **b** Statistical performance analysis with EU effects for RFaT, MNS 2, and $L = 316$

Underlaid to the estimated trajectories given in Fig. 13 is the assumed environment and the boundary of the actual environment. These results have been selected as they show exemplary the characteristics of RFaT.

It can be noted that several agents are (temporarily) located in the barrier domain according to the estimates, cf. e.g., the bottom turning point Fig. 13 a and b as well as c. This is possible due to the fact that the RF does not impose constraints upon the motion of the agents.

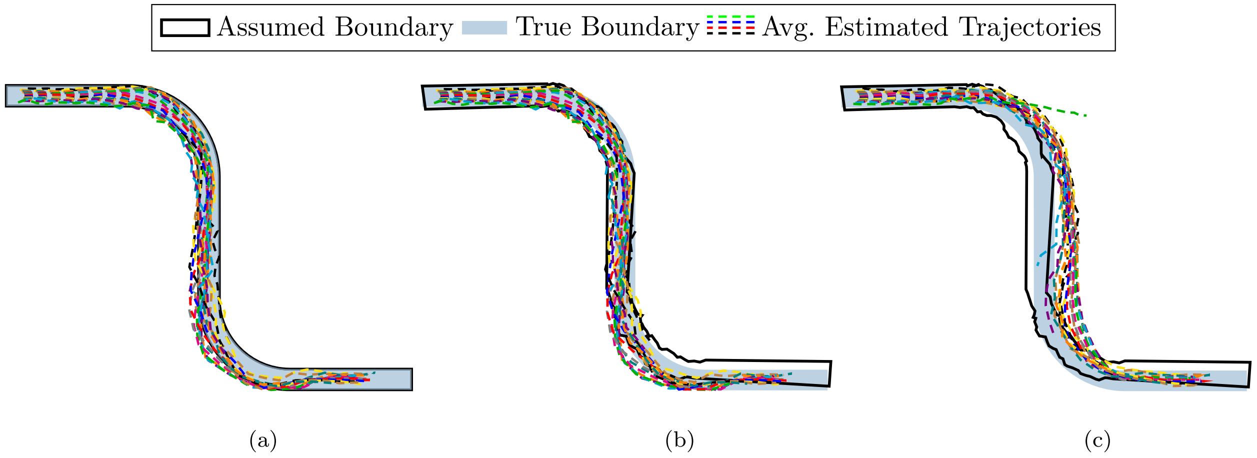


Fig. 13 Averaged trajectories (dashed lines) with different levels of EU and the assumed (meshed) environment in blue for $L_\theta = 750, L = 316$. **a** No EU ($\sigma_{\mathrm{EU}} = 0$), **b**, **c** with EU and assumed environment boundary visualized by black lines

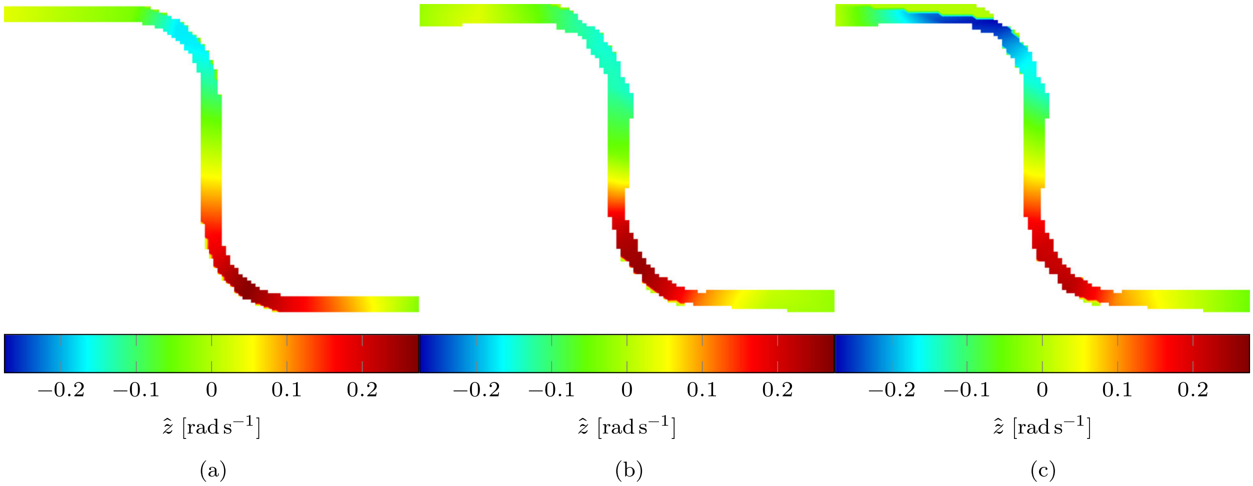


Fig. 14 Estimated RF values for $L_\theta = 750, L = 316$. Individual (i.e., non-averaged) results that have been selected for illustrative purposes are shown. The actual turn rate profile is given in Fig. 5b. **a** RFE = 6.70 × 10^{-3} rad s^{-1}, $\sigma_{\mathrm{EU}} = 0, l_\varphi = 3.50$ m. **b** RFE = 7.50 × 10^{-3} rad s^{-1}, $\sigma_{\mathrm{EU}} = 0.5, l_\varphi = 2.50$ m. **c** RFE = 9.21 × 10^{-3} rad s^{-1}, $\sigma_{\mathrm{EU}} = 0.5, l_\varphi = 3.50$ m

Instead, it is used to steer the agents' direction of motion. Moreover, recall that the covariance grid extents beyond the environment boundary between Ω_n and Ω_b, whereas the mean grid does not cover the barrier domain. More specifically, the agents passing the barrier domain in Fig. 13c are the very first agents inserted into the environment. For these, the RF estimation is still inaccurate due to the online estimation framework, leading to a wrongly estimated turn rate which allows the estimated position to be placed in the barrier domain. The agents inserted afterwards, benefit from more accurate field estimates.

From the trajectories, it can be noticed that RFaT is generally able to infer information about the turning behavior induced by the environment. However, as also visible from the estimated RF given in Fig. 14c, if inaccurate a priori environment information is provided, which is simulated here using EU $\sigma_{\mathrm{EU}} = 0.5$ and no corresponding compensation via increases mesh resolution is adopted (cf. center panels), the turn rate may be underestimated. In Fig. 13c, this leads to increased localization errors.

In summary, Figs. 13 and 14 have shown that EU affects the accuracy of the localization and RF estimation. However, as shown before, increased mesh resolution can effectively limit these effects.

10 Conclusion

In this work, a novel localization framework is presented that extends previous methods (cf. Table 1) in multiple respects. For the first time, the estimation of spatial properties modeled by means of an RF is combined with distance- and/or bearing-measurement-based localization. The objective of the RF model is to capture relevant properties of the underlying environment to improve the localization performance. One such relevant property is the turn rate, which affects the heading direction of the agents. Such models are of particular importance in scenarios with limited beacon coverage, in which the tracking process therefore relies mainly on AAMs rather than ABMs. Correspondingly, more accurate motion modeling is achieved through the RF representation of the turn rate, which compensates for the relative lack of informative measurement information.

Compared to IPF, the method presented here (RFaT) employs a spatial model that has proven to facilitate more accurate modeling of the kinetics imposed on the agents and, thus, more precise localization. This is, among other reasons, achieved through the RF model that considers spatial correlation among the agents' trajectories. The resulting problem addressed in this work is a high-dimensional state and parameter estimation problem that is solved using two layers of SMC methods that comprise of particle filtering and PMCMC methods. Unlike many related works, the scheme presented in this work does not rely on direct field measurements and, thus, is applicable to a wide range of scenarios. This lack of field measurements leads to increased computational complexity of the devised method because it requires the consideration of non-zero-mean fields, which, in turn, significantly increases the parameter space (in the simulations presented in this work, the number of parameters to be estimated is increased by almost tenfold). The proposed algorithm is equipped with an adaptation scheme that allows the number of state particles to be automatically increased on an on-demand basis. This approach not only reduces the configuration complexity but also enables efficient use of computational resources. In the presented results, it has also been shown that the prior information needed within RFaT's FEM approximation is used to reduce the computational complexity. In other words, theoretically, RFaT does not rely on spatial prior information but it is used to reduce the computational complexity.

The presented results have shown that RFaT achieves significant reductions in the localization error of up to 50% compared to IPF. Notably, also IPF achieves significant performance gains compared to related works (RMSE reductions up to 69% compared to MPF). However, these performance gains are achieved at the cost of increased computational complexity. Because the processing is performed by a FC, parallelizing the individual filtering steps for each parameter particle ℓ_θ can play pivotal in reducing the computing time. Since for each ℓ_θ, a separate PF is operated, a corresponding implementation is straightforward.

In future works, we are looking to investigating theoretical convergence criteria and conditions.

Appendix

A Additional background

A.1 Particle filtering

The objective of SMC methods, i.e., of particle filtering is to recursively estimate the posterior distribution

$$p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{y}_{\mathbb{i},1:k}) = \frac{p(\boldsymbol{y}_{\mathbb{i},k}|\boldsymbol{x}_{\mathbb{i},k})p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{y}_{\mathbb{i},1:k-1})}{p(\boldsymbol{y}_{\mathbb{i},k}|\boldsymbol{y}_{\mathbb{i},1:k-1})}, \tag{44}$$

where $\boldsymbol{y}_{\mathbb{i},1:k}$ denotes all measurements of agent $\mathbb{i}$ up to time k. Moreover, with the Chapman-Kolmogorov equation (CKE)

$$\begin{aligned} p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{y}_{\mathbb{i},1:k-1}) = \int & p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{x}_{\mathbb{i},k-1}) \\ & \cdot p(\boldsymbol{x}_{\mathbb{i},k-1}|\boldsymbol{y}_{\mathbb{i},1:k-1})\, d\boldsymbol{x}_{\mathbb{i},k-1} \end{aligned} \tag{45}$$

and

$$p(\boldsymbol{y}_k|\boldsymbol{y}_{\mathbb{i},1:k-1}) = \int p(\boldsymbol{y}_k|\boldsymbol{x}_{\mathbb{i},k})p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{y}_{\mathbb{i},1:k-1})\;\; d\boldsymbol{x}_{\mathbb{i},k}, \tag{46}$$

individual components of the posterior can be given. However, for general nonlinear state space models, these integrals cannot be computed analytically.

In particle filtering, the sought posterior distribution $p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{y}_{\mathbb{i},1:k})$ is approximated using L weighted particles, denoted by $\left\{\left\langle w_{\mathbb{i},k}^{(\ell)}, \boldsymbol{x}_{\mathbb{i},k}^{(\ell)}\right\rangle\right\}_{\ell=1}^{L}$, such that

$$p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{y}_{\mathbb{i},1:k}) \approx \sum_{\ell=1}^{L} w_{\mathbb{i},k}^{(\ell)} \delta\left(\boldsymbol{x}_{\mathbb{i},k} - \boldsymbol{x}_{\mathbb{i},k}^{(\ell)}\right), \tag{47}$$

where $\delta(\cdot)$ is the Dirac delta function. The weights are defined as follows

$$w_{\mathbb{i},k}^{(\ell)} \propto p\left(\boldsymbol{x}_{\mathbb{i},0:k}^{(\ell)}\middle|\boldsymbol{y}_{\mathbb{i},1:k}\right) \Big/ \pi\left(\boldsymbol{x}_{\mathbb{i},0:k}^{(\ell)}\middle|\boldsymbol{y}_{\mathbb{i},1:k}\right), \tag{48}$$

where $\pi\left(\boldsymbol{x}_{\mathbb{i},0:k}\middle|\boldsymbol{y}_{\mathbb{i},1:k}\right)$ denotes a proposal distribution (PD). A PD is used because in most cases, directly sampling from $p(\boldsymbol{x}_{\mathbb{i},0:k}|\boldsymbol{y}_{\mathbb{i},1:k})$ is impossible because it would require solving complex and high-dimensional integrals for which no general analytical solution is known [34].

If the PD is chosen to be factorized such that [35]

$$\pi(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) = \pi(\boldsymbol{x}_0)\prod_{t=1}^{k}\pi(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t}), \tag{49}$$

then the following recursive expression for the weights can be obtained [34]:

$$w_{\mathbb{i},k}^{(\ell)} \propto w_{\mathbb{i},k-1}^{(\ell)} \frac{p\left(\boldsymbol{y}_{\mathbb{i},k}\middle|\boldsymbol{x}_{\mathbb{i},k}^{(\ell)}\right)p\left(\boldsymbol{x}_{\mathbb{i},k}^{(\ell)}\middle|\boldsymbol{x}_{\mathbb{i},k-1}^{(\ell)}\right)}{\pi\left(\boldsymbol{x}_{\mathbb{i},k}^{(\ell)}\middle|\boldsymbol{x}_{\mathbb{i},0:k-1}^{(\ell)}, \boldsymbol{y}_{\mathbb{i},1:k}\right)}. \tag{50}$$

The performance of a PF scheme depends on the choice of the PD $\pi(\cdot)$ and on the number of particles L. Regarding the former, it is known that an incremental variance-optimal PD is given by

$$\pi(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{x}_{\mathbb{i},0:k-1}, \boldsymbol{y}_{\mathbb{i},1:k}) = p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{x}_{\mathbb{i},k-1}, \boldsymbol{y}_{\mathbb{i},1:k});$$

however, in most cases, this PD is not available for sampling [36]. This is because, in the general case, such sampling would require solving an integral without an analytical solution. Consequently, in many cases, the PD is chosen to be

$$\pi(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{x}_{\mathbb{i},0:k-1}, \boldsymbol{y}_{\mathbb{i},1:k}) = p(\boldsymbol{x}_{\mathbb{i},k}|\boldsymbol{x}_{\mathbb{i},k-1}), \tag{51}$$

which further simplifies the PF processing (cf. (50)).

A.2 Sequential Monte Carlo methods for combined state and parameter estimation

Based on the state estimation methods described above, extensions can be devised which can be used for a combined state and parameter estimation. Subsequently, SSMs of kind (2) are considered, where contrary to the previous case, the parameters $\boldsymbol{\theta}$ are not assumed to be known which eventually results in the following alternative description of the dynamic system, where the agent index has been dropped for simplicity

$$\boldsymbol{x}_{k+1} \sim p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{\theta}), \qquad \boldsymbol{y}_k \sim p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{\theta}). \tag{52}$$

While approaches such as state augmentation, where an extended state $\boldsymbol{x}'_k = [\boldsymbol{x}_k, \boldsymbol{\theta}]^\intercal$ in combination with the SMC methods described in the Appendix A.1 are theoretically possible, such approaches are known to be associated with strong convergence problems. This is even the case if artificial dynamics for the time-invariant parameters $\boldsymbol{\theta}$ are adopted $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \boldsymbol{v}_{\boldsymbol{\theta}_k}$, as the PF does not properly explore the parameter space [28, 37].

It is important to note that similar to the motivation for the adoption of particle filtering for state estimation, the estimation of the parameters, e.g., in form of the parameter posterior

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_{1:k}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}_{1:k}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{53}$$

or the joint state and parameter posterior $p(\boldsymbol{x}_k, \boldsymbol{\theta}|\boldsymbol{y}_{1:k})$ is analytically intractable in general. This motivates the use of SMC methods also for the parameter estimation.

To this end, parallel or iterated SMC methods have been proposed, where PFs are used for state estimation that are intertwined with PMCMC method for the parameter estimation [24, 38]. In theses methods, not only state particles $\left\{\left\langle \boldsymbol{x}_k^{(\ell)}, w_k^{(\ell)} \right\rangle\right\}_{\ell=1}^{L}$ are used, but also parameter particles $\left\{\left\langle w_{\theta,k}^{(\ell_\theta)}, \boldsymbol{\theta}_k^{(\ell_\theta)} \right\rangle\right\}_{\ell_\theta=1}^{L_\theta}$. Here, index k for the parameter particles indicates that the current estimate of the parameter may change at time step k despite the fact that the underlying sought parameter is time invariant. To prevent degeneration of the parameter particles, a PMCMC rejuvenation procedure is adopted if the ESS goes below a predefined threshold $L_{\theta\mathrm{th}}$ [28]. The resulting procedure proposed independently by [37] and [28] is henceforth denoted as SMC^2 and exploits, similar to the sequential importance sampling (SIS) methods for state estimation, an iterative batch importance sampling (IBIS) scheme. Details of the SMC^2 are discussed in the subsequent Appendix A.3.

Iterative batch importance sampling The IBIS is a sequential method to approximate the parameter posterior $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:k})$ via a set of particles $\left\{\left\langle w_{\theta,k}^{(\ell_\theta)}, \boldsymbol{\theta}_k^{(\ell_\theta)} \right\rangle\right\}_{\ell_\theta=1}^{L_\theta}$ and uses a procedure similar to the Metropolis-Hastings (MH) algorithm to build a Markov chain where the target distribution equals the parameter posterior. However, contrary to classical MH, which would require a new chain for every time step, IBIS performs importance sampling (IS) to adapt the MH algorithm to time variant systems [39].

Based on a parameter prior $p(\boldsymbol{\theta})$, the procedure initializes with

$$\boldsymbol{\theta}_0^{(\ell_\theta)} \sim p(\boldsymbol{\theta}), \qquad \ell_\theta = 1, \ldots, L_\theta \tag{54}$$

$$w_{\theta,0}^{(\ell_\theta)} = \frac{1}{L_\theta}, \qquad \ell_\theta = 1, \ldots, L_\theta. \tag{55}$$

At subsequent time steps, the parameter weights are updated based on the likelihood, in a similar fashion as for state estimation (cf. (50)) via [39]:

$$w_{\theta,k}^{(\ell_\theta)} \propto w_{\theta,k-1}^{(\ell_\theta)} p\left(\boldsymbol{y}_k \middle| \boldsymbol{y}_{1:k-1}, \boldsymbol{\theta}_k^{(\ell_\theta)}\right), \quad \ell_\theta = 1, \ldots, L_\theta. \tag{56}$$

Similar to resampling in PF-based state estimation to avoid degeneracy, the IBIS also performs a corresponding resampling step equivalent if the ESS goes below a set threshold $L_{\theta\mathrm{th}}$. In this case, new parameter particles are sampled with equal weights. Due to the fact that this step is related to the transition step within the MH algorithm, this procedure is denoted as *resample-move* step. The proposal density in the IBIS for the parameter particles is modeled by a Gaussian kernel $\pi(\cdot|\boldsymbol{\theta}_k) = \mathcal{N}\left(\boldsymbol{\mu}_{\pi(\boldsymbol{\theta}_k)}, \boldsymbol{\Sigma}_{\pi(\boldsymbol{\theta}_k)}\right)$ for simplicity. The mean and covariance matrix of this kernel are computed accordingly to

$$\boldsymbol{\mu}_{\pi(\boldsymbol{\theta}_k)} \equiv \frac{\sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)} \boldsymbol{\theta}_k^{(\ell_\theta)}}{\sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)}}, \tag{57a}$$

$$\boldsymbol{\Sigma}_{\pi(\boldsymbol{\theta}_k)} \equiv \frac{\sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)} \left(\boldsymbol{\theta}_k^{(\ell_\theta)} - \boldsymbol{\mu}_{\pi(\boldsymbol{\theta}_k)}\right)\left(\boldsymbol{\theta}_k^{(\ell_\theta)} - \boldsymbol{\mu}_{\pi(\boldsymbol{\theta}_k)}\right)^{\mathsf{T}}}{\sum_{\ell_\theta=1}^{L_\theta} w_{\theta,k}^{(\ell_\theta)}}. \tag{57b}$$

It is important to note that the likelihood increments $p\left(\boldsymbol{y}_k \middle| \boldsymbol{y}_{1:k-1}, \boldsymbol{\theta}_k^{(\ell_\theta)}\right)$ are typically intractable. To this end, the subsequently defined SMC^2 is used, which couples the IBIS with PFs to approximate this PDF.

A.3 Sequential Monte Carlo squared (SMC^2)

The SMC^2 as proposed in [24] combines the IBIS for parameter estimation with a PF for state estimation. More precisely, for every time step k and parameter particle indexed by ℓ_θ, a PF iteration is performed. Due to this interwoven structure, where a state particle does not only depend on the state parameter index ℓ but also on the parameter particle index ℓ_θ, the state particles are subsequently denoted as $\left\langle w_k^{(\ell,\ell_\theta)}, \boldsymbol{x}_k^{(\ell,\ell_\theta)} \right\rangle$. With this, the likelihood increment needed for the weight update, cf. (56), is approximated as [24]

$$p\left(\boldsymbol{y}_k \middle| \boldsymbol{y}_{1:k-1}, \boldsymbol{\theta}_k^{(\ell_\theta)}\right) = \frac{1}{L} \sum_{\ell=1}^{L} w_k^{(\ell,\ell_\theta)}. \tag{58}$$

Similar to IBIS, SMC^2 performs a PMCMC rejuvenation step in case the ESS in the parameter domain drops below a set threshold. The procedure is briefly summarized below, which is repeated for every parameter particle ℓ_θ.

1. Using kernel (57), a parameter candidate $\tilde{\boldsymbol{\theta}}^{\ell_\theta} \sim \pi(\cdot|\boldsymbol{\theta}_k)$ is sampled.
2. With this parameter particle, a PF is operated to obtain the state particles $\tilde{\boldsymbol{x}}_{1:k}^{(1:L,\ell_\theta)}$
3. Accept the move, as per MH algorithm, to replace the old particles by candidates with probability

$$\min\left(\frac{p\left(\tilde{\boldsymbol{\theta}}^{(\ell_\theta)}\right)\widehat{p}\left(\boldsymbol{y}_{1:k}\middle|\tilde{\boldsymbol{\theta}}^{(\ell_\theta)}\right)\pi\left(\boldsymbol{\theta}_k^{\ell_\theta}\middle|\tilde{\boldsymbol{\theta}}^{(\ell_\theta)}\right)}{p\left(\boldsymbol{\theta}_k^{(\ell_\theta)}\right)\widehat{p}\left(\boldsymbol{y}_{1:k}\middle|\boldsymbol{\theta}_k^{(\ell_\theta)}\right)\pi\left(\tilde{\boldsymbol{\theta}}^{(\ell_\theta)}\middle|\boldsymbol{\theta}_k^{\ell_\theta}\right)}, 1\right). \tag{59}$$

Finally, following the MH algorithm, the set of accepted particles as well as the acceptance ratio, i.e., the relative number of accepted candidates per rejuvenation step is calculated.

The complete SMC^2 for the general case is given in [24] and skipped here for brevity.

Remark 7 *As detailed in the proposed algorithm which builds upon SMC^2 (cf. Algorithm 1 Line 20), the SMC^2 facilitates automatic adaptation of the number of state particles L to improve the overall estimation. In case the number of particles is increased, a PF is operated with the increased number of particles for every parameter particles to eventually replace the current set of particles* [24].

A.4 General spatial random field

A RF is a spatial random process whose domain is a subset of the Euclidean space. A realization of such a field is henceforth denoted as $z(\boldsymbol{s})$, where $\boldsymbol{s} \in \Omega$.

Definition 1 (Random Field, RF, [40]) *A random field $z : \Omega \rightarrow \mathbb{R}$ is a stochastic process with a spatial domain $\Omega \subset \mathbb{R}^{n_\Omega}$:*

$$\{z(\boldsymbol{s}) | \boldsymbol{s} \in \Omega\}. \tag{60}$$

While theoretically also temporal and multivariate fields can be defined, in this work, only two-dimensional time-invariant scalar fields are considered. Moreover, due to their model simplicity, Gaussian RFs are of particular interest. They are defined as follows.

Definition 2 (Gaussian Random Field, [41]) *A Gaussian RF is a RF that possesses finite-dimensional distributions that are all multivariate Gaussian. The statistics of a Gaussian RF $z(\boldsymbol{s})$ are completely described by their mean and covariance functions:*

$$\boldsymbol{\mu}_z(s) \equiv \mathbb{E}[\, z(\boldsymbol{s})] \,, \tag{61a}$$

$$Cov_z[\, \boldsymbol{s}_1, \boldsymbol{s}_1] \equiv \mathbb{E}\left[[\, z(\boldsymbol{s}_1) - \boldsymbol{\mu}_z(\boldsymbol{s}_1)]^\intercal \, [\, z(\boldsymbol{s}_2) - \boldsymbol{\mu}_z(\boldsymbol{s}_2)] \right], \tag{61b}$$

where the covariance function must be a positive definite function, cf. Definition 3 *below.*

Definition 3 (Positive definite covariance function, [42]) *A covariance function is positive definite, if it satisfies*

$$\sum_{j,k=1}^{n} c_j c_k \mathrm{Cov}_z[\, s_j, s_k] \geq 0 \tag{62}$$
$$\forall_n \in \mathbb{N}, \forall \boldsymbol{s}_1 \ldots \boldsymbol{s}_n \in \Omega, \forall c_1 \ldots c_n \in \mathbb{R}.$$

For model and computational simplicity, stationary and isotropic RFs are considered, which respectively are defined as

Definition 4 (Stationary Gaussian RFs [42]) *A real-valued Gaussian RF is called stationary if the covariance depends only on the difference between two positions $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ such that*

$$\mathrm{Cov}_z[\, \boldsymbol{s}_1, \boldsymbol{s}_2] = \mathrm{Cov}_z[\, \boldsymbol{s}_1 - \boldsymbol{s}_2] \,. \tag{63}$$

Definition 5 (Isotropic Gaussian random field [42]) *A real-valued Gaussian RF is called isotropic if the covariance is only dependent on the distance between two positions such that*

$$\mathrm{Cov}_z[\, \boldsymbol{s}_1, \boldsymbol{s}_2] = \mathrm{Cov}[\, \|\boldsymbol{s}_1 - \boldsymbol{s}_2\|_2] \,. \tag{64}$$

Both properties can, respectively, be understood as invariance properties under translation, rotation, and reflection. Moreover, it is important to note that isotropic Gaussian RFs are always stationary.

With these definitions and assumptions, the covariance of a Gaussian RF can be parameterized efficiently, e.g., through the Matérn covariance function discussed subsequently.

A.5 Matérn covariance function

A commonly used approach to model the covariance of RFs is based on the Matérn model which parameterizes the covariance as follows.

Definition 6 (Matérn covariance function, [17]) *The Matérn covariance between two points $\boldsymbol{s}_1, \boldsymbol{s}_2$ is defined by their Euclidean distance $d = \|\boldsymbol{s}_1 - \boldsymbol{s}_2\|$ such that*

$$\mathrm{Cov}_z[\, \boldsymbol{s}_1, \boldsymbol{s}_2] \equiv \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)} (\kappa d)^\nu K_\nu(\kappa d), \tag{65}$$

where $\Gamma(\nu)$ denotes the Gamma function and K_ν is the modified Bessel function of the second kind and order ν.

The Matérn model provides a good compromise between flexibility and parameter reduction, as only three parameters are required to model the field. The parameters are denoted as marginal variance ($\sigma^2 = \mathbb{V}[\, z(\boldsymbol{s})]$), smoothing parameter ($\nu \in \mathbb{R}_{>0}$) and scale ($\kappa \in \mathbb{R}_{>0}$). In literature, κ is often also expressed via $\kappa = \sqrt{8\nu} \,/\, \rho$, where ρ measures how quickly the correlations of the random field decays with spatial distance. For this reason, ρ is also denoted as the range of the field [17, 42].

A.6 Gaussian Markov random field

A special variant of the general RF defined above is the GMRF which combines the Gaussian and Markovian property. It is of particular importance in this work as it aims to reduce the computational complexity that is associated with RF regression for non-Markovian fields.

Definition 7 (Gaussian Markov random field, GMRF, [43]) *A random vector* $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{N_\alpha}]$ *is called a GMRF with respect to a simple undirected graph* $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ *with mean* $\boldsymbol{\mu} \in \mathbb{R}^{N_\alpha}$ *and precision matrix* $\boldsymbol{Q} \succ 0, \boldsymbol{Q} \in \mathbb{R}^{N_\alpha \times N_\alpha}$, *if and only if it is given by* $p(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{Q}^{-1})$ *and it holds that*

$$\langle i,j \rangle \in \mathcal{E} \Leftrightarrow Q_{ij} \neq 0, \quad \forall i \neq j, \tag{66}$$

where $\mathcal{V} = \{1, \ldots, N_\alpha\}$ *are the vertices of the graph.*

It is important to note that (66) describes the Markov property of $\boldsymbol{\alpha}$ which is summarized in Definition 8.

Definition 8 (Markov property, conditional independence, [43]) *Two random variables (RVs)* x *and* y *are conditional independent given a third RV* z*, if and only if*

$$p(x, y|z) = p(x|z)p(y|z). \tag{67}$$

The GMRF is best understood graphically, where each component of the random vector $\boldsymbol{\alpha}$ corresponds to a graph vertex. The graph, in turn, can be interpreted as a not necessarily regular grid, where the edges describe that two components of that random vector are conditionally dependent.

As mentioned above, computational reasons speak in favor of the GMRF compared to regular Gaussian RFs. In particular, if the precision matrix $\boldsymbol{Q}$ is sparse, computationally efficient processing of the GMRF is possible [17].

Remark 8 *The GMRF as per Definition 7 corresponds to a weak solution of a stochastic partial differential equation driven by white Gaussian noise which describes a Gaussian RF with Matérn covariance function.*

Nomenclature

$\lVert x \rVert_2$	Euclidean norm of vector x
$\mathbf{0}_{n\times m}$	All zero matrix of dimension n × m
$\mathcal{A}$	Set of agents
$\boldsymbol{\beta}_k$	Basis function weight vector at time step k for mean modeling
β_k	Basis function weight for basis function k for mean modeling
$\boldsymbol{c}_{\boldsymbol{u}_k}(\boldsymbol{x}_k)$	Constant part of the linearized transition function
$\lvert \cdot \rvert$	Cardinality of set or tuple ·
$\mathrm{Cov}_p[\cdot]$	Covariance matrix with respect to density p
$\delta(x - x_0)$	Dirac delta function centered at x_0
d_e	Extension of the domain to reduce numerical artifacts
$l_\psi(\boldsymbol{s})$	Desired covariance mesh edge length
$l_\varphi(\boldsymbol{s})$	Desired mean mesh edge length
$d_{\Omega_n}(\boldsymbol{s})$	Smallest Euclidean distance between $\boldsymbol{s}$ and the Ω_n
$\mathcal{E}_\mathcal{G}$	Edge set of graph $\mathcal{G}$
$\mathbb{E}_p[\cdot]$	Expectation operator with respect to density p
ϵ	Random field measurement noise
$\mathcal{E}(\lambda)$	Exponential distribution with rate λ
$\boldsymbol{f}(\cdot)$	State evolution function
$\tilde{\boldsymbol{f}}(\cdot)$	Linearized state evolution function
$\mathcal{G}$	Graph
$\boldsymbol{h}(\cdot)$	Measurement function
$\mathbb{i}, \mathbb{j}, \mathbb{k}$	Some agent indices
K_ν	Modified Bessel function of the second kind and order ν
L	Number of particles
ℓ	Particle index
l_φ^*	The mean mesh edge length
l_ψ^*	The covariance mesh edge length
L_θ	Number of parameter particles
ℓ_θ	Parameter particle index
$L_{\theta,\mathrm{th}}$	Effective Sample Size threshold for parameter particles
L_{typ}	Typical length of the domain of interest
$[\boldsymbol{X}]_{i,j}$	(i,j)th element of matrix $\boldsymbol{X}$
$\boldsymbol{\mu}_{\pi(\boldsymbol{\theta})}$	Proposal mean for parameter particle candidates
∇	Gradient
N_α	Number of grid vertices
n_α	Index of a grid vertex
N_β	Number of grid vertices for mean modeling
n_β	Index of a grid vertex for mean modeling
$\boldsymbol{\eta}$	Measurement noise vector
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
n_Ω	Dimension of the domain of the random field
$\boldsymbol{\nu}$	Process noise vector
Ω	Domain of the random field
Ω_k	Subdomain k of the random field
$\lvert \Omega_n \rvert$	Area of the fluid-carrying domain Ω_n within the environment
$p(\cdot)$	Probability density function or probability mass function
$\boldsymbol{\varphi}(\boldsymbol{s})$	Basis function vector evaluated at position $\boldsymbol{s}$
$\varphi_{n_\beta}(\boldsymbol{s})$	Basis function n_β evaluated at position $\boldsymbol{s}$ for mean modeling
$\pi(\cdot)$	Proposal density
$\boldsymbol{\psi}(\boldsymbol{s})$	Basis function vector evaluated at position s
$\psi_n(\boldsymbol{s})$	n-th basis function evaluated at position s
$\boldsymbol{Q}(\boldsymbol{\theta})$	Precision matrix for a given $\boldsymbol{\theta}$
r	Range parameter of the covariance function according to [18]
R_s	Agent sensing range
$\boldsymbol{s}$	Position in the random field domain

$\boldsymbol{s}_{\boldsymbol{u}_k}(\boldsymbol{x}_k)$	Slope part of the linearized transition function
$\boldsymbol{\Sigma}_v$	Process noise covariance matrix
σ_ϵ	Random field measurement noise standard deviation
σ_{EU}	Standard deviation of the boundary noise
$\boldsymbol{\Sigma}_{\pi(\boldsymbol{\theta})}$	Proposal covariance for parameter particle candidates
$\sigma_{u,\omega}$	Marginal standard deviation of turn rate input u_ω
$\sigma_{\dot{v}}$	Process noise covariance matrix element describing acceleration
$\sigma_{\dot{\phi}}$	Process noise covariance matrix element describing change of heading angle
σ_x	Process noise covariance matrix element describing change in Cartesian x position
σ_y	Process noise covariance matrix element describing change in Cartesian Y position
s_{Ω_b}	Slope factor describing the increase in distance between mesh points in Ω_b
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
T	Sampling, i.e., measurement period
$\boldsymbol{\theta}$	State Space Model parameter
$\widehat{\boldsymbol{\theta}}_k$	Parameter estimate at time step k
$\tilde{\boldsymbol{\theta}}_k^{(\ell_\theta)}$	Parameter particles candidate ℓ_θ at time step k
$\boldsymbol{\theta}_k^{(\ell_\theta)}$	Parameter particles ℓ_θ at time step k
u_k^0	Development point of control input
$\boldsymbol{u}_{\mathbb{i},k}$	Control Input vector of agent $\mathbb{i}$ at time instant k
$\mathcal{V}_{\mathcal{G}}$	Vertex set of graph $\mathcal{G}$
$\phi_{\mathbb{i},k}$	Heading angle of agent $\mathbb{i}$ at time step k
$\upsilon_{\mathbb{i},k}$	Speed of agent $\mathbb{i}$ at time step k
$\omega_{\mathbb{i},k}$	Turn rate of agent $\mathbb{i}$ at time step k
$\mathbb{V}_p[\,\cdot\,]$	Variance with respect to density p
$\boldsymbol{\alpha}$	Gaussian Markov Random Field random vector
w	Particle weight
$w_{\theta,k}^{(\ell_\theta)}$	Parameter particle weight ℓ_θ at time step k
$\{x_i\}_i$	Collection, i.e. set, of all x_i
$\boldsymbol{x}_{\mathbb{i},k}$	State vector of agent $\mathbb{i}$ at time instant k
$\mathrm{x}_{\mathbb{i},k}$	Cartesian x coordinate of agent $\mathbb{i}$ at time step k
$\hat{\boldsymbol{x}}_{\mathbb{i},k}$	Estimate of state vector of agent $\mathbb{i}$ at time instant k
$\boldsymbol{y}_{\mathbb{i},k}$	Measurement vector of agent $\mathbb{i}$ at time instant k
$\mathrm{y}_{\mathbb{i},k}$	Cartesian y coordinate of agent $\mathbb{i}$ at time step k
$\mathcal{Z}_{\neg\mathbb{i},k}$	All state vectors of agents other than $\mathbb{i}$ at time instant k
$z(\boldsymbol{s})$	Random field evaluated at the position $\boldsymbol{s}$
$\widehat{z}(\boldsymbol{s})$	Estimate of the random field at location $\boldsymbol{s}$

Abbreviations

AAM: Agent-to-agent measurement; ABM: Agent-to-beacon measurement; ACI: Artificial control input; AR: Acceptance ratio; AWGN: Additive white Gaussian noise; BHM: Bayesian hierarchical model; CFD: Computational fluid dynamics; CI: Control input; CKE: Chapman-Kolmogorov equation; ESS: Effective sample size; EU: Environment uncertainty; ExpMax: Expectation maximization; FC: Fusion center; FEM: Finite element method; GMRF: Gaussian Markov random field; GPS: Global Positioning System; i.i.d.: Independent and identically distributed; IBIS: Iterative batch importance sampling; IPF: Input-aided particle filter; IQR: Interquartile range; IS: Importance sampling; LA: Likelihood approximation; LAP: Likelihood approximation problem; LBM: Lattice Boltzmann method; MC: Monte Carlo; MCA: Monte Carlo approximation; MH: Metropolis-Hastings; MNS: Measurement noise scenario; MPF: Multiple particle filter; PD: Proposal distribution; PDF: Probability density function; PF: Particle filter; PMCMC: Particle Markov Monte Carlo chain; PPA: Particles per agent; RF: Random field; RFaT: Random field-aided tracking; RFE: Random field error; RMSE: Root-mean-squared Error; RV: Random variable; SIS: Sequential importance sampling; SLAM: Simultaneous localization and mapping; SMC: Sequential Monte Carlo; SMC2: Sequential Monte Carlo squared; SSM: State space model; w.r.t.: With respect to; WSN: Wireless sensor network

Acknowledgements

We gratefully acknowledge the computational resources provided by the RWTH Compute Cluster from RWTH Aachen University under project RWTH0118.

Authors' contributions

SS performed the simulations, wrote the majority of the manuscript, and conceived the proposed method. TH implemented significant parts of the algorithm and contributed to the manuscript. GA initiated the research and also commented on and approved the manuscript. All authors read and approved the final manuscript.

Authors' information

All authors are with the Chair for Integrated Signal Processing Systems, RWTH Aachen University, Germany. Email addresses: schlupkothen@ice.rwth-aachen.de, tim.heidenblut@rwth-aachen.de, ascheid@ice.rwth-aachen.de

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 665347.

Availability of data and materials

Not applicable

Competing interests

The authors declare that they have no competing interests.

Received: 26 May 2019 Accepted: 28 November 2019
Published online: 13 February 2020

References

1. Y. Wu, E. Mittmann, C. Winston, K. Youcef-Toumi, in *2019 American Control Conference (ACC)*. A practical minimalism approach to in-pipe robot localization, (2019), pp. 3180–3187. https://doi.org/10.23919/acc.2019.8814648
2. E. H. A. Duisterwinkel, E. Talnishnikh, D. Krijnders, H. J. Wörtche, Sensor motes for the exploration and monitoring of operational pipelines. IEEE Trans. Instrum. Meas. **67**(3), 655–666 (2018). https://doi.org/10.1109/TIM.2017.2775404
3. L. Guan, X. Xu, Y. Gao, F. Liu, H. Rong, M. Wang, A. Noureldin, in *Advances in Human and Machine Navigation Systems*. Micro-inertial-aided high-precision positioning method for small-diameter PIG navigation (IntechOpen, 2019). https://doi.org/10.5772/intechopen.80343
4. E. Talnishnikh, J. van Pol, H. J. Wörtche, in *Intelligent Environmental Sensing, Smart Sensors, Measurement and Instrumentation, vol. 13*, ed. by H. Leung, S. Chandra Mukhopadhyay. Micro motes: A highly penetrating probe for inaccessible environments (Springer, Cham, 2015), pp. 33–49. https://doi.org/10.1007/978-3-319-12892-4_2

5. X. R. Li, V. P. Jilkov, Survey of maneuvering target tracking. part i. dynamic models. IEEE Trans. Aerosp. Electron. Syst. **39**(4), 1333–1364 (2003). https://doi.org/10.1109/TAES.2003.1261132
6. P. M. Djuric, M. F. Bugallo, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Multiple particle filtering with improved efficiency and performance, (2015), pp. 4110–4114. https://doi.org/10.1109/ICASSP.2015.7178744
7. NH. Do, J. Mahdi, T. Mehmet, C. Jongeun, Fully bayesian field slam using gaussian markov random fields. Asian J. Control. **18**(4), 1175–1188 (2015). https://doi.org/10.1002/asjc.1237
8. Y. Xu, J. Choi, S. Dass, T. Maiti, Efficient bayesian spatial prediction with mobile sensor networks using gaussian markov random fields. Automatica. **49**(12), 3520–3530 (2013). https://doi.org/10.1016/j.automatica.2013.09.008
9. M. Jadaliha, Y. Xu, J. Choi, N. S. Johnson, W. Li, Gaussian process regression for sensor networks under localization uncertainty. IEEE Trans. Sig. Process. **61**(2), 223–237 (2013). https://doi.org/10.1109/tsp.2012.2223695
10. S. Choi, M. Jadaliha, J. Choi, S. Oh, Distributed gaussian process regression under localization uncertainty. J. Dyn. Syst. Meas. Control. **137**(3), 031002 (2014). https://doi.org/10.1115/1.4028148
11. H. Braham, S. B. Jemaa, G. Fort, E. Moulines, B. Sayrac, Spatial prediction under location uncertainty in cellular networks. IEEE Trans. Wirel. Commun. **15**(11), 7633–7643 (2016). https://doi.org/10.1109/twc.2016.2605676
12. Z. Song, K. Mohseni, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Autonomous vehicle localization in a vector field: Underwater vehicle implementation (IEEE, 2014). https://doi.org/10.1109/iros.2014.6942872
13. S. Schlupkothen, G. Ascheid, in *2018 6th IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. Particle Filter Based Tracking of Highly Agile Wireless Agents via Random Input Sampling, (2018), pp. 227–232. https://doi.org/10.1109/WiSEE.2018.8637343
14. F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, F. Hlawatsch, Cooperative simultaneous localization and tracking in mobile agent networks (2014). arXiv:1403.1824v2
15. F. Daum, J. Huang, in *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652), vol. 4*. Curse of dimensionality and particle filters, (2003), pp. 1979–1993. https://doi.org/10.1109/AERO.2003.1235126
16. S. Schlupkothen, G. Ascheid, Multiple particle filtering for tracking wireless agents via monte-carlo likelihood approximation. EURASIP J. Adv. Sig. Process. (2019). https://doi.org/10.1186/s13634-019-0643-3
17. F. Lindgren, H. Rue, J. Lindström, An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **73**(4), 423–498 (2011). https://doi.org/10.1111/j.1467-9868.2011.00777.x
18. H. Bakka, J. Vanhatalo, J. B. Illian, D. Simpson, H. Rue, Non-stationary gaussian models with physical barriers. Spat. Stat. **29**, 268–288 (2019). https://doi.org/10.1016/j.spasta.2019.01.002
19. E. T. Krainski, V. Gómez-Rubio, H. Bakka, A. Lenzi, D. Castro-Camilo, D. Simpson, F. Lindgren, H. Rue, *Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA*. (Chapman and Hall/CRC, 2018). https://doi.org/10.1201/9780429031892
20. Y. Bar-Shalom, X. R. Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. (John Wiley & Sons, Inc., 2001). https://doi.org/10.1002/0471221279
21. C. Andrieu, A. Doucet, R. Holenstein, Particle markov chain monte carlo methods. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **72**(3), 269–342 (2010). https://doi.org/10.1111/j.1467-9868.2009.00736.x
22. T. B. Schön, A. Wills, B. Ninness, System identification of nonlinear state-space models. Automatica. **47**(1), 39–49 (2011). https://doi.org/10.1016/j.automatica.2010.10.013
23. M. K. Pitt, R. dos Santos Silva, P. Giordani, R. Kohn, On some properties of markov chain monte carlo simulation methods based on the particle filter. J. Econ. **171**(2), 134–151 (2012). https://doi.org/10.1016/j.jeconom.2012.06.004
24. N. Chopin, P. E. Jacob, O. Papaspiliopoulos, SMC2: an efficient algorithm for sequential analysis of state space models. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **75**(3), 397–426 (2012). https://doi.org/10.1111/j.1467-9868.2012.01046.x
25. A. Papavasiliou, Parameter estimation and asymptotic stability in stochastic filtering. Stoch. Process. Appl. **116**(7), 1048–1065 (2006). https://doi.org/10.1016/j.spa.2006.01.002
26. D. Crisan, J. Míguez, Nested particle filters for online parameter estimation in discrete-time state-space markov models. Bernoulli. **24**(4A), 3039–3086 (2018). https://doi.org/10.3150/17-bej954
27. J. Liu, M. West, *Combined Parameter and State Estimation in Simulation-Based Filtering*. (A. Doucet, N. de Freitas, N. Gordon, eds.) (Springer, New York, 2001), pp. 197–223
28. N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, N. Chopin, On particle methods for parameter estimation in state-space models. Stat. Sci. **30**(3), 328–351 (2015). https://doi.org/10.1214/14-sts511
29. D. Simpson, H. Rue, A. Riebler, T. G. Martins, S. H. Sørbye, Penalising model component complexity: A principled, practical approach to constructing priors. Stat. Sci. **32**(1), 1–28 (2017). https://doi.org/10.1214/16-sts576
30. G.-A. Fuglstad, D. Simpson, F. Lindgren, H. Rue, Constructing priors that penalize the complexity of gaussian random fields. J. Am. Stat. Assoc. **114**(525), 445–452 (2019). https://doi.org/10.1080/01621459.2017.1415907
31. Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice boltzmann bgk model. Phys. Fluids. **9**(6), 1591–1598 (1997). https://doi.org/10.1063/1.869307
32. P.-o. Persson, G. Strang, A simple mesh generator in matlab. SIAM Rev. **46**, 2004 (2004)
33. P. J. Rousseeuw, I. Ruts, J. W. Tukey, The bagplot: A bivariate boxplot. Am. Stat. **53**(4), 382–387 (1999). https://doi.org/10.1080/00031305.1999.10474494
34. S. Särkkä, *Bayesian Filtering and Smoothing*. (Cambridge University Press, New York, 2013)
35. A. Doucet, N. Freitas, N. Gordon, *Sequential Monte Carlo Methods in Practice*. (Springer New York, 2001). https://doi.org/10.1007/978-1-4757-3437-9
36. A. Doucet, S. Godsill, C. Andrieu, On sequential monte carlo sampling methods for bayesian filtering. Stat. Comput. **10**(3), 197–208 (2000). https://doi.org/10.1023/A:1008935410038
37. G. Kitagawa, A self-organizing state-space model. J. Am. Stat. Assoc. **93**(443), 1203 (1998). https://doi.org/10.2307/2669862
38. A. Fulop, J. Li, Efficient learning via simulation: A marginalized resample-move approach. J. Econ. **176**(2), 146–161 (2013). https://doi.org/10.1016/j.jeconom.2013.05.002
39. N. Chopin, A sequential particle filter method for static models. Biometrika. **89**(3), 539–552 (2002). https://doi.org/10.1093/biomet/89.3.539
40. E. Vanmarcke, *Random Fields: Analysis And Synthesis (Revised And Expanded New Edition)*. (WORLD SCIENTIFIC, 2010). https://doi.org/10.1142/5807
41. R. J. Adler, *The Geometry of Random Fields*. (Society for Industrial and Applied Mathematics, 2010). https://doi.org/10.1137/1.9780898718980
42. M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging (Springer Series in Statistics)*. (Springer, 1999). https://doi.org/10.1007/978-1-4612-1494-6
43. H. Rue, L. Held, *Gaussian Markov Random Fields*. (Chapman and Hall/CRC, 2005). https://doi.org/10.1201/9780203492024

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.