

RESEARCH

Open Access



Multiple particle filtering for tracking wireless agents via Monte Carlo likelihood approximation

Stephan Schlupkothen*  and Gerd Ascheid

Abstract

The localization of multiple wireless agents via, for example, distance and/or bearing measurements is challenging, particularly if relying on beacon-to-agent measurements alone is insufficient to guarantee accurate localization. In these cases, agent-to-agent measurements also need to be considered to improve the localization quality. In the context of particle filtering, the computational complexity of tracking many wireless agents is high when relying on conventional schemes. This is because in such schemes, all agents' states are estimated simultaneously using a single filter. To overcome this problem, the concept of multiple particle filtering (MPF), in which an individual filter is used for each agent, has been proposed in the literature. However, due to the necessity of considering agent-to-agent measurements, additional effort is required to derive information on each individual filter from the available likelihoods. This is necessary because the distance and bearing measurements naturally depend on the states of two agents, which, in MPF, are estimated by two separate filters. Because the required likelihood cannot be analytically derived in general, an approximation is needed. To this end, this work extends current state-of-the-art likelihood approximation techniques based on Gaussian approximation under the assumption that the number of agents to be tracked is fixed and known. Moreover, a novel likelihood approximation method is proposed that enables efficient and accurate tracking. The simulations show that the proposed method achieves up to 22% higher accuracy with the same computational complexity as that of existing methods. Thus, efficient and accurate tracking of wireless agents is achieved.

Keywords: Wireless sensor networks, Localization, Tracking, Particle filtering, Multiple particle filtering

1 Introduction

Technological advances have opened the door for a broad variety of applications of mobile agents such as robots, drones, and sensory agents, ranging from surveillance and delivery tasks to aerial and underwater exploration. In many of these applications, accurate localization of the agents is essential for successful task completion. However, such localization often cannot be achieved using satellite-based systems such as GPS. Instead, in many cases, a set of local beacons¹ is used, which facilitate range- and/or bearing-based localization.

However, in many scenarios, access to beacons is limited. This may be because their placement is costly or because good beacon locations cannot be determined a priori. Examples of such scenarios include those involving underwater robots, rescue robots operating in cases of disasters such as earthquakes, and the exploration of subterranean fluid-carrying structures and pipes [1, 2]. In particular, the inspection of pipeline systems, e.g., predictive maintenance, is an emerging application of miniature wireless agents. This approach promises to make obsolete the bulky robots that are currently used, which require maintenance shutdowns and thus result in production downtimes. For many of these pipeline systems, the placement of beacons would be extremely costly. In all these application cases, this beacon access limitation can be overcome through the use of numerous cooperating agents conducting range and/or bearing measurements

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

¹Beacons are stationary at positions that are known a priori.

between pairs of agents. In such application cases, on which this paper focuses henceforth, the number of agents to be tracked is fixed and known a priori. Moreover, due to size constraints imposed on the agents and corresponding energy limitations, offline processing of the agents' readings, including the localization, is considered in this work. Access to the agents' readings is, thus, only possible after the agents have been extracted from their operation domain.

Because many agents are needed to accomplish these tasks, computationally efficient methods are needed for localization. Classical (single) particle filters (PFs), for example, 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 [3]. 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 (MPF), in which one particle filter is employed for each agent individually, has been proposed. However, in scenarios with limited access to beacons, such a localization process relies in large part on measurements between agents (agent-to-agent measurements (AAMs)) rather than measurements between agents and beacons (beacon-to-agent measurements). This results in a nonseparable likelihood $p(y_{i,k} | \mathbf{x}_{i,k}, \mathbf{x}_{j,k})$ because each measurement depends on two filters (the filter for agent i 's state $\mathbf{x}_{i,k}$ at time step k and the filter for agent j 's state $\mathbf{x}_{j,k}$ at time step k), whereas the solution to the localization problem requires a likelihood for each individual filter, $p(y_{i,k} | \mathbf{x}_{i,k})$ and $p(y_{j,k} | \mathbf{x}_{j,k})$. Here, $\mathbf{x}_{i,k}$, $y_{i,k}$ denote the state vector and measurements of agent i at timestep k , respectively.

To address this issue, a new approximation of the sought likelihood is proposed that enables higher tracking accuracy with lower computational complexity compared with what can be achieved using the existing methods in the literature.

1.1 Related works

The literature is rich in PF-based localization algorithms; however, these algorithms are mainly used to track single or non-interacting² agents or targets [4–7]. To mitigate the enormous computational complexity that is associated with tracking many wireless agents, the concept of multiple particle filtering has been proposed. The problem of obtaining likelihood information for each individual filter based on a nonseparable likelihood was first discussed in [8]. The authors proposed a technique based on the replacement of the dependency on $\mathbf{x}_{j,k}$ with a dependency on an estimate of agent j 's predicted state, $\hat{\mathbf{x}}_{j,k} =$

$\sum_{\ell=1}^L w_{j,k-1}^{(\ell)} \mathbf{x}_{j,k}^{(\ell)}$, where the $\{w_{j,k-1}^{(\ell)} \mathbf{x}_{j,k}^{(\ell)}\}_{\ell}$ terms represent the corresponding particles and their weights from agent j 's filter. This method is called point estimate approximation (PE) throughout the remainder of this manuscript because it employs the point estimates $\hat{\mathbf{x}}_{j,k}$.

After a series of additional papers on this topic by the same authors [9–13], in [14], the authors proposed an improved version based on an approximation of the measurement function around the mean of agent j 's states via Taylor approximation. This scheme assumes the likelihood to be Gaussian, leaving only its mean and variance to be determined. The mean is derived from a second-order Taylor approximation, while the variance is derived only from first-order approximations. Moreover, the corresponding equations are provided only for additive Gaussian noise and scalar states. Consequently, to apply this scheme to the problem at hand, the generalization derived in this work is required.

Scenarios with uncertainty in the motion and/or the measurement model are covered in [15, 16], for example. In both works, a finite collection of models is considered to address, e.g., the problem of tracking persons who can change their modes of transportation (cycle, bus, train, or car). Instead of resorting to model switching, both works exploit Bayesian model averaging [17] in combination with sequential Monte Carlo methods. Theoretically, the lack of information in $p(y_{i,k} | \mathbf{x}_{i,k}, \mathbf{x}_{j,k})$ about $\mathbf{x}_{j,k}$ could be interpreted as a lack of information about the measurement function such that the different models \mathcal{M}_q correspond to $p(y_{i,k} | \mathbf{x}_{i,k}, \mathbf{x}_{j,k}^{(q)})$, where $\{\mathbf{x}_{j,k}^{(q)}\}_{q=1,\dots,Q}$ is a finite collection of samples of possible agent states. However, this interpretation, and thus also the methods presented in both works, is inappropriate because it would require each agent model to change at every time step. Moreover, an enormous collection of samples ($q = 1, \dots, Q$) would be needed to cover all possible agent states, which would result in high computational complexity.

The literature on related but different scenarios, in which the number of targets or agents to be tracked is unknown, offers a rich variety of solution approaches, which are often based on random finite sets (RFSs) (see, e.g., [18] for a review). In contrast to our scenario, where the number of, e.g., rescue robots deployed in a zone is known a priori, RFS scenarios include the tracking of unknown or hostile aircraft using radar [19], pedestrian tracking [20], active speaker tracking [21], and extended object tracking, in which multiple objects generate an unknown number of reflections that cannot be assigned a priori to the different targets [22]. In our scenario, however, the number of agents is fixed and known, and thus, there is no need to consider RFS theory or the additional overhead associated with the estimation of the number of targets. In the scenario considered in this work, the

²Interaction here refers to the presence of agent-to-agent measurement.

demand for many agents, their cooperative nature, and the resulting agent-to-agent measurements are the predominant factors necessitating the schemes presented in this work.

1.2 Contribution

To overcome the nonseparability of the likelihood, a new method is proposed in this work that does not rely on Taylor approximation and, thus, on the computation of gradients or Hessians. Such a method is of particular interest in the case of nondifferentiable measurement functions. Instead of relying on derivatives, the proposed method uses an approach that exploits Monte Carlo integration, in which the filter output from the previous time step is exploited to obtain likelihood information for the individual filters. This method is henceforth referred to as Monte Carlo approximation (MCA).

Moreover, to enable this scheme to be compared with state-of-the-art methods, the concept presented in [14] is generalized to support multidimensional states. Additionally, it is generalized to multiplicative noise scenarios, which are commonly encountered in the context of distance-based localization, as such scenarios reflect the property that measurements between agents separated by farther distances are subject to stronger noise [23–26]. This generalized algorithm is henceforth referred to as Gaussian approximation (GA).

Additionally, the GA scheme is extended to a full second-order Taylor approximation. This method is henceforth called extended Gaussian approximation (EGA) and is intended to provide approximations that are more accurate than those obtained using GA. An overview of all discussed methods is given in Table 1.

In the presented simulations, it is shown that the method proposed in this work outperforms all three methods considered for comparison, i.e., PE [8], GA (generalized from [14]), and EGA, the last of which exploits full second-order approximations. Moreover, it is shown that the proposed method achieves higher localization accuracy within the same computing time.

1.3 Organization

This work is structured as follows. Section 2 introduces the concept of using MPF for tracking wireless agents. Section 3.1 presents the GA likelihood approximation technique. Sections 3.1.3 and 3.1.4 extend this scheme to present the EGA method. In Section 4, the proposed method is derived and presented. In Section 5, the simulation setup and the performance metric are discussed. In Section 6, numerical results are presented, and all four considered methods are evaluated and compared. Finally, conclusions are drawn in Section 7.

2 Problem introduction and system model

This paper considers scenarios in which many wireless agents are localized using pairwise measurements, such as distance and/or bearing measurements. By the nature of the abovementioned scenarios (cf. Section 1), a predefined and known number of agents are employed which are localized offline, after all agents' readings have been extracted in a fusion center. Consequently, no uncertainty regarding the number of agents is considered. Moreover, access to beacons is assumed to be available but limited, i.e., insufficient for agents to be accurately localized solely by utilizing beacon-to-agent measurements. For this reason, measurements between mobile agents (AAMs) become increasingly relevant.

Because distance and bearing measurements as well as realistic agent motions are nonlinear, PF approaches are considered in this work. Subsequently, the important properties of such approaches are briefly summarized.

In [27], the convergence of Monte Carlo approximation in terms of the mean square error was shown to be of $\mathcal{O}(L^{-1})$, where L is the number of particles. Moreover, it has been shown that the error is (in theory) independent of the state dimensionality. In practice, however, the dimensionality plays a significant role in determining the performance of PF techniques [3, 28]. For example, [3] reported an exponential relationship between the number

Table 1 Features of the considered likelihood approximation schemes

	Applicable to vector states	Based on Gaussian and Taylor approx.	2nd-order approx. for mean	2nd-order approx. for variance	Exploits full particle description
PE, [8]	✓ [§]	×	n.a.	n.a.	×
[14]	×	✓	✓	×	×
GA	✓ [‡]	✓	✓	×	×
EGA	✓	✓	✓	✓ [*]	×
MCA	✓	n.a.	n.a.	n.a.	✓

n.a. not applicable

[§]Uses point estimates only

[†]Equations are given only for scalar states

[‡]Generalization of [14] to vector states

^{*}Extension of GA to a full second-order Taylor approximation

of particles and the state dimensions (“curse of dimensionality”). Due to the high state-space dimensionality, classical PF approaches demand unreasonable resources for tracking many agents. Consequently, the concept of multiple particle filtering, in which one filter is used for each agent to be tracked, is considered instead.

A brief summary of the concepts relevant to particle filter (PF) and multiple particle filter (MPF) is presented below.

2.1 Particle filtering

In an agent tracking scenario, the objective is to sequentially estimate the agents’ states, which are assumed to be Markovian and discrete in time. Correspondingly, the following state-space model for agent i is considered:

$$\mathbf{x}_{i,k+1} = \tilde{f}(\mathbf{x}_{i,k}, \mathbf{v}_k), \quad \mathbf{y}_{i,k} = \tilde{h}(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}, \boldsymbol{\eta}_k), \quad (1)$$

where $\mathbf{x}_{i,k} \in \mathbb{R}^{n_x}$ represents the state of agent i at discrete time k and $\mathbf{y}_{i,k} \in \mathbb{R}^{n_y}$ represents the corresponding measurements of this agent, each of which, by the nature of AAMs, also depends on the state vector of the other agent involved in that measurement, $\mathbf{x}_{j,k}$ (henceforth, the notation j will be used to denote an agent other than i). Moreover, \mathbf{v}_k and $\boldsymbol{\eta}_k$ denote the process noise and measurement noise, respectively. The generally nonlinear functions $\tilde{f}(\cdot)$ and $\tilde{h}(\cdot)$ denote the state evolution and measurement models, respectively, with noise included.

Equivalently, (1) can be written as

$$\mathbf{x}_{i,k} \sim p(\mathbf{x}_{i,k} | \mathbf{x}_{i,k-1}), \quad (2a)$$

$$\mathbf{y}_{i,k} \sim p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k}, \mathbf{x}_{j,k}). \quad (2b)$$

In the context of sequential Bayesian filtering in general and the tracking of wireless agents in particular, the objective is to recursively estimate the posterior distribution

$$p(\mathbf{x}_{i,k} | \mathbf{y}_{i,1:k}) = \frac{p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k}) p(\mathbf{x}_{i,k} | \mathbf{y}_{i,1:k-1})}{p(\mathbf{y}_{i,k} | \mathbf{y}_{i,1:k-1})}, \quad (3)$$

with the Chapman-Kolmogorov equation

$$p(\mathbf{x}_{i,k} | \mathbf{y}_{i,1:k-1}) = \int p(\mathbf{x}_{i,k} | \mathbf{x}_{i,k-1}) \cdot p(\mathbf{x}_{i,k-1} | \mathbf{y}_{i,1:k-1}) d\mathbf{x}_{i,k-1}$$

and

$$p(\mathbf{y}_{i,k} | \mathbf{y}_{i,1:k-1}) = \int p(\mathbf{y}_k | \mathbf{x}_{i,k}) p(\mathbf{x}_{i,k} | \mathbf{y}_{i,1:k-1}) d\mathbf{x}_{i,k}.$$

For general nonlinear state-space models, these integrals cannot be computed analytically.

In sequential importance resampling (particle filtering), the posterior distribution $p(\mathbf{x}_{i,k} | \mathbf{y}_{i,1:k})$ is therefore approximated using L weighted particles, denoted by

$\left\{ \left(w_{i,k}^{(\ell)}, \mathbf{x}_{i,k}^{(\ell)} \right) \right\}_{\ell=1}^L$, such that

$$p(\mathbf{x}_{i,k} | \mathbf{y}_{i,1:k}) \approx \sum_{\ell=1}^L w_{i,k}^{(\ell)} \delta(\mathbf{x}_{i,k} - \mathbf{x}_{i,k}^{(\ell)}), \quad (4)$$

where $\delta(\cdot)$ is the Dirac delta function and $\ell \in \{1, \dots, L\}$ denotes a particle index. The weights are defined as

$$w_{i,k}^{(\ell)} \propto p(\mathbf{x}_{i,0:k}^{(\ell)} | \mathbf{y}_{i,1:k}) / \pi(\mathbf{x}_{i,0:k}^{(\ell)} | \mathbf{y}_{i,1:k}), \quad (5)$$

where $\pi(\mathbf{x}_{i,0:k} | \mathbf{y}_{i,1:k})$ denotes a proposal distribution (PD). A PD is used because in most cases, directly sampling from $p(\mathbf{x}_{i,0:k} | \mathbf{y}_{i,1:k})$ is impossible because it would require solving complex and high-dimensional integrals for which no general analytical solution is known [28].

If the PD is chosen to be factorized such that [29]

$$\begin{aligned} \pi(\mathbf{x}_{0:k} | \mathbf{y}_{1:k}) &= \pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) \pi(\mathbf{x}_{0:k-1} | \mathbf{y}_{1:k-1}) \\ &= \pi(\mathbf{x}_0) \prod_{t=1}^k \pi(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}), \end{aligned} \quad (6)$$

then the following recursive expression for the weights can be obtained [28]:

$$w_{i,k}^{(\ell)} \propto w_{i,k-1}^{(\ell)} \frac{p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k}^{(\ell)}) p(\mathbf{x}_{i,k}^{(\ell)} | \mathbf{x}_{i,k-1}^{(\ell)})}{\pi(\mathbf{x}_{i,k}^{(\ell)} | \mathbf{x}_{i,0:k-1}^{(\ell)}, \mathbf{y}_{i,1:k})}. \quad (7)$$

The performance of a PF scheme depends on the choice of the PD $\pi(\cdot)$ and, as discussed in Section 2, on the number of particles L . Regarding the former, it is known that an incremental variance-optimal PD is given by

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

however, in most cases, this PD is not available for sampling [30]. 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

$$\pi(\mathbf{x}_{i,k} | \mathbf{x}_{i,0:k-1}, \mathbf{y}_{i,1:k}) = p(\mathbf{x}_{i,k} | \mathbf{x}_{i,k-1}), \quad (8)$$

which further simplifies the PF (cf. (7)).

2.2 Multiple particle filtering

Multiple particle filtering aims to mitigate the curse of dimensionality by employing a bank of parallel PFs. An outline of the MPF procedure is given in Algorithm 1. The differences with respect to the particle representations are illustrated in Fig. 1. Most importantly, in classical PFs (cf. Fig. 1a), a single particle is used to represent the states of all M agents, making the computation of the likelihood readily available.

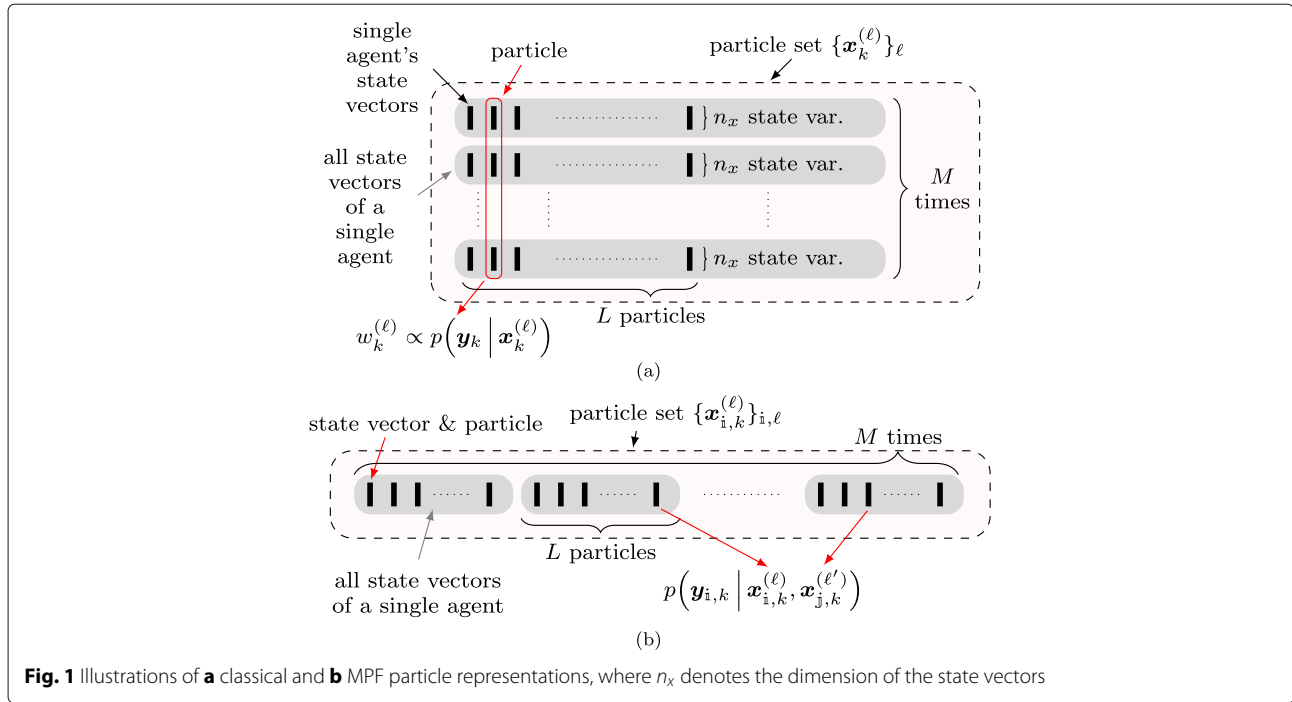


Fig. 1 Illustrations of **a** classical and **b** MPF particle representations, where n_x denotes the dimension of the state vectors

By contrast, in MPF, a separate PF is employed for every agent, resulting in the particle representation illustrated in Fig. 1b. The benefits are a reduced state dimensionality on a per-PF basis and the resulting improved convergence properties. However, the separation of the PFs in MPF causes the likelihood, which is required to compute the weight update, for example (cf. (7)), to depend on two particles representing two parallel, decoupled PFs (i.e., the likelihood is nonseparable). In fact, in the context of distance- and/or bearing-based tracking in our scenario, the likelihood is nonseparable for the following two reasons: first, the necessity of considering agent-to-agent measurements, and second, the fact that both types of measurements inherently depend on the states of both agents conducting the measurements and their corresponding filters.

In terms of likelihoods, the use of MPF in the context of distance- and/or bearing-based tracking is associated with the following problem: In MPF, only $p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k}, \mathbf{x}_{j,k})$ is readily available, where $\mathbf{x}_{i,k}$ and $\mathbf{x}_{j,k}$ are described by particles of different PFs. However, for the weight update in each PF (cf. (7) and line 5 in Algorithm 1), the full likelihood $p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k})$ is required. Therefore, special procedures are required to approximate the required likelihood. The proposal of new procedures for this task is the main topic of this work.

To this end, the following sections discuss three different techniques for approximating $p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k})$ for each PF based on the particle information from all parallel PFs $\{p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k}, \mathbf{x}_{j,k})\}_i$.

3 Gaussian likelihood approximation via Taylor approximation

The general idea presented in [14] is to approximate the sought likelihood $p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k})$ by a Gaussian distribution, leaving only the mean $\mathbb{E}[\mathbf{y}_{i,k} | \mathbf{x}_{i,k}]$ and variance $\mathbb{V}[\mathbf{y}_{i,k} | \mathbf{x}_{i,k}]$ to be computed. In this case, $\mathbf{x}_{j,k}$ is treated as a random variable, and its moments are deduced from the particles.

The scalar measurements are subsequently assumed to result from $\tilde{h}(\cdot)$:

$$y_{i,k} = \tilde{h}(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}, \eta_k), \quad (9)$$

via Taylor expansion of the measurement function around the mean of $\mathbf{x}_{j,k}$. In [14], corresponding equations for scalar states $x_{i,k}$ and $x_{j,k}$ are given under the assumption

Algorithm 1: Outline of multiple particle filtering

- 1 Sample prior $p(\mathbf{x}_{i,0})$ for all agents;
 - 2 **foreach** $k = 1, \dots, K$ **do** // Loop over all time steps
 - 3 **foreach** $i = 1, \dots, M$ **do** // Loop over all agents
 - 4 Time update using PD, cf. (8);
 - 5 Weight update, cf. (7), for which an approximation to $p(\mathbf{y}_{i,k} | \mathbf{x}_{i,k})$ is required;
 - 6 Resampling;
 - 7 **end**
 - 8 **end**
-

of additive measurement noise and for first-order approximations of the variance only. In the remaining part of this section, this approach will be extended to cover the general case of vector states that is required for the described application case. With this approximation of the bootstrap filter, the weights are updated (cf. (7)) using:

$$p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}) = \mathcal{N}\left(y_{i,k} \mid \mathbb{E}[y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}], \mathbb{V}[y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}]\right). \quad (10)$$

Subsequently, the derivations for additive white Gaussian noise (AWGN)- and multiplicative Gaussian noise-corrupted measurement are given in Sections 3.1.3 and 3.1.4, respectively. Note that both approximations are needed due to the assumption of multiplicative Gaussian noise- and AWGN-corrupted distance and bearing measurements, respectively.

3.1 Taylor approximation

First, consider the following noise-free measurements:

$$y_{i,k} = h(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}). \quad (11)$$

Subsequently, a second-order Taylor approximation around the mean of $\mathbf{x}_{j,k}$ is derived for $h(\mathbf{x}_{i,k}, \mathbf{x}_{j,k})$.

$$\begin{aligned} h(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}) &\approx \hat{h}(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}) \\ &\equiv h(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k}) + \mathbf{g}(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k}) \tilde{\mathbf{x}}_{j,k} \\ &\quad + \frac{1}{2} \tilde{\mathbf{x}}_{j,k}^T \mathbf{H}(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k}) \tilde{\mathbf{x}}_{j,k} \end{aligned} \quad (12)$$

where $\tilde{\mathbf{x}}_{j,k} \equiv \mathbf{x}_{j,k} - \bar{\mathbf{x}}_{j,k}$ and $\mathbb{E}[\mathbf{x}_{j,k}] \equiv \bar{\mathbf{x}}_{j,k}$. Moreover, let $\mathbf{g}(\cdot)$ denote the gradient and $\mathbf{H}(\cdot)$ denote the Hessian evaluated at $\bar{\mathbf{x}}_{j,k}$, which are henceforth denoted simply by \mathbf{g} and \mathbf{H} , respectively.

3.1.1 First moment of Taylor approximation

Lemma 1 (Expectation value of a quadratic form [31]) Let $\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}]$ and $\boldsymbol{\Sigma} = \text{Cov}[\mathbf{x}]$; then,

$$\mathbb{E}[\mathbf{x}^T \mathbf{A} \mathbf{x}] = \text{tr}[\mathbf{A} \boldsymbol{\Sigma}] + \boldsymbol{\mu}^T \mathbf{A} \boldsymbol{\mu}, \quad (13)$$

where \mathbf{A} is a real square matrix.

The conditional expectation value of the Taylor approximant $\hat{h}(\cdot)$ (cf. (12)) can be derived by using Lemma 1 in step (†) below:

$$\begin{aligned} \mathbb{E}[\hat{h}(\cdot) | \cdot] &\approx h(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k}) + h(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k}) \text{tr}[\mathbf{H} \boldsymbol{\Sigma}_{\mathbf{x}_{j,k}}] \\ &\quad + \text{tr}[\mathbf{g} \mathbf{g}^T] \end{aligned} \quad (14)$$

where $\boldsymbol{\Sigma}_{\mathbf{x}_{j,k}} \equiv \text{Cov}[\mathbf{x}_{j,k}]$. Consequently, $\mathbb{E}[\tilde{\mathbf{x}}_{j,k}] = \mathbf{0}$.

Lemma 2 (Covariance of a quadratic form [32]) Let \mathbf{x} be multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$; then,

$$\text{Cov}[\mathbf{x}^T \mathbf{A} \mathbf{x}] = 2 \text{tr}[(\mathbf{A} \boldsymbol{\Sigma})^2] + 4 \boldsymbol{\mu}^T \mathbf{A} \boldsymbol{\Sigma} \boldsymbol{\mu}, \quad (15)$$

where $(\mathbf{A} \boldsymbol{\Sigma})^2 = \mathbf{A} \boldsymbol{\Sigma} \mathbf{A} \boldsymbol{\Sigma}$ and \mathbf{A} is a real square matrix.

3.1.2 Second moment of Taylor approximation

Based on Eq. 13, the second noncentral moment can be calculated as follows:

$$\begin{aligned} \mathbb{E}[\hat{h}^2(\cdot) | \cdot] &\approx h(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k})^2 + h(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k}) \text{tr}[\mathbf{H} \boldsymbol{\Sigma}_{\mathbf{x}_{j,k}}] \\ &\quad + \text{tr}[\mathbf{g} \mathbf{g}^T \boldsymbol{\Sigma}_{\mathbf{x}_{j,k}}] + \mathbb{E}[\mathbf{g} \tilde{\mathbf{x}}_{j,k} \tilde{\mathbf{x}}_{j,k}^T \mathbf{H} \tilde{\mathbf{x}}_{j,k} | \cdot] \\ &\quad + \frac{1}{4} \mathbb{E}[(\tilde{\mathbf{x}}_{j,k}^T \mathbf{H} \tilde{\mathbf{x}}_{j,k})^2 | \cdot], \end{aligned} \quad (16)$$

where the last summand in (16) can be simplified using

$$\begin{aligned} \mathbb{E}[(\tilde{\mathbf{x}}_{j,k}^T \mathbf{H} \tilde{\mathbf{x}}_{j,k})^2 | \cdot] &= \text{Cov}[\tilde{\mathbf{x}}_{j,k}^T \mathbf{H} \tilde{\mathbf{x}}_{j,k}] + (\mathbb{E}[\tilde{\mathbf{x}}_{j,k}^T \mathbf{H} \tilde{\mathbf{x}}_{j,k} | \cdot])^2 \\ &\stackrel{(\ddagger)}{=} 2 \text{tr}[(\mathbf{H} \boldsymbol{\Sigma}_{\mathbf{x}_{j,k}})^2] + (\mathbb{E}[\tilde{\mathbf{x}}_{j,k}^T \mathbf{H} \tilde{\mathbf{x}}_{j,k} | \cdot])^2. \end{aligned} \quad (17)$$

In step (‡), Lemma 2 is used, and consequently, the assumption of Gaussian states $\mathbf{x}_{j,k}$ is made. Moreover, under the same assumptions, Appendix A shows that the third-order moment in Eq. 16 vanishes; that is:

$$\mathbb{E}[\mathbf{g} \tilde{\mathbf{x}}_{j,k} \tilde{\mathbf{x}}_{j,k}^T \mathbf{H} \tilde{\mathbf{x}}_{j,k} | \mathbf{x}_{i,k}] = 0.$$

Consequently, the second noncentral moment is described by

$$\begin{aligned} \mathbb{E}[\hat{h}^2(\cdot) | \cdot] &\approx h(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k})^2 + h(\mathbf{x}_{i,k}, \bar{\mathbf{x}}_{j,k}) \text{tr}[\mathbf{H} \boldsymbol{\Sigma}_{\mathbf{x}_{j,k}}] \\ &\quad + \text{tr}[\mathbf{g} \mathbf{g}^T \boldsymbol{\Sigma}_{\mathbf{x}_{j,k}}] + \frac{1}{2} \text{tr}[(\mathbf{H} \boldsymbol{\Sigma}_{\mathbf{x}_{j,k}})^2] + \frac{1}{4} (\text{tr}[\mathbf{H} \boldsymbol{\Sigma}_{\mathbf{x}_{j,k}}])^2, \end{aligned} \quad (18)$$

3.1.3 Synthesis: Additive white Gaussian noise measurements

For AWGN-corrupted measurements of the form:

$$y_{i,k} = h(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}) + \eta_k, \quad \eta_k \sim \mathcal{N}(0, \sigma_\eta^2), \quad (19)$$

the desired moments can be found as follows:

$$\mathbb{E}[y_{i,k} | \mathbf{x}_{i,k}] \approx \mathbb{E}[\hat{h}(\cdot) | \mathbf{x}_{i,k}] \quad \text{cf. (14)}, \quad (20)$$

$$\mathbb{E}[(y_{i,k})^2 | \mathbf{x}_{i,k}] \approx \mathbb{E}[\hat{h}^2(\cdot) | \mathbf{x}_{i,k}] + \sigma_\eta^2 \quad \text{cf. (18)}. \quad (21)$$

The required variance $\mathbb{V}[y_{i,k} | \mathbf{x}_{i,k}]$ can then be easily inferred via

$$\mathbb{V}[y_{i,k} | \mathbf{x}_{i,k}] = \mathbb{E}[(y_{i,k})^2 | \mathbf{x}_{i,k}] - (\mathbb{E}[y_{i,k} | \mathbf{x}_{i,k}])^2. \quad (22)$$

3.1.4 Synthesis: Multiplicative Gaussian noise measurements

Similarly, in the case of multiplicative noise in the form

$$y_{i,k} = h(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}) (1 + \eta_k), \quad \eta_k \sim \mathcal{N}(0, \sigma_\eta^2), \quad (23)$$

the following is found:

$$\begin{aligned} \mathbb{E}[y_{i,k} | \mathbf{x}_{i,k}] &\approx \mathbb{E}[\hat{h}(\cdot) | \mathbf{x}_{i,k}] \quad \text{cf. (14)}, \quad (24) \\ \mathbb{E}[(y_{i,k})^2 | \mathbf{x}_{i,k}] &\approx \mathbb{E}[\hat{h}^2(\cdot) | \mathbf{x}_{i,k}] (1 + \sigma_\eta^2) \quad \text{cf. (18)}. \quad (25) \end{aligned}$$

The variance can again be obtained via (22).

4 Monte Carlo-based likelihood approximation

In the following, a Monte Carlo (MC)-based approximation of the sought likelihood is derived. This approximation is believed to be more accurate than the presented GA-based methods because it does not rely on the assumption that the corresponding likelihood can be modeled by a Gaussian distribution. Moreover, our MCA method does not exploit only the first- and second-order statistics of the particles; instead, it uses the complete information carried by the particles.

The derivation is based on the following approximation of the sought likelihood:

$$\begin{aligned} p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}) &\stackrel{(\mathbb{I})}{\approx} p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{y}_{j,1:k-1}) \\ &= \int p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{x}_{j,k}^{(\ell)}, \mathbf{y}_{j,1:k-1}) d\mathbf{x}_{j,k} \\ &= \int p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{x}_{j,k}^{(\ell)}, \mathbf{y}_{j,1:k-1}) \\ &\quad \cdot p(\mathbf{x}_{j,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{y}_{j,1:k-1}) d\mathbf{x}_{j,k} \\ &= \int p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{x}_{j,k}^{(\ell)}) \underbrace{p(\mathbf{x}_{j,k} | \mathbf{y}_{j,1:k-1})}_{\equiv (\star)} d\mathbf{x}_{j,k}, \quad (26) \end{aligned}$$

where (\star) is given by the Chapman-Kolmogorov equation and consequently can be computed as follows:

$$\begin{aligned} p(\mathbf{x}_{j,k} | \mathbf{y}_{j,1:k-1}) &= \int p(\mathbf{x}_{j,k} | \mathbf{x}_{j,k-1}, \mathbf{y}_{j,1:k-1}) \\ &\quad \cdot p(\mathbf{x}_{j,k-1} | \mathbf{y}_{j,1:k-1}) d\mathbf{x}_{j,k-1}. \quad (27) \end{aligned}$$

Note that step (\mathbb{I}) in (26) assumes that the current measurements of agent i and the collection of past measurements of agent j are approximately conditionally independent given the current state of agent i . Using the

importance sampling description of the previous posterior, $p(\mathbf{x}_{j,k} | \mathbf{y}_{j,1:k-1})$ equals:

$$\begin{aligned} &\int p(\mathbf{x}_{j,k} | \mathbf{x}_{j,k-1}) \cdot \underbrace{p(\mathbf{x}_{j,k-1} | \mathbf{y}_{j,1:k-1})}_{\stackrel{(\S)}{\approx} \sum_{\ell'=1}^L w_{j,k-1}^{(\ell')} \delta(\mathbf{x}_{j,k-1} - \mathbf{x}_{j,k-1}^{(\ell')})} d\mathbf{x}_{j,k-1} \\ &\approx \sum_{\ell'=1}^L w_{j,k-1}^{(\ell')} p(\mathbf{x}_{j,k} | \mathbf{x}_{j,k-1}^{(\ell')}), \quad (28) \end{aligned}$$

where the particle-based posterior description was used in step (\S) . Thus, the sought likelihood can be written as follows (cf. (26)):

$$\begin{aligned} p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}) &\approx \sum_{\ell'=1}^L w_{j,k-1}^{(\ell')} \int p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{x}_{j,k}^{(\ell')}) \\ &\quad \cdot p(\mathbf{x}_{j,k} | \mathbf{x}_{j,k-1}^{(\ell')}) d\mathbf{x}_{j,k}. \quad (29) \end{aligned}$$

Because the integral cannot be analytically computed in general, MC integration techniques are adopted, i.e., the integral is approximated as follows:

$$\begin{aligned} &\int p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{x}_{j,k}^{(\ell')}) p(\mathbf{x}_{j,k} | \mathbf{x}_{j,k-1}^{(\ell')}) d\mathbf{x}_{j,k} \\ &\approx \frac{1}{N} \sum_{n=1}^N p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{x}_{j,k}^{(n|\ell')}), \quad (30) \end{aligned}$$

with samples

$$\mathbf{x}_{j,k}^{(n|\ell')} \sim p(\mathbf{x}_{j,k} | \mathbf{x}_{j,k-1}^{(\ell')}), \quad n = 1, \dots, N. \quad (31)$$

Here, the notation $(n|\ell')$ is used to illustrate the dependency on the particle ℓ' . Finally, the likelihood that is sought is obtained as follows:

$$p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}) \approx \sum_{\ell'=1}^L w_{j,k-1}^{(\ell')} \cdot \frac{1}{N} \sum_{n=1}^N p(y_{i,k} | \mathbf{x}_{i,k}^{(\ell)}, \mathbf{x}_{j,k}^{(n|\ell')}). \quad (32)$$

Because additional sampling and corresponding evaluations of the probability density function given in (32) is computationally demanding, a simplification is proposed: Instead of sampling new particles $\mathbf{x}_{j,k}^{(n|\ell')}$ as per (31), the time update samples (34), which are sampled from the same probability density function, are reused. With this procedure, approximation (35) is obtained which can be interpreted as a particular instance of (32) for $n = 1$. Therefore, an easily implementable approximation of the sought likelihood is obtained that avoids the computation of gradients and Hessians, which are needed in GA-based methods. The full bootstrap-like algorithm employing the proposed approximation is described in Algorithm 2, where $\mathcal{A}_{y_{i,i}}$ denotes the set of agents from which measurements have been recorded by agent i at the current time step.

Algorithm 2: Particle Filter with proposed Likelihood Approximation for Multiple Particle Filter

input : Prior, evolution and measurement densities $p(\mathbf{x}_{i,0})$, $p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1})$, and $p(\mathbf{y}_{i,k}|\mathbf{x}_{i,k}, \mathbf{x}_{j,k})$

- 1 Draw L samples $\mathbf{x}_{i,0}^{(\ell)}$ from the prior

$$\mathbf{x}_{i,0}^{(\ell)} \sim p(\mathbf{x}_{i,0}), \quad \ell = 1, \dots, L; \forall i \in \mathcal{A} \quad (33)$$
 and set $w_{i,0}^{(\ell)} = \frac{1}{L}, \forall \ell = 1, \dots, L; \forall i \in \mathcal{A}$;
- 2 **foreach** $k = 1, \dots, K$ **do** // Loop over all time steps
 - 3 **foreach** $i \in \mathcal{A}$ **do** // Loop over all agents
 - 4 Draw samples $\mathbf{x}_{i,k}^{(\ell)}$ from the importance distribution

$$\mathbf{x}_{i,k}^{(\ell)} \sim p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1}^{(\ell)}), \quad \ell = 1, \dots, L; \quad (34)$$
 - 5 **end**
 - 6 **foreach** $j \in \mathcal{A}_{y_i,i}$ **do** // Loop over all agents
 - 7 **foreach** $j \in \mathcal{A}_{y_i,i}$ **do** // Loop over measurements
 - 8 Calculate new weights according to

$$\gamma_j^\ell \leftarrow \sum_{\ell'=1}^L w_{j,k-1}^{(\ell')} p(\mathbf{y}_{i,k}|\mathbf{x}_{i,k}^{(\ell)}, \mathbf{x}_{j,k}^{(\ell')}), \quad \ell = 1, \dots, L;$$
 - 9 **end**
 - 10 Obtain unnormalized weights as

$$w_{i,k}^{(\ell)} \leftarrow w_{i,k-1}^{(\ell)} \prod_{j \in \mathcal{A}_{y_i,i}} \gamma_j^\ell \quad (36)$$

and normalize them to sum to unity via

$$w_{i,k}^{(\ell)} \leftarrow w_{i,k}^{(\ell)} / \sum_{\ell=1}^L w_{i,k}^{(\ell)}; \quad (37)$$
 - 11 **end**
 - 12 **end**
 - 13 **end**

5 Simulation setup and method

This section introduces the chosen simulation setups, including the environment, as well as the state evolution and measurement models.

5.1 System model

Although the proposed procedure is also applicable to other models, the following Gaussian constant-velocity, constant-turn model is considered here for each agent, with $\mathbf{x}_{i,k+1} = f(\mathbf{x}_{i,k}) + \mathbf{v}_k = [x_{i,k+1}, y_{i,k+1}, v_{i,k+1}, \phi_{i,k+1}, \omega_{i,k+1}]^T$ [33]:

$$\mathbf{x}_{i,k+1} = \begin{bmatrix} x_{i,k} + \frac{2v_{i,k}}{\omega_{i,k}} \sin(\omega_{i,k}T/2) \cos(\phi_{i,k} + \omega_{i,k}T/2) \\ y_{i,k} + \frac{2v_{i,k}}{\omega_{i,k}} \sin(\omega_{i,k}T/2) \sin(\phi_{i,k} + \omega_{i,k}T/2) \\ v_{i,k} \\ \phi_{i,k} + \omega_{i,k}T \\ \omega_{i,k} \end{bmatrix} + \mathbf{v}_k, \quad (38)$$

where $x_{i,k}$ and $y_{i,k}$ are the agent's Cartesian x and y positions, respectively; $v_{i,k} = \sqrt{\dot{x}_{i,k}^2 + \dot{y}_{i,k}^2}$ is the agent's speed; and $\phi_{i,k} = \text{atan}(\dot{y}_{i,k}/\dot{x}_{i,k})$ and $\omega_{i,k} = \dot{\phi}_{i,k}$ are the agent's heading angle and turning rate, respectively.

The process noise $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \Sigma)$ is Gaussian, with the following covariance matrix:

$$\Sigma = \text{blkdiag} \left[\sigma_x^2 \quad \sigma_y^2 \quad T^2 \sigma_v^2 \quad \sigma_\omega^2 \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T^2 \end{bmatrix} \right], \quad (39)$$

where T is the sampling period and σ_v and σ_ω are the standard deviations of the noise processes that correspond to the linear acceleration and angular acceleration, respectively.

Consequently, it is assumed that $p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1}) = \mathcal{N}(f(\mathbf{x}_{i,k-1}), \Sigma)$, where $f(\cdot)$ is the noise-free state evolution model given in Eq. 38.

In the following, bearing and distance measurements between agents or beacons i and j of the form

$$y_{i,j,k} = \left[\frac{\sqrt{(x_{i,k} - x_{j,k})^2 + (y_{i,k} - y_{j,k})^2} \cdot (1 + \eta_{d,k})}{\text{atan2}(y_{i,k} - y_{j,k}, x_{i,k} - x_{j,k}) + \eta_{b,k}} \right] \quad (40)$$

are considered, where $\boldsymbol{\eta}_k = [\eta_{d,k}, \eta_{b,k}]^T \sim \mathcal{N}(\mathbf{0}, \Gamma)$, with $\Gamma = \text{diag}(\sigma_d^2, \sigma_b^2)$, 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 i is then obtained as follows:

$$\mathbf{y}_{i,k} = \begin{bmatrix} \dots & \mathbf{y}_{i,j,k}^T & \dots \end{bmatrix}^T. \quad (41)$$

To obtain the results presented in the following section, the sampling period was set to $T = 1$ s, and the process noise parameters were set to $\sigma_v = 3.16 \times 10^{-3}$ m/s², $\sigma_\omega = 3.16 \times 10^{-3}$ rad/s², and $\sigma_x = \sigma_y = 0.30$ m.

5.2 Simulation setup I

The first simulation setup is visualized in Fig. 2. It consists of four mobile agents and four beacons, each with a sensing radius of 4 m. In this setup, multiple different maneuvers, consisting of turns and velocity changes, need to be captured. A variety of trajectories are used to test the algorithms' performance on these two main types of maneuvers. The agents follow the model given in (38), but the turn rate and speed are abruptly changed.

- For agent 1:

$$[\omega_1(kT)]_k = \begin{bmatrix} \mathbf{0}_{15}, \Pi_{\frac{1}{2}}^{10}, -\Pi_{\frac{1}{2}}^{10}, \mathbf{0}_{10}, -\Pi_{\frac{1}{2}}^{12}, \Pi_{\frac{1}{2}}^{22}, \mathbf{0}_9 \end{bmatrix}$$

$$[v_1(kT)]_k = [0.15 \cdot \mathbf{1}_5, 0.075 \cdot \mathbf{1}_{83}]$$

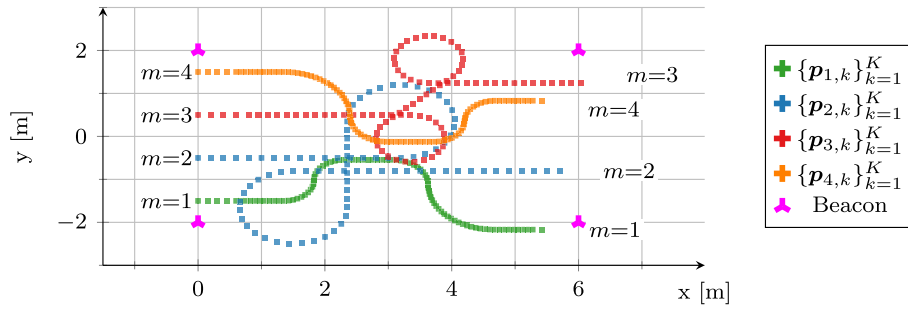


Fig. 2 Simulation setup with four agents and four beacons. The agents' true trajectories are shown

- For agent 2:

$$[\omega_2(kT)]_k = \begin{bmatrix} \mathbf{0}_{17}, \Pi_{\frac{3}{2}}^{20}, \mathbf{0}_{10}, -\Pi_{\frac{3}{2}}^{20}, \mathbf{0}_{21} \end{bmatrix}$$

$$[v_2(kT)]_k = [0.2 \cdot \mathbf{1}_{88}]$$

- For agent 3:

$$[\omega_3(kT)]_k = \begin{bmatrix} \mathbf{0}_{22}, \Pi_{\frac{7}{4}}^{20}, \mathbf{0}_{10}, -\Pi_{\frac{7}{4}}^{20}, \mathbf{0}_{16} \end{bmatrix}$$

$$[v_3(kT)]_k = [0.2 \cdot \mathbf{1}_5, 0.15 \cdot \mathbf{1}_{83}]$$

- For agent 4:

$$[\omega_4(kT)]_k = \begin{bmatrix} \mathbf{0}_{15}, -\Pi_{\frac{1}{2}}^{22}, \Pi_{\frac{1}{2}}^{12}, \mathbf{0}_{10}, \Pi_{\frac{1}{2}}^{10}, -\Pi_{\frac{1}{2}}^{10}, \mathbf{0}_9 \end{bmatrix}$$

$$[v_4(kT)]_k = [0.15 \cdot \mathbf{1}_5, 0.075 \cdot \mathbf{1}_{83}]$$

Here, $\mathbf{0}_n$ and $\mathbf{1}_n$ denote a sequence of n zeros and a sequence of n ones, respectively. Moreover, Π_a^b denotes the sequence that yields a $a \cdot \pi$ turn in b steps.

Due to the limited sensing range, this setup is relevant to the considered scenario of limited beacon access. All figures show results averaged over 100 simulations. All algorithms listed in Table 2 are evaluated. This table also summarizes the key features and properties of the algorithms.

The setup is chosen for the following reasons: First, the agents follow predefined trajectories that, apart from the introduced abrupt changes with respect to velocity and turn rate, are fully consistent with the chosen motion model (38). Consequently, any tracking inaccuracy resulting from model mismatch, which will inherently arise in any real-world scenario, is strongly limited. This, in turn, facilitates a more in-depth analysis and fairer comparison of the different likelihood approximation methods, which target only the measurement function (cf. (2a)). Second, despite the use of predefined trajectories, the maneuvers are complex and challenging due to the variation in their motion profiles, which include a broad set of different turns and acceleration phases. Thus, adequate likelihood approximation schemes are required to enable accurate localization.

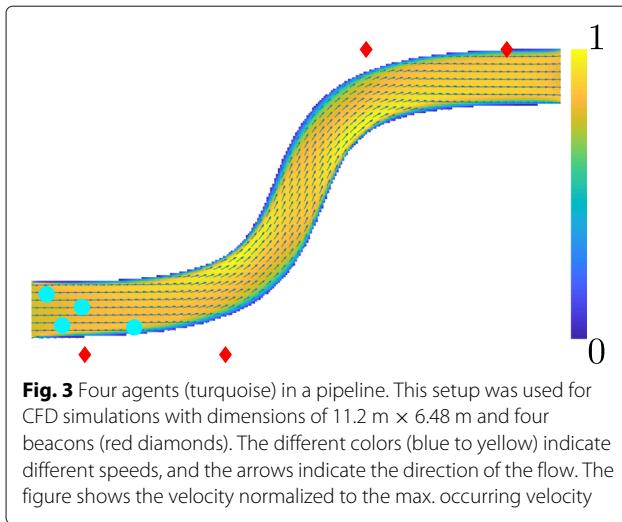
The reported results below are related to the following two measurement noise scenarios (MNSs): *MNS 1*: Multiplicative measurement noise with a standard deviation of $\sigma_d = 0.04$ and bearing measurement noise with a standard deviation of $\sigma_b = 5^\circ$ and *MNS 2*: $\sigma_d = 0.06$, $\sigma_b = 10^\circ$.

5.3 Simulation setup II

Additional simulation results, in which the agents' motion is obtained via computational fluid dynamics (CFD) simulations, are presented. In contrast to the results presented in the context of the first simulation setup, the resulting agent motion cannot, to a significant extent, be properly modeled by (38) (cf. Section 5). For this reason, the results presented and discussed for this setup exhibit more variation. The CFD results were obtained by simulating the pipe model shown in Fig. 3 using a D2Q9 lattice Boltzmann model (LBM). The pipe is filled with water at 25 °C. The LBM parameters used for the CFD simulation are listed in Table 3. Details such as the boundary method implemented for the D2Q9 LBM are given in [34], where

Table 2 Overview of the evaluated algorithms

Abbr.	Proposed in	Discussed in	Comment
PE	[8]	—	Eliminates the dependency of the likelihood on the other agent's state by means of the point estimate $\hat{\mathbf{x}}_{j,k} = \sum_{\ell=1}^L w_{j,k-1}^\ell \mathbf{x}_{j,k}^\ell$.
GA	[14] (basic form)	Section 3.1 (generalized form)	Approximates the likelihood as a Gaussian density using first- and second-order terms for the variance and mean, respectively. Treats $\mathbf{x}_{j,k}$ as a random variable in the likelihood.
EGA	This work	Sections 3.1.3 and 3.1.4	Extension of GA to a complete second-order approx. for both additive and multiplicative noise.
MCA	This work	Section 4	MC-based likelihood approx.



this method is reported to be accurate to approximately second order.

These simulation results are thus consistent with the application scenarios discussed in Section 1, in which miniature agents may be employed to survey underground structures or pipeline systems, such as those that exist in chemical plants. The chosen setup, however, is designed to challenge the algorithms (cf. Table 2), which is achieved through a combination of fast agent motion and limited beacon coverage.

5.4 Evaluation metric

The following time- and agent-averaged localization error metric is used to evaluate the algorithms' performance: $MAE = \frac{1}{M \cdot K} \sum_{i=1}^M \sum_{k=1}^K \|p_{i,k} - \hat{p}_{i,k}\|_2$, where $p_{i,k}$ is the true position of agent i at time step k and $\hat{p}_{i,k}$ is the corresponding estimate, which refers to the first two states of the corresponding state vector.

6 Numerical results and discussion

The discussion of the simulation results is split into three main parts, with each focusing on different behavior and properties of the methods. This discussion is presented in the following subsections.

Table 3 D2Q9 LBM simulation parameters

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

For all results evaluating a performance as a function of particles per agent (PPA), the 99% confidence intervals (CIs) are shown as well. These results are visualized as a transparent hose around the average performance with the same line style and visualize the variation in the individual simulations.

6.1 Setup I: Localization error vs. number of particles and run time

The first set of results is presented in Fig. 4, which shows the localization performance as a function of the number of PPA, L . MNS 1 and MNS 2 are represented in the top and middle panels, respectively. Moreover, in Figs. 4 and 5, a third MNS is presented in which multiplicative distance measurement noise is increased to $\sigma_d = 0.1$ compared to MNS 2.

The left panels of Fig. 4 clearly show that PE exhibits significantly reduced performance compared to that of the other three methods. The right row of Fig. 4 focuses on the differences between the GA-based algorithms (GA and EGA) and the proposed MCA method. In these panels, the differences between the former two are generally small. However, the performance of MCA is significantly improved compared to that of the other methods. For example, for $L = 1000$, an error reduction of 16% compared to EGA is achieved. The middle panels show that in MNS 2, EGA is slightly superior to GA. Most notably, the localization error reduction of 21% is achieved by MCA over EGA. Similarly, for MNS 3, MCA achieves error reductions of 24%.

Interestingly, the mean absolute error (MAE) of the PE method decreases from 0.46 (MNS 1) to 0.39 (MNS 2) to 0.35 (MNS 3) with increased measurement noise. In contrast, the MAE of the other methods increases at the same time. For example, for MCA, the MAE increases from 0.10 to 0.15 to 0.18 at these measurement noise configurations. Despite this increase in absolute terms, the *relative* improvement of MCA over EGA increases from 16% (MNS 1) to 22% (MNS 2) to 24% (MNS 3), as mentioned above. Importantly, despite the improvement in PE with increasing noise intensity, the absolute performance gains through MCA remain significant. Additionally, CIs show that the performance of PE is the least consistent.

A comparison of performance versus run time for the MNS 1 and MNS 2 results shown in Fig. 4 is presented in Fig. 5. Note that for each algorithm, an increase in computing time is directly linked to an increase in PPA value. These results show the trade-off between the different algorithms in terms of the achievable MAE performance and required computing time. Given certain requirements of either performance or computing time, these results aid in selecting algorithms for particular application scenarios. As discussed subsequently, some algorithms are limited in either of these figures, i.e., MAE performance

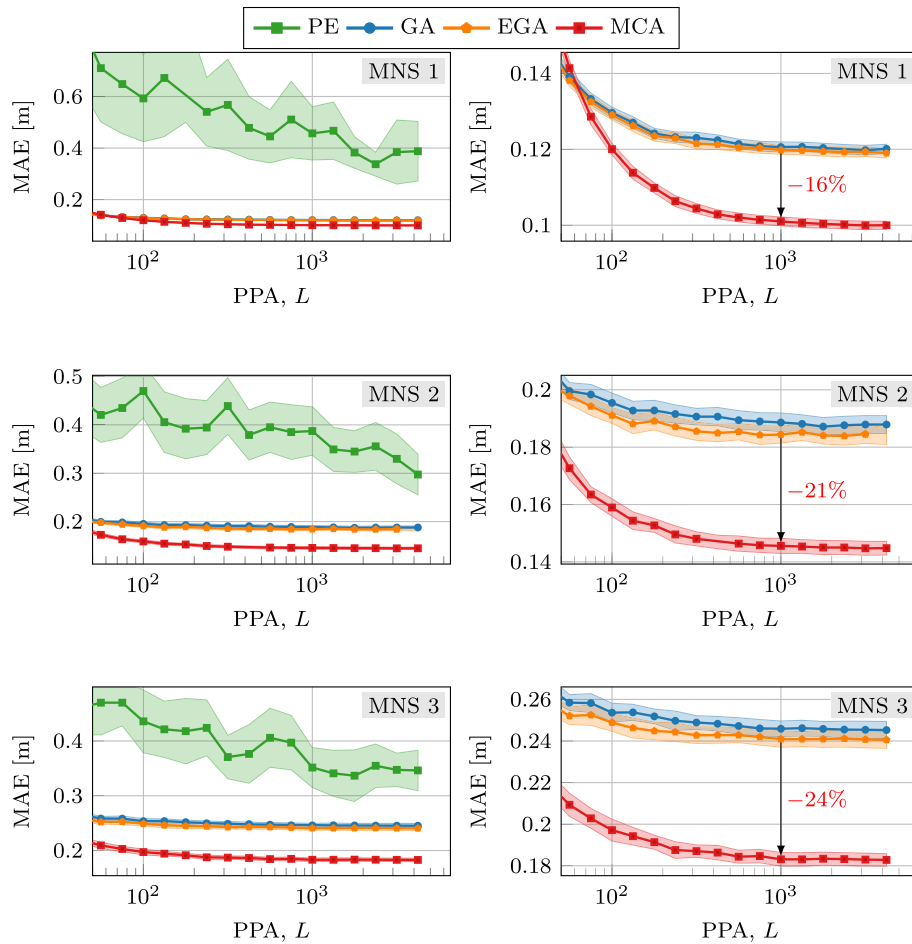


Fig. 4 Localization error as a function of PPA. The plots for MNS 1, MNS 2, and MNS 3 are at the top, middle, and bottom, respectively. In all panels, $M = 4$ agents are considered. The right panels show magnified views of the results presented in the upper panels but without the PE results for improved readability

or computing time. Similar to the previous results, the left panels of Fig. 5 show a comparison among all four algorithms, whereas the right panels of Fig. 5 focus on the GA-based methods and MCA. Because the results in Fig. 5 are deduced from the data depicted in Fig. 4, the supports for the curves are different. As such, the left panels of Fig. 5 show low run times for PE because no data are available for run times longer than 10 s. However, these low run times are not accompanied by performance values better than a localization error of 0.34 m.

The right panels of Fig. 5 (together with the right panels of Fig. 4) show that MCA not only yields improved localization performance, with a lower localization error than those of all other methods, but also achieves improved performance within the same computing time. For example, in MNS 1, a reduction in the error metric of 14% is achieved for a run time of 14 s. Similarly, for MNS 2 and MNS 3, reductions in 20% and 23% are obtained, respectively.

6.2 Setup I: Performance over time

The second set of results, illustrating the performance of the algorithms over the different time steps, is visualized in Figs. 6, and 7.

Figure 6 visualizes the performance for MNS 2 with 422 PPA, which has been chosen as a compromise between a low PPA (where MCA performs similarly to EGA) and a high PPA (where MCA is able to significantly improve compared to EGA and the other methods). The estimated positions are overlaid on the agents' actual trajectories, which are shown in red, and are plotted as markers only in the same colors as those used in Fig. 2.

It can be observed that the performance of PE (cf. Fig. 6b) deteriorates in the case of maneuvers. The agent profiles visualized with red and orange markers are particularly strongly affected. GA and EGA (cf. Fig. 6c, d) show improved localization performance compared to that of PE; this improvement is also reflected by the observed

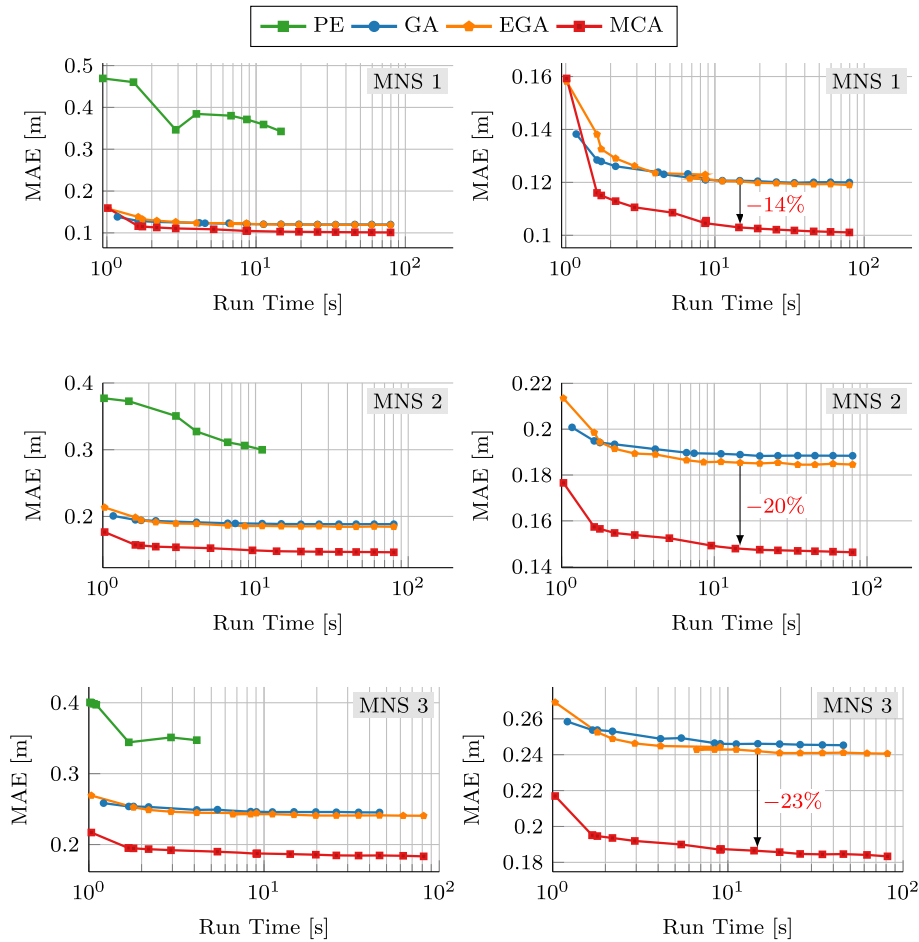


Fig. 5 Localization error as a function of run time. The plots for MNS 1 and MNS 2 are at the top and bottom, respectively. In both MNSs, $M = 4$ agents are considered. The right panels show magnified views of the results presented in the upper panels but without the PE results for improved readability

reduction of almost 50% in the localization error. However, both of these algorithms also show reduced tracking accuracy during maneuvers (cf. in particular the estimates represented by red markers in the top loop and by green and orange markers at the saddle points of the maneuvers). Figure 6a shows the performance of MCA, which encounters only minor issues with tracking (cf. the estimates represented by red markers in the top loop). This superior performance is also evident from the reduction of -22% in the localization error compared with that of GA.

Figure 7b clearly shows a decreasing tracking accuracy, with the peak being in the transition zone between the coverage areas of the left and right beacons. Figure 7c and d, for GA and EGA, respectively, show significantly improved performance compared with that of PE, with some strong fluctuations (note the differences in scaling for reasons of readability). Figure 7a shows the performance of MCA. At the beginning of tracking, MCA

exhibits a significant reduction in error from the initial value. Compared to all other methods, not only reduced error but also lower fluctuations in the performance over time are apparent.

6.3 Setup I: Localization error vs. number of agents

The third and last set of results for the first setup is presented in Figs. 8 and 9, which analyze the tracking performance for varying numbers of agents for MNS 1 and MNS 2, respectively. The results were obtained with the same setup depicted in Fig. 2, with only the first $i = 1, \dots, M$ agents being simulated in cases with fewer than four agents.

Figure 8 shows that the performance of PE decreases significantly with an increasing number of agents. Similar behavior can be observed for GA and EGA, with the error metric increasing by approximately 11% when four agents need to be localized instead of two. Only MCA exhibits the opposite trend, with the

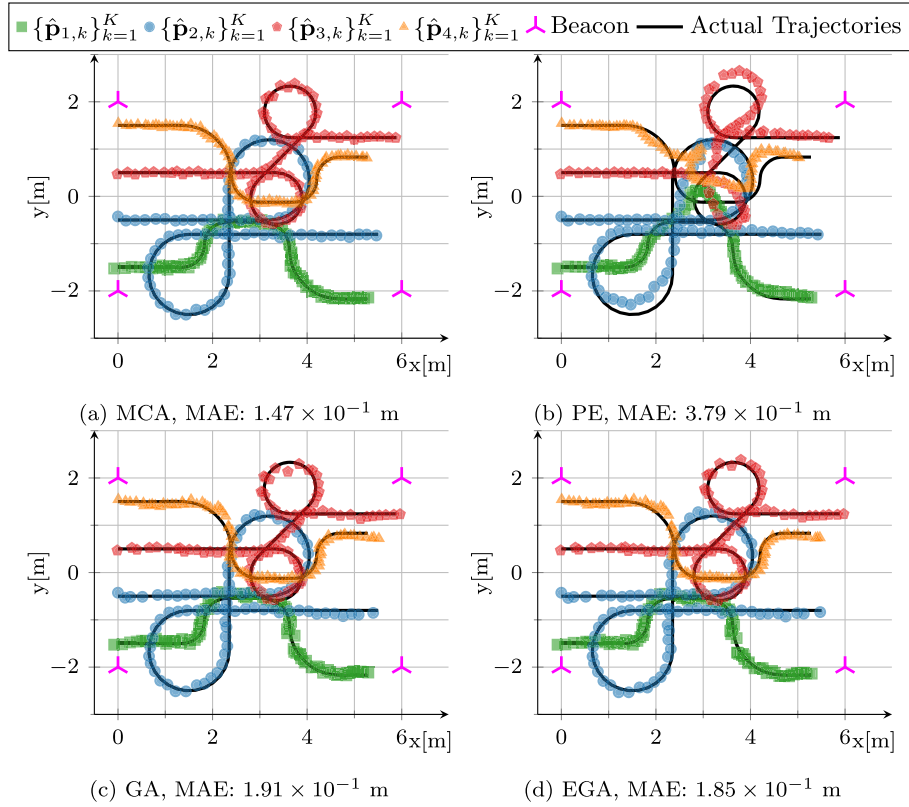


Fig. 6 Estimated averaged trajectories (markers only) for all methods, with the actual trajectories underlaid in black (cf. Fig. 2). These results were obtained for MNS 2 and 422 PPA

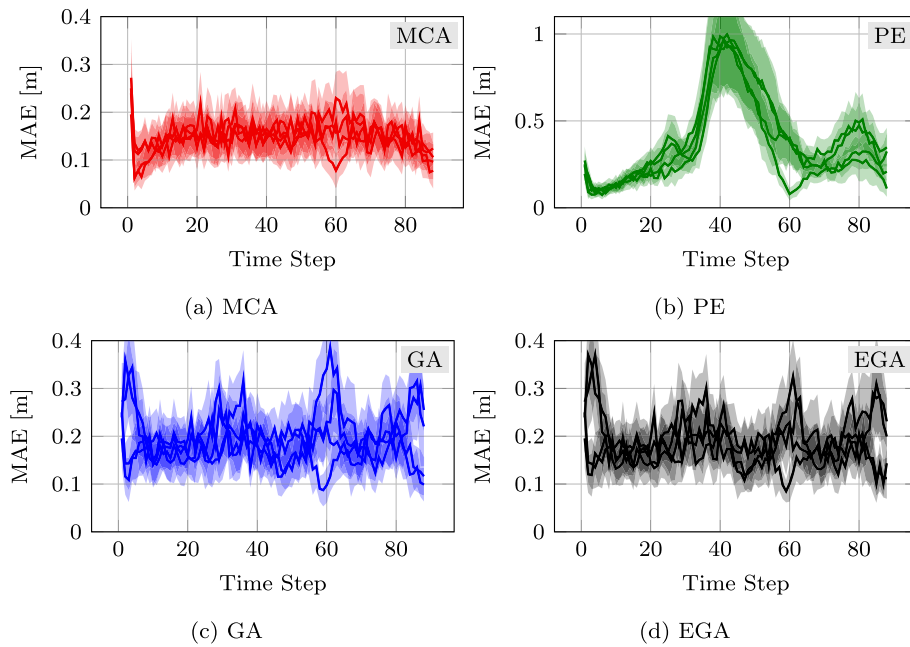


Fig. 7 Localization error as a function of the time step for $M = 4$ agents in MNS 2. The individual solid lines in each of the plots correspond to one of the agents. The CIs are illustrated as transparent hoses around the mean performance. **a** MCA. **b** PE. **c** GA. **d** EGA

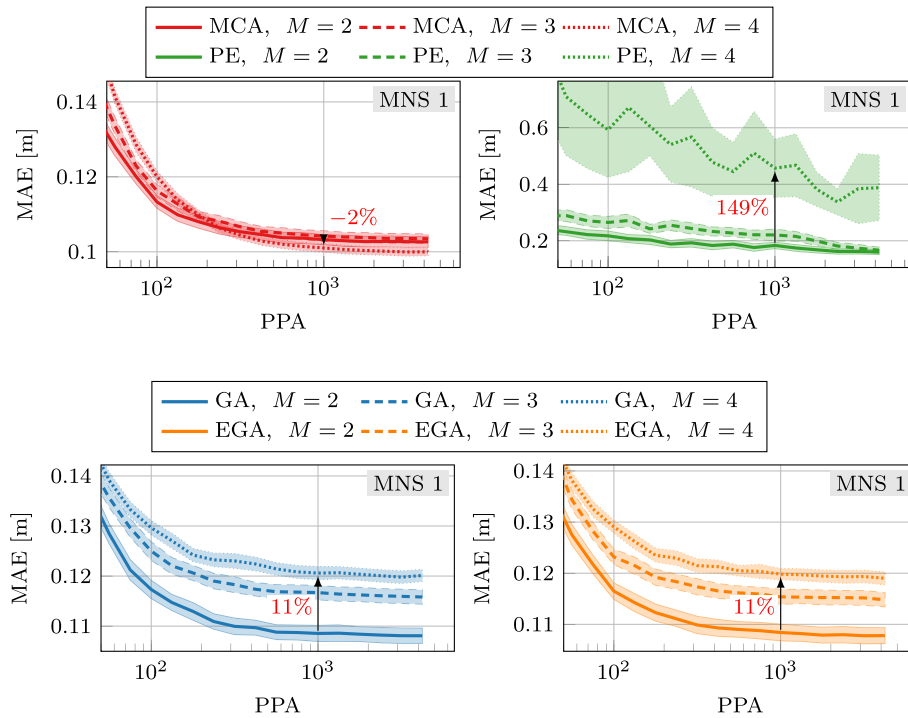


Fig. 8 Localization error as a function of the number of PPA for different numbers of agents to be tracked for MNS 1

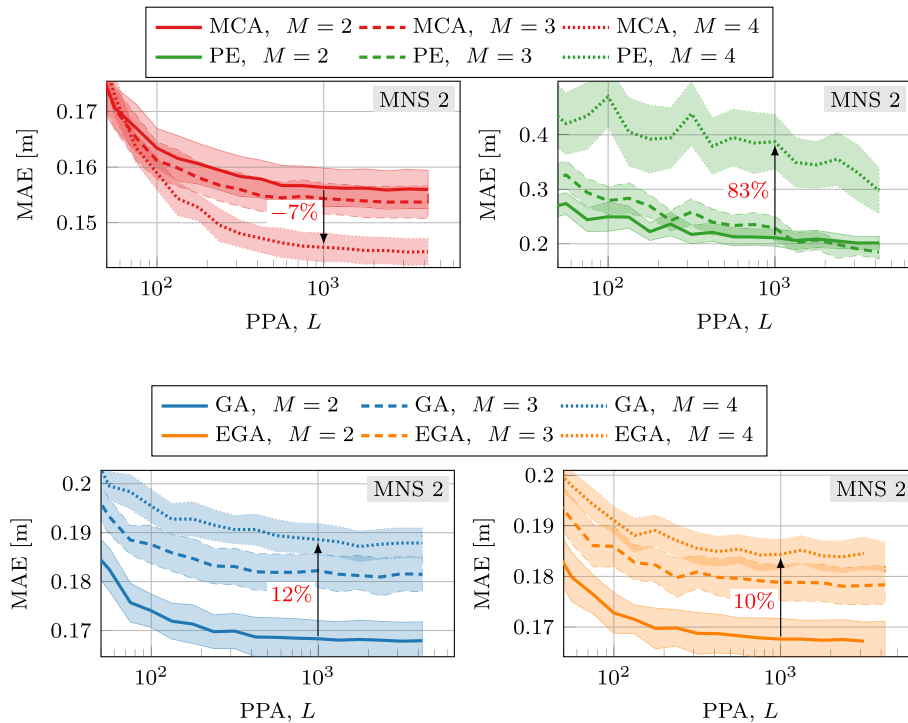


Fig. 9 Localization error as a function of the number of PPA for different numbers of agents to be tracked for MNS 2

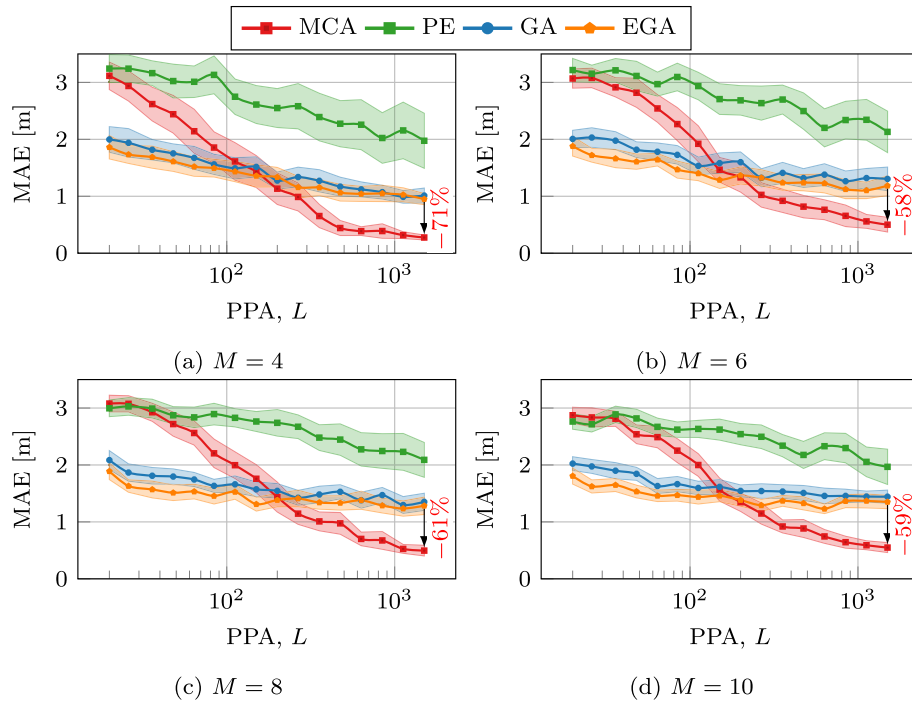


Fig. 10 Localization performance as a function of PPA for different numbers of agents: **a** four agents, **b** six agents, **c** eight agents, and **d** ten agents

error metric decreasing by approximately 2% in the same case.

Similar trends can be observed in Fig. 9 for increasing measurement noise. Here, the error metric of MCA is reduced by 6.88% when four agents are tracked instead of two.

In both MNSs, this behavior can be explained by the additional agent-to-agent measurements that are provided by the additional agents. In the cases of GA and EGA, for example, these additional measurements result in significant error accumulation because the likelihoods are approximated as Gaussian distributions and, in the weight update (cf. (7)), the product of these likelihoods is taken³. For MCA, however, the approximation error is lower, allowing the additional measurements provided by the extra agents to be successfully exploited.

6.4 Setup II: Localization error vs. number of particles and run time

For brevity, the second setup is evaluated only for the following simulation configuration. The MNS was scenario 2, and the agent and beacon coverage ranges were both $R_s = 10$ m.

The first set of results is presented in Fig. 10, which compares the performance of the different algorithms as a function of the number of PPA. The different panels of

Fig. 10 present the results for $M = 4, 6, 8$, and 10 agents. For low PPA values, GA and EGA outperform MCA. However, GA and EGA do not achieve a MAE lower than 1 m, which can be considered critical depending on the precise application requirements. For more than approx. 200 PPA, MCA achieves the best performance, resulting in MAE reductions of between 71% (for $M = 4$) and 59% (for $M = 10$) compared to EGA. Interestingly, not only the average MAE performance of MCA improves for increasing PPA: For example, for four agents, the size of the CI for MCA decreases from 0.55 m (200 PPA) to 0.08 m (1500 PPA). Similarly, for ten agents, the CI size decreases from 0.32 m to 0.17 m at the same time.

Similar to the results discussed in Section 6, EGA outperforms GA, although marginally, throughout the simulations.

The second set of results is given in Fig. 11, which compares the run times of the algorithms with their achieved MAE values. With the same run-time budget, the following reductions in MAE are achieved: for $M = 4$ agents and 62 s, 63%; for $M = 6$ agents and 145 s, 32%; for $M = 8$ agents and 254 s, 45%; and for $M = 10$ agents and 389 s, a reduction of 45%.

6.5 Setup II: Localization error vs. number of agents

The third set of results is given in Fig. 12, which compares the change in performance of each algorithm as the number of agents to be tracked changes. The fluctuations with

³Recall that the measurement noise is assumed to be independent and identically distributed in nature.

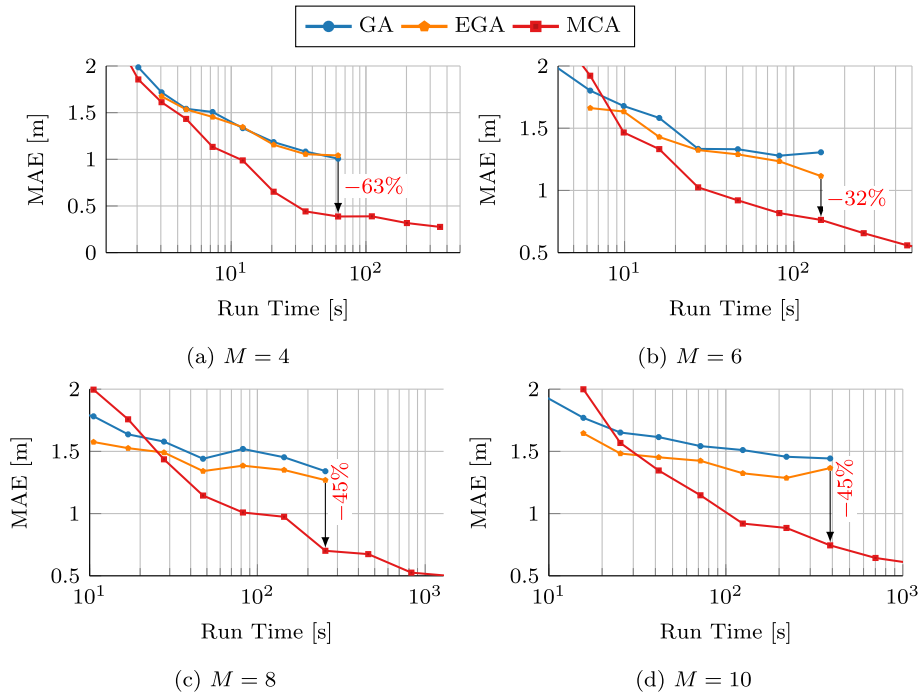


Fig. 11 Localization performance as a function of run time for 1500 PPA and for different numbers of agents: **a** four agents, **b** six agents, **c** eight agents, and **d** ten agents

variations in the number of agents and the PPA value for MCA are very low in the absolute sense. In fact, the performance degradation associated with the requirement to track 10 instead of 4 agents corresponds to an increase of only approximately 0.27 m relative to the initial value of 0.275 m ($M = 4$). In contrast, not only the fluctuations but also the absolute MAE values for GA and EGA are much higher. For example, for 1500 PPA, the MAE increases by 0.43 m from an initial value of 1 m (GA) and by 0.4 m from an initial value of 0.95 m (EGA).

The fourth and final set of results is presented in Figs. 13 and 14, which show the actual agents' trajectories as solid lines and the average estimated positions as markers only for $M = 4$ and 8 agents, respectively. For $M = 4$ agents (cf. Fig. 13), MCA can accurately track the agents' fast motion and achieves an average MAE of 0.28 m. In contrast, the other methods quickly lose track of the agents' motion and cannot achieve an MAE better than 0.95 m.

In the case that 8 agents are to be tracked (cf. Fig. 14), MCA tends to overestimate the turn rates of the agents, resulting in an average MAE of 0.5 m. In contrast, all other methods are, on average, unable to track the agents' motion at all, resulting in a poor accuracy of 1.28 m at best.

6.6 Setup II: Summary of results

In the second simulation setup, a close-to-reality scenario is considered in which up to 10 agents are carried by fluid dynamics alone through a pipe branch.

Because the agents' motion is difficult to describe with a single motion model, the results, as expected, show more fluctuations and are more dependent on the accuracy of the motion model compared to results of the first simulation setup. Nevertheless, even in this scenario, the proposed MCA method outperforms the PE, GA, and EGA methods, achieving MAE reductions of up to 63% when the computational costs for all methods are fixed.

7 Conclusion and future work

In this work, a new approximation scheme for individual filter likelihoods for the tracking of wireless agents has been proposed. The proposed MCA scheme has been compared to the existing PE method and to a generalization of the Gaussian approximation concept presented in [14]. Moreover, this generalization (GA) has further been extended to the EGA method for additional comparisons. The simulation results showed clear improvements in the localization accuracy when MCA was employed. In particular, a reduction in the localization error by up to 22.81% was achieved for a fixed number of particles. Additionally, it was shown that given the same computing time, MCA achieves a tracking accuracy that is 22.35% higher than that of GA. Moreover, the tracking accuracy of MCA was shown to be more consistent over time, with lower fluctuations in performance. In contrast to the other considered methods, MCA was able to successfully exploit

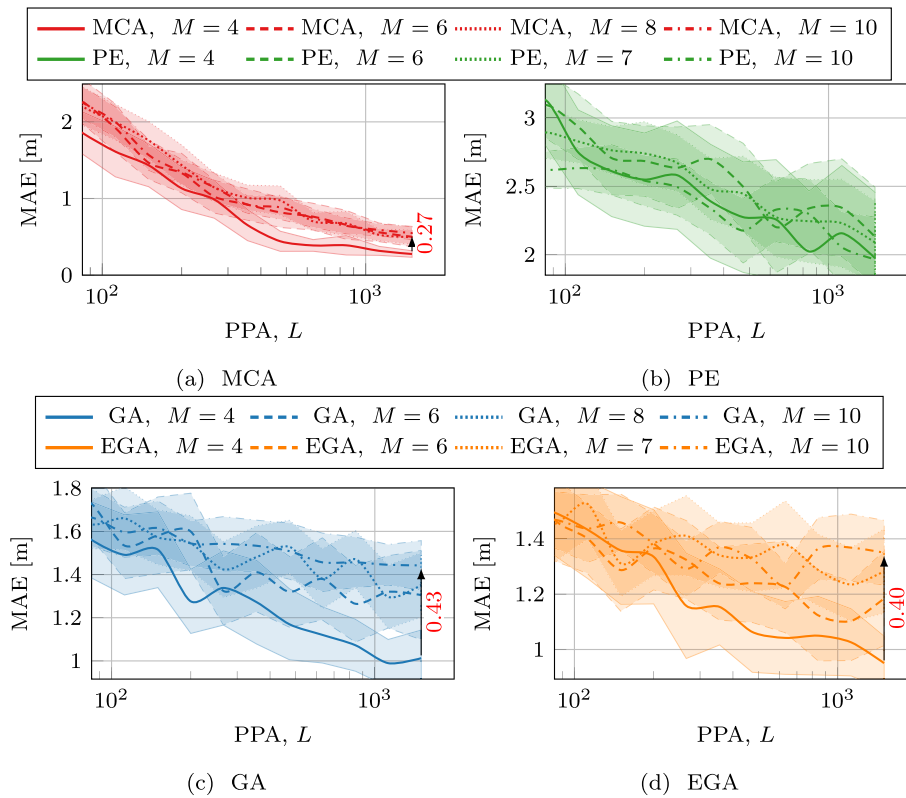


Fig. 12 Localization performance as a function of the PPA value and as a function of the number of agents to be tracked. **a** MCA. **b** PE. **c** GA. **d** EGA

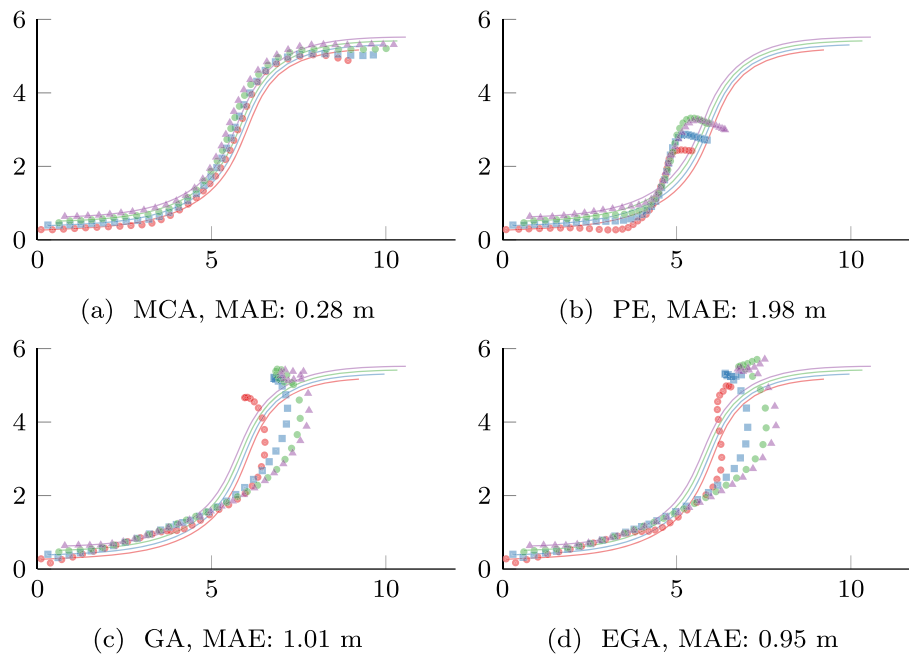


Fig. 13 Average estimated positions (markers) and actual trajectories (solid lines) in the case of $M = 4$ agents and for 1500 PPA. **a** MCA, MAE 0.28 m. **b** PE, MAE 1.98 m. **c** GA, MAE 1.01 m. **d** EGA, MAE 0.9 m

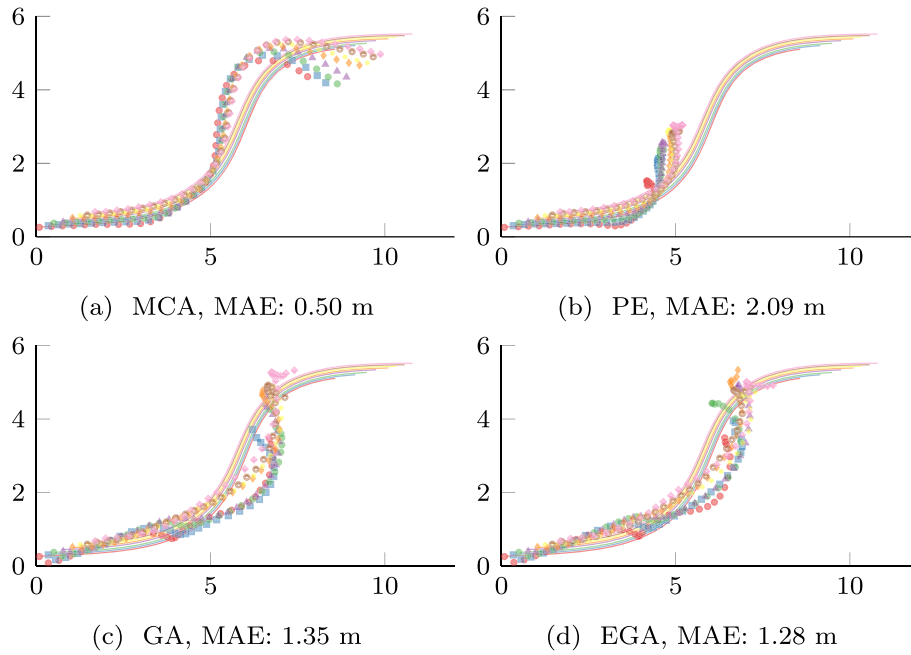


Fig. 14 Average estimated positions (markers) and actual trajectories (solid lines) in the case of $M = 8$ agents and for 1500 PPA. **a** MCA, MAE 0.50 m. **b** PE, MAE 2.09 m. **c** GA, MAE 1.35 m. **d** EGA, MAE 1.28 m

additional measurements introduced with an increase in the number of agents to be tracked in the synthetic benchmark (first setup). Similarly, for the second set of close-to-reality simulations (second setup), MCA was least affected by an increase in the number of agents. In numbers, MCA achieved MAE reductions between 58 and 71% compared to EGA.

In summary, the MCA scheme presented in this study offers significant improvements in localization accuracy and/or computing time. The scheme has shown potential for many application cases in which a known number of cooperating agents are to be localized using distance and/or bearing measurements if access to beacons is limited and/or AAMs are essential for accurate localization. Examples thereof are rescue robots employed for surveying collapsed buildings due to earthquakes and miniature agents used for pipeline inspection.

In future works, we will investigate the proposed scheme on a broader scale, including the effect of the MC parameter n on performance and run time. Moreover, an in-depth theoretical analysis of the computational complexity of all discussed algorithms is of interest in this context.

Appendix A: Proof of the third-order moment of a Gaussian approximation component

As mentioned in Section 3, the state $\tilde{\mathbf{x}}_{j,k}$ is assumed to be Gaussian. For simplicity, the agent and time indices are

dropped in the following, such that $\tilde{\mathbf{x}}_{j,k} = \tilde{\mathbf{x}}$ and $\mathbf{x}_{i,k} = \mathbf{x}$. Generally, the i -th element of a vector \mathbf{x} is denoted by $[\mathbf{x}]_i$, and the (i, j) -th element of a matrix \mathbf{X} is denoted by $[\mathbf{X}]_{i,j}$. For simplicity, $[\tilde{\mathbf{x}}_{j,k}]_i$ is denoted by \tilde{x}_i .

With this new notation,

$$\begin{aligned} & \mathbb{E} \left[\mathbf{g}_{\tilde{\mathbf{x}}_{j,k}} \tilde{\mathbf{x}}_{j,k}^T \mathbf{H} \tilde{\mathbf{x}}_{j,k} \mid \mathbf{x}_{i,k} \right] \\ &= \sum_i \sum_j \sum_l \mathbf{g}_i [\mathbf{H}]_{j,l} \mathbb{E} [\tilde{x}_i \tilde{x}_j \tilde{x}_l \mid \mathbf{x}_{i,k}], \end{aligned} \quad (42)$$

and the expected value that is sought reduces to the sum of the higher-order moments of a Gaussian random vector. Subsequently, the link between the higher-order moments and the multivariate Hermite polynomials is exploited in accordance with the following theorem.

Theorem 1 (Higher-order moments of multivariate Gaussian random variables [35]) *For a Gaussian random vector $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n} \succeq 0$, it holds that*

$$\mathbb{E} [x_1^{v_1} \cdots x_n^{v_n}] = \text{He}_{v_1, \dots, v_n}(-\boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}, -\boldsymbol{\Sigma}), \quad (43)$$

where the v_i are nonnegative integers. The Hermite polynomials are defined as

$$\text{He}_{v_1, \dots, v_n}(\mathbf{z}, \mathbf{A}) = \exp(\mathbf{Q}/2) \prod_{i=1}^n \left(-\frac{\partial}{\partial z_i} \right)^{v_i} \exp(-\mathbf{Q}/2), \quad (44)$$

where $\mathbf{Q} = \mathbf{z}^T \mathbf{A} \mathbf{z}$.

Noting that $\tilde{\mathbf{x}}$ is zero-mean by definition, it holds that

$$\mathbb{E}[\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_l] = -\frac{\partial^3}{\partial z_i \partial z_j \partial z_l} \exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z}\right) \Big|_{\mathbf{z}=\mathbf{0}}. \quad (45)$$

The following three cases need to be addressed for (42), without loss of generality: $i \neq j \neq l$, $i = j \neq l$, and $i = j = l$. For all of them, the following derivative is important:

$$\begin{aligned} \frac{\partial}{\partial z_q} \exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z}\right) = \\ -\exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z}\right) \left(\underbrace{A_{q,q} z_q + \frac{1}{2} \sum_{n \neq q} A_{n,q} z_n}_{\equiv \gamma(q)} \right). \end{aligned} \quad (46)$$

Case 1: $i \neq j \neq k$

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_k] = -\exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z}\right) \left\{ \gamma(i) \gamma(j) \gamma(k) \right. \\ \left. + \frac{1}{2} (\gamma(i) A_{j,k} + \gamma(j) A_{i,k} + \gamma(k) A_{i,j}) \right\} = 0 \end{aligned} \quad (47)$$

Case 2: $i = j \neq k$

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{x}}_i^2 \tilde{\mathbf{x}}_k] = -\exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z}\right) \\ \{ \gamma(i)^2 \gamma(k) + \gamma(k) A_{i,i} + \gamma(i) A_{k,i} \} = 0 \end{aligned} \quad (48)$$

Case 2: $i = j = k$

$$\mathbb{E}[\tilde{\mathbf{x}}_i^3] = -\exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z}\right) \{ \gamma(i)^3 + 3\gamma(i) A_{i,i} \} = 0 \quad (49)$$

List of symbols

$\mathbf{0}_{n \times m}$: All zero matrix of dimension $n \times m$; \mathbf{A} : Set of agents; $\text{Cov}_p[\cdot]$: Covariance matrix with respect to density p ; $\delta(\mathbf{x} - \mathbf{x}_0)$: Dirac delta function centered at \mathbf{x}_0 ; $\mathbb{E}_p[\cdot]$: Expectation operator with respect to density p ; $\mathbf{i}, \mathbf{j}, \mathbf{l}$: Some agent indices; L : Number of particles; ℓ : Particle index; $[\mathbf{X}]_{ij}$: (i, j) -th element of matrix \mathbf{X} ; $\boldsymbol{\eta}$: Measurement noise vector; $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: Normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$; \mathbf{v} : Process noise vector; $\mathcal{O}(\cdot)$: Computational complexity in terms of big O notation; $p(\cdot)$: Probability density function or probability mass function; Π_a^b : Sequence that yields a $a \cdot \pi$ turn in b steps; $\pi(\cdot)$: Proposal density; R_s : Agent sensing range; \mathbb{R} : Set of real numbers; $\text{tr}[\cdot]$: Trace Operator; $\mathbb{V}_p[\cdot]$: Variance with respect to density p ; ω : Particle weight; $\mathbf{x}_{i,k}$: State vector of agent i at time instant k ; $\mathbf{x}_{i,k}$: Cartesian x coordinate of agent i at time step k ; $\hat{\mathbf{x}}_{i,k}$: Estimate of state vector of agent i at time instant k ; $\mathbf{y}_{i,k}$: Measurement vector of agent i at time instant k ; $y_{i,k}$: Cartesian y coordinate of agent i at time step k

Abbreviations

AAM: Agent-to-agent measurement; AWGN: Additive white Gaussian noise; CFD: Computational fluid dynamics; CI: Confidence interval; EGA: Extended Gaussian approximation; GA: Gaussian approximation; LBM: Lattice Boltzmann model; MAE: Mean absolute error; MC: Monte Carlo; MCA: Monte Carlo approximation; MNS: Measurement noise scenario; MPF: Multiple particle filter;

PD: Proposal distribution; PE: Point-estimate approximation; PF: Particle filter; PPA: Particles per agent; RFS: Random finite set

Acknowledgments

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 and wrote the majority of the manuscript. GA initiated the research and also commented on and approved the manuscript. Both 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: schlupkothén@ice.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: 29 April 2019 Accepted: 2 September 2019

Published online: 07 November 2019

References

1. S. Schlupkothén, G. Dartmann, G. Ascheid, A novel low-complexity numerical localization method for dynamic wireless sensor networks. *IEEE Trans. Signal Process.* **63**(15), 4102–4114 (2015). <https://doi.org/10.1109/TSP.2015.2422685>
2. S. Schlupkothén, A. Hallawa, G. Ascheid, in *2018 International Conference on Computing, Networking and Communications (ICNC): Wireless Ad Hoc and Sensor Networks (ICNC'18 WAHS)*. Evolutionary algorithm optimized centralized offline localization and mapping (IEEE, Maui, 2018)
3. F. Daum, J. Huang, in *2003 IEEE Aerospace Conference Proceedings (Cat. No. 03TH8652)*. Curse of dimensionality and particle filters, vol. 4, (2003), pp. 1979–1993. <https://doi.org/10.1109/AERO.2003.1235126>
4. F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, P. J. Nordlund, Particle filters for positioning, navigation, and tracking. *IEEE Trans. Signal Process.* **50**(2), 425–437 (2002). <https://doi.org/10.1109/78.978396>
5. D. Chang, M. Fang, Bearing-only maneuvering mobile tracking with nonlinear filtering algorithms in wireless sensor networks. *IEEE Syst. J.* **8**(1), 160–170 (2014). <https://doi.org/10.1109/JSYST.2013.2260641>
6. P. Yang, Efficient particle filter algorithm for ultrasonic sensor-based 2D range-only simultaneous localisation and mapping application. *IET Wirel. Sensor Syst.* **2**(4), 394–401 (2012). <https://doi.org/10.1049/iet-wss.2011.0129>
7. B. F. Wu, C. L. Jen, Particle-filter-based radio localization for mobile robots in the environments with low-density WLAN APs. *IEEE Trans. Ind. Electron.* **61**(12), 6860–6870 (2014). <https://doi.org/10.1109/TIE.2014.2327553>
8. P. M. Djurić, T. Lu, M. F. Bugallo, in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. Multiple particle filtering, vol. 3, (2007), pp. 1181–1184. <https://doi.org/10.1109/ICASSP.2007.367053>
9. M. F. Bugallo, T. Lu, P. M. Djurić, in *2007 IEEE Aerospace Conference*. Target tracking by multiple particle filtering, (2007), pp. 1–7. <https://doi.org/10.1109/AERO.2007.353042>
10. M. F. Bugallo, P. M. Djurić, in *2010 IEEE Aerospace Conference*. Target tracking by symbiotic particle filtering, (2010), pp. 1–7. <https://doi.org/10.1109/AERO.2010.5446681>
11. P. M. Djurić, M. F. Bugallo, in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. Particle filtering for high-dimensional systems, (2013), pp. 352–355. <https://doi.org/10.1109/CAMSAP.2013.6714080>

12. M. F. Bugallo, P. M. Djurić, in *2014 IEEE Workshop on Statistical Signal Processing (SSP)*. Gaussian particle filtering in high-dimensional systems, (2014), pp. 129–132. <https://doi.org/10.1109/SSP.2014.6884592>
13. M. F. Bugallo, P. M. Djurić, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Particle filtering in high-dimensional systems with gaussian approximations, (2014), pp. 8013–8017. <https://doi.org/10.1109/ICASSP.2014.6855161>
14. P. M. Djurić, 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>
15. L. Martino, J. Read, V. Elvira, F. Louzada, Cooperative parallel particle filters for online model selection and applications to urban mobility. *Digit. Signal Process.* **60**, 172–185 (2017). <https://doi.org/10.1016/j.dsp.2016.09.011>
16. I. Urteaga, M. F. Bugallo, P. M. Djurić, in *2016 IEEE Statistical Signal Processing Workshop (SSP)*. Sequential Monte Carlo methods under model uncertainty, (2016), pp. 1–5. <https://doi.org/10.1109/SSP.2016.7551747>
17. J. A. Hoeting, D. Madigan, A. E. Raftery, C. T. Volinsky, Bayesian model averaging: a tutorial. *Stat. Sci.* **14**(4), 382–401 (1999)
18. B.-N. Vo, S. Singh, A. Doucet, Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Trans. Aerosp. Electron. Syst.* **41**(4), 1224–1245 (2005). <https://doi.org/10.1109/TAES.2005.1561884>
19. M. B. Guldogan, Consensus bernoulli filter for distributed detection and tracking using multi-static doppler shifts. *IEEE Signal Process. Lett.* **21**(6), 672–676 (2014). <https://doi.org/10.1109/LSP.2014.2313177>
20. S. Reuter, K. Dietmayer, in *14th International Conference on Information Fusion*. Pedestrian tracking using random finite sets, (2011), pp. 1–8
21. W.-K. Ma, B.-N. Vo, S. S. Singh, A. Baddeley, Tracking an unknown time-varying number of speakers using TDOA measurements: a random finite set approach. *IEEE Trans. Signal Process.* **54**(9), 3291–3304 (2006). <https://doi.org/10.1109/TSP.2006.877658>
22. A. Eryildirim, M. B. Guldogan, A Bernoulli filter for extended target tracking using random matrices in a UWB sensor network. *IEEE Sensors J.* **16**(11), 4362–4373 (2016). <https://doi.org/10.1109/JSEN.2016.2544807>
23. P. Biswas, Y. Ye, in *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium On*. Semidefinite programming for ad hoc wireless sensor network localization, (2004), pp. 46–54. <https://doi.org/10.1109/IPSIN.2004.1307322>
24. P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, T.-C. Wang, Semidefinite programming approaches for sensor network localization with noisy distance measurements. **3**(4), 360–371 (2006). <https://doi.org/10.1109/TASE.2006.877401>
25. P. Biswas, T.-C. Lian, T.-C. Wang, Y. Ye, Semidefinite programming based algorithms for sensor network localization. *ACM Trans. Sen. Netw.* **2**(2), 188–220 (2006). <https://doi.org/10.1145/1149283.1149286>
26. S. Schlupkothén, G. Ascheid, in *2016 IEEE International Conference on Wireless for Space and Extreme Environments (WISEE) (WISEE'16)*. Joint localization and transmit-ambiguity resolution for ultra-low energy wireless sensors, (2016). <https://doi.org/10.1109/wisee.2016.7877302>
27. D. Crisan, A. Doucet, A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Process.* **50**(3), 736–746 (2002). <https://doi.org/10.1109/78.984773>
28. S. Särkkä, *Bayesian Filtering and Smoothing*. (Cambridge University Press, New York, 2013)
29. A. Doucet, A. Smith, N. de Freitas, N. Gordon, *Sequential Monte Carlo Methods in Practice*, ser. *Information Science and Statistics*. (Springer, 2001). <https://doi.org/10.1007/978-1-4757-3437-9>
30. 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>
31. A. M. Mathai, S. B. Provost, *Quadratic Forms in Random Variables*, ser. *Statistics: A Series of Textbooks and Monographs*. (Taylor & Francis, 1992)
32. A. C. Rencher, G. B. Schaalje, *Linear Models in Statistics*. (Wiley, 2008)
33. 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>
34. 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>
35. C. S. Withers, The moments of the multivariate normal. *Bull. Aust. Math. Soc.* **32**(1), 103–107 (1985). <https://doi.org/10.1017/S000497270000976X>

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)