Multisensor Multiobject Tracking With Improved Sampling Efficiency

Wenyu Zhang ⓘ, *Student Member, IEEE* and Florian Meyer ⓘ, *Member, IEEE*

Abstract—**Passive monitoring of acoustic or radio sources has important applications in modern convenience, public safety, and surveillance. A key task in passive monitoring is multiobject tracking (MOT). This paper presents a Bayesian method for multisensor MOT for challenging tracking problems where the object states are high-dimensional, and the measurements follow a nonlinear model. Our method is developed in the framework of factor graphs and the sum-product algorithm (SPA) and implemented using random samples or "particles". The multimodal probability density functions provided by the SPA are effectively represented by a Gaussian mixture model (GMM). To perform the operations of the SPA with improved sample efficiency, we make use of particle flow (PFL). Here, particles are migrated towards regions of high likelihood based on the solution of a partial differential equation. This makes it possible to obtain good object detection and tracking performance even in challenging multisensor MOT scenarios with single sensor measurements that have a lower dimension than the object positions. We perform a numerical evaluation in a passive acoustic monitoring scenario where multiple sources are tracked in 3-D from 1-D time-difference-of-arrival (TDOA) measurements provided by pairs of hydrophones. Our numerical results, obtained by processing synthetic and real data, demonstrate favorable detection and estimation accuracy compared to state-of-the-art reference techniques.**

Index Terms—**Multiobject tracking, particle flow, factor graphs, sum-product algorithm.**

I. INTRODUCTION

MULTIOBJECT tracking (MOT) is an important capability for a variety of applications, including surveillance, autonomy, and marine mammal research. MOT is a high-dimensional nonlinear filtering problem complicated by measurement-origin uncertainty (MOU), i.e., the associations between measurements and objects, and an unknown number of objects to be tracked. In this paper, we develop a sequential Bayesian MOT framework for the particularly challenging scenarios where object states are high-dimensional and measurement models are nonlinear. We expect that our approach is particularly useful for the passive monitoring of acoustic [1] or radio [2] sources in 3-D.

A. State-of-the-Art

Traditional methods for MOT include probabilistic data association (PDA) [3], multi-hypothesis tracking (MHT) [4], and methods based on random finite sets (RFS) [5], [6], [7], [8]. Most of these traditional approaches suffer from a computational complexity that is exponential in important system parameters, including the number of measurements, objects, and sensors. MOT methods that are scalable with respect to these parameters have been recently developed in the framework of factor graphs and the SPA [9], [10], [11], [12], [13]. Factor graphs represent statistical independencies of random variables. The SPA is known to provide accurate solutions to high-dimensional Bayesian estimation problems efficiently. In particular, by performing local operations ("messages") on the factor graph, accurate approximations ("beliefs") of the marginal posterior pdfs of unknown states [14] are computed. SPA-based methods are versatile and have been successfully applied to a variety of applications, including cooperative localization [15], [16], [17], [18], simultaneous localization and mapping (SLAM) [19], [20], [21], and focalization for underwater localization [22].

To calculate messages that, due to nonlinearities in the system model, cannot be evaluated in closed form, SPA-based methods for MOT typically rely on particle-based computations that closely follow the bootstrap particle filter (BPF) [23], [24], [25] and rely on importance sampling. A known drawback of this approach is that it typically fails in tracking problems where (i) the states of individual objects have dimensions higher than four, (ii) measurements are very informative compared to the predicted/prior pdfs. In particular, tracking of objects in 3-D Cartesian coordinates or employing sensors that yield low measurement variance often leads to a failure of particle-based computations due to particle degeneracy [26]. The particle degeneracy problem is related to the fact that predicted pdfs are used as proposal pdfs for sampling. Since predicted pdfs can have completely different shapes than the posterior pdfs, this sampling strategy is highly inefficient, i.e., few or none of the generated particles are suitable to represent the posterior
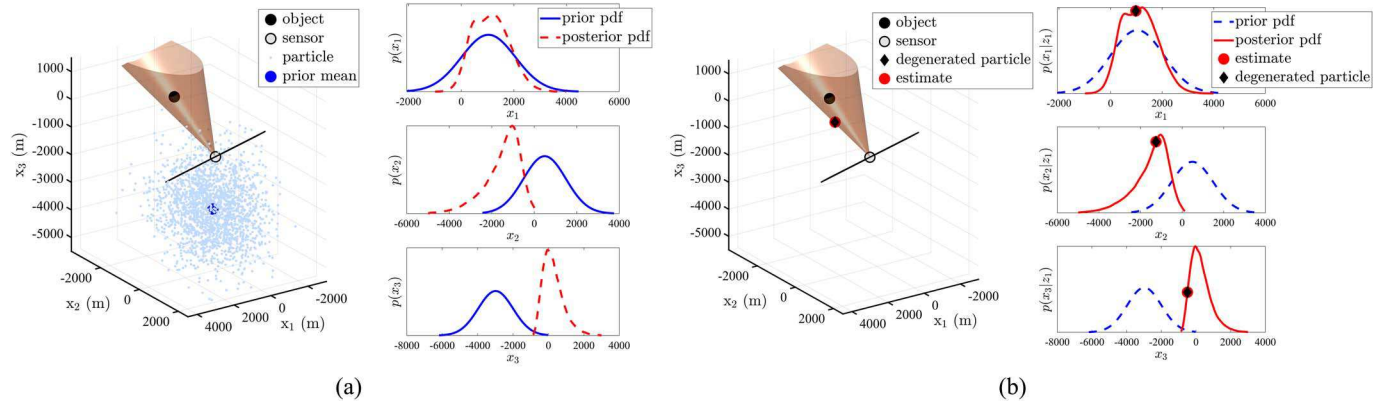
Fig. 1. Particle degeneracy in a tracking scenario with 3-D object state, $\boldsymbol{x} = [x_1 \, x_2 \, x_3]^T$, and a single 1-D TDOA measurement z_1. A single time step is considered. The 1-D TDOA measurement is generated by the sensor shown as gray circle. Assuming no measurement noise, the 1-D TDOA measurement describes potential 3-D object locations on the hyperboloid shown in red. The object, shown in black, is located on the hyperboloid. Note that any other location on the hyperboloid will lead to the same measurement in the case without noise. (a): The prior pdf, $f(\boldsymbol{x})$, is Gaussian, with the mean depicted as a big blue dot. 2000 particles, shown as small light blue dots, are drawn from the prior distribution. On the right, the prior and posterior pdfs for the case with measurement noise are shown in three separate 2-D plots. Each of these plots is obtained by depicting the prior and posterior pdfs along the three axes of the coordinate system. (b): After importance sampling, as performed by the conventional "bootstrap" particle filter, only a single particle has a nonzero weight. This single particle does not accurately represent the posterior pdf $p(\boldsymbol{x}|z_1)$ for future processing, e.g., of a measurement, z_2, provided by a second sensor.

pdfs. Fig. 1 shows an example of particle degeneracy in a 3-D tracking scenario with a single object and a single TDOA measurement. Particle degeneracy is exacerbated in high dimensional problems and in problems with low measurement variance. In particular, it can lead to the unwanted behavior that filter performance degrades as measurement variance is reduced. As the dimension of the problem increases, or measurement variance is reduced, the likelihood function becomes "peakier," and it becomes more unlikely that a particle, sampled from the prior distribution, is located in a region of high likelihood.

Sometimes particle degeneracy can be avoided by using vast numbers of particles or by implementing regularization strategies [16], [25], [27]. A straightforward approach to improve sampling efficiency and avoid particle degeneracy is to design proposal pdfs that are similar to the posterior pdfs [23], [24], [25]. However, finding a distribution that is easy to sample from and simultaneously similar to posterior pdfs [23], [24], [25] is often challenging. To improve sampling efficiency, adaptive importance sampling can be employed [28], [29]. In particular, auxiliary particle filters use a delayed resampling strategy to increase the number of particles with significant weights after importance sampling [30]. This approach can improve sampling efficiency but can only be applied in combination with a prediction step, which may be unavailable for newly introduced object states in MOT scenarios. Furthermore, multiple particle filtering [31], similar to a particle-based implementation of the SPA, aims to increase sample efficiency by exploiting factorization of the underlying statistical model [28]. Since it relies on a suitable factorization of the conditional posterior, its applicability is restricted. Incorporating sequential Markov chain Monte Carlo (SMCMC) methods into particle filters [32], [33] is another general approach for nonlinear sequential Bayesian estimation, which is known to be very computationally expensive in high-dimensional state spaces. An alternative approach to improve

sampling efficiency in sequential estimation is to perform the update step of an unscented Kalman filter [34] and use the resulting Gaussian pdf as a proposal pdf for particle filtering. The unscented particle filter [35], [36] combines this idea with a Gaussian mixture representation of predicted and posterior pdfs. To the best of our knowledge, the unscented particle filtering approach has not yet been extended to problems with MOU and an unknown number of states to be estimated.

PFL [37], [38], [39], [40], [41] is a promising strategy for challenging nonlinear estimation problems that has recently received significant attention [42]. It has the potential to avoid particle degeneracy due to its ability to actively move particles representing a prior or predicted pdf to locations of high likelihood.[1]

This active motion is illustrated in Fig. 2. For PFL a homotopy function is defined to formulate a pdf that can be smoothly deformed from the predicted pdf (or prior pdf) to the posterior pdf. PFL then makes use of the homotopy function to incrementally move a set of particles sampled from the predicted pdf. In particular, a partial differential equation (PDE) for particle velocity is obtained by combining the homotopy function with the Fokker-Planck equation. The particle velocity solution to the PDE can be discretized and used as a transport equation for particle migration. After migration, the set of particles represents the posterior pdf. There are two different types of PFL resulting in the exact Daum and Huang (EDH) filter and the localized exact Daum and Huang (LEDH) filter. In the EDH filter, the PFL equations are computed once for the mean of all particles. In contrast, in the computationally

---

[1]Conventional strategies that rely on resampling also move particles but do so in a more passive way. In particular, in conventional strategies, after particles are randomly drawn, only those that correspond to locations of high likelihood remain after resampling. In challenging problems, this is prone to particle degeneracy, i.e., the number of remaining particles can become too low to be a representative description of the underlying posterior pdf.
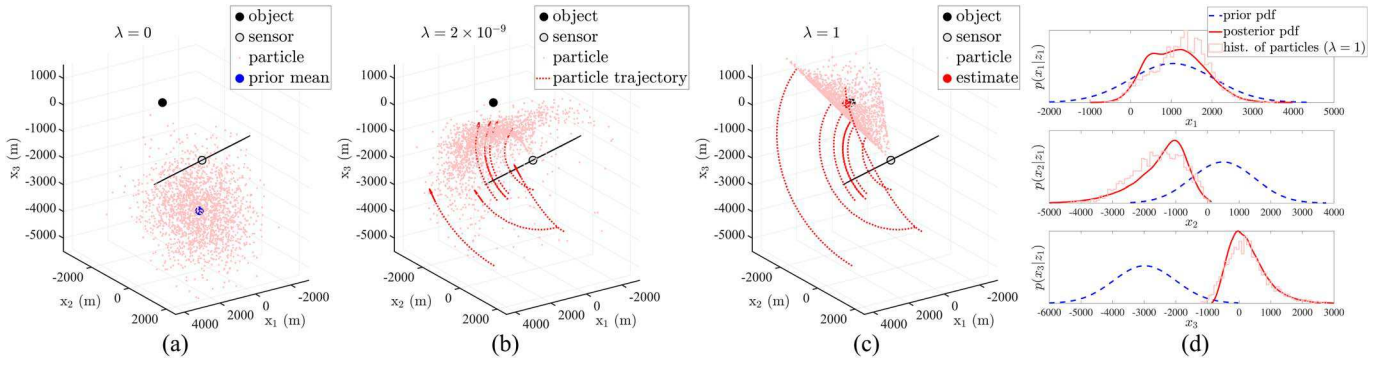
Fig. 2. Example of PFL in the tracking scenario with 3-D object state and a single 1-D TDOA measurement discussed in Fig. 1. (a): 2000 particles represent the prior pdf at the onset of the flow, i.e., at pseudo time $\lambda = 0$, are depicted. (b): An intermediate flow state corresponding to $\lambda = 2 \times 10^{-9}$ is shown. The tracks of 8 selected particles are indicated as red dashed line with arrows. (c): At pseudo time $\lambda = 1$, particle migration is completed and the resulting particles represent the hyperboloid-shaped posterior pdf. (d): The histogram of the flowed particles together with 1-D prior and posterior pdfs is drawn. The representation of the posterior pdf provided by the particles after the flow is much more accurate than the single "degenerated" particle resulting from conventional particle filtering discussed in Fig. 1. Due to approximations performed in PFL, there can be a small mismatch of particles after the flow and the true posterior pdf. Such a mismatch can also be seen in (d) by comparing the posterior pdf with the histogram of particles at $\lambda = 1$. Invertible PFL can eliminate such mismatch and provide an asymptotical optimal representation of the posterior pdf by making it possible to compute particle weights for importance sampling.

more demanding LEDH filter, the PFL equations are computed for each particle individually. PFL has been demonstrated to achieve a superior performance complexity tradeoff compared to existing approaches that aim at improving sampling efficiency [42]. PFL is highly parallelizable [38], [39], [40] and thus ideal for real-time processing on graphical processing units (GPUs).

Traditional PFL methods avoid importance sampling and can only provide an approximate representation of posterior pdfs in general nonlinear systems [37], [38], [39], [40], [41]. Nevertheless, these "proposal-free" methods often lead to accurate estimation results at a significantly reduced computational complexity compared to BPF [42]. Recently, it has been shown that PFL can be described by an invertible mapping and can thus be used as a measurement-driven proposal pdf for importance sampling [42]. The resulting invertible PFL filter [42] is an asymptotically optimal approach to nonlinear filtering that avoids particle degeneracy and can provide accurate estimation results in high-dimensional and nonlinear problems.

A significant limitation of the PFL filter presented in [42] is that it assumes that the prior or predicted pdfs follow Gaussian distributions. It is thus unsuitable for problems that involve multimodal pdfs. For problems where the measurement noise follows a Gaussian mixture pdf, [43] introduces the Gaussian sum PFL filter. Here, the means of the Gaussian mixture components are updated by performing an update step similar to the LEDH. On the other hand, the covariance matrices of the components are updated by extended Kalman filters that also run in parallel. An extension of [43] to the case where both driving noise and measurement noise are distributed by a Gaussian mixture pdf is presented in [44]. Here, invertible flow is used for particle weight update in an importance sampling step. For problems where both driving noise and measurement noise can be multimodal, [45] combines the invertible PFL with a SMCMC method that relies on the Metropolis-Hastings

approach, i.e., a Metropolis-Hastings kernel is constructed using a PFL algorithm based on a GMM. However, aforementioned PFL approaches that can represent multimodal pdfs [43], [44], [45] are unsuitable for MOT since neither model MOU nor an unknown number of states to be estimated. For the cooperative localization problem, a method that relies on invertible PFL is presented in [46], [47]. This method is not suitable for the more challenging MOT problems since it can only be applied to problems without MOU, known number of states to be estimated, and posterior pdfs with simple, unimodal shapes. A variant of the PFL filter has been proposed for MOT [8]. In particular, EDH and LEDH variants of the single-sensor δ-Generalized Labeled Multi-Bernoulli filter [7] with invertible flow are presented. These approaches are unsuitable for MOT problems where measurements are provided by multiple sensors.

B. Contributions, Paper Organization, and Notation

We develop a method for multisensor MOT with improved sample efficiency that can be used in scenarios with high-dimensional object states and informative measurements. Of particular interest are multisensor MOT problems, where inexpensive sensors are used and the tracking of objects in Cartesian coordinates is impossible based on the measurements provided by a single sensor. In this type of tracking problems, the measurement of a single sensor typically has a lower dimension than the positions of objects. Consider a scenario where object positions are 3-D, but sensors only provide 1-D measurements, e.g., times of arrival (TOAs), time differences of arrival (TDOAs), or directions of arrival (DOAs). In this type of MOT problem, prior or predicted pdfs can have complicated multimodal shapes, e.g., spheres, hyperboloids, or cones at the initial step after the appearance of a new object. As an example, Figs. 1 and 2 show the hyperboloid-shaped pdfs resulting from a TDOA measurement model in a 3-D tracking scenario.

Our approach performs SPA-based message passing on the factor graph for scalable multisensor MOT developed in [10]. The messages of the SPA are computed sequentially across sensors. To improve sampling efficiency, we embed invertible PFL into SPA computations. For the evaluations of particle weights, invertible PFL relies on a Gaussian representation of the prior or predicted pdf at the onset of the flow. To represent beliefs of object states with complicated non-Gaussian shapes, such as, e.g., hyperboloids, as in the example in Figs. 1 and 2, we make use of a GMM representation that is known to be asymptotically optimal [48]. Combining a GMM with an efficient sampling approach to represent pdfs with complicated shapes in high dimensions is inspired by unscented particle filtering [35], [36]. A general proposal pdf that takes MOU into account and consists of a mixture of pdfs related to different particle flows is developed. The resulting computations are asymptotically optimal. In particular, since particles are migrated towards regions of high likelihood, an accurate approximation of SPA messages with a relatively small number of particles is obtained.

The technical novelty of the proposed method lies in a new method for multitarget tracking that can achieve a superior runtime–estimation accuracy tradeoff in nonlinear and high-dimensional problems by improving sampling efficiency. The improved tradeoff is obtained by carefully embedding invertible PFL. In particular, to address MOU, association probabilities are computed by performing parallel flows, one for each component of the GMM and each possible measurement-to-object association. The particles of the parallel flows are weighted based on association probabilities and combined into a mixture of flows. The mixture of flows provides samples of the proposal for importance sampling. Since all flows are invertible, it is possible to evaluate the proposal pdf represented by the mixture of flows at each particle. Thus, the resulting SPA-based computation of beliefs is asymptotically optimal in the sense that the resulting particle representation of the beliefs provided by the SPA is arbitrarily accurate for an increasingly large number of Gaussian components and a number of particles. Our method, for the first time, performs MOT with probabilistic data association (DA) based on PFL.

We further demonstrate that the proposed multisensor MOT can outperform reference methods based on conventional ("bootstrap") and unscented particle filtering in a 3-D passive source tracking scenario. In particular, in the considered realistic source tracking scenario, graph-based MOT based on conventional particle filtering [9] cannot provide acceptable estimation accuracy. The also considered, yet unpublished, implementation of graph-based MOT based on unscented particle filtering, has a lower estimation accuracy but a higher runtime compared to the proposed method that embeds invertible PFL.

Key contributions of this paper are as follows.

- We develop a graph-based MOT method based on a GMM and invertible PFL for challenging scenarios with high-dimensional object states and arbitrarily shaped posterior pdfs.
- We demonstrate that the proposed method can significantly outperform reference techniques in a challenging 3-D

passive source multisensor MOT scenario and show tracking results using real passive acoustic data.

This paper advances over the preliminary account of our method provided in the conference publication [49] by (i) introducing a GMM for multimodal state distribution with dynamic kernel resampling; (ii) considering the multisensor MOT problem; (iii) presenting an improved proposal distribution based on PFL; (iv) performing a comprehensive numerical evaluation in a 3-D passive source tracking scenario; and (v) applying the proposed method to an underwater acoustic dataset[2]. Contrary to the approach presented in [8], the proposed method is suitable for multisensor scenarios. In addition, SPA-based processing makes our approach scalable with respect to relevant system parameters.

Notation: Random variables are displayed in sans serif, upright fonts and their realizations in serif, italic fonts. Vectors and matrices are denoted by bold lowercase and uppercase letters, respectively. For example, a random variable and its realization are denoted by x and x, respectively, and a random vector and its realization by $\mathbf{x}$ and $\boldsymbol{x}$, respectively. Furthermore, $\|\boldsymbol{x}\|$ and $\boldsymbol{x}^{\mathrm{T}}$ denote the Euclidean norm and the transpose of vector $\boldsymbol{x}$, respectively; and $\propto$ indicates equality up to a normalization factor. $\mathcal{N}(\boldsymbol{x}; \boldsymbol{x}^*, \boldsymbol{P})$ denotes the Gaussian pdf (of random vector $\mathbf{x}$) with mean $\boldsymbol{x}^*$ and covariance matrix $\boldsymbol{P}$. The trace of matrix $\boldsymbol{M}$ is denoted as $\mathrm{Tr}\{\boldsymbol{M}\}$. Finally, $1(a)$ denotes the indicator function of the event $a = 0$, i.e., $1(a) = 1$ if $a = 0$ and 0 otherwise.

II. REVIEW OF INVERTIBLE PFL

We consider the general setting of calculating the posterior pdf based on Bayes' rule $f(\boldsymbol{x}|\boldsymbol{z}) \propto f(\boldsymbol{x}) f(\boldsymbol{z}|\boldsymbol{x})$ with the state of interest $\boldsymbol{x}$ and the observed (fixed) measurement $\boldsymbol{z}$. If the prior pdf $f(\boldsymbol{x})$ follows a Gaussian distribution and the likelihood function $f(\boldsymbol{z}|\boldsymbol{x})$ represents a linear measurement model $\mathbf{z} = \boldsymbol{H}\mathbf{x} + \mathbf{v}$ with Gaussian measurement noise $\mathbf{v}$, the posterior pdf $f(\boldsymbol{x}|\boldsymbol{z})$ also follows a Gaussian distribution. In this special case, the mean and covariance of the Gaussian posterior pdf $f(\boldsymbol{x}|\boldsymbol{z})$ can be calculated in closed form by the Kalman update step [51].

If the measurement model is nonlinear, e.g., $\mathbf{z} = \boldsymbol{h}(\mathbf{x}) + \mathbf{v}$, a popular approach is to approximate the posterior pdf $f(\boldsymbol{x}|\boldsymbol{z})$ by a set of N_{P} weighted particles $\{(\boldsymbol{x}^{(i)}, w^{(i)})\}_{i=1}^{N_{\mathrm{P}}}$. Note that the weights are normalized to one, i.e., $\sum_{i=1}^{N_{\mathrm{P}}} w^{(i)} = 1$ and can be computed based on the importance sampling principle [25] as follows

$$w^{(i)} \propto \frac{f(\boldsymbol{x}^{(i)}) f(\boldsymbol{z}|\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)}|\boldsymbol{z})}. \tag{1}$$

Here, the proposal pdf $q(\boldsymbol{x}|\boldsymbol{z})$ is used to sample the particles $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N_{\mathrm{P}}}$. It is an arbitrary pdf that has the same support as $f(\boldsymbol{x}|\boldsymbol{z})$. Importance sampling is asymptotically optimal if $q(\boldsymbol{x}|\boldsymbol{z})$ is "heavier tailed", i.e., less informative, than $f(\boldsymbol{x}|\boldsymbol{z})$ [24]. In particular, importance sampling can provide

---

[2]More details on the application of the proposed method to the problem of tracking multiple whales underwater by performing TDOA measurements of their echolocation clicks, is presented in the companion paper [50].

an approximation of $f(\boldsymbol{x}|\boldsymbol{z})$ that can be made arbitrarily good by choosing N_{p} sufficiently large [25]. For N_{p} fixed, if the proposal $q(\boldsymbol{x}|\boldsymbol{z})$ is "more similar" to the posterior $f(\boldsymbol{x}|\boldsymbol{z})$ [24], importance sampling is "more accurate".

A simple choice for the proposal pdf used in the update step of the conventional "bootstrap" particle filter [23], [25] is the prior pdf $f(\boldsymbol{x})$. However, for a feasible number of particles N_{p} and most choices of the proposal pdf, importance sampling can suffer from particle degeneracy [26]. Particle degeneracy is especially severe if the state $\boldsymbol{x}$ is high-dimensional and the measurement $\boldsymbol{z}$ is informative (i.e., the likelihood function has narrow and sharp peaks).

A. Particle Flow (PFL)

PFL [37], [38], [39], [40] is an approach that aims at avoiding particle degeneracy. Here, particles are smoothly migrated in the state space from a representation of the prior pdf to a representation of the posterior pdf by solving a PDE. Let us introduce the homotopy function $\pi_\lambda(\boldsymbol{x}) = f(\boldsymbol{x})l^\lambda(\boldsymbol{x})$ where $\lambda \in [0,1]$ is the pseudo time of the flow process and $l(\boldsymbol{x}) = f(\boldsymbol{z}|\boldsymbol{x})$ is the likelihood function. Note that for $\lambda = 1$, the homotopy function is equal to the unnormalized posterior pdf, i.e., $\pi(\boldsymbol{x}) \triangleq \pi_1(\boldsymbol{x}) = f(\boldsymbol{x})l(\boldsymbol{x})$. The log-homotopy function is then given by [37], [38],

$$\phi(\boldsymbol{x}, \lambda) = \log f(\boldsymbol{x}) + \lambda \log l(\boldsymbol{x}). \tag{2}$$

The log-homotopy function is a pseudo posterior pdf in the log domain that defines a smooth and continuous deformation from $\phi(\boldsymbol{x}, 0) = \log f(\boldsymbol{x})$ to $\phi(\boldsymbol{x}, 1) = \log \pi(\boldsymbol{x})$. This deformation describes the PFL process.

It can be shown that the stochastic process defined by homotopy function $\pi_\lambda(\boldsymbol{x})$ satisfies the Fokker-Planck equation [39], [40], [41]. Combining the Fokker-Planck equation for the zero-diffusion case with (2) results in the following PDE [39], [40]

$$\frac{\partial \phi(\boldsymbol{x}, \lambda)}{\partial \boldsymbol{x}} \boldsymbol{\zeta}(\boldsymbol{x}, \lambda) + \log l(\boldsymbol{x}) = -\mathrm{Tr}\left(\frac{\partial \boldsymbol{\zeta}(\boldsymbol{x}, \lambda)}{\partial \boldsymbol{x}}\right) \tag{3}$$

where $\boldsymbol{\zeta}(\boldsymbol{x}, \lambda) = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\lambda}$ describes particle velocity (samples of $\mathbf{x}$) as the pseudo time λ increases from 0 to 1, i.e., as the homotopy function is deformed from the prior pdf to the posterior pdf. This migration is referred to as the PFL.

B. PFL Update Step

If $f(\boldsymbol{x})$ and $l(\boldsymbol{x})$ are Gaussians or in another exponential family, then an exact and closed form solution for (3) is available. The EDH filter [39], [52] makes use of this closed-form solution in its update step. More precisely, let $f(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}_0^*, \boldsymbol{P})$ and $\mathbf{z} = \boldsymbol{H}\mathbf{x} + \mathbf{v}$ be a linear measurement model with measurement noise $\mathbf{v} \sim \mathcal{N}(\boldsymbol{v}; 0, \boldsymbol{R})$. The exact flow solution [39], [52] now reads $\boldsymbol{\zeta}(\boldsymbol{x}, \lambda) = \boldsymbol{A}(\lambda)\boldsymbol{x} + \boldsymbol{b}(\lambda)$ where we introduce

$$\boldsymbol{A}(\lambda) = -\frac{1}{2}\boldsymbol{P}\boldsymbol{H}^{\mathrm{T}}(\lambda \boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1}\boldsymbol{H} \tag{4}$$

and

$$\boldsymbol{b}(\lambda) = (\boldsymbol{I} + 2\lambda\boldsymbol{A}(\lambda))\big[(\boldsymbol{I} + \lambda\boldsymbol{A}(\lambda))\boldsymbol{P}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{z} + \boldsymbol{A}(\lambda)\boldsymbol{x}_0^*\big]. \tag{5}$$

Algorithm 1: PFL Update Step

1   $\left[\{\boldsymbol{x}_1^{(i)}\}_{i=1}^{N_{\mathrm{P}}}, \{A_l\}_{l=1}^{N_\lambda}\right] = \mathrm{ParticleFlow}\left(\{\boldsymbol{x}_0^{(i)}\}_{i=1}^{N_{\mathrm{P}}}, \boldsymbol{x}_0^*, \boldsymbol{P}, \boldsymbol{z}\right)$

2   Define pseudo time steps $0 = \lambda_0 < \lambda_1 < ... < \lambda_{N_\lambda} = 1$;

3   **for** $l = 1 : N_\lambda$ **do**

4   Calculate the linearized measurement model $\boldsymbol{H}_l$ according to (9);

5   Compute $\boldsymbol{A}_l$ and $\boldsymbol{b}_l$ according to (7) and (8);

6   **for** $i = 1 : N_{\mathrm{P}}$ **do**

7         $\tilde{\boldsymbol{\zeta}}(\boldsymbol{x}_{\lambda_{l-1}}^{(i)}, \lambda_l) = \boldsymbol{A}_l\boldsymbol{x}_{\lambda_{l-1}}^{(i)} + \boldsymbol{b}_l$;

8         $\boldsymbol{x}_{\lambda_l}^{(i)} = \boldsymbol{x}_{\lambda_{l-1}}^{(i)} + \tilde{\boldsymbol{\zeta}}(\boldsymbol{x}_{\lambda_{l-1}}^{(i)}, \lambda_l)(\lambda_l - \lambda_{l-1})$;

9      $\tilde{\boldsymbol{\zeta}}(\boldsymbol{x}_{\lambda_{l-1}}^*, \lambda_l) = \boldsymbol{A}_l\boldsymbol{x}_{\lambda_{l-1}}^* + \boldsymbol{b}_l$;

10    $\boldsymbol{x}_{\lambda_l}^* = \boldsymbol{x}_{\lambda_{l-1}}^* + \tilde{\boldsymbol{\zeta}}(\boldsymbol{x}_{\lambda_{l-1}}^*, \lambda_l)(\lambda_l - \lambda_{l-1})$;

11   Output: $\{\boldsymbol{x}_1^{(i)}\}_{i=1}^{N_{\mathrm{P}}} \triangleq \{\boldsymbol{x}_{\lambda_{N_\lambda}}^{(i)}\}_{i=1}^{N_{\mathrm{P}}}$ and $\{A_l\}_{l=1}^{N_\lambda}$

Note that in (5), $\boldsymbol{z}$ is the observed and thus fixed measurement.

This solution is extended to the nonlinear measurement model $\mathbf{z} = \boldsymbol{h}(\mathbf{x}) + \mathbf{v}$ by performing a suboptimal linearization step. In particular, in a first-order approximation, a Jacobian matrix is computed, i.e. $\boldsymbol{H}(\lambda) = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_\lambda^*}$ where $\boldsymbol{x}_\lambda^*$ is the approximated mean of $\boldsymbol{x}$ at pseudo time λ.

In a practical implementation, we calculate $\boldsymbol{\zeta}(\boldsymbol{x}, \lambda)$ at N_λ discrete values of λ, i.e., $0 = \lambda_0 < \lambda_1 < ... < \lambda_{N_\lambda} = 1$, to perform the PFL. Here, we first sample N_{p} particles $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^{N_{\mathrm{P}}} \triangleq \{\boldsymbol{x}_{\lambda_0}^{(i)}\}_{i=1}^{N_{\mathrm{P}}}$ from $f(\boldsymbol{x})$. Next, at each discrete pseudo time step $l \in \{1, \ldots, N_\lambda\}$, particles are migrated according to

$$\boldsymbol{x}_{\lambda_l}^{(i)} = \boldsymbol{x}_{\lambda_{l-1}}^{(i)} + \tilde{\boldsymbol{\zeta}}(\boldsymbol{x}_{\lambda_{l-1}}^{(i)}, \lambda_l)(\lambda_l - \lambda_{l-1}) \tag{6}$$

for all $i \in \{1, \ldots, N_{\mathrm{p}}\}$. Here, the linearized flow solution $\tilde{\boldsymbol{\zeta}}(\boldsymbol{x}_{\lambda_{l-1}}^{(i)}, \lambda_l) = \boldsymbol{A}_l\boldsymbol{x}_{\lambda_{l-1}}^{(i)} + \boldsymbol{b}_l$ is computed based on $\boldsymbol{A}_l$ and $\boldsymbol{b}_l$ given by (cf. (4) and (5))

$$\boldsymbol{A}_l = -\frac{1}{2}\boldsymbol{P}\boldsymbol{H}_l^{\mathrm{T}}(\lambda_l\boldsymbol{H}_l\boldsymbol{P}\boldsymbol{H}_l^{\mathrm{T}} + \boldsymbol{R})^{-1}\boldsymbol{H}_l \tag{7}$$

$$\boldsymbol{b}_l = (\boldsymbol{I} + 2\lambda_l\boldsymbol{A}_l)[(\boldsymbol{I} + \lambda_l\boldsymbol{A}_l)\boldsymbol{P}\boldsymbol{H}_l^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{z} - \boldsymbol{e}_l) + \boldsymbol{A}_l\boldsymbol{x}_0^*]. \tag{8}$$

Note that $\boldsymbol{e}_l = \boldsymbol{h}(\boldsymbol{x}_{\lambda_{l-1}}^*, 0) - \boldsymbol{H}_l\boldsymbol{x}_{\lambda_{l-1}}^*$ is the error of linearization and that the linearized measurement model, $\boldsymbol{H}_l$, is computed based on the mean of the last step $l - 1$, i.e.,

$$\boldsymbol{H}_l = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{\lambda_{l-1}}^*}. \tag{9}$$

The mean at which the measurement model is linearized is typically propagated in parallel to the particles, i.e., $\boldsymbol{x}_{\lambda_l}^* = \boldsymbol{x}_{\lambda_{l-1}}^* + \tilde{\boldsymbol{\zeta}}(\boldsymbol{x}_{\lambda_{l-1}}^*, \lambda_l)(\lambda_l - \lambda_{l-1})$.

After the last discrete pseudo time step, $l = N_\lambda$, particles $\{\boldsymbol{x}_1^{(i)}\}_{i=1}^{N_{\mathrm{P}}} \triangleq \{\boldsymbol{x}_{\lambda_{N_\lambda}}^{(i)}\}_{i=1}^{N_{\mathrm{P}}}$ that approximately represent the unnormalized posterior pdf $\pi(\boldsymbol{x})$ are finally obtained. Pseudocode for the PFL update step is provided in **Algorithm 1**. PFL based on a linearized model has no optimality guarantees. However, it has been demonstrated numerically to typically provide an accurate representation of the posterior pdf $f(\boldsymbol{x}|\boldsymbol{z})$ [37], [38], [39], [40], [41], [42]. In what follows, the PFL related to the measurement $\boldsymbol{z}$ as defined by (6)–(8), is denoted as $\mathbf{x}_0 \rightarrow \boldsymbol{z} \rightarrow \mathbf{x}_1$ or for notational convenience in future derivations as $\mathbf{x}_0 \rightarrow \boldsymbol{z} \rightarrow \mathbf{x}$.

As an alternative to EDH PFL as discussed above, LEDH PFL has been introduced in [42], [53]. Here, a linearization is performed for each particle location individually instead of only at the mean particle location, i.e., individual flow parameters $\boldsymbol{H}_l^{(i)}$, $\boldsymbol{A}_l^{(i)}$, and $\boldsymbol{b}_l^{(i)}$ are computed to migrate the particles along their individual flows. Although the LEDH flow usually outperforms the EDH flow, it suffers from considerable computational complexity since the main computational burden is related to calculating the flow parameters.

In Fig. 2, it is depicted how PFL actively move particles representing a prior or predicted pdfs to locations of high likelihood. Active motion of particles leads to a significantly improved approximation of the posterior pdf compared to conventional importance sampling shown in Fig. 1.

C. Importance Sampling With Invertible Flow

PFL can be used to compute a measurement-driven proposal pdf $q(\boldsymbol{x}|\boldsymbol{z})$ for importance sampling (cf. (1)) to perform asymptotically optimal estimation [42]. Here, the mapping as performed by PFL $\mathsf{x}_0 \rightarrow \boldsymbol{z} \rightarrow \mathsf{x}$ is invertible, i.e., there exists an invertible mapping of the particles after the flow $\big\{\boldsymbol{x}^{(i)}\big\}_{i=1}^{N_{\mathrm{P}}}$ to the particles $\big\{\boldsymbol{x}_0^{(i)}\big\}_{i=1}^{N_{\mathrm{P}}}$ if certain constraints on the differences of consecutive discrete pseudo times $\lambda_l - \lambda_{l-1}$, $l \in \{1, \ldots, N_\lambda\}$ are satisfied [42].

By exploiting the invertible mapping, the proposal pdf resulting from PFL can be evaluated at the particles as [42]

$$q_{\mathrm{PFL}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) = \frac{f(\boldsymbol{x}_0^{(i)})}{\theta}. \tag{10}$$

Here, the "mapping factor" θ is defined as

$$\theta = \prod_{l=1}^{N_\lambda} \big| \det\big[\boldsymbol{I} + (\lambda_l - \lambda_{l-1})\boldsymbol{A}_l\big]\big|. \tag{11}$$

By plugging (10) into (1) the weight of the particle $\boldsymbol{x}^{(i)}$ is obtained as

$$w^{(i)} \propto \frac{\theta f(\boldsymbol{z}|\boldsymbol{x}^{(i)})f(\boldsymbol{x}^{(i)})}{f(\boldsymbol{x}_0^{(i)})}. \tag{12}$$

The resulting particle set $\{\boldsymbol{x}^{(i)}, w^{(i)}\}_{i=1}^{N_{\mathrm{P}}}$ is an asymptotically optimal sample representation of the posterior pdf $f(\boldsymbol{x}|\boldsymbol{z})$ that can often provide accurate estimation results in nonlinear and high-dimensional estimation problems even if the number of particles is moderate [42]. Pseudocode for importance sampling with invertible PFL is provided in **Algorithm 2**. Note that since the PFL used for the measurement-driven proposal pdf is typically based on the EDH filter update step, a Gaussian prior pdf is assumed.

An approximate Gaussian representation of this posterior distribution can be subsequently obtained by applying **Algorithm 3** which calculates a mean $\boldsymbol{x}_1^*$ and a covariance matrix $\boldsymbol{P}_1$ from the unnormalized weighted particles $\big\{\boldsymbol{x}_1^{(i)}, w_1^{(i)}\big\}_{i=1}^{N_{\mathrm{P}}}$. Note that in Algorithm 2, the same mapping factor θ is used to calculate all particle weights. If Algorithm 3 is applied after Algorithm 2, this factor is irrelevant since all weights are normalized in Algorithm 3. However, making use of θ is important

---

Algorithm 2: Importance Sampling with Invertible Flow

1 $\Big[\big\{\boldsymbol{x}_1^{(i)}, w_1^{(i)}\big\}_{i=1}^{N_{\mathrm{P}}}\Big] = \mathrm{InvertibleFlow}(\boldsymbol{x}^*, \boldsymbol{P}, \boldsymbol{z})$

2 **for** $i = 1 : N_{\mathrm{P}}$ **do**

3   Draw $\boldsymbol{x}_0^{(i)} \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}^*, \boldsymbol{P})$;

4 Perform PF according
    $\Big[\{\boldsymbol{x}_1^{(i)}\}_{i=1}^{N_{\mathrm{P}}}, \{A_l\}_{l=1}^{N_\lambda}\Big] = \mathrm{ParticleFlow}\Big(\{\boldsymbol{x}_0^{(i)}\}_{i=1}^{N_{\mathrm{P}}}, \boldsymbol{x}^*, \boldsymbol{P}, \boldsymbol{z}\Big)$;

5   // see **Alg. 1**

6 Compute the mapping factor θ from $\{A_l\}_{l=1}^{N_\lambda}$ following (11);

7 **for** $i = 1 : N_{\mathrm{P}}$ **do**

8   Perform weight update according to (12), i.e.,
    $w_1^{(i)} = \frac{\theta f\big(\boldsymbol{z}|\boldsymbol{x}_1^{(i)}\big)\mathcal{N}\big(\boldsymbol{x}_1^{(i)};\boldsymbol{x}^*,\boldsymbol{P}\big)}{\mathcal{N}\big(\boldsymbol{x}_0^{(i)};\boldsymbol{x}^*,\boldsymbol{P}\big)}$;

9 Output: $\big\{\boldsymbol{x}_1^{(i)}, w_1^{(i)}\big\}_{i=1}^{N_{\mathrm{P}}}$

Algorithm 3: Computation of Gaussian Representation

1 $\big[\boldsymbol{x}^*, \boldsymbol{P}\big] = \mathrm{GaussianRepresentation}\Big(\big\{\boldsymbol{x}^{(i)}, \omega^{(i)}\big\}_{i=1}^{N_{\mathrm{P}}}\Big)$

2 Normalize particles, i.e.,

3 **for** $i = 1 : N_{\mathrm{P}}$ **do**

4      $w^{(i)} = \frac{\omega^{(i)}}{\sum_{i'=1}^{N_{\mathrm{P}}} \omega^{(i')}}$;

5 Compute mean and covariance matrix from particles, i.e.,

6 $\boldsymbol{x}^* = \sum_{i=1}^{N_{\mathrm{P}}} w^{(i)}\boldsymbol{x}^{(i)}$

7 $\boldsymbol{P} = \sum_{i=1}^{N_{\mathrm{P}}} w^{(i)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(i)\,\mathrm{T}} - \boldsymbol{x}^*\boldsymbol{x}^{*\,\mathrm{T}}$

8 Output: $\boldsymbol{x}^*, \boldsymbol{P}$

---

if multiple flows are performed in parallel, as will be discussed in Section III.

Note that instead of a particle-based covariance matrix computation as performed in Line 7 of Algorithm 3, an extended or unscented Kalman update step can be used [42]. For a large number of particles N_{P}, a particle-based computation is more accurate than a computation based on the extended or unscented Kalman update step.

III. GAUSSIAN MIXTURE REPRESENTATION FOR NONLINEAR ESTIMATION IN HIGH-DIMENSIONS

In this section, we use GMM representations [54], [55], [56] for nonlinear estimation in high dimensions. In particular, we developed methods for updating the parameters of GMMs based on PFL. This approach is suitable for high-dimensional pdfs that are multimodal and thus relevant for estimation problems in MOT and SLAM [19].

A. GMM Importance Sampling With Invertible Flow

As discussed in the previous Section II, for the evaluations of particle weights, invertible particle flow relies on a Gaussian representation of the prior pdf at the onset of the flow. In challenging multisensor MOT problems, the complicated multimodal shapes of prior and posterior pdfs (see, e.g., the hyperboloid-shaped posterior pdf in Fig. 2(c)) can often not be approximated accurately by a single Gaussian. A GMM aims at representing multimodal distributions based on an additively weighted combination of multiple Gaussian components. Each Gaussian component is typically referred to as

---

Algorithm 4: GMM Importance Samp. with Invertible Flow

---

1   $\left[\{\boldsymbol{x}_+^{*(h)}, \boldsymbol{P}_+^{(h)}\}_{h=1}^{N_\mathrm{k}}\right] = \mathrm{InvertibleFlowGMM}\left(\{\boldsymbol{x}^{*(h)}, \boldsymbol{P}^{(h)}\}_{h=1}^{N_\mathrm{k}}, \boldsymbol{z}\right)$

2   **for** $h = 1 : N_\mathrm{k}$ **do**

3      $\left[\{\boldsymbol{x}_1^{(i,h)}, w_1^{(i,h)}\}_{i=1}^{N_\mathrm{P}}\right] = \mathrm{InvertibleFlow}\left(\boldsymbol{x}^{*(h)}, \boldsymbol{P}^{(h)}, \boldsymbol{z}\right)$

4   // see **Alg. 2**

5      $\left[\sim, \boldsymbol{P}_1^{(h)}\right] = \mathrm{GaussianRepresentation}\left(\{\boldsymbol{x}_1^{(i,h)}, w_1^{(i,h)}\}_{i=1}^{N_\mathrm{P}}\right)$

6   // see **Alg. 3**

7   $\left[\{\boldsymbol{x}_+^{*(h)}, \boldsymbol{P}_+^{(h)}\}_{h=1}^{N_\mathrm{k}'}\right] = \mathrm{Resampling}\left(\left\{\boldsymbol{P}_1^{(h)}, \{\boldsymbol{x}_1^{(i,h)}, w_1^{(i,h)}\}_{i=1}^{N_\mathrm{P}}\right\}_{h=1}^{N_\mathrm{k}}\right)$

8   // see **Alg. 5**

9   Output: $\{\boldsymbol{x}_+^{*(h)}, \boldsymbol{P}_+^{(h)}\}_{h=1}^{N_\mathrm{k}'}$

Algorithm 5: Resampling

---

1   $\left[\{\boldsymbol{x}_+^{*(h')}, \boldsymbol{P}_+^{(h')}\}_{h'=1}^{N_\mathrm{k}'}\right] = \mathrm{Resampling}\left(\left\{\boldsymbol{P}_1^{(h)}, \{\boldsymbol{x}_1^{(i,h)}, w_1^{(i,h)}\}_{i=1}^{N_\mathrm{P}}\right\}_{h=1}^{N_\mathrm{k}}\right)$

2   **for** $h = 1 : N_\mathrm{k}$ **do**

3   **for** $i = 1 : N_\mathrm{P}$ **do**

4          $w_1^{(i,h)} = \dfrac{w_1^{(i,h)}}{\sum_{h'=1}^{N_\mathrm{k}} \sum_{i'=1}^{N_\mathrm{P}} w_1^{(i',h')}}$;

5   **for** $h' = 1 : N_\mathrm{k}'$ **do**

6   Sample index $(i', \hbar)$ using $\left\{\{w_1^{(i,h)}\}_{i=1}^{N_\mathrm{P}}\right\}_{h=1}^{N_\mathrm{k}}$;

7   Set $\boldsymbol{x}_+^{*(h')} = \boldsymbol{x}_1^{(i',\hbar)}$ and $\boldsymbol{P}_+^{(h')} = \boldsymbol{P}_1^{(\hbar)}$;

8   Output: $\{\boldsymbol{x}_+^{*(h')}, \boldsymbol{P}_+^{(h')}\}_{h'=1}^{N_\mathrm{k}'}$

---

a "kernel". Let N_k be the total number of kernels and let $h \in \{1, \dots, N_\mathrm{k}\}$ be the kernel index. A multimodal prior pdf that follows a GMM representation can then be written as $f(\boldsymbol{x}) = \frac{1}{N_\mathrm{k}} \sum_{h=1}^{N_\mathrm{k}} \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}^{*(h)}, \boldsymbol{P}^{(h)})$. The corresponding multimodal posterior pdf $f(\boldsymbol{x}|\boldsymbol{z})$ can be computed by performing Algorithm 2 and Algorithm 3 N_k times in parallel, i.e., one instance of both algorithms is performed for each kernel $\mathcal{N}(\boldsymbol{x}; \boldsymbol{x}^{*(h)}, \boldsymbol{P}^{(h)})$, $h \in \{1, \dots, N_\mathrm{k}\}$. To obtain a GMM representation composed of an arbitrary number N_k' of kernels, a resampling step is then performed, i.e., N_k' particles are drawn from the overall $N_\mathrm{k} N_\mathrm{P}$ particles based on their weights $w_1^{(h,i)}$. The resampled particles represent the mean of N_k' new kernels. The covariance of the new kernels is inherited from the original kernel the mean was sampled from. Pseudocode for GMM importance sampling with invertible PFL is provided in **Algorithm 4**. Importance sampling with invertible PFL makes use of resampling as presented in **Algorithm 5**.

For $N_\mathrm{k} = 1$ and $N_\mathrm{P} > 1$, this importance sampling approach is equivalent to invertible PFL based on the EDH update step. Furthermore, if $N_\mathrm{P} = 1$, $N_\mathrm{k} > 1$, this importance sampling approach is equivalent to invertible PFL based on the LEDH update step. Note that for $N_\mathrm{P} = 1$, as performed by the LEDH, an additional extended or unscented Kalman update step needs to be used to calculate an approximate covariance matrix [42].

B. Measurement-Origin Uncertainty (MOU)

In a variety of estimation problems, the measurement model suffers from a deficiency beyond measurement noise referred

to as MOU [3]. Here, there is a single object but multiple measurements and it is not known which measurement was generated by the object. Consider a single object with state $\mathbf{x}$ and measurements $\mathbf{z}^{(m)}$, $m \in \{1, \dots, M\}$. If m' is the measurement that was generated by the object, the corresponding measurement model is given by $\mathbf{z}^{(m')} = \boldsymbol{h}(\mathbf{x}) + \mathbf{v}$. Based on this model, the conditional pdf of the object-originated measurement $\boldsymbol{z}^{(m')}$ reads $f_\mathrm{o}(\boldsymbol{z}^{(m')}|\boldsymbol{x})$. All the other measurements are false positives (FPs) that follow the FP pdf $f_\mathrm{fp}(\boldsymbol{z}^{(m')})$. It is assumed that at most one measurement originates from the object. The probability that the object generates a measurement is p_d, and the mean number of FPs is Poisson distributed with mean μ_fp. Since it is unknown which measurement was generated by the object, a discrete and random association variable $\mathsf{a} \in \{0, 1, \dots, M\}$ is introduced. Here, $\mathsf{a} = 0$ describes the event where no measurement originated from the object and $\mathsf{a} = m$, $m \in \{1, \dots, M\}$ describes the event where measurement $\mathbf{z}^{(m)}$ was originated from the object. Let $\mathbf{z} = [\mathbf{z}^{(1)\mathrm{T}}, \dots, \mathbf{z}^{(M)\mathrm{T}}]^\mathrm{T}$ be the joint measurement vector. Following common assumptions [3], conditioned on $\mathbf{x}$, the joint pdf of $\mathbf{z}$ and a is given by

$$f(\boldsymbol{z}, a|\boldsymbol{x}) \propto \begin{cases} \dfrac{p_\mathrm{d} f(\boldsymbol{z}^{(m)}|\boldsymbol{x})}{\mu_\mathrm{fp} f_\mathrm{fp}(\boldsymbol{z}^{(m)})}, & a = m \in \{1, \dots, M\} \\ 1 - p_\mathrm{d}, & a = 0. \end{cases} \quad (13)$$

For $\boldsymbol{z}$ fixed, one can use this conditional pdf to directly compute the MOU likelihood function

$$f(\boldsymbol{z}|\boldsymbol{x}) = \sum_{a=0}^{M} f(\boldsymbol{z}, a|\boldsymbol{x})$$
$$\propto 1 - p_\mathrm{d} + \sum_{a=1}^{M} \frac{p_\mathrm{d} f(\boldsymbol{z}^{(a)}|\boldsymbol{x})}{\mu_\mathrm{fp} f_\mathrm{fp}(\boldsymbol{z}^{(a)})} \quad (14)$$

and the marginal probability mass function (pmf)

$$p(a|\boldsymbol{z}) \propto f(a, \boldsymbol{z})$$
$$= \int f(\boldsymbol{z}, a|\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \quad (15)$$

The values of the pmf $p(a|\boldsymbol{z})$ are also referred to as marginal association probabilities [3], i.e., they represent the probability of a particular association event $a \in \{0, \dots, M\}$ conditioned on an observed $\boldsymbol{z}$.

C. Importance Sampling With Invertible PFL for Problems With MOU

In principle, the MOU likelihood function in (14) can be directly used for importance sampling as in (1). However, in problems with MOU, importance sampling based on invertible PFL is complicated by the fact that there are multiple measurements, and it is thus not clear which measurement should be used to compute PFL parameters (7) and (8), i.e., the PFL proposal pdf $q_\mathrm{PFL}(\boldsymbol{x}_1^{(i)}|\boldsymbol{z})$ in (10) cannot be directly used. To address this problem, we propose the combined proposal pdf

$$q(\boldsymbol{x}|\boldsymbol{z}) = p(a = 0|\boldsymbol{z}) f(\boldsymbol{x})$$
$$+ \sum_{m=1}^{M} p(a = m|\boldsymbol{z}) q_\mathrm{PFL}(\boldsymbol{x}|\boldsymbol{z}^{(m)}) \quad (16)$$

where we used the marginal association probabilities $p(a|z)$ to weight the proposal pdfs $q_{\text{PFL}}(x|z^{(m)})$ related to PFL based on measurements $z^{(m)}$, $m \in \{1, \ldots, M\}$ (cf. (10)) and the Gaussian prior pdf $f(x) = \mathcal{N}(x; x_0^*, P)$. In particular, recall that the proposal distribution $q_{\text{PFL}}(x|z^{(m)})$ related to the PFL $\mathbf{x}_0 \to z^{(m)} \to \mathbf{x}^{(m)}$, can be evaluated as

$$q_{\text{PFL}}(x^{(i,m)}|z^{(m)}) = \frac{\mathcal{N}(x_0^{(i,m)}; x_0^*, P)}{\theta^{(m)}} \quad (17)$$

where $\theta^{(m)}$ is the mapping factor.

A total of $(M+1)N_{\text{p}}$ particles representing the proposal pdf in (16) is obtained by drawing N_{p} particles for each of the $M+1$ components in (16) and calculating corresponding marginal association probabilities and weights. For the first component related to association event $\mathsf{a} = 0$, N_{p} particles $\{x^{(i,0)}\}_{i=1}^{N_{\text{p}}}$ are directly drawn from $f(x)$, the corresponding marginal association probability is obtained by using (13) in (15), i.e., $p(a=0|z) \propto 1 - p_{\text{d}}$, and the corresponding combined proposal weights are set to $\omega'^{(i,0)} = p(a=0|z)\mathcal{N}(x_0^{(i,0)}; x_0^*, P)$ (cf. (16)). For each other component related to association event $\mathsf{a} = m$, $m \in \{1, \ldots, M\}$, first N_{p} particles $\{x_0^{(i,m)}\}_{i=1}^{N_{\text{p}}}$ are drawn from $f(x)$. Next, the PFL $\mathbf{x}_0 \to z^{(m)} \to \mathbf{x}^{(m)}$ is applied to the particles $\{x_0^{(i,m)}\}_{i=1}^{N_{\text{p}}}$ and new particles $\{x^{(i,m)}\}_{i=1}^{N_{\text{p}}}$ are obtained for each $m \in \{1, \ldots, M\}$. An approximation of each marginal association probability $\tilde{p}(a=m|z)$, $m \in \{1, \ldots, M\}$ is finally calculated from these particles by using (13) in (15) and performing Monte Carlo integration [24] based on the proposal pdf $q_{\text{PFL}}(x^{(i,m)}|z^{(i,m)})$ in (17), i.e.,

$$\tilde{p}(a=m|z) \propto \sum_{i=1}^{N_{\text{p}}} \frac{p_{\text{d}}\theta^{(m)} f(z^{(m)}|x^{(i,m)})\mathcal{N}(x^{(i,m)}; x_0^*, P)}{N_{\text{p}}\mu_{\text{fp}} f_{\text{fp}}(z^{(m)})\mathcal{N}(x_0^{(i,m)}; x_0^*, P)}. \quad (18)$$

The corresponding combined proposal weights are set according to (cf. (16) and (17))

$$\omega'^{(i,m)} = \tilde{p}(a=m|z)\frac{\mathcal{N}(x_0^{(i,m)}; x_0^*, P)}{\theta^{(m)}}. \quad (19)$$

Finally, we reindex the resulting particles and weights $\{(\omega'^{(i,a)}, x^{(i,a)})\}_{i=1}^{N_{\text{p}}}$, $a \in \{0, \ldots, M\}$ to obtain $\{(\omega^{(l)}, x^{(l)})\}_{l=1}^{L}$ where $l = i(a+1)$ and $L = N_{\text{p}}(M+1)$.

Following the importance sampling principle, we next aim to compute particles $\{(w^{(l)}, x^{(l)})\}_{l=1}^{L}$ that represent the posterior pdf $f(x|z)$. In particular, by plugging (14) into (1) and by using $\{(\omega^{(l)}, x^{(l)})\}$ (cf. (19)) to represent $q(x|z)$ in (1), we obtain

$$w^{(l)} \propto \frac{\mathcal{N}(x^{(l)}; x_0^*, P)\left(1 - p_{\text{d}} + \sum_{m=1}^{M} \frac{p_{\text{d}} f(z^{(m)}|x^{(l)})f(x^{(l)})}{\mu_{\text{fp}} f_{\text{fp}}(z^{(m)})}\right)}{\omega^{(l)}}. \quad (20)$$

The resulting set of particles $\{(w^{(l)}, x^{(l)})\}_{l=1}^{L}$ is an asymptotically optimal representation of $f(x|z)$ for scenarios with MOU.

Algorithm 6: GMM Importance Sampling with Invertible Flow and DA

1 $\left[\{x_+^{*(h)}, P_+^{(h)}\}_{h=1}^{N_{\text{k}}}\right] = \text{InvertibleFlowGMMwithDA}\left(\{x^{*(h)}, P^{(h)}\}_{h=1}^{N_{\text{k}}}, z\right)$

2 $\left[\left\{\left\{\{x^{(i,h,a)}, \omega^{(i,h,a)}\}_{i=1}^{N_{\text{P}}}, \theta^{(h,a)}\right\}_{h=1}^{N_{\text{k}}}, \beta^{(a)}\right\}_{a=0}^{M}\right] =$
$\quad \text{Evaluation}\left(\{x^{*(h)}, P^{(h)}\}_{h=1}^{N_{\text{k}}}, 1, z\right)$

3 $\qquad\qquad\qquad\qquad\qquad\qquad$ // see **Alg. 7**

4 $\left[\{x_+^{*(h)}, P_+^{(h)}\}_{h=1}^{N_{\text{k}}}, \sim\right] =$
$\quad \text{Update}\left(\left\{\left\{\{x^{(i,h,a)}, \omega^{(i,h,a)}\}_{i=1}^{N_{\text{P}}}, \theta^{(h,a)}\right\}_{h=1}^{N_{\text{k}}}, \beta^{(a)}, 1\right\}_{a=0}^{M}\right)$

5 $\qquad\qquad\qquad\qquad\qquad\qquad$ // see **Alg. 8**

6 Output: $\{x_+^{*(h)}, P_+^{(h)}\}_{h=1}^{N_{\text{k}}}$

Algorithm 7: GMM Importance Sampling with Invertible Flow and DA – Measurement Evaluation

1 $\left[\left\{\left\{\{x^{(i,h,a)}, \omega^{(i,h,a)}\}_{i=1}^{N_{\text{P}}}, \theta^{(h,a)}\right\}_{h=1}^{N_{\text{k}}}, \beta^{(a)}\right\}_{a=0}^{M}\right] =$
$\quad \text{Evaluation}\left(\{x^{*(h)}, P^{(h)}\}_{h=1}^{N_{\text{k}}}, p, z\right)$

2 Initialize association variables as $\beta^{(0)} = (1 - p_{\text{d}})p + (1-p)$;

3 **for** $h = 1 : N_{\text{k}}$ **do**

4 $\quad$ **for** $i = 1 : N_{\text{P}}$ **do**

5 $\qquad$ Draw $x_0^{(i,h)} \sim \mathcal{N}(x^{(h)}; x^{*(h)}, P^{(h)})$;

6 $\quad$ Initialize particles as $\{x^{(i,h,0)}\}_{i=1}^{N_{\text{P}}} = \{x_0^{(i,h)}\}_{i=1}^{N_{\text{P}}}$;

7 $\quad$ Initialize proposal weights according to

8 $\quad$ $\omega^{(i,h,0)} = \mathcal{N}(x^{(i,h,0)}; x^{*(h)}, P^{(h)})$, $i = 1, \ldots, N_{\text{P}}$;

9 Initialized mapping factor $\theta^{(h,0)} = 1$;

10 **for** $m = 1 : M$ **do**

11 $\quad$ **for** $h = 1 : N_{\text{k}}$ **do**

12 $\qquad$ Perform PFL, i.e.,

13 $\qquad$ $\left[\{x^{(i,h,m)}\}_{i=1}^{N_{\text{P}}}, \{A_l^{(h,m)}\}_{l=1}^{N_{\lambda}}\right]$

14 $\qquad\quad = \text{ParticleFlow}\left(\{x_0^{(i,h)}\}_{i=1}^{N_{\text{P}}}, x^{*(h)}, P^{(h)}, z^{(m)}\right)$

15 $\qquad\qquad\qquad\qquad\qquad$ // see **Alg. 1**

16 $\qquad$ Compute mapping factor $\theta^{(h,m)}$ from $\{A_l^{(h,m)}\}_{l=1}^{N_{\lambda}}$ as in (11);

17 $\quad$ Precompute weights (cf. (19) and (20)), i.e.,

18 $\quad$ $\omega^{(i,h,m)} = \frac{\mathcal{N}(x^{(i,h,0)}; x^{*(h)}, P^{(h)})}{\mathcal{N}(x^{(i,h,m)}; x^{*(h)}, P^{(h)})\theta^{(h,m)}}$, $i = 1, \ldots, N_{\text{P}}$;

19 $\quad$ Compute approximate association variables following (18), i.e.,
$\qquad$ $\beta^{(m)} = \frac{p\, p_{\text{d}}}{N_{\text{P}} N_{\text{k}} \mu_{\text{fp}}} \sum_{h=1}^{N_{\text{k}}} \sum_{i=1}^{N_{\text{P}}} \frac{f(z^{(m)}|x^{(i,h,m)})}{f_{\text{fp}}(z^{(m)})\omega^{(i,h,m)}}$;

20 Output: $\left\{\left\{\{x^{(i,h,a)}, \omega^{(i,h,a)}\}_{i=1}^{N_{\text{P}}}, \theta^{(h,a)}\right\}_{h=1}^{N_{\text{k}}}, \beta^{(a)}\right\}_{a=0}^{M}$

D. GMM Importance Sampling With Invertible Flow for Problems With MOU

For problems where the prior distribution is non-Gaussian and potentially multimodal, GMM PFL with invertible flow discussed in Section III-A can be directly applied to problems with MOU. Pseudocode for GMM importance sampling with the invertible flow for MOU problems is provided in **Algorithm 6**. Algorithm 6 relies on the measurement evaluation presented in **Algorithm 7** and the measurement update presented in **Algorithm 8**. Note that in Algorithm 8, for future reference, we have also introduced extrinsic DA information denoted as $\kappa^{(a)}$, $a \in \{0, \ldots, M\}$. In the single object tracking considered

---

Algorithm 8: GMM Importance Sampling with Invertible Flow and DA – Measurement Update

1 $\left[\left\{\boldsymbol{x}_+^{*(h)}, \boldsymbol{P}_+^{(h)}\right\}_{h=1}^{N_\mathrm{k}}, p_+\right] =$

    $\mathrm{Update}\Big(\Big\{\Big\{\big\{\boldsymbol{x}^{(i,h,a)}, \omega^{(i,h,a)}\big\}_{i=1}^{N_\mathrm{P}}, \theta^{(h,a)}\Big\}_{h=1}^{N_\mathrm{k}}, \beta^{(a)}, \kappa^{(a)}\Big\}_{a=0}^{M}, p\Big)$

2 Compute association probabilities $\tilde{p}(a|\boldsymbol{z}) = \frac{\beta^{(a)}\kappa^{(a)}}{\sum_{a=0}^{M}\beta^{(a)}\kappa^{(a)}}, a = 0,\ldots,M$

3 **for** $h = 1 : N_\mathrm{k}$ **do**

4   **for** $a = 0 : M$ **do**

5   Update proposal weights (cf. (16)–(20)), i.e.,

6      $\omega'^{(i,h,a)} = \tilde{p}(a|\boldsymbol{z})\omega^{(i,h,a)}, i = 1,\ldots,N_\mathrm{P};$

7   Reindex particles and proposal weights using $l = i(a+1)$ to obtain $\big\{\big(\omega^{(l,h)}, \boldsymbol{x}^{(l,h)}\big)\big\}_{l=1}^{L}$ from $\big\{\big(\omega'^{(i,h,a)}, \boldsymbol{x}^{(i,h,a)}\big)\big\}_{i=1}^{N_\mathrm{P}},$ $a \in \{0,\ldots,M\};$

8   Compute final particle weights according to (cf. (20))

9   **for** $l = 1 : L$ **do**

10      $w^{(l,h)} = \frac{\left(\kappa^{(0)}(1-p_\mathrm{d}) + \sum_{m=1}^{M}\frac{\kappa^{(m)}p_\mathrm{d}f\big(\boldsymbol{z}^{(m)}\big|\boldsymbol{x}^{(l,h)}\big)}{\mu_\mathrm{fp}f_\mathrm{fp}\big(\boldsymbol{z}^{(m)}\big)}\right)}{\omega^{(l,h)}};$

11      $\left[\sim, \boldsymbol{P}^{(h)}\right] = \mathrm{GaussianRepresentation}\Big(\big\{\boldsymbol{x}^{(l,h)}, w^{(l,h)}\big\}_{l=1}^{L}\Big)$

12   // see **Alg. 3**

13 $w = \frac{p}{N_\mathrm{k}L}\sum_{h=1}^{N_\mathrm{k}}\sum_{l=1}^{L}w^{(l,h)}$

14 $p_+ = \frac{w}{(1-p)\tilde{p}(a=0|\boldsymbol{z})+w}$

15 $\left[\big\{\boldsymbol{x}_+^{*(h)}, \boldsymbol{P}_+^{(h)}\big\}_{h=1}^{N_\mathrm{k}}\right] = \mathrm{Resampling}\Big(\Big\{\boldsymbol{P}^{(h)}, \big\{\boldsymbol{x}^{(l,h)}, w^{(l,h)}\big\}_{l=1}^{L}\Big\}_{h=1}^{N_\mathrm{k}}\Big)$

16   // see **Alg. 5**

17 Output: $\big\{\boldsymbol{x}_+^{*(h)}, \boldsymbol{P}_+^{(h)}\big\}_{h=1}^{N_\mathrm{k}}, p_+$

---

here, we have $\kappa^{(a)} = 1$. Note that for later use in Section V-D, Algorithm 7 performs measurements evaluation by also taking a probability of existence, p, into account. Similarly, Algorithm 8 also updates p. By using $p = 1$ as input for Algorithm 7 (see Algorithm 6, line 2), Algorithm 6 is equivalent to the estimation method discussed in Sections III-B and III-C. Furthermore, note that for consistency with Section V-D, we introduced the notation $\beta^{(a)} \propto \tilde{p}(a|\boldsymbol{z})$, $a \in \{0,\ldots,M\}$ in Algorithm 6, Algorithm 7, and Algorithm 8.

IV. REVIEW OF GRAPH-BASED MULTISENSOR MOT

We will first discuss the concept of potential object (PO) states and then review the SPA messages that will later be calculated based on PFL. A summary of the system model and corresponding factor graph can be found in the supplementary material [57]. Graph-based MOT will be combined with a Gaussian mixture representation and PFL-based processing in Section V.

A. PO States

As in [9], [10], we consider MOT for an unknown, time-varying number of objects by introducing PO states. The number of POs J_{k-1} at discrete time $k - 1 \geqslant 0$ is the maximum possible number of objects that have generated a measurement up to time $k - 1$. At time k, a new PO is introduced for each of the M_k observed measurements, and the total number of POs is updated as $\mathsf{J}_k = \mathsf{J}_{k-1} + \mathsf{M}_k$. All POs that have been introduced

at previous time steps are referred to as legacy POs, i.e., at time k, there are J_{k-1} legacy POs and M_k new POs.

The augmented state of PO $j \in \{1,\ldots,J_k\}$ is given by $\mathbf{y}_k^{(j)} \triangleq \big[\mathbf{x}_k^{(j)\mathrm{T}} \mathbf{r}_k^{(j)}\big]^{\mathrm{T}}$, where the state $\mathbf{x}_k^{(j)}$ of PO j consists of the position and possibly further parameters of the object represented by the PO. Furthermore, the existence variable $\mathrm{r}_k^{(j)} \in \{0,1\}$ models the existence/nonexistence of PO j in the sense that PO j exists at time k if and only if $\mathrm{r}_k^{(j)} = 1$. For nonexistent POs, i.e., $\mathrm{r}_k^{(j)} = 0$, the state $\mathbf{x}_k^{(j)}$ is obviously irrelevant. Thus, all pdfs of augmented PO states $f\big(\mathbf{y}_k^{(j)}\big) = f\big(\mathbf{x}_k^{(j)}, \mathrm{r}_k^{(j)}\big)$ can be expressed as $f\big(\mathbf{x}_k^{(j)}, 0\big) = f_k^{(j)} f_\mathrm{D}\big(\mathbf{x}_k^{(j)}\big)$, where $f_\mathrm{D}\big(\mathbf{x}_k^{(j)}\big)$ is an arbitrary "dummy pdf" and $f_k^{(j)} \in [0,1]$ is a constant. To distinguish between legacy and new POs, we denote by $\underline{\mathbf{y}}_k^{(j)}$ and by $\overline{\mathbf{y}}_k^{(m)}$ the augmented state of a legacy PO and a new PO states, respectively.

The concept of legacy and new POs can be extended to scenarios with S sensors as follows. Let $\mathsf{M}_{k,s}$ be the number of measurements at time k and sensor $s \in (1,\ldots,S)$, where $(1,\ldots,S)$ is an arbitrary processing order of the sensors. The maximum possible number of objects that generated a measurement up to time k and sensor s is $\mathsf{J}_{k,s} = \mathsf{J}_{k,s-1} + \mathsf{M}_{k,s}$, with $\mathsf{J}_{k,0} \triangleq \mathsf{J}_{k-1}$.

B. Problem Formulation and Selected Messages of the SPA

At each time step $k \geqslant 1$, we consider the tracking of an unknown number of objects based on measurements $\boldsymbol{z}_{1:k}$. Object detection is performed by comparing the existence probability $p\big(r_k^{(j)} = 1\big|\boldsymbol{z}_{1:k}\big)$ with a threshold P_th, i.e., PO $j \in \{1,\ldots,J_k\}$ is declared to exist if $p\big(r_k^{(j)} = 1\big|\boldsymbol{z}_{1:k}\big) > P_\mathrm{th}$. Note that $p\big(r_k^{(j)} = 1\big|\boldsymbol{z}_{1:k}\big) = \int f\big(\boldsymbol{x}_k^{(j)}, r_k^{(j)} = 1\big|\boldsymbol{z}_{1:k}\big)\mathrm{d}\boldsymbol{x}_k^{(j)}$. For existent POs, state estimation is performed by calculating the minimum mean-square error (MMSE) estimate [58] as $\hat{\boldsymbol{x}}_k^{(j)} \triangleq \int \boldsymbol{x}_k^{(j)} f\big(\boldsymbol{x}_k^{(j)}\big|r_k^{(j)} = 1, \boldsymbol{z}_{1:k}\big)\mathrm{d}\boldsymbol{x}_k^{(j)}$, where $f\big(\boldsymbol{x}_k^{(j)}\big|r_k^{(j)} = 1, \boldsymbol{z}_{1:k}\big) = f\big(\boldsymbol{x}_k^{(j)}, r_k^{(j)} = 1\big|\boldsymbol{z}_{1:k}\big)/p\big(r_k^{(j)} = 1\big|\boldsymbol{z}_{1:k}\big)$.

Both object detection and estimation require the marginal posterior pdfs $f\big(\boldsymbol{x}_k^{(j)}, r_k^{(j)}\big|\boldsymbol{z}_{1:k}\big) \triangleq f\big(\boldsymbol{y}_k^{(j)}\big|\boldsymbol{z}_{1:k}\big)$, $j \in \{1,\ldots,J_k\}$. However, calculating $f\big(\boldsymbol{x}_k^{(j)}, r_k^{(j)}\big|\boldsymbol{z}_{1:k}\big)$ by direct marginalization is infeasible due to the large number parameters in the joint posterior distribution in [57, Eq. (1)].

As in [9], [10], we consider approximate calculation by performing the loopy SPA on the factor graph in Fig. 1 of [57] and passing messages only forward in time. This makes it possible to efficiently calculate so-called beliefs $\tilde{f}\big(\boldsymbol{x}_k^{(j)}, r_k^{(j)}\big) \triangleq \tilde{f}\big(\boldsymbol{y}_k^{(j)}\big), j \in \{1,\ldots,J_k\}$ which accurately approximate the marginal posterior pdfs $f\big(\boldsymbol{x}_k^{(j)}, r_k^{(j)}\big|\boldsymbol{z}_{1:k}\big), j \in \{1,\ldots,J_k\}$ needed for object detection and estimation. To keep computational complexity feasible, at the end of each time k with all sensors processed, a suboptimal pruning step has to be performed. Here, POs with probability of existence $p_k^{(j)} \triangleq \tilde{p}(r_k^{(j)} = 1|\boldsymbol{z}_{1:k})$ below a threshold P_pr are removed from the state space.

Next, we review the SPA messages that will later be calculated based on PFL. We will limit our discussion to messages and beliefs related to legacy PO states. Messages and beliefs

related to new PO states are obtained by performing similar steps. (A complete description of message passing for MOT is provided in [10, Section IX-A].)

We consider sequential sensor processing, where the SPA incorporates sensor measurements sequentially and in an arbitrary processing order at each time step. The part of the factor graph that represents the processing of the measurements of one sensor at one time step is shown in Fig. 1 of [57]. At the processing step related to time k and sensor s, the "prior messages" of legacy PO states are denoted as $\alpha_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}) = \alpha_{k,s}^{(j)}(\underline{\boldsymbol{y}}_{k,s}^{(j)})$. At time k and sensor $s = 1$, these message are computed by a prediction step [10, Section IX-A1], i.e.,

$$
\alpha_{k,1}^{(j)}(\underline{\boldsymbol{x}}_{k,1}^{(j)}, \underline{r}_{k,1}^{(j)}) = \sum_{r_{k-1}^{(j)} \in \{0,1\}} \int f(\underline{\boldsymbol{x}}_{k,1}^{(j)}, \underline{r}_{k,1}^{(j)} | \boldsymbol{x}_{k-1}^{(j)}, r_{k-1}^{(j)})
$$
$$
\times \tilde{f}(\boldsymbol{x}_{k-1}^{(j)}, r_{k-1}^{(j)}) \mathrm{d}\boldsymbol{x}_{k-1}^{(j)} \quad (21)
$$

that makes use of the state-transition function $f(\underline{\boldsymbol{x}}_{k,1}^{(j)}, \underline{r}_{k,1}^{(j)} | \boldsymbol{x}_{k-1}^{(j)}, r_{k-1}^{(j)}) \triangleq f(\underline{\boldsymbol{y}}_{k,1}^{(j)} | \boldsymbol{y}_{k-1}^{(j)})$. At the time k and sensor $s > 1$, the "prior message" is the same as the belief after the previous sensor update, i.e., $\alpha_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}) \triangleq \tilde{f}(\boldsymbol{x}_{k,s-1}^{(j)}, r_{k,s-1}^{(j)})$. For future reference, we also introduce $\underline{p}_{k,s}^{(j)} = \int \alpha_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)} = 1) \mathrm{d}\underline{\boldsymbol{x}}_{k,s}^{(j)}$ as the predicted probability of existence for each legacy PO. Note that $\underline{p}_{k,s}^{(j)} + \int \alpha_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)} = 0) \mathrm{d}\underline{\boldsymbol{x}}_{k,s}^{(j)} = 1$.

After the prior messages have been computed, a "measurement evaluation" step for each legacy and each new PO is performed. Here, we denote by $a_{k,s}^{(j)}$ the association variable related to PO j at the update step related to sensor s at time k and by $b_{k,s}^{(m)}$ the association variable related to measurement m at the update step related to sensor s at time k (see [57, Sec. 1.2] for details). The SPA messages that are passed from the factor nodes $q(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s})$ in [57, Eq. (2)] and $v(\overline{\boldsymbol{x}}_{k,s}^{(m)}, \overline{r}_{k,s}^{(m)}, b_{k,s}^{(m)}; \boldsymbol{z}_{k,s}^{(m)})$ in [57, Eq. (3)] to the adjacent variables nodes $a_{k,s}^{(j)}$ and $b_{k,s}^{(m)}$, respectively, are computed. For legacy POs, these messages are given by (see [10, Section IX])

$$
\beta_{k,s}^{(j)}(a_{k,s}^{(j)}) = \int q(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s}) \alpha_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1) \mathrm{d}\underline{\boldsymbol{x}}_{k,s}^{(j)}
$$
$$
+ 1(a_{k,s}^{(j)})(1 - \underline{p}_{k,s}^{(j)}). \quad (22)
$$

For new POs, the corresponding messages are denoted as $\xi_{k,s}^{(m)}(b_{k,s}^{(m)})$, $m \in \{1, \ldots, M_{k,s}\}$ and are calculated similarly (see [10, Section IX]).

Next, probabilistic DA is performed by means of iterative SPA message passing with input messages $\beta_{k,s}^{(j)}(a_{k,s}^{(j)})$, $j \in \{1, \ldots, J_{k,s-1}\}$ and $\xi_{k,s}^{(m)}(b_{k,s}^{(m)})$, $m \in \{1, \ldots, M_{k,s}\}$ (see [10, Section IX-A3] for details). After convergence, corresponding output messages $\kappa_{k,s}^{(j)}(a_{k,s}^{(j)})$, $j \in \{1, \ldots, J_{k,s-1}\}$ and $\iota_{k,s}^{(m)}(b_{k,s}^{(m)})$, $m \in \{1, \ldots, M_{k,s}\}$ are available for legacy POs and new POs, respectively. Probabilistic DA is followed by a "measurement update" step. Here, for legacy POs, messages

$\gamma_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)})$ passed from $q(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s})$ to $\underline{\boldsymbol{y}}_{k,s}^{(j)}$ are calculated as

$$
\gamma_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1) = \sum_{a_{k,s}^{(j)}=0}^{M_{k,s}} q(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s}) \kappa_{k,s}^{(j)}(a_{k,s}^{(j)}) \quad (23)
$$

and as $\gamma_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 0) = \gamma_{k,s}^{(j)} = \kappa_{k,s}^{(j)}(0)$. Measurement update for new POs is performed by following similar steps [10, Section IX-A].

Finally, beliefs are calculated to approximate the posterior pdfs of POs. For legacy POs, beliefs $\tilde{f}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)})$ approximating $f(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)} | \boldsymbol{z}_{1:k})$ are obtained as

$$
\tilde{f}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1) = \frac{1}{\underline{C}_{k,s}^{(j)}} \alpha_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1) \gamma_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1) \quad (24)
$$

and as $\tilde{f}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 0) = \underline{f}_{k,s}^{(j)} f_{\mathrm{D}}(\underline{\boldsymbol{x}}_{k,s}^{(j)})$ with $\underline{f}_{k,s}^{(j)} = (1 - \underline{p}_{k,s}^{(j)}) \gamma_{k,s}^{(j)}/\underline{C}_{k,s}^{(j)}$. The constant $\underline{C}_{k,s}^{(j)}$ is given by $\underline{C}_{k,s}^{(j)} \triangleq \int \alpha_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1) \gamma_{k,s}^{(j)}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1) \mathrm{d}\underline{\boldsymbol{x}}_{k,s}^{(j)} + (1 - \underline{p}_{k,s}^{(j)}) \gamma_{k,s}^{(j)}$.

Calculating the beliefs $\tilde{f}(\overline{\boldsymbol{x}}_{k,s}^{(m)}, \overline{r}_{k,s}^{(m)})$, $m \in \{1, \ldots, M_{k,s}\}$ for new POs is performed by following similar steps [10, Section IX-A]. Note that this calculation of new POs involves the messages $\iota_{k,s}^{(m)}(b_{k,s}^{(m)})$ and $\varsigma_{k,s}^{(m)}(\overline{\boldsymbol{y}}_{k,s}^{(m)})$, $m \in \{1, \ldots, M_{k,s}\}$ also shown in Fig. 1 of [57]. The resulting beliefs for legacy and new POs are used as the prior messages for measurement update of sensor $s + 1$ as discussed above, i.e., $\alpha_{k,s+1}^{(j)}(\underline{\boldsymbol{x}}_{k,s+1}^{(j)}, \underline{r}_{k,s+1}^{(j)}) \triangleq \tilde{f}(\boldsymbol{x}_{k,s}^{(j)}, r_{k,s}^{(j)})$, $j \in \{1, \ldots, J_{k,s}\}$. When the measurements of the last sensor in the sequence have been processed, i.e., $s = S$, the resulting beliefs are used in the prediction steps (21) of the next time step $k + 1$.

V. GRAPH-BASED MULTISENSOR MOT WITH INVERTIBLE PFL

In nonlinear MOT scenarios, calculation of $\beta_{k,s}^{(j)}(a_{k,s}^{(j)})$ in (22) and $\tilde{f}(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)})$ in (24) related to legacy PO states as well as their counterparts $\xi_{k,s}^{(m)}(b_{k,s}^{(m)})$ and $\tilde{f}(\overline{\boldsymbol{x}}_{k,s}^{(m)}, \overline{r}_{k,s}^{(m)})$, related to new PO states cannot be performed in closed form. We propose a particle-based implementation where a proposal pdf is established using invertible PFL as introduced in Section II-B. This makes it possible to implement multisensor MOT with high dimensional states and nonlinear measurement models. A single time step of the proposed particle-based implementation is discussed next. At first, we assume a single Gaussian kernel as the prior knowledge for each PO. An extension to GMM is also presented. In what follows, we consider a single time step, remove the time index k, and use the index $-$ short for $k - 1$.

A. Prediction

It is assumed that the beliefs of legacy POs at time $k - 1$ are represented by a single Gaussian distribution, i.e., $\tilde{f}(\boldsymbol{x}_-^{(j)}, r_-^{(j)} = 1) = p_-^{(j)} \mathcal{N}(\boldsymbol{x}_-^{(j)}; \boldsymbol{x}_-^{*(j)}, \boldsymbol{P}_-^{(j)})$, $j \in \{1, \ldots, J_-\}$. In MOT problems, the state transition function underlying the

state transition model $f\big(\underline{x}_1^{(j)}\big|x_-^{(j)}\big)$ discussed in [10, Section VIII-C] is typically linear with additive Gaussian noise, i.e., $\mathbf{x}_1^{(j)} = \boldsymbol{G}\mathbf{x}_-^{(j)} + \mathbf{u}^{(j)}$ where $\boldsymbol{G}$ is the state transition matrix and $\mathbf{u}^{(j)}$ is an additive Gaussian noise vector with mean $\boldsymbol{u}^*$ and covariance matrix $\boldsymbol{P}_{\mathbf{u}}$. Consequently, the messages computed in the prediction step are also represented by a Gaussian distribution, i.e., $\alpha_1^{(j)}\big(\underline{x}_1^{(j)}, \underline{r}_1^{(j)}\big) = \underline{p}_1^{(j)}\mathcal{N}\big(\underline{x}_1^{(j)}; \underline{x}_1^{*(j)}, \underline{P}_1^{(j)}\big)$, $j \in \{1, \ldots, J_-\}$ with mean, covariance matrix, and existence probability given by $\underline{x}_1^{*(j)} = \boldsymbol{G}x_-^{*(j)} + \boldsymbol{u}^*$, $\underline{P}_1^{(j)} = \boldsymbol{G}\underline{P}_-^{(j)}\boldsymbol{G}^{\mathrm{T}} + \boldsymbol{P}_{\mathbf{u}}$, and $\underline{p}_1^{(j)} = p_{\mathrm{su}}\underline{p}_-^{(j)}$, respectively. Here, p_{su} is the survival probability, i.e., the probability that an object that exists at time step $k-1$, still exists at time step k. Here, if the state transition function is not linear with additive Gaussian noise, for each $j \in \{1, \ldots, J_-\}$, N_{P} particles are drawn from $\mathcal{N}\big(x_-^{(j)}; x_-^{*(j)}, \boldsymbol{P}_-^{(j)}\big)$, the prediction step of a conventional particle filter is performed [25], and a predicted Gaussian representation $\mathcal{N}\big(\underline{x}_1^{(j)}; \underline{x}_1^{*(j)}, \underline{P}_1^{(j)}\big)$ is computed from the resulting particles using Algorithm 3.

B. Measurement Evaluation

The following steps are performed sequentially for each sensor $s = 1, \ldots, S$. First, a particle representation $\big\{\big(\underline{x}_{0,s}^{(i,j)}, \omega_{0,s}^{(i,j)}\big)\big\}_{i=1}^{N_{\mathrm{P}}}$ is obtained by drawing particles $\underline{x}_{0,s}^{(i,j)}$, $i \in \{1, \ldots, N_{\mathrm{P}}\}$ from $\mathcal{N}\big(\underline{x}_s^{(j)}; \underline{x}_s^{(j)*}, \underline{P}_s^{(j)}\big)$ and setting the corresponding weights to $\omega_{0,s}^{(i,j)} = \underline{p}_s^{(j)}/N_{\mathrm{P}}$. Next, we compute an extended set $\big\{\big\{\big(\underline{x}_s^{(i,a,j)}, \underline{\omega}_s^{(i,a,j)}\big)\big\}_{i=1}^{N_{\mathrm{P}}}\big\}_{a=0}^{M_s}$, that consists of N_{P} particles and weights for each value of $a_s^{(j)} \in \{0, \ldots, M_s\}$ and $j \in \{1, \ldots, J_{s-1}\}$. (Note that measurement gating [3] can be employed to reduce the number of measurements used for PFL.) For $a_s^{(j)} = 0$, we perform no flow, i.e., we set $\big\{\big(\underline{x}_s^{(i,0,j)}, \underline{\omega}_s^{(i,0,j)}\big)\big\}_{i=1}^{N_{\mathrm{P}}} = \big\{\big(\underline{x}_{0,s}^{(i,j)}, \underline{\omega}_{0,s}^{(i,j)}\big)\big\}_{i=1}^{N_{\mathrm{P}}}$. For $a_s^{(j)} = m \in \{1, \ldots, M_s\}$, the PFL $\underline{\mathbf{x}}_{0,s}^{(j)} \to z_s^{(m)} \to \underline{\mathbf{x}}_s^{(j,m)}$ is applied to obtain new particles $\big\{\underline{x}_s^{(i,m,j)}\big\}_{i=1}^{N_{\mathrm{P}}}$ by migrating the particles $\big\{\underline{x}_{0,s}^{(i,m,j)}\big\}_{i=1}^{N_{\mathrm{P}}}$. By making use of the invertible PFL principle (cf. (10)) [42], the weights $\underline{\omega}_s^{(i,m,j)}$ corresponding to the migrated particles $\underline{x}_s^{(i,m,j)}$ are obtained as

$$\underline{\omega}_s^{(i,m,j)} = \frac{\mathcal{N}\big(\underline{x}_s^{(i,m,j)}; \underline{x}_s^{(j)*}, \underline{P}_s^{(j)}\big)\theta_{m,s}^{(j)}}{\mathcal{N}\big(\underline{x}_{0,s}^{(i,j)}; \underline{x}_s^{(j)*}, \underline{P}_s^{(j)}\big)}\underline{\omega}_{0,s}^{(i,j)}, i \in \{1, \ldots, N_{\mathrm{P}}\}$$

with mapping factor $\theta_{m,s}^{(j)}$ (cf. (11)). Note that the sets of weighted particles $\big\{\underline{x}_s^{(i,m,j)}, \underline{\omega}_s^{(i,m,j)}\big\}_{i=1}^{N_{\mathrm{P}}}$, $m \in \{1, \ldots, M_s\}$, despite all being based on a different proposal pdf $q_{\mathrm{PFL}}\big(\underline{x}_s^{(j)}|z_s^{(m)}\big)$, $m \in \{1, \ldots, M_s\}$, still represent $\alpha_s^{(j)}\big(\underline{x}_s^{(j)}, 1\big)$. The result of this particle migration along the flow defined by a measurement $z_s^{(m)}$, $m \in \{1, \ldots, M_s\}$, is that the particles are now at locations where the evaluation of the corresponding likelihood function $f\big(z_s^{(m)}\big|\underline{x}_s^{(j)}\big)$ will produce a significant particle weight. The migrated particles can thus approximate message-passing operations accurately even if the dimension of the state is high [9].

The measurement evaluation step can now be performed on weighted particles $\big\{\big\{\big(\underline{x}_s^{(i,a,j)}, \underline{\omega}_s^{(i,a,j)}\big)\big\}_{i=1}^{N_{\mathrm{P}}}\big\}_{a=0}^{M_s}$ by calculating an approximation $\tilde{\beta}_s^{(j)}(a)$ of the messages $\beta_s^{(j)}(a)$ in (22) for all $j \in \{1, \ldots, J_{s-1}\}$, $a \in \{0, \ldots, M_s\}$ as

$$\tilde{\beta}_s^{(j)}\big(a_s^{(j)} = a\big) = \sum_{i=1}^{N_{\mathrm{P}}} q\big(\underline{x}_s^{(i,a,j)}, 1, a; z_s\big)\underline{\omega}_s^{(i,a,j)} + 1(a)\big(1 - \tilde{\underline{p}}_s^{(j)}\big).$$

For the computation of a particle representation of new PO states, we first draw particles $\overline{x}_{0,s}^{(i)}$, $i \in \{1, \ldots, N_{\mathrm{P}}\}$ from $f_{\mathrm{b}}\big(\overline{x}\big)$ as introduced in [Sec. 1.1] [57]. Next, for each new PO $j = J_{s-1} + m$, $m \in \{1, \ldots, M_s\}$, new particles and corresponding weights $\big\{\overline{x}_s^{(m,i)}, \overline{w}_s^{(m,i)}\big\}_{i=1}^{N_{\mathrm{P}}}$ are obtained from $\big\{\overline{x}_{0,s}^{(i)}\big\}_{i=1}^{N_{\mathrm{P}}}$ by performing the invertible PFL $\overline{\mathbf{x}}_{0,s} \to z_s^{(m)} \to \overline{\mathbf{x}}_s^{(m)}$. Note that this flow relies on the mean $\overline{x}_{\mathrm{b}}$ and covariance matrix $\overline{P}_{\mathrm{b}}$ of $f_{\mathrm{b}}\big(\overline{x}\big)$. Finally, for each $m \in \{1, \ldots, M_s\}$ approximate messages $\tilde{\xi}_s^{(m)}\big(b_s^{(m)}\big)$ are calculated from $\big\{\overline{x}_s^{(m,i)}, \overline{w}_s^{(m,i)}\big\}_{i=1}^{N_{\mathrm{P}}}$ by performing the same steps as described above for the calculation of $\tilde{\beta}_s^{(j)}\big(a_s^{(j)}\big)$. These messages are used as an input for the iterative SPA for DA [9], [10], [59] performed next (see [10, Sec. VI] for details).

C. Measurement Update and Belief Calculation

After the iterative loopy SPA for DA has been converged, the messages $\tilde{\kappa}_s^{(j)}\big(a_s^{(j)}\big)$, $j \in \{1, \ldots, J_{s-1}\}$ and $\tilde{\iota}_s^{(m)}\big(b_s^{(m)}\big)$, $m \in \{1, \ldots, M_s\}$ are available. These messages are used to obtain an approximation $\tilde{\gamma}_s^{(j)}\big(\underline{x}_s^{(j)}, 1\big)$ of the messages $\gamma_s^{(j)}\big(\underline{x}_s^{(j)}, 1\big)$, $j \in \{1, \ldots, J_{s-1}\}$ in (23) as well as an approximation $\tilde{\varsigma}_s^{(m)}\big(\overline{x}_s^{(m)}, 1\big)$ of the messages $\varsigma_s^{(m)}\big(\overline{x}_s^{(m)}, 1\big)$, $m \in \{1, \ldots, M_s\}$ in [10, Section IX].

Beliefs approximating the posterior pdf of POs are now computed by means of importance sampling. As in Section III-C, we use the marginal association probabilities $p(a_s^{(j)}|z_s)$ to weight the proposal pdfs $q_{\mathrm{PFL}}\big(\underline{x}_s^{(j)}|z_s^{(m)}\big)$ related to PFL based on measurements $z_s^{(m)}$, $m \in \{1, \ldots, M_s\}$. Note that in MOT scenarios, accurate approximations of $p(a_s^{(j)}|z_s)$ can be obtained as $\tilde{p}(a_s^{(j)}|z_s) \propto \tilde{\beta}_s^{(j)}\big(a_s^{(j)}\big)\tilde{\kappa}_s^{(j)}\big(a_s^{(j)}\big)$ (see [10, Section IX] for details). Consequently, we obtain a new set of reindex particles and weights $\big\{\big(\underline{x}_s^{(l,j)}, \underline{\omega}_s^{(l,j)}, \big)\big\}_{l=1}^{L_s}$ from $\big\{\big(\underline{x}_s^{(i,a,j)}, \underline{\omega}_s^{(i,a,j)}\big)\big\}_{i=1}^{N_{\mathrm{P}}}$, $a \in \{0, \ldots, M_s\}$ by using $l = i(a+1)$, $\underline{\omega}_s^{(l,j)} = \underline{\omega}_s^{(i,a,j)}/\tilde{p}(a_s^{(j)} = a|z_s)$, and $L_s = N_{\mathrm{P}}(M_s + 1)$.

Next, based on (24), we update the particle weights of the legacy POs $j \in \{1, \ldots, J_{s-1}\}$ by first computing

$$\underline{w}_s'^{(l,j)} = \tilde{\gamma}_s^{(j)}\big(\underline{x}_s^{(l,j)}, 1\big)\underline{\omega}_s^{(l,j)}, \quad l \in \{1, \ldots, L_s\}$$

and then calculating normalized weights as

$$\underline{w}_s^{(l,j)} = \frac{\underline{w}_s'^{(l,j)}}{\sum_{l=1}^{L_s} \underline{w}_s'^{(l,j)} + \big(1 - \tilde{\underline{p}}_s^{(j)}\big)\tilde{\gamma}_s^{(j)}}, \quad l \in \{1, \ldots, L_s\}. \tag{25}$$

Note that denominator of (25) is a particle-based approximation of $\underline{C}_s^{(j)}$ in (24). The resulting particles and weights

---

Algorithm 9: Gaussian Mixture Implementation of Multisensor MOT – Single Sensor Update Step

1. $\left[\left\{\left\{\boldsymbol{x}_+^{*(h,j)}, \boldsymbol{P}_+^{(h,j)}\right\}_{h=1}^{N_k}, p_+^{(j)}\right\}_{j=1}^{J_s}\right] =$
   SingleSensorUpdate$\left(\left\{\left\{\underline{\boldsymbol{x}}^{*(h,j)}, \underline{\boldsymbol{P}}^{(h,j)}\right\}_{h=1}^{N_k}, \underline{p}^{(j)}\right\}_{j=1}^{J_{s-1}}, \boldsymbol{z}_s\right)$

2. $\left[\left\{\left\{\overline{\boldsymbol{x}}_+^{*(h,m)}, \overline{\boldsymbol{P}}_+^{(h,m)}\right\}_{h=1}^{N_k}, \xi^{(m)}\right\}_{m=1}^{M_s}\right] = \text{NewObjects}(\boldsymbol{z}_s)$

3.   // see **Alg. 10**

4. **for** $j = 1 : J_{s-1}$ **do**

5. $\left[\left\{\left\{\left\{\underline{\boldsymbol{x}}^{(i,h,a,j)}, \underline{w}^{(i,h,a,j)}\right\}_{i=1}^{N_P}, \theta^{(h,a,j)}\right\}_{h=1}^{N_k}, \beta^{(a,j)}\right\}_{a=0}^{M_s}\right] =$
     Evaluation$\left(\left\{\underline{\boldsymbol{x}}^{*(h,j)}, \underline{\boldsymbol{P}}^{(h,j)}\right\}_{h=1}^{N_k}, \underline{p}^{(j)}, \boldsymbol{z}_s\right)$

6.   // see **Alg. 7**

7. $\left[\left\{\left\{\kappa^{(a,j)}\right\}_{a=0}^{M_s}\right\}_{j=1}^{J_{s-1}}, \left\{\iota^{(a)}\right\}_{a=0}^{M_s}\right] =$

8. DataAssociation$\left(\left\{\left\{\beta^{(a,j)}\right\}_{a=0}^{M_s}\right\}_{j=1}^{J_{s-1}}, \left\{\xi^{(a)}\right\}_{a=0}^{M_s}\right)$

9.   // see [10, Sec. VI]

10. Compute $\overline{p}_+^{(m)} = \frac{(\xi^{(m)}-1)\iota^{(m)}}{(\xi^{(m)}-1)\iota^{(m)}+1}$, $m \in \{1,\ldots,M_s\}$

11. **for** $j = 1 : J_{s-1}$ **do**

12. $\left[\left\{\underline{\boldsymbol{x}}_+^{*(h,j)}, \boldsymbol{P}_+^{(h,j)}\right\}_{h=1}^{N_k}, \underline{p}_+^{(j)}\right] = \text{Update}\Big(\left\{\left\{\left\{\underline{\boldsymbol{x}}^{(i,h,a,j)},\right.\right.\right.$

13. $\left.\left.\left.\underline{w}^{(i,h,a,j)}\right\}_{i=1}^{N_P}, \theta^{(h,a,j)}\right\}_{h=1}^{N_k}, \beta^{(a,j)}, \kappa^{(a,j)}\right\}_{a=0}^{M}, \underline{p}^{(j)}\Big)$

14.   // see **Alg. 8**

15. For $j=1,\ldots,J_{s-1}$, reindex legacy object state information according to $\left\{\boldsymbol{x}_+^{*(h,j)}, \boldsymbol{P}_+^{(h,j)}\right\}_{h=1}^{N_k} = \left\{\underline{\boldsymbol{x}}_+^{*(h,j)}, \underline{\boldsymbol{P}}_+^{(h,j)}\right\}_{h=1}^{N_k}$ and $p_+^{(j)} = \underline{p}_+^{(j)}$.

16. For $j=J_{s-1}+1,\ldots,J_s$, reindex new object state information following $\left\{\boldsymbol{x}_+^{*(h,j)}, \boldsymbol{P}_+^{(h,j)}\right\}_{h=1}^{N_k} = \left\{\overline{\boldsymbol{x}}_+^{*(h,m)}, \overline{\boldsymbol{P}}_+^{(h,m)}\right\}_{h=1}^{N_k}$ and $p_+^{(j)} = \overline{p}_+^{(m)}$ using $m = j - J_{s-1}$ and $J_s = J_{s-1} + M_s$.

17. Output: $\left\{\left\{\boldsymbol{x}_+^{*(h,j)}, \boldsymbol{P}_+^{(h,j)}\right\}_{h=1}^{N_k}, p_+^{(j)}\right\}_{j=1}^{J_s}$

Algorithm 10: Gaussian Mixture Implementation of Multisensor MOT – Generation of New POs

1. $\left[\left\{\left\{\overline{\boldsymbol{x}}_+^{*(h,m)}, \overline{\boldsymbol{P}}_+^{(h,m)}\right\}_{h=1}^{N_k}, \xi^{(m)}\right\}_{m=1}^{M}\right] = \text{NewObjects}(\boldsymbol{z}_s)$

2. Draw $\overline{\boldsymbol{x}}_b^{*(h)}, h=1,\ldots,N_k$ from $f_b(\overline{\boldsymbol{x}})$;

3. Compute covariance matrix $\overline{\boldsymbol{P}}_b$ from $f_b(\overline{\boldsymbol{x}})$;

4. **for** $m = 1 : M_s$ **do**

5. **for** $h = 1 : N_k$ **do**

6. $\left[\left\{\overline{\boldsymbol{x}}^{(i,h,m)}, \overline{w}^{(i,h,m)}\right\}_{i=1}^{N_P}\right] =$

7.   InvertibleFlow$\left(\overline{\boldsymbol{x}}_b^{*(h)}, \overline{\boldsymbol{P}}_b, \boldsymbol{z}_s^{(m)}\right)$;

8.   // see **Alg. 4**

9. $\left[\sim, \overline{\boldsymbol{P}}^{(h,m)}\right] =$
      GaussianRepresentation$\left(\left\{\overline{\boldsymbol{x}}^{(i,h,m)}, \overline{w}^{(i,h,m)}\right\}_{i=1}^{N_P}\right)$;

10.   // see **Alg. 3**

11. $\xi^{(m)} = 1 + \frac{\mu_b}{N_k N_P \mu_{fp} f_{fp}(\boldsymbol{z}_s^{(m)})} \sum_{h=1}^{N_k} \sum_{i=1}^{N_P} \overline{w}^{(i,h,m)}$;

12. $\left[\left\{\overline{\boldsymbol{x}}_+^{*(h,m)}, \overline{\boldsymbol{P}}_+^{(h,m)}\right\}_{h=1}^{N_k}\right] =$

13.   Resampling$\left(\left\{\overline{\boldsymbol{P}}^{(h,m)}, \left\{\overline{\boldsymbol{x}}^{(i,h,m)}, \overline{w}^{(i,h,m)}\right\}_{i=1}^{N_P}\right\}_{h=1}^{N_k}\right)$;

14.   // see **Alg. 5**

15. Output: $\left\{\left\{\overline{\boldsymbol{x}}_+^{*(h,m)}, \overline{\boldsymbol{P}}_+^{(h,m)}\right\}_{h=1}^{N_k}, \xi^{(m)}\right\}_{m=1}^{M}$

$\left\{\left(\underline{\boldsymbol{x}}_s^{(l,j)}, \underline{w}_s^{(l,j)}\right)\right\}_{l=1}^{L_s}$ represent the belief $\tilde{f}(\underline{\boldsymbol{x}}_s^{(j)}, 1)$ of legacy PO $j \in \{1,\ldots,J_{s-1}\}$. These particles can be used to calculate an approximation of the existence probability as $\tilde{p}_s^{(j)} = \sum_{l=1}^{L_s} \underline{w}_s^{(l,j)}$. A Gaussian representation of the belief $\tilde{f}(\boldsymbol{x}_s^{(j)}, r_s^{(j)}) \triangleq \tilde{f}(\underline{\boldsymbol{x}}_s^{(j)}, \underline{r}_s^{(j)})$ is furthermore obtained by applying Algorithm 3, which calculates a mean $\boldsymbol{x}_s^{(j)*}$ and a covariance matrix $\boldsymbol{P}_s^{(j)}$ from $\left\{\left(\underline{\boldsymbol{x}}_s^{(l,j)}, \underline{w}_s^{(l,j)}/\tilde{p}_s^{(j)}\right)\right\}_{l=1}^{L_s}$.

For new POs $j \in \{J_{s-1}+1,\ldots,J_s\}$, approximate existence probabilities $\tilde{p}_s^{(j)}$ and a Gaussian representation of the beliefs $\tilde{f}(\boldsymbol{x}_s^{(j)}, r_s^{(j)}) \triangleq \tilde{f}(\overline{\boldsymbol{x}}_s^{(m)}, \overline{r}_s^{(m)})$, for $m = j - J_{s-1}$ are calculated by performing similar steps as discussed above for legacy POs. Existence probabilities and Gaussian representations of the beliefs related to PO that have not been pruned are then used as input for processing measurements of the next sensor $s + 1$ or, in case $s = S$, for processing at the next time step.

For POs that have been declared to exist after the last sensor update, i.e., $\tilde{p}_S^{(j)} > P_{th}$, an approximate MMSE state estimate is directly given by the mean of the Gaussian representation, i.e., $\hat{\boldsymbol{x}}^{(j)} \approx \boldsymbol{x}_S^{*(j)}$.

D. The Proposed Multisensor MOT Method

In multisensor MOT problems with nonlinear measurement models, object beliefs are non-Gaussian and potentially multimodal. Here, graph-based multisensor MOT with invertible PFL has to be combined with a GMM discussed in Section III-A. Pseudocode for a single time step of the resulting multisensor MOT method is provided in **Algorithm 11**. Algorithm 11 relies on the single sensor update step provided in **Algorithm 9** which, in turn, relies on the introduction of new POs presented in **Algorithm 10**, measurement evaluation presented in Algorithm 7 and measurement update presented in Algorithm 8. Note that at time $k = 0$, Algorithm 11 is typically initialized by setting $J_- = 0$. However, if prior information is available, it can be incorporated in the form of the set $\left\{\left\{\boldsymbol{x}_-^{*(h,j)}, \boldsymbol{P}_-^{(h,j)}\right\}_{h=1}^{N_k}, p_-^{(j)}\right\}_{j=1}^{J_-}$. Note that in a GMM, an approximate MMSE estimate can be obtained as $\hat{\boldsymbol{x}}^{(j)} \approx \frac{1}{N_k} \sum_{h=1}^{N_k} \boldsymbol{x}_S^{*(h,j)}$.

Note that for the execution of the proposed method, for each time step and each object, we need to update each particle per kernel, pseudo-time, measurement, and sensor. The asymptotic complexity with respect to these parameters, per object and time step, thus reads $\mathcal{O}\left(N_\lambda N_k N_P\left(\sum_{s=1}^{S} M_s\right)\right)$. The complexity of a conventional bootstrap implementation per object and time step is $\mathcal{O}\left(N_b\left(\sum_{s=1}^{S} M_s\right)\right)$, where N_b is the number of particles. The improved runtime-complexity tradeoff of the proposed method is due to the fact that, in challenging problems as the one considered in Section VI, for $N_\lambda N_k N_P = N_b$, the proposed method strongly outperforms graph-based MOT that relies on conventional particle filtering. Since memory requirements per time step and object are $N_k N_P$ for the proposed method, and N_b for bootstrap particle filtering, and since $N_\lambda \gg 1$, for $N_\lambda N_k N_P = N_b$, the memory requirements of the proposed method are significantly lower compared to conventional particle filtering.

Algorithm 11: Single Time Step of Multisensor MOT with GMM and Invertible Flow

---

1   $\left[\left\{\left\{\boldsymbol{x}^{*(h,j)}, \boldsymbol{P}^{(h,j)}\right\}_{h=1}^{N_{\mathrm{k}}}, p^{(j)}\right\}_{j=1}^{J}\right] =$
   $\mathrm{MultisensorMOT}\left[\left\{\left\{\boldsymbol{x}_{-}^{*(h,j)}, \boldsymbol{P}_{-}^{(h,j)}\right\}_{h=1}^{N_{\mathrm{k}}}, p_{-}^{(j)}\right\}_{j=1}^{J_{-}}, \boldsymbol{z}\right]$

2   Perform prediction step according to

3   **for** $j = 1 : J_{-}$ **do**

4   **for** $h = 1 : N_{\mathrm{k}}$ **do**

5         $\underline{\boldsymbol{x}}_1^{*(h,j)} = \boldsymbol{G}\boldsymbol{x}_{-}^{*(h,j)} + \boldsymbol{u}^{*}$

6         $\underline{\boldsymbol{P}}_1^{(h,j)} = \boldsymbol{G}\boldsymbol{P}_{-}^{(h,j)}\boldsymbol{G}^{\mathrm{T}} + \boldsymbol{P}_{\mathsf{u}}$

7      $\underline{p}_1^{(j)} = p_{\mathrm{su}}\, p_{-}^{(j)}$

8   Perform single sensor update steps sequentially, i.e.,

9   **for** $s = 1 : S$ **do**

10     $\left[\left\{\left\{\boldsymbol{x}_s^{*(h,j)}, \boldsymbol{P}_s^{(h,j)}\right\}_{h=1}^{N_{\mathrm{k}}}, p_s^{(j)}\right\}_{j=1}^{J_s}\right] =$
      $\mathrm{SingleSensorUpdate}\left[\left\{\left\{\underline{\boldsymbol{x}}_s^{*(h,j)}, \underline{\boldsymbol{P}}_s^{(h,j)}\right\}_{h=1}^{N_{\mathrm{k}}}, \underline{p}_s^{(j)}\right\}_{j=1}^{J_s-1}, \boldsymbol{z}_s\right]$

11   // see **Alg. 9**

12   Set $\left\{\left\{\boldsymbol{x}^{*(h,j)}, \boldsymbol{P}^{(h,j)}\right\}_{h=1}^{N_{\mathrm{k}}}, p^{(j)}\right\}_{j=1}^{J} = \left\{\left\{\boldsymbol{x}_S^{*(h,j)}, \boldsymbol{P}_S^{(h,j)}\right\}_{h=1}^{N_{\mathrm{k}}}, p_S^{(j)}\right\}_{j=1}^{J_S}$
   where $J = J_S$

13   Output: $\left\{\left\{\boldsymbol{x}^{*(h,j)}, \boldsymbol{P}^{(h,j)}\right\}_{h=1}^{N_{\mathrm{k}}}, p^{(j)}\right\}_{j=1}^{J}$

VI. Numerical Results

Next, we report simulation results assessing the performance of our method and comparing it with that of two reference methods for multisensor MOT.

A. Tracking Scenario and Reference Methods

We consider an underwater 3-D surveillance scenario where eight objects are tracked by two static sonar hydrophone arrays. 200 time steps are considered. The hydrophones are deployed about 1300 m below sea level. The object states at time k consist of 3-D position and velocity, i.e., $\mathbf{x}_k^{(j)} = [\mathsf{x}_{1,k}^{(j)}\, \mathsf{x}_{2,k}^{(j)}\, \mathsf{x}_{3,k}^{(j)}\, \dot{\mathsf{x}}_{1,k}^{(j)}\, \dot{\mathsf{x}}_{2,k}^{(j)}\, \dot{\mathsf{x}}_{3,k}^{(j)}]^{\mathrm{T}}$, $j = 1, \ldots, 8$ and evolve according to a constant-velocity model [60, Sec. 6.3.2], where the dynamic noise has the physical interpretation as an acceleration with variance σ_{w}^2. The region of interest (ROI) is $[-1000\mathrm{m}, 1000\mathrm{m}] \times [-1000\mathrm{m}, 1000\mathrm{m}] \times [-1500\mathrm{m}, -500\mathrm{m}]$. Objects appear at $k \in \{1, 10, 20, 30, 40, 50, 60, 70\}$ and disappear at $k \in \{130, 140, 150, 160, 170, 180, 190, 200\}$. To simulate a tracking scenario with challenging DA, we generate the initial state of the objects as follows. The initial state of the first object is randomly generated by setting its position at a circle centered at the origin with a radius of 50 m and a depth of 1000 m, i.e., for its position, we have $\sqrt{\mathsf{x}_{1,0}^{(1)2} + \mathsf{x}_{2,0}^{(1)2}} = 50$ m and $\mathsf{x}_{3,0}^{(1)} = -1000$ m. The velocity is obtained by setting the vertical speed to zero and the horizontal velocity vector pointing to the circle center with $\sqrt{\dot{\mathsf{x}}_{1,0}^{(1)2} + \dot{\mathsf{x}}_{2,0}^{(1)2}} = 0.3\mathrm{m/s}$. For the appearance of each further object $j = 2, \ldots, 8$, an initial position is randomly generated near the initial position of the previously appeared object, i.e., around a circle with radius 50 m centered at $[x_{1,k}^{(j-1)}\, x_{2,k}^{(j-1)}\, x_{3,k}^{(j-1)}]$. The initial velocity vector is set with respect to the center circle, as discussed above. As a result of this initialization procedure, tracks start in close proximity in time and space. This makes it challenging to perform DA and declare the existence of newborn objects. The times when objects appear and disappear, as well as the states of appearing objects, are unknown to all simulated tracking methods. All methods aim to detect the presence of a new object and sequentially estimate its state across time merely based on TDOA measurements and the statistical model. The prior intensity of object birth is modeled at each timestep by a Poisson point process with mean $\mu_{\mathrm{b}} = 0.05$. The prior pdf for newborn object, $f_{\mathrm{b}}(\overline{\boldsymbol{x}}_k)$, is uniform on the ROI for the 3-D position and Gaussian with zero mean and covariance matrix $5\mathrm{m}^2/\mathrm{s}^2\boldsymbol{I}_3$ for the 3-D velocity. The survival probability is $p_{\mathrm{su}} = 0.95$. The object declaration threshold is set to $P_{\mathrm{th}} = 0.5$ and the pruning threshold to $P_{\mathrm{pr}} = 10^{-4}$.

The two hydrophone arrays have the geometry of the array described in [61] and are located at $[519\,\mathrm{m}\ 137\,\mathrm{m}\ -1300\,\mathrm{m}]^{\mathrm{T}}$ and $[-519\,\mathrm{m}\ -137\,\mathrm{m}\ -1300\,\mathrm{m}]^{\mathrm{T}}$, respectively. Each hydrophone array consists of 4 receivers. Hence, there are six receiver pairs at each array. Each receiver pair generates TDOA measurements and is considered a sensor for MOT. This means that multisensor measurements $\mathbf{z}_{k,s}^{(m)}$, $m \in \{1, \ldots, \mathsf{M}_{k,s}\}$ and $s \in \{1, \ldots, 12\}$ are obtained by the two arrays at time k. At each sensor s, a random number of $\mathsf{M}_{k,s}$ measurements are generated. In particular, the TDOA measurement $\mathsf{z}_{k,s}^{(m)}$ of a detected object with state $\mathbf{x}_k^{(j)}$ is modeled as $\mathsf{z}_{k,s}^{(m)} = \frac{1}{c}\left(\left\|[\mathsf{x}_{1,k}^{(j)}\, \mathsf{x}_{2,k}^{(j)}\, \mathsf{x}_{3,k}^{(j)}]^{\mathrm{T}} - \boldsymbol{p}_{s_{\mathrm{L}}}\right\| - \left\|[\mathsf{x}_{1,k}^{(j)}\, \mathsf{x}_{2,k}^{(j)}\, \mathsf{x}_{3,k}^{(j)}]^{\mathrm{T}} - \boldsymbol{p}_{s_{\mathrm{R}}}\right\|\right) + \mathsf{v}_{k,s}^{(m)}$ where $\boldsymbol{p}_{s_{\mathrm{L}}}$ and $\boldsymbol{p}_{s_{\mathrm{R}}}$ are the paired receiver positions of sensor s, $c = 1500$ m/s is the propagation speed, and $\mathsf{v}_{k,s}^{(m)}$ is additive zero-mean Gaussian noise with standard deviation σ_{v} that is assumed statistically independent across s, k, and m. The pdf of FP measurements, $f_{\mathrm{fp}}(z_{k,s}^{(m)})$, is uniform on $\frac{1}{c}\left[-\|\boldsymbol{p}_{s_{\mathrm{L}}} - \boldsymbol{p}_{s_{\mathrm{R}}}\|, \|\boldsymbol{p}_{s_{\mathrm{L}}} - \boldsymbol{p}_{s_{\mathrm{R}}}\|\right]$.

We compare the proposed SPA-based MOT with the embedded particle flow sampling strategy ("SPA-PF") with two reference sampling strategies. The first ("SPA-PM") follows the sampling strategy of the bootstrap particle filter [23], [25] and uses predicted beliefs as proposal pdf [9]. The second (yet unpublished) ("SPA-UT") follows the sampling strategy of the unscented particle filter [35], i.e., it uses a Gaussian mixture representation and the unscented transformation to calculate an informative proposal pdfs. For SPA-PM, we use $N_{\mathrm{b}} = 10^6$ particles for newborn POs and $N_{\mathrm{b}} = 6 \cdot 10^4$ particles for legacy POs. For all other simulated methods, we use $N_{\mathrm{k}} = 100$ kernels. For each kernel representing a newborn PO, we set $N_{\mathrm{p}} = 500$ for SPA-UT and SPA-PF and for each kernel representing a legacy PO, we set $N_{\mathrm{p}} = 30$ for SPA-UT and SPA-PF. Since $N_{\mathrm{b}}/N_{\mathrm{p}}N_{\mathrm{k}} = 20$ the memory requirements of SPA-PM are 20 times higher compared to SPA-PF and SPA-UT.

We also simulate two variants of the second reference method to obtain a similar runtime for SPA-PF. In particular, for "SPA-UT-1", we set $N_{\mathrm{p}} = 4000$ for kernels representing newborn POs and $N_{\mathrm{p}} = 250$ for kernels representing legacy POs. Furthermore, for "SPA-UT-2", we set $N_{\mathrm{p}} = 6000$ for kernels representing newborn PO and $N_{\mathrm{p}} = 30$ for kernels representing legacy PO. Note that the memory requirements of
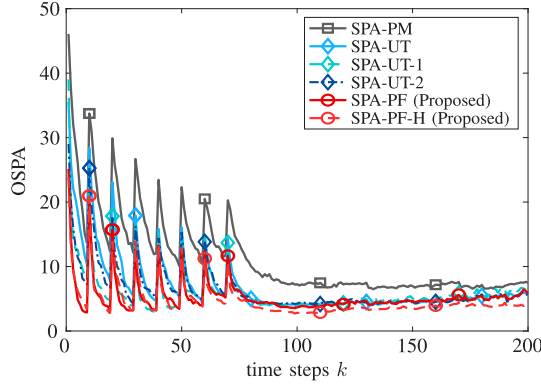
Fig. 3. MOSPA performance for high uncertainty of prior with $\sigma_\mathbf{w} = 1$ m/s^2. Other parameters are set as $\mu_\mathrm{fp} = 5$, $\sigma_\mathbf{v} = 10^{-6}$ s, $p_\mathrm{d} = 0.9$.

Fig. 4. MOSPA performance for informative measurement model with $\sigma_\mathbf{v} = 5 \times 10^{-7}$ s. Other parameters are set as $\mu_\mathrm{fp} = 5$, $\sigma_\mathbf{w} = 0.1$ m/s^2, $p_\mathrm{d} = 0.9$.

SPA-UT-1 and SPA-UT-2 are higher than the ones of SPA-UT and SPA-PF due to their higher values of N_p. Finally, we also simulate a method ("SPA-PF-H") that uses the sampling strategy of SPA-PF for kernels representing newborn POs and the sampling strategy of SPA-PM for kernels representing legacy POs. Using fewer samples for the kernels representing legacy PO, or even using the strategy of SPA-PF, is motivated by the fact that the beliefs of legacy POs are typically unimodal and quite informative. We set $N_\lambda = 20$ for SPA-PF. For all considered methods, 100 simulation runs are performed. All methods are implemented in MATLAB, and each simulation run is processed on a single core of a 2.6GHz Intel Xeon Gold 6240 processor.

The performance of the six MOT methods is evaluated w.r.t. to changes in four system parameters (i) object driving noise standard deviation $\sigma_\mathbf{w}$, (ii) mean number of FPs μ_fp, (iii) detection probability p_d and (iv) measurement noise standard deviation $\sigma_\mathbf{v}$. Note for the setting $\sigma_\mathbf{v} = 5 \times 10^{-7}$ s, the number of samples N_p was doubled for all methods to yield high tracking performance. The tracking accuracy of the various methods is measured by the Euclidean distance-based OSPA metric with cutoff parameter $C = 50$ [62].

B. Performance Comparison

In what follows, we discuss two scenarios where particle degeneracy is particularly pronounced. Particle degeneracy can be caused by uninformative prior information or a very informative likelihood function. To obtain a scenario with uninformative prior information, we consider the case $\sigma_\mathbf{w} = 1$ m/s^2. In addition, to obtain a scenario with an informative likelihood function, we consider the case $\sigma_\mathbf{v} = 5 \times 10^{-7}$ s. Fig. 3 and Fig. 4 show the mean (MOSPA) error—averaged over 100 simulation runs—of all methods versus time k for these two scenarios. It can be seen that all methods yield error peaks at time steps where the objects appear. This is because the MOT methods do not know when a new object appears and often need a few time steps after the appearance of an object to declare its existence based on TDOA measurements and the statistical model. However, the proposed methods, i.e., SPA-PF and SPA-PF-H, have lower error peaks at the time steps when
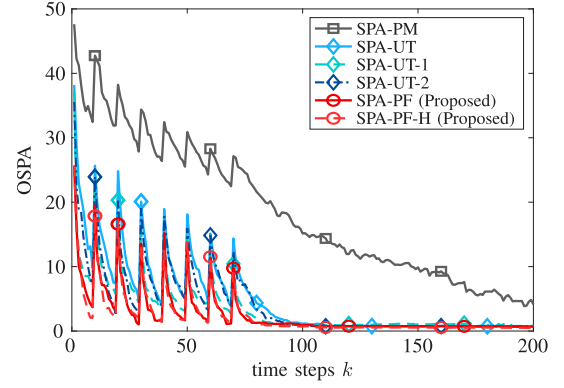
objects appear, i.e., SPA-PF and SPA-PF-H can often declare the existence of an object faster. In addition, SPA-PF and SPA-PF-H can outperform the other reference methods at almost all time steps. SPA-PM, in particular performs very poorly due to particle degeneracy. It can be noted that SPA-UT-1 and SPA-UT-2 have improved performance compared to SPA-UT but are still outperformed by SPA-PF and SPA-PF-H despite their more extensive memory requirements.

Furthermore, the MOSPA error and the runtime per time step of the six methods for different system parameter values are shown in Table I. The default value of the four parameters are $\sigma_\mathbf{w} = 0.1$ m/s^2, $\sigma_\mathbf{v} = 1 \times 10^{-6}$ s, $\mu_\mathrm{fp} = 5$ and $p_\mathrm{d} = 0.9$. For each row corresponding to a particular system parameter value, the value of the other three parameters is set to the default value. For each system parameter value corresponding to one row, the MOSPA and runtime are averaged over 100 simulation runs and 200 time steps. The best and the second best MOSPA value corresponding to each system parameter value is marked by an underline and dashed underline, respectively. As can be noted, the performance of SPA-PM significantly degrades when measurement noise standard deviation is reduced to $\sigma_\mathbf{v} = 5 \times 10^{-7}$ s. This unwanted and counterintuitive behavior is a clear indicator of particle degeneracy in SPA-PM. Only with the proposed method is it possible to yield improved tracking performance as measurement noise variance is reduced.

It can be seen that for almost all system parameter values, the proposed SPA-PF and SPA-PF-H outperform the reference methods. At the same time, their runtime is comparable with SPA-UT-1 and SPA-UT-2, which yield higher memory requirements. The only scenario where SPA-PF and SPA-PF-H do not result in the lowest MOSPA is when $\sigma_\mathbf{v} = 2 \times 10^{-6}$ s. In this case, since measurements are not very informative, SPA-PM does not suffer from particle degeneracy. It can also be seen that SPA-PF-H outperforms SPA-PF both in terms of MOSPA and runtime, while both SPA-PF and SPA-PF-H typically outperform SPA-UT. We can thus conclude that the main challenge for the sampling method is the initial time step after a new object appears in the scene. Here, beliefs of new POs are highly uninformative and have complicated shapes. At later time steps, beliefs become informative and unimodal and

TABLE I
MOSPA AND RUNTIME PER TIME STEP OF DIFFERENT ALGORITHMS FOR THE CONSIDERED TRACKING SCENARIO W.R.T RELATED PARAMETERS

Parameter	Value	SPA-PM		SPA-UT		SPA-UT-1		SPA-UT-2		SPA-PF		SPA-PF-H	
		OSPA	Runtime (s)	OSPA	Runtime (s)	OSPA	Runtime (s)	OSPA	Runtime (s)	OSPA	Runtime (s)	OSPA	Runtime (s)
$\sigma_{\mathbf{w}}$	1 m/s²	11.38	20.94	7.02	3.22	6.06	19.98	6.45	25.84	5.31	21.66	4.64	13.19
	0.5 m/s²	9.13	23.54	5.84	3.35	4.50	21.42	5.11	28.10	3.80	22.54	3.45	13.90
	0.1 m/s²	7.61	23.99	4.82	3.35	3.45	21.59	4.35	29.10	2.96	22.48	2.67	13.68
$\sigma_{\mathbf{v}}$	2×10^{-6} s	4.51	21.69	5.25	3.19	4.78	19.85	5.31	25.64	5.37	22.53	5.13	14.43
	1×10^{-6} s	7.61	20.77	4.69	3.05	3.44	19.33	4.45	25.95	3.02	20.64	2.69	12.62
	5×10^{-7} s	18.60	42.83	5.10	5.06	3.59	37.94	4.28	51.82	2.61	31.72	2.40	21.26
μ_{fp}	10	7.79	33.88	5.85	5.29	4.89	33.13	6.74	41.76	4.30	41.21	4.10	29.48
	5	7.61	22.66	4.78	3.15	3.48	20.20	4.38	27.02	3.28	21.58	2.82	13.21
	2	9.70	15.66	5.03	2.40	2.36	14.20	4.23	19.22	2.67	13.42	2.33	7.27
p_{d}	0.85	11.78	20.72	8.52	3.27	4.89	20.15	8.00	26.36	4.32	22.31	5.02	14.89
	0.9	7.61	22.15	5.01	3.02	3.29	18.98	4.65	24.97	3.09	21.15	2.66	12.50
	0.95	8.06	21.80	3.90	3.26	2.58	20.77	3.01	27.83	2.56	21.10	2.35	12.40

Fig. 5. MOSPA performance of SPA-PM and SPA-PF-H for different number of kernels, N_k.

can thus be computed accurately with fewer samples and the sampling strategy of the bootstrap particle filter. For real-time processing, it is expected that an adaptation of the proposed method for execution on GPUs, can strongly reduce runtimes by exploiting the highly parallelizable nature of PFL.

In what follows, we numerically investigate the effect of the number of kernels N_k on system performance. In particular, we set the number of kernels as $N_{\mathrm{k}} \in \{1, 5, 10, 50, 100\}$ in SPA-PF-H and compare SPA-PF-H for these different values of N_{k} with SPA-PM. The parameters $\sigma_{\mathbf{v}}, \mu_{\mathrm{fp}}, \sigma_{\mathbf{w}}$, and p_{d} are set as dicussed above. For comparable runtimes of all SPA-PF-H variants, we set $N_{\mathrm{p}} \in \{100000, 20000, 8000, 1300, 500\}$ for newborn object and $N_{\mathrm{p}} \in \{15000, 1000, 400, 70, 30\}$ for legacy objects. As expected, it can be seen in Fig. 5 that the accuracy of SPA-PF-H improves with increasing N_{k}. While SPA-PF-H performs very poorly for $N_{\mathrm{k}} = 1$, notably, it can already outperform SPA-PM for $N_{\mathrm{k}} = 5$. The average runtime per time step of SPA-PF-H is 27.14 s, 15.06 s, 10.85 s, 11.35 s, and 12.72 s for $N_{\mathrm{k}} = 1, N_{\mathrm{k}} = 5$, $N_{\mathrm{k}} = 10$, $N_{\mathrm{k}} = 50$, and $N_{\mathrm{k}} = 100$, respectively, as well as 22.66 s for SPA-PM.

C. Real Data Experiment: Echolocation in Oceanography

We further validate our proposed MOT method in an underwater acoustic tracking scenario. The acoustic signals "clicks" emitted by two Cuvier's beaked whales are recorded by two high-frequency acoustic recording package (HARP) [61], each of which is equipped with four hydrophones. HARPs are deployed at a depth of 1330 m and approximately 1 km apart. Preprocessing of acoustic data is described in [50]. Each pair of hydrophones on each HARP acts as a sensor that provides TDOA measurements every 7 s. Since there are 6 pairs of hydrophones on each HARP, there are a total of $S = 12$ sensors providing TDOA measurements. We use a dataset that consists of 172 time steps and has a total duration of roughly 20 minutes. It was recorded on July 1st, 2018, in Southern California. Tracking results are shown in Fig. 6. The red solid lines show the estimated tracks of two whales provided by SPA-PF-H. The two whales were initially detected at depths of about 450 m and then kept diving until a depth of 1300 m. For obvious reasons, no ground truth information exists for this scenario. However, we have added reference tracks of the two whales that are the result of a trained operator hand-annotating preprocessed acoustic data.

In Fig. 7, we show the estimate tracks of SPA-PF-H compared with SPA-PM in 2-D. SPA-PM does break and merge tracks, which makes it very difficult to determine how many whales are actually there. Most importantly SPA-PF-H can potentially replace the human operator, while SPA-PM cannot. The overall number of particles of SPA-PM is 20 times that of SPA-PF-H. The runtime per time step of SPA-PM is 4.65 s while 2.35 s of SPA-PF-H, i.e., SPA-PF-H is faster than SPA-PM. Since the measurement interval is 7 s, both SPA-PF-H and SPA-PM can be used in real-time. For larger scenarios with
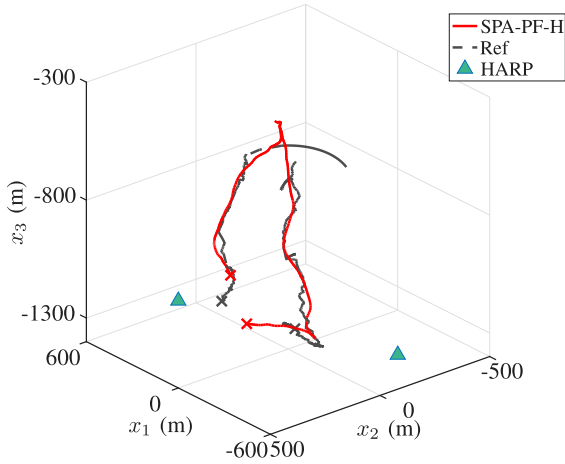
Fig. 6. Underwater acoustic tracking scenario with two Cuvier's beaked whales.
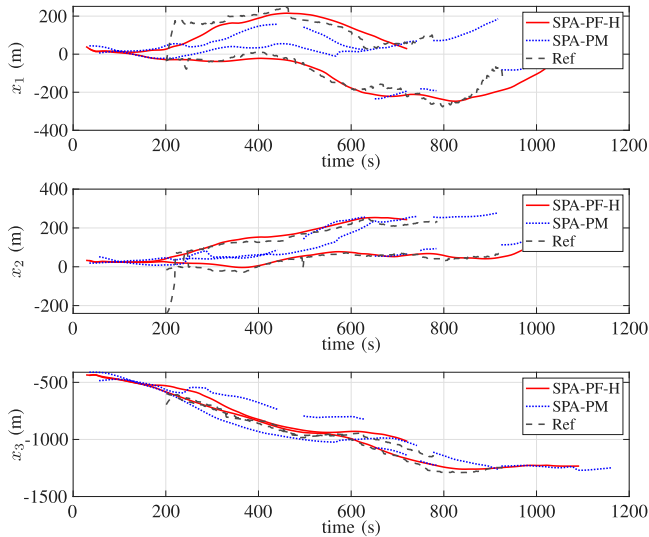

Fig. 7. A comparison between the estimated tracks provided by SPA-PF-H and SPA-PM. The hand-annotated tracks of the two whales are marked for reference (gray dashed line). Each axis of the 3-D domain is shown individually.

more than two whales, a GPU implementation is required for real-time processing.

VII. CONCLUSION

We presented a graph-based Bayesian method for multisensor MOT with high-dimensional object states. Particle degeneracy is avoided by performing operations on the graph using PFL. Our numerical results indicate that the main challenge for sampling is representing the posterior distribution at the initial time step after a new object appears in the scene. Compared to state-of-the-art reference methods, we show favorable tracking performance in a 3-D MOT scenario. The introduced approach is expected to be particularly appealing for passive surveillance problems [63]. Future research avenues include graph-based processing with embedded stochastic PFL [64], [65] and applications including extended object tracking [13], [66], simultaneous localization and object tracking [16], and

information-seeking control [67]. We also aim to demonstrate real-time processing capabilities of the proposed approach by execution on GPUs, exploiting the highly parallelizable nature of PFL.

REFERENCES

[1] M. Zimmer, S. Dolman, and L. Weilgart, *Passive Acoustic Monitoring* of Cetaceans. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[2] K. L. Cummins and M. J. Murphy, "An overview of lightning locating systems: History, techniques, and data uses, with an in-depth look at the U.S. NLDN," *IEEE Trans. Electromagn. Compat.*, vol. 51, no. 3, pp. 499–518, Aug. 2009.

[3] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT, USA: Yaakov Bar-Shalom, 2011.

[4] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[5] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, 2007.

[6] J. L. Williams, "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA and association-based MeMBer," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1664–1687, Jul. 2015.

[7] B.-N. Vo, B.-T. Vo, and H. G. Hoang, "An efficient implementation of the generalized labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1975–1987, Apr. 2017.

[8] A. Saucan, Y. Li, and M. Coates, "Particle flow SMC delta-GLMB filter," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 4381–4385.

[9] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch, "A scalable algorithm for tracking an unknown number of targets using multiple sensors," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3478–3493, Jul. 2017.

[10] F. Meyer et al., "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[11] G. Soldi, F. Meyer, P. Braca, and F. Hlawatsch, "Self-tuning algorithms for multisensor-multitarget tracking using belief propagation," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 3922–3937, Aug. 2019.

[12] A. Tesei, F. Meyer, and R. Been, "Tracking of multiple surface vessels based on passive acoustic underwater arrays," *J. Acoust. Soc. Am.*, vol. 147, no. 2, pp. EL87–EL92, 2020.

[13] F. Meyer and J. L. Williams, "Scalable detection and tracking of geometric extended objects," *IEEE Trans. Signal Process.*, vol. 69, pp. 6283–6298, 2021.

[14] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[15] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[16] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, "Distributed localization and tracking of mobile networks including noncooperative objects," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 1, pp. 57–71, Mar. 2016.

[17] M. Z. Win, F. Meyer, Z. Liu, W. Dai, S. Bartoletti, and A. Conti, "Efficient multisensor localization for the Internet of Things: Exploring a new class of scalable localization algorithms," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 153–167, 2018.

[18] B. Teague, Z. Liu, F. Meyer, A. Conti, and M. Z. Win, "Network localization and navigation with scalable inference and efficient operation," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2072–2087, Jun. 2022.

[19] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Z. Win, "A belief propagation algorithm for multipath-based SLAM," *IEEE Trans. Wireless Commun.*, vol. 18, no. 12, pp. 5613–5629, Dec. 2019.

[20] X. Li, E. Leitinger, A. Venus, and F. Tufvesson, "Sequential detection and estimation of multipath channel parameters using belief propagation," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8385–8402, Oct. 2022.

[21] E. Leitinger, A. Venus, B. Teague, and F. Meyer, "Data fusion for multipath-based SLAM: Combing information from multiple propagation paths," 2022, *arXiv: 2211.09241*.

[22] F. Meyer and K. L. Gemba, "Probabilistic focalization for shallow water localization," *J. Acoust. Soc. Am.*, vol. 150, no. 2, pp. 1057–1066, 2021.

[23] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F*, vol. 140, no. 2, pp. 107–113, 1993.

[24] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001.

[25] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[26] P. Bickel, B. Li, and T. Bengtsson, "Sharp failure rates for the bootstrap particle filter in high dimensions," in *Pushing the Limits of Contemporary Statistics*: Contributions in Honor of Jayanta K. Ghosh, vol. 3. Beachwood, OH, USA: Inst. Math. Statist., 2008, pp. 318–329.

[27] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularised particle filters," in *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001, pp. 247–271.

[28] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, "Adaptive importance sampling: The past, the present, and the future," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 60–79, Jul. 2017.

[29] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric, "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 145–152, Nov. 2019.

[30] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Am. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, Jun. 1999.

[31] P. M. Djuric, T. Lu, and M. F. Bugallo, "Multiple particle filtering," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, vol. 3, 2007, pp. 1181–1184.

[32] C. Berzuini, N. G. Best, W. R. Gilks, and C. Larizza, "Dynamic conditional independence models and Markov chain Monte Carlo methods," *J. Am. Stat. Assoc.*, vol. 92, no. 440, pp. 1403–1412, 1997.

[33] W. R. Gilks and C. Berzuini, "Following a moving target-Monte Carlo inference for dynamic Bayesian models," *J. Roy. Statist. Soc. Ser. B Statist. Methodol.*, vol. 63, no. 1, pp. 127–146, 2001.

[34] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[35] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2000, pp. 584–590.

[36] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wang, "Four hydrophone array for acoustic three-dimensional location," Cambridge Univ. Eng. Dept., Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR 380, Aug. 2000.

[37] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proc. Signal Data Process. Small Targets (SPIE)*, Aug. 2007, pp. 423–437.

[38] F. Daum and J. Huang, "Nonlinear filters with particle flow induced by log-homotopy," in *Proc. Signal Process., Sensor Fusion, Target Recognit. XVIII (SPIE)*, May 2009, pp. 76–87.

[39] F. Daum, J. Huang, and A. Noushin, "Exact particle flow for nonlinear filters," in *Proc. Signal Process., Sensor Fusion, Target Recognit. XIX (SPIE)*, Apr. 2010, pp. 92–110.

[40] F. Daum and J. Huang, "Particle flow with non-zero diffusion for nonlinear filters," in *Proc. Signal Process., Sensor Fusion, Target Recognit. XXII (SPIE)*, May 2013, pp. 226–238.

[41] P. Bunch and S. Godsill, "Approximations of the optimal importance density using Gaussian particle flow importance sampling," *J. Am. Stat. Assoc.*, vol. 111, no. 514, pp. 748–762, Aug. 2016.

[42] Y. Li and M. Coates, "Particle filtering with invertible particle flow," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4102–4116, Aug. 2017.

[43] S. Pal and M. Coates, "Gaussian sum particle flow filter," in *Proc. IEEE 7th Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process. (CAMSAP)*, Curacao, Dutch Antilles, Dec. 2017, pp. 1–5.

[44] S. Pal and M. Coates, "Particle flow particle filter for Gaussian mixture noise models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, Canada, 2018, pp. 4249–4253.

[45] Y. Li, S. Pal, and M. J. Coates, "Invertible particle-flow-based sequential MCMC with extension to Gaussian mixture noise models," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2499–2512, May 2019.

[46] L. Wielandner, E. Leitinger, F. Meyer, and K. Witrisal, "Message passing-based 9-D cooperative localization and navigation with embedded particle flow," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 9, pp. 95–109, 2023.

[47] L. Wielandner, E. Leitinger, F. Meyer, B. Teague, and K. Witrisal, "Message passing-based cooperative localization with embedded particle flow," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, pp. 5652–5656.

[48] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.

[49] W. Zhang and F. Meyer, "Graph-based multiobject tracking with embedded particle flow," in *Proc. IEEE Radar Conf.*, Atlanta, GA, USA, May 2021, pp. 1–6.

[50] J. Jang, F. Meyer, E. R. Snyder, S. M. Wiggins, S. Baumann-Pickering, and J. A. Hildebrand, "Bayesian detection and tracking of odontocetes in 3-D from their echolocation clicks," *J. Acoust. Soc.*, vol. 153, no. 5, pp. 2690–2705, May 2023.

[51] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2002.

[52] F. Daum and J. Huang, "Exact particle flow for nonlinear filters: Seventeen dubious solutions to a first order linear underdetermined PDE," in *Proc. Conf. Rec. 44th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2010, pp. 64–71.

[53] T. Ding and M. J. Coates, "Implementation of the Daum-Huang exact-flow particle filter," in *Proc. IEEE Statist. Signal Process. Workshop (SSP)*, Ann Arbor, MI, USA, 2012, pp. 257–260.

[54] B. Silverman, *Density Estimation for Statistics and Data Analysis*. New York, NY, USA: Taylor & Francis, 1986.

[55] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, "Finite mixture models," *Annu. Rev. Statist. Appl.*, vol. 6, no. 1, pp. 355–378, 2019.

[56] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*. New York, NY, USA: Cambridge Univ. Press, 2007.

[57] W. Zhang and F. Meyer, "Multisensor multiobject tracking with improved sampling efficiency: Supplementary Material," 2024. Accessed: Mar. 1, 2024. [Online]. Available: https://fmeyer.ucsd.edu/TSP-ZhaMey-2024-SM.pdf

[58] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York, NY, USA: Springer-Verlag, 1994.

[59] J. L. Williams and R. Lau, "Approximate evaluation of marginal association probabilities with belief propagation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 4, pp. 2942–2959, Oct. 2014.

[60] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001.

[61] S. M. Wiggins and J. A. Hildebrand, "High-frequency acoustic recording package (HARP) for broad-band, long-term marine mammal monitoring," in *Proc. IEEE Int. Symp. Underwater Technol.*, Apr. 2007, pp. 551–557.

[62] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.

[63] G. Ferri et al., "Cooperative robotic networks for underwater surveillance: An overview," *IET Radar Sonar Navig.*, vol. 11, no. 12, pp. 1740–1761, Dec. 2017.

[64] L. Dai and F. Daum, "Stiffness mitigation in stochastic particle flow filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 4, pp. 3563–3577, Aug. 2022.

[65] L. Dai and F. Daum, "On the design of stochastic particle flow filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 3, pp. 2439–2450, Jun. 2023.

[66] F. Meyer and M. Z. Win, "Scalable data association for extended object tracking," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 491–507, 2020.

[67] F. Meyer, H. Wymeersch, M. Fröhle, and F. Hlawatsch, "Distributed estimation with information-seeking control in agent networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2439–2456, Nov. 2015.

Wenyu Zhang (Student Member, IEEE) received the B.E. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017, and the M.S. degree in computer engineering from the University of California San Diego (UCSD), La Jolla, CA, USA, in 2020. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, UCSD. His research interests include high dimensional sampling and filtering, Bayesian inference, information seeking control, and reinforcement learning. In 2022, he was a recipient of an Qualcomm Innovation Fellowship. He serves as a Reviewer for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, and the *ISIF Journal of Advances in Information Fusion*.

Florian Meyer (Member, IEEE) received the M.Sc. and Ph.D. degrees (with highest honors) in electrical engineering from TU Wien, Vienna, Austria, in 2011 and 2015, respectively. He is an Assistant Professor with the University of California San Diego, La Jolla, CA, USA, jointly between Scripps Institution of Oceanography and the Electrical and Computer Engineering Department. From 2017 to 2019, he was a Postdoctoral Fellow and Associate with the Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA, and from 2016 to 2017, he was a Research Scientist with the NATO Centre for Maritime Research and Experimentation, La Spezia, Italy. His research interests include statistical signal processing, high-dimensional and nonlinear estimation, inference on graphs, machine perception, and graph neural networks. He is the recipient of the 2021 ISIF Young Investigator Award, an 2022 NSF CAREER Award, an 2022 DARPA Young Faculty Award, and an 2023 ONR Young Investigator Award. He is currently an Associate Editor with IEEE TRANSACTIONS ON SIGNAL PROCESSING and also served as an Associate Editor for IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS from 2021 to 2023 and the *ISIF Journal of Advances in Information Fusion* from 2019 to 2022.