

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Localization and Tracking in Challenging Environments: Particle Flow and Active Planning

Permalink

<https://escholarship.org/uc/item/3fs0d4z5>

ISBN

9798273318229

Author

Zhang, Wenyu

Publication Date

2025-01-13

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Localization and Tracking in Challenging Environments: Particle Flow and Active Planning

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Computer Engineering)

by

Wenyu Zhang

Committee in charge:

Professor Florian Meyer, Chair
Professor Nikolay A. Atanasov
Professor Sicun Gao
Professor Peter Gerstoft

2025

Copyright
Wenyu Zhang, 2025
All rights reserved.

The Dissertation of Wenyu Zhang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2025

DEDICATION

*To my family,
whose love and support carried me through every chapter of this work.*

EPIGRAPH

All models are wrong,
But some are useful.

George E. P. Box

TABLE OF CONTENTS

Dissertation Approval Page	iii
Dedication	iv
Epigraph	v
Table of Contents	vi
List of Figures	ix
List of Tables	xi
Preface	xii
Acknowledgements	xiii
Vita	xiv
Abstract of the Dissertation	xv
Chapter 1 Introduction	1
1.1 Bayesian Estimation and Engineering Applications	1
1.2 State-of-the-Art Techniques for Localization and Tracking	4
1.2.1 Tracking with Sequential Bayesian Estimation and Fusion	4
1.2.2 Multisensor MOT with Graph-based BP	5
1.2.3 Curse of Dimensionality and PFI	6
1.2.4 Cooperative Localization and Information-Seeking Control	10
1.3 Motivation, Contribution and Outline	12
Chapter 2 Background	16
2.1 Bayesian Estimation	17
2.2 Sequential Bayesian Estimation	19
2.3 Multiobject Tracking and Data Association	21
2.3.1 Potential Object States	22
2.3.2 Measurement-Object Uncertainty	23
2.3.3 Joint Posterior PDF and Factor Graph	25
2.3.4 Problem Formulation and Sum-Product Message Passing	25
2.4 Summary	29
Chapter 3 Particle Flow for Importance Sampling	30
3.1 Curse of Dimensionality and Particle Flow	30
3.1.1 Curse of Dimensionality Revisited	30
3.1.2 Particle Flow as a Homotopy Transformation	33
3.1.3 Exact Particle Flow for Gaussian Distribution	34

3.2	Importance Sampling with Invertible Flow	38
3.3	Improving Sampling Efficiency for A Single Object with Measurement-Origin Uncertainty	40
3.3.1	GMM Importance Sampling with Invertible Flow	41
3.3.2	PFI-Based Proposal for Problems with MOU	41
3.3.3	GMM Importance Sampling with Invertible Flow for Problems with MOU	46
3.4	Graph-Based Multisensor MOT with Invertible PFI	46
3.4.1	Nonparametric Estimation of General Integral Functions	49
3.4.2	Prediction	50
3.4.3	Measurement Evaluation	50
3.4.4	Measurement Update and Belief Calculation	52
3.4.5	The Proposed Multisensor MOT Method	55
3.5	Experiments and Numerical Results	57
3.5.1	Tracking Scenario and Reference Methods	57
3.5.2	Performance Comparison	59
3.5.3	Real Data Experiment: Echolocation in Oceanography	62
3.6	Summary	65
Chapter 4	Stochastic Particle Flow	67
4.1	Stochastic PFI	68
4.2	IS With Stochastic PFI	69
4.3	Numerical Experiments and Results	72
4.3.1	Simulation Parameters of Source Localization Scenario	73
4.3.2	Performance of Gromov's Flow	75
4.3.3	Performance of General Stochastic PFI with Diffusion Optimization	79
4.4	Summary	82
Chapter 5	FIM-Based Active Planning	83
5.1	System Model and Estimation Problem	84
5.1.1	System Model	84
5.1.2	Estimation for Cooperative Localization	87
5.2	Active Planning	88
5.2.1	Equivalent FIM and Myopic Control Policy	88
5.2.2	RH Control Policy and Tree-Search	90
5.3	Results	92
5.3.1	Simulation Setup and Example Realization	93
5.3.2	Numerical Results and Computational Complexity	95
Chapter 6	Conclusion	98
Appendix A	Derivation PDE of PFI	103
A.1	Review of Fokker-Planck equation	103
A.1.1	Kramers-Moyal Expansion	103
A.1.2	Fokker-Planck Equation	104

A.2 PDE of Bayesian Particle Flow	105
Bibliography	108

LIST OF FIGURES

Figure 2.1.	Diagram of machine perception. The block in red is the topic we focus in this work.	16
Figure 2.2.	Sequential Bayesian estimation.	21
Figure 2.3.	Factor graph for multiobject tracking (MOT) of an unknown, time-varying number of objects, corresponding to the propagation of the joint probability density function (PDF) $f(\mathbf{y}_{1:k}, \mathbf{a}_{1:k}, \mathbf{b}_{1:k} \mathbf{z}_{1:k})$ in (2.11). One time step k is shown.	26
Figure 3.1.	Particle degeneracy in a tracking scenario with 3-D object state.	32
Figure 3.2.	Example of particle flow (PFI) in the tracking scenario with 3-D object state and a single 1-D time-difference of arrival (TDOA) measurement discussed in Fig. 3.1.	37
Figure 3.3.	OSPA performance for high uncertainty of prior with $\sigma_w = 1 \text{ m/s}^2$. Other parameters are set as $\mu_{fp} = 5$, $\sigma_v = 10^{-6} \text{ s}$, $p_d = 0.9$	60
Figure 3.4.	OSPA performance for informative measurement model with $\sigma_v = 5 \times 10^{-7} \text{ s}$. Other parameters are set as $\mu_{fp} = 5$, $\sigma_w = 0.1 \text{ m/s}^2$, $p_d = 0.9$	61
Figure 3.5.	OSPA performance of BP-PM and BP-PF-H for different number number of kernels, N_k	64
Figure 3.6.	Underwater acoustic tracking scenario with two Cuvier's beaked whales.	65
Figure 3.7.	A comparison between the estimated tracks provided by BP-PF-H and BP-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.	66
Figure 4.1.	(a). Source position and hyperboloids resulting from the TDOA measurements of two sensors. Each sensor consists of two receiver pairs. A dashed red line indicates the intersection of the two hyperboloids. (b). Receiver and sensor topology used in our simulation.	74
Figure 4.2.	Particle-based representation after the exact Daum and Huang (EDH) flow of the spatial PDFs by Gaussian mixture model (GMM) model. There exist three signal sources in a 3-D environment with clutters, and the state is sequentially filtered by sensor.	76
Figure 4.3.	Statistics of the optimal sub-pattern assignment (OSPA) error for different algorithms. Each column corresponds to the algorithm with the same ID as in Table 4.1.	78

Figure 4.4.	Log condition number of the Jacobian matrix (a) and log 2-norm of the diffusion matrix (b) of the drift term of different stochastic particle flows. .	81
Figure 5.1.	Cooperative localization scenario with mobile agents, mobile anchors, and static anchors.	85
Figure 5.2.	Block diagram of the proposed estimation and control framework for cooperative localization.	86
Figure 5.3.	3-D range direction vector between position of anchor j and position of agent i	88
Figure 5.4.	Example realization of the simulated 3-D localization scenario.	94
Figure 5.5.	Logarithmic mean-square error (MSE) of agent position versus time. The proposed active planning methods result in a significant reduction of the MSE.	96

LIST OF TABLES

Table 3.1.	Particle degeneracy of traditional importance sampling (IS) methods with TDOA measurements. Effective sample size (ESS) of IS and auxiliary importance sampling (AIS) using 2000 samples and different measurement standard deviations are shown for the tracking scenario in Fig. 3.1.	31
Table 3.2.	Mean OSPA (MOSPA) and runtime per time step of different algorithms for the considered tracking scenario w.r.t related parameters.	63
Table 4.1.	Simulated mean OSPA error and runtime per run for different algorithms and system parameters.	78
Table 4.2.	Simulated mean OSPA error and runtime per run for different methods and system parameters.	79
Table 4.3.	Mean OSPA error of different PFI-based IS method and different integration step sizes.	80
Table 5.1.	Percentage of lost tracks for all simulated methods.	97

PREFACE

This dissertation is the result of my Ph.D. research conducted at the Department of Electrical and Computer Engineering, University of California, San Diego, from September 2020 to August 2025.

Unlike many scholars who begin their Ph.D. studies with a clearly defined research interest, I did not have a strong sense of what I would pursue—even after completing my M.S. degree. During my undergraduate studies at SJTU and my master’s program at UCSD, I took a wide range of courses spanning electrical engineering and computer science, and I gained some research experience in different groups. However, a Ph.D. is a completely different experience—it requires deep focus on a long-term project filled with uncertainty.

I was fortunate to meet my Ph.D. advisor, Prof. Florian Meyer, who gave me the opportunity to join his group and guided me into the academic world of statistical signal processing, including Bayesian estimation algorithms, multi-object tracking, and more. To me, the core logic of statistical signal processing is about efficiently extracting useful information from signals to describe the world in a simplified yet meaningful way. Achieving this involves many aspects—such as signal preprocessing, causal inference, and modeling. My research is just a small piece of this broader picture, but I am motivated to explore more in this direction.

The challenges in a Ph.D. journey extend beyond research—they also come from life. I am deeply grateful to my advisor for his support and encouragement during moments of hesitation and difficulty. He introduced me to the academic community, allowing me to meet many colleagues with shared interests. I feel lucky to have completed my Ph.D. in an open and supportive environment, free from excessive external pressure. I also want to thank my family for their unwavering support of my decisions, and my colleagues and friends—it has always been a joy, whether discussing research or celebrating together.

At this point, I’m happy to see the progress I’ve made in both my research abilities and personal growth. It has been a great gift in my life, and the memories from this journey will always be precious.

ACKNOWLEDGEMENTS

I would like to sincerely acknowledge my Ph.D. advisor, Professor Florian Meyer, for his unwavering support and for serving as the chair of my committee. His guidance through multiple drafts and many long nights has been invaluable to the completion of this dissertation.

I would also like to thank Professors Peter Gerstoft, Nikolay A. Atanasov, and Sicun Gao for serving as members of my committee. I am grateful for their time, feedback, and support throughout my doctoral studies.

I further acknowledge Professor Mohammad Javad Khojasteh, Professor Nikolay A. Atanasov, Professor Eric Leitinger, and Mr. Bryan Teague for their collaboration and constructive comments on my work. Their insights greatly contributed to the development of my research. Many thanks also go to my lab colleagues: Mingchao Liang, Junsu Jang, Shaoxiu Wei, Luisa Watkins, Thomas Kropfreiter and Xuhong Li, as well as to my senior colleagues Kuan-Lin Chen, Chinghua Lee, and Harinath Garudadri. I deeply appreciate the academic discussions and camaraderie we shared.

I gratefully acknowledge the funding agencies that supported my work: the Under Secretary of Defense for Research and Engineering under Air Force Contract FA8702-15-D-0001, the National Science Foundation (NSF) under CAREER Award 2146261, and the Qualcomm Innovation Fellowship under Grant 492866.

Finally, I want to thank my family—my mother, father, and grandparents. Your love and support have given me the strength and confidence to pursue every decision I’ve made. I also want to thank my girlfriend, Ying Chi, for her love, care, and trust throughout this journey. Special thanks to my friends Yujie Zhu, Ruochuan Xu, and others for being wonderful companions along the way.

VITA

2013-2017 Bachelor of Engineering, Shanghai Jiao Tong University
2017-2021 Master of Science, University of California San Diego
2021–2025 Doctor of Philosophy, University of California San Diego

PUBLICATIONS

W. Zhang and F. Meyer, “Multisensor multiobject tracking with improved sampling efficiency,” *IEEE Trans. Signal Process.*, vol. 72, pp. 2036–2053, 2024.

W. Zhang, M. J. Khojasteh, N. A. Atanasov, and F. Meyer, “Importance sampling with stochastic particle flow and diffusion optimization,” *IEEE Signal Process. Lett.*, vol. 32, pp. 2304–2308, 2025.

W. Zhang, M. J. Khojasteh, and F. Meyer, “Particle flows for source localization in 3-D using TDOA measurements,” in *Proc. FUSION-24*, Venice, Italy, Jul. 2024. **Best Student Paper Award, 2nd Place.**

W. Zhang, B. Teague, and F. Meyer, “Active planning for cooperative localization: A Fisher information approach,” in *Proc. Asilomar-22*, Pacific Grove, CA, USA, Oct. 2022, pp. 795–800.

W. Zhang and F. Meyer, “Graph-based multiobject tracking with embedded particle flow,” in *Proc. IEEE RadarConf-21*, Atlanta, GA, USA, May 2021.

E. Leitinger, B. Teague, W. Zhang, M. Liang, and F. Meyer, “Data fusion for radio frequency SLAM with robust sampling,” in *Proc. FUSION-22*, Linöping, Sweden, 2022.

FIELDS OF STUDY

Major Field: Engineering (Electrical Engineering & Computer Science)

Studies in Signal Processing
Professors Florian Meyer

Studies in Applied Ocean Sciences
Professors Florian Meyer and Peter Gerstoft

Studies in Robotics and Automation
Professors Nikolay A. Atanasov and Sicun Gao

ABSTRACT OF THE DISSERTATION

Localization and Tracking in Challenging Environments: Particle Flow and Active Planning

by

Wenyu Zhang

Doctor of Philosophy in Electrical Engineering (Computer Engineering)

University of California San Diego, 2025

Professor Florian Meyer, Chair

Modern estimation theory for engineering applications is increasingly characterized by multimodal, high-dimensional state distributions and nonlinear, high-accuracy measurements with complex data associations. In this dissertation, we address the challenges of sampling-based multi-object tracking (MOT) and localization in complex 3D environments. I first focus on enhancing sampling efficiency in passive MOT scenarios. To this end, I incorporate particle flow (PF) techniques into a belief propagation (BP) framework for sequential estimation of an unknown number of states in the presence of measurement-origin uncertainty. In particular, I develop a “flow-induced” proposal PDF for importance sampling that consists of a weighted mixture of parallel flows performed for multiple candidate measurements. Particle flows are

either modeled using deterministic or stochastic spatio-temporal processes. The resulting method for MOT is evaluated with synthetic data and in a challenging passive acoustic undersea tracking problem where the 6-D state of multiple whales is tracked based on 1-D time-difference of arrival (TDOA) measurements provided by pairs of hydrophones on a volumetric array. This thesis also introduces an active planning method for cooperative localization. Here, sensor positions are dynamically controlled to maximize the information gain of future measurements. The control policy is derived by minimizing the trace of approximate inverse Bayesian Fisher information matrices (FIMs). Numerical results demonstrate that particle flow significantly improves sampling efficiency and estimation accuracy, while active planning leads to enhanced localization performance.

Chapter 1

Introduction

1.1 Bayesian Estimation and Engineering Applications

A key task in modern engineering is the estimation of the state of a system. Important applications include MOT in surveillance [1, 2], underwater localization in applied ocean sciences [3, 4, 5], simultaneous localization and mapping (SLAM) in machine perception [6, 7, 8, 9], information seeking and multiagent control in robotics [10, 11, 12], speech enhancement and restoration [13, 14], health-care monitoring [15], pollution source localization [16], chemical plume tracking [17], and risk assessment in quantitative finances [18]. In many applications, state estimation relies on signals recorded by a sensor that provide informations about the environment. Ubiquitous types of signals include acoustic [19] and radio-frequency (RF) waveforms [20], LiDAR reflections or “point clouds” [21], as well as camera images [22].

Different types of signals and their corresponding sensors have distinct advantages and limitations. Camera images typically contain the richest information, making them ideally suited for large-scale machine learning and driving progress in computer vision. They are the most straightforward signals for visual-based object detection and environmental monitoring. However, their quality can deteriorate significantly under poor lighting or visibility conditions. LiDAR is more robust across a range of weather and lighting environments, which makes it essential for autonomous driving. Yet, LiDAR sensors are generally expensive and not well-suited for mass-market applications. Radar [23] and RF antennas serve as the foundation of wireless

communication and are widely used for distance measurement and localization due to their high-frequency characteristics. With mobile phones, routers, and base stations now ubiquitous, RF signals offer great potential for large-scale sensing and localization. Although RF signals carry less detailed information—often limiting them to identity recognition in certain frequency band—their flexibility makes them well-suited for localization and navigation, particularly in indoor environments with non-line-of-sight (NLOS) conditions or high-mobility scenarios such as UAV tracking. Sonar [24], by contrast, is primarily used for underwater detection and localization. RF signals degrade rapidly in water, and visual sensors are ineffective in the absence of light in deep-sea environments. Even with artificial lighting, light cannot propagate far underwater. Sonar remains the best choice for underwater applications because low-frequency acoustic waves travel long distances through water. Therefore, selecting the appropriate sensing modality depends on the specific engineering task and the environmental conditions.

In this thesis, I address localization and tracking for environments where global navigation satellite systems (GNSS) such as GPS[25, 26] have limited performance or are generally unavailable. Important environments include indoor, aerospace, and underwater. The localization task focuses on detecting an unknown number of objects and estimating their positions in uncertain environments. During localization, the objects are typically assumed to be static. In contrast, the tracking task aims to sequentially estimate object positions and motions over time, given a series of measurements sampled at specific time intervals. In standard GPS-based systems, the position of a target relative to known landmarks or signal patterns is determined by remote sensing satellites. For more accurate and robust navigation, measurements from inertial measurement units (IMUs) or odometers can be integrated and fused with GNSS data [27]. However, remote RF communications are severely attenuated underwater and can degrade or be blocked in indoor environments. Additionally, in scenarios such as tracking a hostile aircraft or missile, the target is external to the GNSS system used by the detector and does not cooperate with the navigation infrastructure.

Measurements, which contain information about the system state to be estimated, are

typically derived from signal samples (e.g., image frames or the impulse response of RF signals) using signal and image processing methods. Although visual estimation has been extensively studied for tracking object motion and mapping environments in the absence of GNSS (e.g., optical flow [28], monocular camera localization [29], or multi-camera tracking [30]), this thesis focuses on RF and acoustic signal models for measurement generation, primarily using radar and sonar sensors. Compared to cameras, radar offers significant advantages for indoor localization: it can detect motion through walls and obstacles, operate in low-light or dark conditions, and provide velocity information, all while preserving privacy since it does not capture visual images. These features make radar ideal for environments where visual privacy is a concern. Radar arrays are also the preferred choice for tracking fast-moving targets such as aircraft and missiles, which are too fast for standard cameras due to limitations in frame rate and exposure time. Another drawback of cameras is the difficulty of accurately estimating depth from 2D images. For underwater localization, sonar is the preferred option due to its ability to propagate reliably through water. However, because radar and sonar signals are relatively low-resolution and susceptible to multipath interference, they must be carefully preprocessed and denoised [4]. Signal preprocessing steps typically include modulation and digital resampling [31, 32] to extract useful measurements. These measurements carry state information about the targets of interest. For localization and tracking tasks, this state information often includes position, velocity, acceleration, and steering angle. Depending on the signal type, commonly used measurements include time of arrival (TOA), time difference of arrival (TDOA) [33], direction of arrival (DOA) [32], frequency difference of arrival (FDOA), and joint angle-delay estimation (JADE) [34]. TDOA measurements are typically extracted from sonar or acoustic signals, while DOA and JADE are used with RF signals on radar and antenna arrays. Using these measurements—along with any available prior information about the state—we aim to estimate or infer the ground truth with minimal estimation error.

Promising approaches to indoor localization include ranging based on received signal strength (RSS) [35] or ultra-wideband (UWB) [36, 37] measurements. In UWB-based local-

ization, non-line-of-sight (NLOS) conditions can degrade accuracy and robustness. Multipath-assisted indoor navigation [38] addresses this limitation by incorporating floorplan information and leveraging additional localization cues provided by NLOS paths and multipath components, resulting in improved navigation performance. For underwater localization, time-difference-of-arrival (TDOA) measurements can be obtained through spectrogram analysis of the cross-correlation of acoustic signals [4]. In tracking applications, remote sensing devices such as radar or sonar are used to determine the positions of moving objects over time. The position of the sensing device is typically assumed to be perfectly known if it is fixed, or irrelevant if the external object or phenomenon is tracked relative to the sensor’s own position [1].

1.2 State-of-the-Art Techniques for Localization and Tracking

In this section, we focus on state-of-the-art estimation methods for localization and tracking and discuss open challenges associated with these techniques.

1.2.1 Tracking with Sequential Bayesian Estimation and Fusion

The core task of tracking is to estimate the state of one or more mobile objects over time. Sequential Bayesian estimation can be formulated as a Markov chain, where the estimate at each time step depends on the previous estimate and the current measurement. Various filtering algorithms have been developed for estimation at each time step. The Kalman filter (KF) [39] provides a theoretically optimal estimate for linear Gaussian problems. For nonlinear systems, the extended Kalman filter (EKF) [40, 41] applies linearization of the measurement model around the prior mean. A more general approach is the particle filter [42, 43], which represents the probability density function (PDF) of the state using a set of weighted particles. Distributed particle filters are especially appealing for tracking in arbitrary nonlinear and non-Gaussian systems; see, for example, [44, 45]. However, particle filters often require a large number of particles for high-dimensional problems [46]. The unscented Kalman filter (UKF) [47], also

known as the sigma-point filter, is another sequential Bayesian method designed for nonlinear systems. It generally provides better performance than the EKF while being less computationally intensive than the particle filter.

1.2.2 Multisensor MOT with Graph-based BP

For multisource localization or MOT, sensors such as radar and sonar receive signals transmitted or reflected by objects. Of particular interest are multisensor MOT problems, where inexpensive sensors are employed, and due to the underdetermined nature of sensor measurements, object tracking in Cartesian coordinates is not feasible using a single sensor. In such cases, multiple sensors combined with advanced MOT methods are required.

Traditional approaches to MOT include probabilistic data association (PDA) [48], multiple hypothesis tracking (MHT) [49], and methods based on random finite sets (RFS) [50, 51, 52, 53]. Most of these classical methods suffer from computational complexity that grows combinatorially with both the number of measurements and the number of objects. Note that the dimensionality of the total state (comprising all objects) increases with the number of objects. Factorizing the joint posterior probability density function (PDF) is significantly more complex than in standard sequential estimation problems and is often described using a graphical model [54, 55, 56]. For general loopy factor structures, an iterative message-passing scheme can be employed to approximately compute the marginalizations required for Bayesian inference [57]. Under mild assumptions, the factor graph [57, 56] representing the joint posterior PDF reflects object-to-object and object-to-measurement dependencies while improving computational scalability.

Sequential Bayesian estimation without data association (DA) uncertainty in distributed tracking is a case where the joint posterior PDF forms a tree-structured factor graph. Gaussian BP [58] and nonparametric belief propagation (BP) [59] are two variants of BP that extend the Kalman filter (KF) [39] and the particle filter [60], respectively, to general factor graph structures, depending on how variable distributions are modeled and messages are evaluated on

the factor graph.¹ Similar to the comparison between the KF and particle filter, Gaussian BP assumes a linear measurement model with Gaussian-distributed variables, while nonparametric BP is more general and handles nonlinear, non-Gaussian systems. Since measurement models in multisensor MOT are typically nonlinear, nonparametric BP is preferred over Gaussian BP in most scenarios. Message-passing algorithms have been proposed in [59] and [61] for localizing static and mobile objects, respectively. As an alternative to BP, a message-passing algorithm based on the mean-field approximation is presented in [62, 63] for static source localization.

In real-world applications, tracking multiple moving objects is often complicated due to several challenges:

1. The number of objects is unknown and varies over time.
2. The source of each measurement is uncertain and not all measurements are necessarily related to the objects or phenomena of interest .
3. The object state is high-dimensional and sensor measurements follow a nonlinear model .
4. Sensor measurements follow a nonlinear undetermined model, and due to the undetermined nature of the model, multiple sensors are required.

Challenges 1 and 2 are addressed by modern MOT methods [64, 65, 48, 66, 50, 67, 49, 68, 69, 70, 71, 72, 73, 53]. Of particular interest in this thesis is a recently proposed method [64, 65] based on the framework of factor graphs and the BP. The method is highly scalable in the number of objects, number of measurements, and number of sensors. It uses particle-based computations to calculate messages that, due to nonlinearities in the system model, can not be evaluated in closed form. Challenges 3 and 4 will be addressed in Chapters 3 and 4

1.2.3 Curse of Dimensionality and PFI

A major issue for sampling-based estimation—especially in the era of high-accuracy sensors—is particle degeneracy [46]. This phenomenon arises because predicted PDFs are often

¹KFs are parameter-based, relying on the mean and covariance, whereas particle filters are sampling-based and rely on Monte Carlo approximations.

used as proposal distributions for sampling. However, predicted PDFs can differ substantially from the posterior PDFs, making this strategy inefficient; few, if any, sampled particles end up representing the posterior accurately. The problem worsens in high-dimensional settings or when measurements have low variance. In particular, the likelihood function becomes increasingly "peaked" under such conditions, making it unlikely that particles drawn from the prior fall within regions of high likelihood.

Some mitigation strategies include using a very large number of particles or applying regularization techniques [74, 60, 75]. A more principled solution is to design proposal distributions that closely resemble the posterior [42, 76, 60]. However, finding a proposal that is both easy to sample from and similar to the posterior remains a difficult task. Adaptive importance sampling (IS) techniques [77, 78] have been introduced to improve efficiency. For example, auxiliary particle filters use a delayed resampling strategy to increase the number of particles with significant weights after IS [79]. This method improves sampling efficiency but relies on a prediction step, which may not be available when new object states appear in multi-object tracking (MOT).

Multiple particle filtering [80], similarly to a particle-based implementation of BP, improves sample efficiency by exploiting factorization in the statistical model [77]. However, its effectiveness is limited to models with suitable conditional posterior factorizations. Another general approach for nonlinear sequential Bayesian estimation involves integrating sequential Markov chain Monte Carlo (SMCMC) methods into particle filters [81, 82], though these are often computationally prohibitive in high-dimensional spaces.

An alternative to improve sampling efficiency is to use the update step of an unscented Kalman filter (UKF) [47] to construct a Gaussian proposal PDF for particle filtering. The resulting unscented particle filter [83] incorporates a Gaussian mixture representation of predicted and posterior PDFs. However, to the best of our knowledge, this method has not yet been extended to problems involving multiple unknown objects (MOU) and an unknown number of states to be estimated.

PFI [84, 85, 86, 87, 88] has recently gained attention [89] as a promising strategy for nonlinear estimation. PFL addresses particle degeneracy by actively transporting particles from the predicted (or prior) distribution toward regions of high likelihood in the posterior.²

Connections between PFI and variational inference have recently been explored [90], linking it to particle-based variational inference methods such as Stein variational gradient descent [91] and diffusion-based approaches [92, 93].

In PFL, a homotopy function defines a smooth transformation from the predicted PDF to the posterior PDF. A partial differential equation (PDE) for particle velocity is then derived by combining this homotopy with the Fokker–Planck equation. This PDE is discretized and used to define a transport equation that governs the migration of particles. After migration, the particle set represents the posterior distribution.

There are two main variants of PFL: the exact Daum–Huang (EDH) filter and the localized EDH (LEDH) filter. In the EDH filter, the PFL equations are computed once using the mean of all particles, while in the more computationally intensive LEDH filter, they are computed individually for each particle. PFL has demonstrated superior tradeoffs between performance and computational complexity compared to other approaches that aim to improve sampling efficiency [89]. Contrary to other particle filtering methods, PFL is highly parallelizable [85, 86, 87] and therefore well-suited for real-time implementation on graphical processing units (GPUs).

Traditional PFI methods avoid IS and can only provide an approximate representation of posterior PDFs in general nonlinear systems [84, 85, 86, 87, 88]. Nevertheless, these “proposal-free” methods often lead to accurate estimation results at a significantly reduced computational complexity compared to bootstrap particle filter (BPF) [89]. Recently, it has been shown that PFI can be described by an invertible mapping and can thus be used as a measurement-driven proposal PDF for IS [89]. The resulting invertible PFI filter [89] is an asymptotically optimal approach to nonlinear filtering that avoids particle degeneracy and can provide accurate estimation results in

²Traditional resampling methods also move particles but do so passively—only those located in high-likelihood regions survive resampling, often leaving too few particles to represent the posterior adequately.

high-dimensional and nonlinear problems.

A significant limitation of the PFI filter presented in [89] is its assumption that the prior or predicted PDFs are Gaussian. This makes it unsuitable for problems involving multimodal PDFs. For cases where the measurement noise follows a Gaussian mixture PDF, [94] introduced the Gaussian sum PFI filter. In this method, the means of the Gaussian mixture components are updated using a procedure similar to the localized exact Duam and Huang (LEDH), while the covariance matrices are updated in parallel by extended Kalman filters. An extension to the case where both the driving noise and the measurement noise follow Gaussian mixture PDFs was proposed in [95], where invertible flow is used for particle weight updates in an IS step.

In [90], a variational formulation of the particle flow filter is presented, where the transient gradient is interpreted as a time-scaled trajectory of the Fisher–Rao gradient flow minimizing the Kullback–Leibler divergence. This variational approach eliminates the Gaussian assumptions on the prior and likelihood, thereby providing the flexibility to address estimation problems with arbitrary prior–likelihood pairs.

For problems where both the driving noise and the measurement noise are multimodal, [96] combines invertible PFI with an sequential Markov chain Monte Carlo (SMCMC) method that relies on the Metropolis–Hastings algorithm. Specifically, a Metropolis–Hastings kernel is constructed using a PFI algorithm based on a GMM. However, these PFI approaches for multimodal PDFs [94, 95, 96] are not suitable for MOT, since they neither model measurement-origin uncertainty (MOU) nor handle an unknown number of states to be estimated.

For cooperative localization problems, methods based on invertible PFI have been proposed in [97, 98]. These methods are not suitable for more challenging multiobject tracking problems, as they assume no measurement origin uncertainty, a known number of states, and unimodal posterior PDFs. A variant of the PFI filter tailored to MOT was presented in [52], where the EDH and LEDH variants of the single-sensor δ -Generalized Labeled Multi-Bernoulli filter [99] are combined with invertible flow. However, these approaches are not applicable to multiobject tracking scenarios with measurements provided by multiple sensors.

1.2.4 Cooperative Localization and Information-Seeking Control

In the aforementioned multisensor MOT problem, the objects are assumed to be detectable at all sensors so that all objects are filtered in a centralized way. In addition, the positions of sensors are assumed to be well known and configured in geometry to ensure an accurate estimation. However, this is not always the case, especially for sensing in a mobile network. In such mobile networks, sensors are often referred to as cooperative agents (CAs). For estimation, each CA measures quantities about its neighboring CAs as well as external objects relative to its own position (e.g., involving distances or angles). By cooperating with other CAs and exchanging position-related information, each CA is able to estimate its own position (i.e., cooperative localization) while providing measurements of external objects to be tracked. At each CA, estimates of the object states are cooperatively calculated from all CA measurements in a distributed way (i.e., distributed tracking). A key challenge in tracking using wireless sensor networks is the dissemination of data in the network [2]. While most existing work on estimation methods for cooperative localization assumes that CAs in the network are static, a variety of emerging networks, e.g., in the context of the Internet of Things (IoT) [8], search-and-rescue [100], and underwater surveillance [101] and unmanned aerial vehicles (UAV) application [102], consist of mobile CAs. A smart *control* (moving CAs to different positions or adapting sensor characteristics) can be expected to lead to more informative measurements and, consequently, an improved estimation performance for both localization and tracking. Since the estimation tasks involve the measurements of all CAs, the smart controller moving and adapting the individual CAs needs to be derived from a global objective function and implemented in a distributed way to be effective. This calls for the development of controlled networked sensing approaches for cooperative localization.

In controlled networked sensing, a control policy is used to maximize the information gained from observations with respect to the unknown states of interest. The use of information measures for the control of the movement of a single CA or a network of CAs was introduced

in [103] and [104], respectively. Here, the CAs' control objective is to maximize the joint information about a global state that is carried by the measurements of all CAs about a global state. Suitable measures of information include negative posterior entropy [105], mutual information (MI) [105], and scalar-valued functions of the Fisher information matrix (FIM) [41]. In particular, the determinant, trace, and spectral norm of the FIM are considered in [106], where the control objective is to maximize the information related to the positions of the CAs and of an object. The maximization of negative posterior entropy is considered in [107, 108, 102, 109, 110, 111]. In [108], a central controller steers CAs with known positions along the gradient of negative posterior entropy with the goal of optimally sensing and estimating a static global state. A distributed solution for global state estimation is proposed in [107] based on a pairwise neighboring-CAs approximation of MI and in [102, 109] by using a consensus algorithm. Methods that rely on negative posterior entropy or MI are suitable for nonlinear and non-Gaussian system models but perform computationally intensive operations based on particle representations of probability distributions. In case the unknown states are high-dimensional, due to the curse of dimensionality, these particle-based operations are unfeasible. FIM-based methods, on the other hand, are more computationally efficient but rely on Gaussian representations of probability distributions that are typically less accurate [106, 111]. In [110], BP-based cooperative localization is combined with myopic information-seeking control of some CAs in the network. Here, information is quantified by the negative conditional differential entropy [105] of all unknown states conditioned on all measurements collected by the network. For each controlled node, a myopic control policy based on partial derivatives of the negative conditional differential entropy is developed. The computation of partial derivatives is performed using Monte Carlo integration and is thus very costly. In addition, the resulting gradient-based myopic control policy is unsuitable for receding horizon (RH) control [111].

For myopic control, the control policy depends only on the current state of CAs and objects. Due to the uncertainty of object states, a myopic control policy can lead to highly suboptimal control decisions that also negatively influence estimation and control at future

timesteps. This “getting stuck” at a local minimum often happens at initial timesteps where the localization uncertainty of the network is significant. To avoid local minima, RH control [112] uses model-based prediction of future costs within a sliding window that consists of multiple timesteps. The resulting information-seeking control over a receding planning horizon becomes a tree-structured path searching and planning problem. To improve searching efficiency, Dijkstra’s or A* algorithms can be used. One can also train a heuristic function as a predicted cost from current state to the goal. If the model-based policy is biased, a potential data-based method, such as reinforcement learning, can be used to keep optimizing the policy [113, 114].

1.3 Motivation, Contribution and Outline

As introduced in Section 1.2.3, the curse of dimensionality is a pervasive challenge in high-dimensional estimation problems, including belief propagation (BP)-based approaches for localization and tracking. As discussed in Section 1.2.2, nonparametric BP, also known as importance sampling (IS)-based BP, is more general and suitable for nonlinear, non-Gaussian problems. However, a key bottleneck for online estimation using IS-based methods—especially when working with highly informative measurements from modern sensors—is low sampling efficiency. This inefficiency arises primarily because a simple prior PDF is often used as the proposal distribution for sampling particles. As we will demonstrate later in this thesis, traditional sampling-based estimation methods that perform adequately in 2D scenarios break down in 3D settings due to severe particle degeneracy. Thus, sampling-efficient estimation techniques are crucial for state-of-the-art localization and tracking.

The key to improving sampling efficiency is to construct a proposal distribution that is sufficiently close to, but heavier-tailed than, the true posterior PDF [115, 116]. PFL is a well-known method for efficiently migrating particles from an uninformative prior to the posterior. Although PFL has been shown to overcome particle degeneracy in a variety of nonlinear and high-dimensional problems [84, 85, 86, 117], it does not offer asymptotic optimality guarantees

when applied directly to the particles [89]. In contrast, IS is asymptotically optimal under relatively weak conditions [115, 116]. As a result, recent work has explored the use of PFL to generate proposal distributions within the IS framework [89, 94, 95, 96]. However, most of this research has focused on improving sampling efficiency and estimation accuracy in single particle filters.

In contrast, improving sampling efficiency in a BP framework—while maintaining accurate belief approximations, especially under data association (DA) uncertainty in complex multi-object tracking (MOT) problems—remains an open challenge. Furthermore, information-seeking control for cooperative localization calls for robust active planning algorithms, particularly using receding horizon (RH) control policies, to continually enhance estimation accuracy. To the best of our knowledge, active planning, i.e., RH information-seeking control, in cooperative localization settings remains an open research problem.

In this thesis, we propose computationally feasible and sampling-efficient estimation algorithms for 3D localization and multi-object tracking (MOT). In Chapter 2, we formulate the MOT problem using a statistical framework and graphical model representation. Chapters 3 and 4 present our proposed 3D MOT algorithms with enhanced sampling efficiency, incorporating deterministic and stochastic particle flow filtering (PFL), respectively. Chapter 5 focuses on receding horizon (RH) information-seeking control, using the Fisher Information Matrix (FIM) as a heuristic. An outline of the thesis and a summary of the main contributions are provided below.

- In **Chapter 2**, we formulate our MOT problem as a statistical Bayesian estimation problem. We then introduce a scalable graph-based BP method for data association (DA) uncertainty. We also present how to approximate BP operations by IS.
- In **Chapter 3**, 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. Our approach performs BP-based message passing on the factor graph

for scalable multisensor MOT. To improve sampling efficiency, we embed invertible PFI for IS and further use it as a “flow-induced” proposal PDF for BP computations. For the evaluation of particle weights, invertible PFI 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, we make use of a GMM representation that is known to be asymptotically optimal [118]. A general proposal PDF that takes MOU into account and consists of a mixture of PDFs related to different PFIs is developed. In particular, since particles are migrated towards regions of high likelihood, an accurate approximation of BP messages with a relatively small number of particles is obtained.

We further demonstrate that the proposed multisensor MOT can outperform reference methods based on conventional (“bootstrap”) [60] and unscented particle filtering [83] 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 [64] 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 PFI.

- In **Chapter 4**, we focus on design of stochastic PFI for IS to improve performance in localization and tracking problems. The stochastic PFI outperforms deterministic PFI for improving the transient dynamics. In some applications, their random nature can avoid numerical issues, referred to as stiffness of the flow [119]. By using stochastic differential equation (SDE) to model the particle dynamics, stochastic PFI introduces a process noise modeled as a Brownian motion, the covariance matrix of which is called diffusion. However, using a general form of stochastic PFI for IS and further particle evaluation is still an open problem. We start with an existing popular stochastic PFI called Gromov’s flow. Then we introduce a IS method using general stochastic flows that include

optimizable diffusion terms. Our method makes it possible to evaluate a “flow-induced” proposal PDF after the kernel parameters have been migrated by stochastic PFI. The resulting IS framework is combined with a GMM and used within a BP framework for the detection and localization of an unknown number of sources in 3-D. Compared to conventional stochastic PFI without IS, the resulting processing step is asymptotically optimal. Within our method, it is possible to optimize the diffusion matrix that describes the diffusion term of the ordinary differential equation (ODE) to improve the accuracy computational complexity tradeoff. Our simulation results in a highly nonlinear 3-D source localization scenario showcase a reduced stiffness of the ODE and an improved estimating accuracy compared to state-of-the-art deterministic and stochastic PFIs.

- In **Chapter 5**, we introduce active planning for cooperative localization networks that aims at increasing the information gain of future measurements. Contrary to [110], which develops a control policy based on negative conditional differential entropy, we propose a control policy that minimizes traces of approximate inverse Bayesian FIMs [120]. In this way, costly Monte Carlo integration can be avoided, and the computational complexity related to a myopic control, i.e., a control strategy with a planning horizon of one timestep, can be strongly reduced. Due to this reduced computational complexity, we can develop a RH control, i.e., a control strategy with a planning horizon that consists of multiple future timesteps. In particular, we develop a cost function that accumulates Bayesian FIM contributions and state the tree search problem that results from cost function minimization. Approximations for solving the resulting tree-search problem efficiently are also discussed.
- In **Chapter 6**, we summarize our work with conclusion and suggest possible directions for future research.

Chapter 2

Background

In general, MOT can be formulated as a multivariable estimation problem. In a typical machine perception system, as illustrated in Fig. 2.1, a sequence of sensing data (e.g., images, LiDAR, or radar) is processed to perform object detection, which generates either point measurements [121] or extended measurements [122]. These measurements can be represented by a conditional likelihood PDF, referred to as the measurement model.

In this chapter, we review a scalable Bayesian probabilistic estimation method for MOT, formulated within a probabilistic graphical model framework. Specifically, we derive the posterior PDF of the state sequences of an unknown number of objects given a batch of measurements over discrete time steps. We then construct a factor graph model in which factorization and marginalization of the multivariate posterior PDF can be carried out efficiently. This formulation also enables the quantification of DA uncertainty in a fully probabilistic manner.

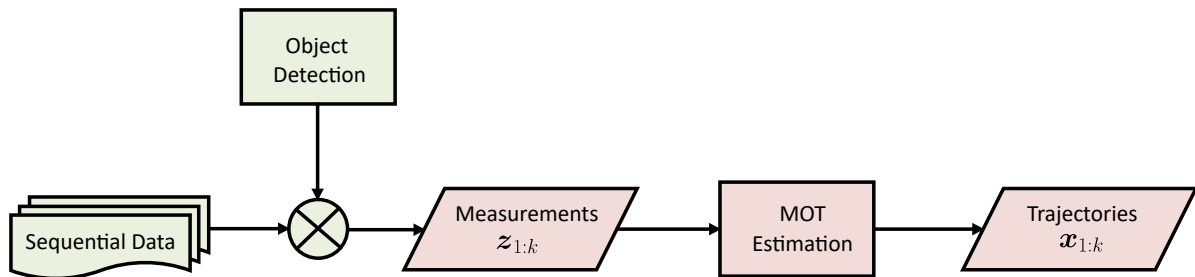


Figure 2.1. Diagram of machine perception. The block in red is the topic we focus in this work.

2.1 Bayesian Estimation

In this section, we formulate the model of Bayesian estimation for MOT. As introduced in the previous section, we want to estimate the state of an unknown number of objects from the associated measurements. We denote $\mathbf{x} \in \mathbb{R}^N$ the state of an object we want to estimate, and $\mathbf{z} \in \mathbb{R}^M$ as the associated measurement. Then we want to get an optimal estimate $\hat{\mathbf{x}}$ given $\mathbf{z}$. In statistics, considering the uncertainty of $\mathbf{x}$, if we know the posterior PDF of $\mathbf{x}$ given $\mathbf{z}$, i.e., $p(\mathbf{x}|\mathbf{z})$, we can derive the estimate as the mean of $\mathbf{x}$ based on $p(\mathbf{x}|\mathbf{z})$, i.e., $\hat{\mathbf{x}} = \mathbb{E}_{p(\mathbf{x}|\mathbf{z})}(\mathbf{x}) = \int_{-\infty}^{\infty} \mathbf{x} p(\mathbf{x}|\mathbf{z}) d\mathbf{x}$ as well as the uncertainty of $\mathbf{x}$ as the covariance matrix $\mathbf{P} = \text{Cov}_{p(\mathbf{x}|\mathbf{z})}(\mathbf{x}) = \int_{-\infty}^{\infty} p(\mathbf{x}|\mathbf{z}) (\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^T d\mathbf{x}$.

Bayes' rule [123] is probably the most famous theorem for statistical estimation and learning. Bayes' rule describes the PDF of a variable, based on prior knowledge of the variable as well as the likelihood of the given measurement/observation associated with it. Denote the prior PDF of a variable as $p(\mathbf{x})$, the likelihood of the measurement as $p(\mathbf{z}|\mathbf{x})$, and the posterior PDF of the variable as $p(\mathbf{x}|\mathbf{z})$. Bayes' rule can be written as

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})}, \quad (2.1)$$

And if there is no knowledge of the prior PDF of the variable, the Bayes' rule coordinates with the maximum likelihood estimation (MLE), of which the prior is a uniform distribution. Given Bayes' rule, i.e., the posterior PDF of the variable is well-defined, we can easily get the minimum mean-square error (MMSE) estimate of the variable $\hat{\mathbf{x}}_{\text{MMSE}}(\mathbf{z}) = \int \mathbf{x} p(\mathbf{x}|\mathbf{z}) d\mathbf{x}$ or the maximum a posteriori (MAP) estimate $\hat{\mathbf{x}}_{\text{MAP}}(\mathbf{z}) = \arg \max_{\mathbf{x}} p(\mathbf{x}|\mathbf{z})$. In practice, the likelihood PDF is given by the measurement model $\mathbf{z} = h(\mathbf{x}, \mathbf{v})$, where $\mathbf{v}$ is the measurement noise.

There are two mainstreams of solving the optimal estimating problem. One is the parametric method, and the other is the nonparametric method. The parametric method tries to get the analytical solution of the posterior PDF, i.e., mean and covariance. The benefit is

obvious, it's concise and easy for calculation in vector space. This method is preferred for certain cases. A typically simplified case occurs when the prior PDF of the variable is Gaussian, i.e. $p(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{P})$ and the likelihood of the measurements is linear with additive Gaussian noise, i.e. $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$ where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. In this case, the posterior PDF of the variable is also Gaussian. The algorithm to get the mean and covariance of the posterior PDF is given by the well-known Kalman filter (KF) [124]. The estimate is given as $\hat{\mathbf{x}} = \boldsymbol{\mu} + \mathbf{K}(\mathbf{z} - \mathbf{H}\boldsymbol{\mu})$, where $\mathbf{K} = \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}$ is the Kalman gain. It's proved that the KF provides the optimal estimate of the posterior PDF in this linear Gaussian case. If the prior PDF is non-Gaussian or the measurement model is not linear, one can still model the PDF by Gaussian and perform "first-order" approximation of the nonlinear measurement model, i.e., calculating the Jacobian matrix of the measurement function at the mean of the state as the linear measurement matrix of KF $\mathbf{H} \approx \left. \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{v})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\boldsymbol{\mu}}$, with sacrifices of the accuracy. This is known as the extended Kalman filter (EKF). There are other variants of the KF to handle the nonlinear measurement model. For example, the unscented Kalman filter (UKF) [47] use sigma points as multiple supports. When propagated through the true nonlinear measurement model, the sigma points capture the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any nonlinearity [125]. Another analytic method is the information filter [126], which is a dual of the KF in the information space, where an information matrix is propagated as the inverse of the covariance matrix of the state vector.

The nonparametric method, on the other hand, tries to represent the posterior PDF by a set of weighted samples. The benefit of the nonparametric method is that it's more general for theoretically any class of posterior PDF, and can handle the non-Gaussian nonlinear problem, where an analytic solution is intractable to achieve. The nonparametric method typically relies on IS [42, 76, 60]. If we sample a set of particles i.i.d. from an arbitrary *proposal* PDF i.e., $\{\mathbf{x}^{(i)}\}_{i=1}^{N_p} \sim q(\mathbf{x})$ in a Monte Carlo simulation, we can approximate the posterior PDF by this

sample set with weights appended, i.e., $\{(\mathbf{x}^{(i)}, w^{(i)})\}_{i=1}^{N_p} \sim p(\mathbf{x}|\mathbf{z})$, with

$$w^{(i)} = \frac{p(\mathbf{z}|\mathbf{x}^{(i)})p(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}. \quad (2.2)$$

Here, we embed the Bayes' rule to replace the posterior PDF at samples with prior times likelihood. After normalizing the sum of weights to 1, the MMSE estimate of the variable is then $\hat{\mathbf{x}}_{\text{MMSE}}(\mathbf{z}) = \sum_{i=1}^{N_p} w^{(i)} \mathbf{x}^{(i)}$. Theoretically, the weighted sample set can give an arbitrary close approximation to the true posterior PDF with arbitrary proposal PDF, if only N_p is large enough. For easy sampling, one usually use the prior PDF as the proposal and hence the weights are just the measurement likelihood at the sample, and this is the well-known bootstrap particle filter [60].

2.2 Sequential Bayesian Estimation

In most engineering applications, measurements are generated in time series, i.e., $\mathbf{z}_k, k \in \{1, \dots, K\}$ or $\mathbf{z}_t, t \in (0, T]$, either in discrete or continuous time. And we want to estimate the object state at time k or t by the measurements so far, of which the PDF can be represented as $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ or $p(\mathbf{x}_t|\mathbf{z}_{0:t})$. In general, the discrete way is more challenging for non-continuous transformation and most continuous case can be derived as a special case of discrete case, not to say that most digital devices rely on discrete sampling of signals in the time domain. Hence we focus on a discrete time system in the remainder of the thesis, i.e., the estimation of the joint posterior PDF $p(\mathbf{x}_{1:k}|\mathbf{z}_{1:k})$. We can see that the complexity of this joint PDF is exponential to k . To reduce computational complexity, a key assumption of sequential Bayesian estimation is that the estimate of the current state depends only on the state of the last time step and the measurement of the current time step, as illustrated in Figure 2.2. Hence the system model is given by the state transition and measurement function as

$$\mathbf{x}_k = g(\mathbf{x}_{k-1}, \mathbf{w}) \quad (2.3)$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}) \quad (2.4)$$

Which is denoted as the Markov process or hidden Markov process [1]. The chained structure as Fig. 2.2 is denoted as the Markov chain. And indeed, the joint posterior can be decompose as a chain in that

$$p(\mathbf{x}_{1:k}|\mathbf{z}_{1:k}) = p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k})p(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k}) \quad (2.5)$$

$$\propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k-1}) \quad (2.6)$$

...

$$\propto p(\mathbf{x}_0)\prod_{k'=1}^k p(\mathbf{z}_{k'}|\mathbf{x}_{k'})p(\mathbf{x}_{k'}|\mathbf{x}_{k'-1}) \quad (2.7)$$

Note that the transformation from (2.5) to (2.6) is exactly Bayes' rule in that $p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_{1:k}, \mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})}$ and $p(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})p(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}$, where $p(\mathbf{z}_k|\mathbf{x}_k) \triangleq p(\mathbf{z}_k|\mathbf{x}_{1:k}, \mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{x}_{k-1}) \triangleq p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})$.

Various sequential estimation methods following system model defined by (2.3) and (2.4) are also called “filtering” algorithms. Filters generally achieve this through a two-step process: propagation and measurement update, corresponding to (2.3) and (2.4) respectively. Note that in Bayes' statistics, the state transition function $g(\cdot)$ provides relationship between the posterior knowledge of the last time step and the prior knowledge of the current time step by prediction. If the state transition model is linear as $\mathbf{x}_k = \mathbf{G}\mathbf{x}_{k-1} + \mathbf{w}$, where $\mathbf{w} \sim \mathcal{N}(0, \mathbf{W})$, and the posterior PDF is Gaussian, i.e. $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\boldsymbol{\mu}_{k-1}, \mathbf{P}_{k-1})$, the predicted prior of $\mathbf{x}_k$ is a Gaussian as $p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{P}_k)$ where $\boldsymbol{\mu}_k = \mathbf{G}\boldsymbol{\mu}_{k-1}$ and $\mathbf{P}_k = \mathbf{G}\mathbf{P}_{k-1}\mathbf{G}^T + \mathbf{W}$. For sampling-based estimation method, which fits a more general state transition model, we can propagate the samples from previous $\{(\mathbf{x}_{k-1}^{(i)}, \mathbf{w}_{k-1}^{(i)})\}_{i=1}^{N_p} \approx p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ to $\{(\mathbf{x}_k^{(i)}, \mathbf{w}_{k-1}^{(i)})\}_{i=1}^{N_p} \approx p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ in the prediction step by $\mathbf{x}_k^{(i)} = g(\mathbf{x}_{k-1}^{(i)}, \mathbf{w})$. By using the predicted posterior $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ as the prior PDF introduced in Section 2.1, parametric and nonparametric Bayesian methods can be

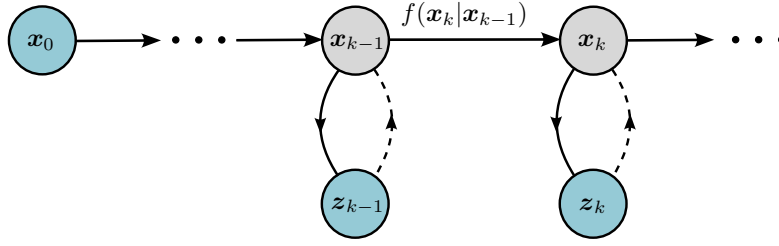


Figure 2.2. Sequential Bayesian estimation.

used for posterior estimate of $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ in a single time step. Specifically, KF has a solution as $\hat{\mathbf{x}}_k = \boldsymbol{\mu}_k + \mathbf{K}(\mathbf{z}_k - \mathbf{H}\boldsymbol{\mu}_k)$, where $\mathbf{K} = \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1}$, while particle filter updates particle weights as

$$w_k^{(i)} \propto p(\mathbf{z}_k | \mathbf{x}_k^{(i)}) w_{k-1}^{(i)} \quad (2.8)$$

Note that here for simplicity, we choose proposal $q(\mathbf{x})$ as the predicted posterior $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$. A resampling step [60] can be used to generate particles that have even weights, which improves the sample efficiency for the next time step.

2.3 Multiobject Tracking and Data Association

In Section 2.1 and 2.2, we assume that one measurement $\mathbf{z}_k$ is known to be generated by one object $\mathbf{x}_k$ at time k . However, for MOT, there are typically an unknown number of objects, and the object number changes by time. What's more, the measurements received at a sensor is typically not labeled so that we don't know which measurement corresponds to which object. In this section, we introduce the system model of MOT to make the problem tractable. To clarify, we make the following further assumptions [48]:

- (A₁) **Independency of object's state transition:** The multiobject state at time k (defined as an ordered or unordered list of all the single-object states $\mathbf{x}_k^{(j)}$, $j = 1, \dots, J_k$, where j is the label of objects) evolves according to a first-order Markov process. The single-object states $\mathbf{x}_k^{(j)}$ evolves independently. The evolution of state $\mathbf{x}_k^{(j)}$ is described by the single

object state-transition PDF $p(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)})$.

(A₂) **MOU**: The measurements $\mathbf{z}_k^{(m)}, m = 1, \dots, M_k$ (denote m the label of measurement) produced at time k have no order, i.e., they are randomly shuffled. At time k , each measurement labeled m originates from either an object or from a false positive (FP) (clutter measurement), and it cannot originate from more than one object simultaneously. In contrast, at time k , an object labeled j can generate at most one measurement labeled m . The prior probability that a measurement originates from an object (i.e., “object j is detected” at time k) is independent across j and defined by a known function of the object state $\mathbf{x}_k^{(j)}$, denoted as $p_d(\mathbf{x}_k^{(j)})$. Consequently, the probability that no measurement originates from object j (i.e., “object j is missed” at time k) is $1 - p_d(\mathbf{x}_k^{(j)})$. We assume that $0 \leq p_d(\mathbf{x}_k^{(j)}) < 1$.

(A₃) **Independency of FPs**: The number of FPs at time k is Poisson distributed with mean μ_{fp} , which is independent of the number of objects that are detected. The FPs at time k are i.i.d. with PDF $f_{\text{fp}}(\mathbf{z}_k^{(m)}) \neq 0$.

(A₄) **Conditional independency of object-originated measurements**: The object-originated measurements are conditionally independent of each other and also conditionally independent of all the clutter measurements. Furthermore, given that the measurement m originates from the object j , the conditional PDF of this measurement is defined by $f(\mathbf{z}_k^{(m)}|\mathbf{x}_k^{(j)})$.

In the remainder of this section, we will first discuss the concept of potential object (PO) states, then derive the statistical model of MOU, and finally review the BP messages that will later be calculated based on PFI in Chapter 3. Additional details can be found in [127].

2.3.1 Potential Object States

As in [64, 65], we consider MOT for an unknown, time-varying number of objects by introducing PO states. Let M_k be the number of measurements at time k . According to assumption (A₂), at time k , a new PO is introduced for each of the M_k observed measurements.

Then the maximum possible number of objects that generated a measurement up to time k is $J_k = J_{k-1} + M_k$, where J_{k-1} at discrete time $k-1 \geq 0$ is the maximum possible number of objects that have generated a measurement up to time $k-1$. All POs that have been introduced at previous time steps are referred to as legacy POs. Hence there are J_k legacy POs and M_k new POs at time k .

We also define the augmented state of PO $j \in \{1, \dots, J_k\}$ is given by $\mathbf{y}_k^{(j)} \triangleq [\mathbf{x}_k^{(j)\top} \mathbf{r}_k^{(j)}]^\top$, 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 $\mathbf{r}_k^{(j)} \in \{0, 1\}$ models the existence/nonexistence of PO j in the sense that PO j exists at time k if and only if $\mathbf{r}_k^{(j)} = 1$. For nonexistent POs, i.e., $\mathbf{r}_k^{(j)} = 0$, the state $\mathbf{x}_k^{(j)}$ is obviously irrelevant. Thus, all PDFs of augmented PO states $f(\mathbf{y}_k^{(j)}) = f(\mathbf{x}_k^{(j)}, \mathbf{r}_k^{(j)})$ can be expressed as $f(\mathbf{x}_k^{(j)}, 0) = f_k^{(j)} f_D(\mathbf{x}_k^{(j)})$, where $f_D(\mathbf{x}_k^{(j)})$ 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 $\underline{\mathbf{y}}_k^{(j)}$, $j = 1, \dots, J_{k-1}$ and by $\bar{\mathbf{y}}_k^{(m)}$, $m = 1, \dots, M_k$ the augmented state of a legacy PO and a new PO states, respectively. The single-object state transition PDF $f(\underline{\mathbf{y}}_k^{(j)} | \underline{\mathbf{y}}_{k-1}^{(j)})$ that models the motion and disappearance of objects and involves the probability of object survival p_s is presented in [65, Section VIII-C]. At time $k=0$, the prior augmented states $\mathbf{y}_0^{(j)}$ are statistically independent across POs j . Often no prior information is available, i.e., $J_0 = 0$. For new PO at time k , i.e., $\bar{\mathbf{y}}_k^{(m)} \triangleq [\bar{\mathbf{x}}_k^{(m)\top} \bar{\mathbf{r}}_k^{(m)}]^\top$, $m \in \{1, \dots, M_k\}$. Here, $\bar{\mathbf{r}}_k^{(m)} = 1$ means that measurement $\mathbf{z}_k^{(m)}$ originated from an object that never generated a measurement before, and $\bar{\mathbf{r}}_k^{(m)} = 0$ otherwise. After the observation, the joint measurement vector at time k is fixed and denoted as $\mathbf{z}_k \triangleq [\mathbf{z}_k^{(1)\top} \dots \mathbf{z}_k^{(M_k)\top}]^\top$. The total joint PO state at time k is denoted as $\mathbf{y}_k \triangleq [\underline{\mathbf{y}}_k^{(1)\top} \dots \underline{\mathbf{y}}_k^{(J_{k-1})\top}, \bar{\mathbf{y}}_k^{(1)\top} \dots \bar{\mathbf{y}}_k^{(M_k)\top}]^\top \triangleq [\mathbf{y}_k^{(1)\top} \dots \mathbf{y}_k^{(J_k)\top}]^\top$.

2.3.2 Measurement-Object Uncertainty

According to assumption (A₂)-(A₄) and definition of PO, each measurement can originate from one of the three sources: (i) a legacy PO, which represents an object that has generated at least one measurement before; (ii) a new PO, which models an object that generates a

measurement for the first time; and (iii) FP. For a legacy PO $j = 1, \dots, J_{k-1}$, a discrete and random “object-oriented” association variable $a_k^{(j)} \in \{0, 1, \dots, M_k\}$ is introduced. Here, $a_k^{(j)} = 0$ describes the event where no measurement originated from the object j and $a_k^{(j)} = m$, $m \in \{1, \dots, M_k\}$ describes the event where measurement $z_k^{(m)}$ was originated from the object. Then for each legacy PO $j = 1, \dots, J_{k-1}$, the joint PDF of z_k and $a_k^{(j)}$ conditioned on $\underline{x}_k^{(j)}$ is given by

$$\begin{aligned}
& f(\mathbf{z}_k, a_k^{(j)} | \underline{\mathbf{x}}_k^{(j)}, r_k^{(j)} = 1) \\
& \propto q(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)} = 1, a_k^{(j)}; \mathbf{z}_k) \\
& \triangleq \begin{cases} \frac{p_d(\underline{\mathbf{x}}_k^{(j)}) f(\mathbf{z}_k^{(m)} | \underline{\mathbf{x}}_k^{(j)})}{\mu_{fp} f_{fp}(\mathbf{z}_k^{(m)})}, & a_k^{(j)} = m \in \{1, \dots, M_k\} \\ 1 - p_d(\underline{\mathbf{x}}_k^{(j)}), & a_k^{(j)} = 0 \end{cases}, \quad \text{for } j = 1, \dots, J_{k-1}. \quad (2.9)
\end{aligned}$$

Here, we define a pseudo likelihood function $q(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)}, a_k^{(j)}; \mathbf{z}_k)$ and $q(\underline{\mathbf{x}}_k^{(j)}, 0, a_k^{(j)}; \mathbf{z}_k) \triangleq 1(a_k^{(j)})$, where $1(a)$ denotes the indicator function of the event $a = 0$ (i.e., $1(a) = 1$ if $a = 0$ and 0 otherwise).

We also introduce the “measurement-oriented” association variable $b_k^{(m)} \in \{0, 1, \dots, J_{k-1}\}$ for each measurement $m = 1, \dots, M_k$ to obtain a scalable and efficient message passing algorithm (see [128, 65] for details). The measurement-oriented association variable $b_k^{(m)}$ is $j \in \{1, \dots, J_{k-1}\}$ if measurement m originated from legacy PO j and zero if it originated from a FP or a newly detected PO. Furthermore, the new PO pseudo likelihood function $v(\bar{\mathbf{x}}_k^{(m)}, \bar{r}_k^{(m)}, b_k^{(m)}; \mathbf{z}_k^{(m)})$ reads

$$\begin{aligned}
& f(\mathbf{z}_k^{(m)}, b_k^{(m)} | \bar{\mathbf{x}}_k^{(m)}, \bar{r}_k^{(m)} = 1) \\
& \propto v(\bar{\mathbf{x}}_k^{(m)}, \bar{r}_k^{(m)} = 1, b_k^{(m)}; \mathbf{z}_k^{(m)}) \\
& \triangleq \begin{cases} 0, & b_k^{(m)} \in \{1, \dots, J_{k-1}\} \\ \frac{p_d(\bar{\mathbf{x}}_k^{(m)}) \mu_b f_b(\bar{\mathbf{x}}_k^{(m)}) f(\mathbf{z}_k^{(m)} | \bar{\mathbf{x}}_k^{(m)})}{\mu_{fp} f_{fp}(\mathbf{z}_k^{(m)})}, & b_k^{(m)} = 0 \end{cases}, \quad \text{for } m = 1, \dots, M_k. \quad (2.10)
\end{aligned}$$

and $v(\bar{\mathbf{x}}_k^{(m)}, 0, b_k^{(m)}; \mathbf{z}_k^{(m)}) \triangleq f_D(\bar{\mathbf{x}}_k^{(m)})$. Here, we model the birth of a new PO as a Poisson point process with mean μ_b and prior state PDF $f_b(\bar{\mathbf{x}}_k^{(m)})$.

2.3.3 Joint Posterior PDF and Factor Graph

We denote the “object-oriented” vector $\mathbf{a}_k = [\mathbf{a}_k^{(1)\top} \dots \mathbf{a}_k^{(J_k-1)\top}]^\top$ and “measurement-oriented” vector $\mathbf{b}_k = [\mathbf{b}_k^{(1)\top} \dots \mathbf{b}_k^{(M_k)\top}]^\top$. As introduced in [64, 65], the joint posterior PDF of $\mathbf{y}_{1:k}$, $\mathbf{a}_{1:k}$, and $\mathbf{b}_{1:k}$ conditioned on observed and thus fixed $\mathbf{z}_{1:k}$ can be obtained as

$$\begin{aligned}
& f(\mathbf{y}_{1:k}, \mathbf{a}_{1:k}, \mathbf{b}_{1:k} | \mathbf{z}_{1:k}) \\
& \propto \left(\prod_{j''=1}^{J_0} f(\mathbf{y}_0^{(j'')}) \right) \prod_{k'=1}^k \left(\prod_{j'=1}^{J_{k'-1}} f(\mathbf{y}_{k'}^{(j')} | \mathbf{y}_{k'-1}^{(j')}) \right) \\
& \quad \times \left(\prod_{j=1}^{J_{k'-1}} q(\mathbf{x}_{k'}^{(j)}, \mathbf{r}_{k'}^{(j)}, a_{k'}^{(j)}; \mathbf{z}_{k'}) \prod_{m'=1}^{M_{k'}} \Psi_{j,m'}(a_{k'}^{(j)}, b_{k'}^{(m')}) \right) \\
& \quad \times \prod_{m=1}^{M_{k'}} v(\bar{\mathbf{x}}_{k'}^{(m)}, \bar{\mathbf{r}}_{k'}^{(m)}, b_{k'}^{(m)}; \mathbf{z}_{k'}^{(m)}). \tag{2.11}
\end{aligned}$$

Here, $q(\mathbf{x}_k^{(j)}, \mathbf{r}_k^{(j)}, a_k^{(j)}; \mathbf{z}_k)$ and $v(\bar{\mathbf{x}}_k^{(m)}, \bar{\mathbf{r}}_k^{(m)}, b_k^{(m)}; \mathbf{z}_k^{(m)})$ are define in (2.9) and (2.10). The binary indicator function $\Psi_{j,m}(a_k^{(j)}, b_k^{(m)})$ checks association consistency of a pair of object-oriented and measurement-oriented variables $(a_k^{(j)}, b_k^{(m)})$ in that $\Psi_{j,m}(a_k^{(j)}, b_k^{(m)})$ is zero if $a_k^{(j)} = m, b_k^{(m)} \neq j$ or $b_k^{(m)} = j, a_k^{(j)} \neq m$ and one otherwise (see [128, 65] for details). The joint posterior in (2.11) can be represented by the factor graph in Fig. 2.3 for each time step k . A detailed derivation of this joint posterior is provided in [65, Section VIII-G].

2.3.4 Problem Formulation and Sum-Product Message Passing

At each time step $k \geq 1$, we consider the tracking of an unknown number of objects based on measurements $\mathbf{z}_{1:k}$. Object detection is performed by comparing the existence probability $p(r_k^{(j)} = 1 | \mathbf{z}_{1:k})$ with a threshold P_{th} , i.e., PO $j \in \{1, \dots, J_k\}$ is declared to exist if $p(r_k^{(j)} =$

factor graph in Fig. 2.3 and passing messages only forward in time. This makes it possible to efficiently calculate so-called beliefs $\tilde{f}(\mathbf{x}_k^{(j)}, r_k^{(j)}) \triangleq \tilde{f}(\mathbf{y}_k^{(j)})$, $j \in \{1, \dots, J_k\}$ which accurately approximate the marginal posterior PDFs $f(\mathbf{x}_k^{(j)}, r_k^{(j)} | \mathbf{z}_{1:k})$, $j \in \{1, \dots, J_k\}$ 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 $p_k^{(j)} \triangleq \tilde{p}(r_k^{(j)} = 1 | \mathbf{z}_{1:n})$ below a threshold P_{pr} are removed from the state space.

Next, we review the message passing procedure for a time step k based on Fig. 2.3. We will use the BP of factor graph [57] to derive the inference of messages. 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 [65, Section IX-A].)

(S₁) **Prediction:** At time k , the “prior messages” of legacy PO states are denoted as

$\alpha_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)}) = \alpha_k^{(j)}(\underline{\mathbf{y}}_k^{(j)})$. These message are computed by a prediction step

$$\alpha_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)}) = \sum_{r_{k-1}^{(j)} \in \{0,1\}} \int f(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)} | \underline{\mathbf{x}}_{k-1}^{(j)}, r_{k-1}^{(j)}) \times \tilde{f}(\underline{\mathbf{x}}_{k-1}^{(j)}, r_{k-1}^{(j)}) d\underline{\mathbf{x}}_{k-1}^{(j)} \quad (2.13)$$

where $\tilde{f}(\underline{\mathbf{x}}_{k-1}^{(j)}, r_{k-1}^{(j)})$ is the approximated beliefs of legacy PO's state at previous time step.

For future reference, we also introduce $\underline{p}_k^{(j)} = \int \alpha_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)} = 1) d\underline{\mathbf{x}}_k^{(j)}$ as the predicted probability of existence for each legacy PO. Note that $\underline{p}_k^{(j)} + \int \alpha_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)} = 0) d\underline{\mathbf{x}}_k^{(j)} = 1$.

(S₂) **Measurement evaluation:** For legacy POs, the messages $\beta_k^{(j)}(a_k^{(j)})$ passed from factor nodes $q(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)}, a_k^{(j)}; \mathbf{z}_k)$ to variable nodes $a_k^{(j)}$ are calculated as

$$\begin{aligned} \beta_k^{(j)}(a_k^{(j)}) &= \int q(\underline{\mathbf{x}}_k^{(j)}, 1, a_k^{(j)}; \mathbf{z}_k) \alpha_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, 1) d\underline{\mathbf{x}}_k^{(j)} \\ &\quad + 1(a_k^{(j)}) (1 - \underline{p}_k^{(j)}). \end{aligned} \quad (2.14)$$

Similarly, messages of new POs $\xi_k^{(m)}(b_k^{(m)})$ from factor nodes $v(\bar{\mathbf{x}}_k^{(m)}, \bar{r}_k^{(m)}, b_k^{(m)}; \mathbf{z}_k^{(m)})$ to variable nodes $b_k^{(m)}$ are calculated as

$$\xi_k^{(m)}(b_k^{(m)}) = \int v(\bar{\mathbf{x}}_k^{(m)}, 1, b_k^{(m)}; \mathbf{z}_k^{(m)}) d\bar{\mathbf{x}}_k^{(m)} + 1 \quad (2.15)$$

(S₃) **Probabilistic DA:** Next, probabilistic DA is performed by means of iterative BP message passing with input messages $\beta_k^{(j)}(a_k^{(j)})$, $j \in \{1, \dots, J_k\}$ and $\xi_k^{(m)}(b_k^{(m)})$, $m \in \{1, \dots, M_k\}$ (see [65, Section IX-A3] for details). After convergence, corresponding output messages $\kappa_k^{(j)}(a_k^{(j)})$, $j \in \{1, \dots, J_k\}$ and $\iota_k^{(m)}(b_k^{(m)})$, $m \in \{1, \dots, M_k\}$ are available for legacy POs and new POs, respectively.

(S₄) **Measurement update:** Here, for legacy POs, messages $\gamma_{k,s}^{(j)}(\underline{\mathbf{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)})$ passed from $q(\underline{\mathbf{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; \mathbf{z}_{k,s})$ to $\underline{\mathbf{y}}_{k,s}^{(j)}$ are calculated as

$$\gamma_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, 1) = \sum_{a_k^{(j)}=0}^{M_k} q(\underline{\mathbf{x}}_k^{(j)}, 1, a_k^{(j)}; \mathbf{z}_k) \kappa_k^{(j)}(a_k^{(j)}) \quad (2.16)$$

and as $\gamma_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, 0) = \gamma_k^{(j)} = \kappa_k^{(j)}(0)$.

For new POs, messages $\varsigma_k^{(m)}(\bar{\mathbf{x}}_k^{(m)}, \bar{r}_k^{(m)})$ passed from $v(\bar{\mathbf{x}}_k^{(m)}, \bar{r}_k^{(m)}, b_k^{(m)}; \mathbf{z}_k^{(m)})$ to $\bar{\mathbf{y}}_k^{(m)}$ are calculated as

$$\varsigma_k^{(m)}(\bar{\mathbf{x}}_k^{(m)}, 1) = v(\bar{\mathbf{x}}_k^{(m)}, 1, 0; \mathbf{z}_k^{(m)}) \iota_k^{(m)}(0). \quad (2.17)$$

and $\varsigma_k^{(m)}(\bar{\mathbf{x}}_k^{(m)}, 0) = \varsigma_k^{(m)} f_D(\bar{\mathbf{x}}_k^{(m)})$ with $\varsigma_k^{(m)}$ calculated as $\varsigma_k^{(m)} = \sum_{b_k^{(m)}=0}^{J_k-1} \iota_k^{(m)}(b_k^{(m)})$.

(S₅) **Belief update:** Finally, beliefs are calculated to approximate the posterior PDFs of POs.

For legacy POs, beliefs $\tilde{f}(\underline{\mathbf{x}}_k^{(j)}, \underline{r}_k^{(j)})$ approximating $f(\underline{\mathbf{x}}_k^{(j)}, \underline{r}_k^{(j)} | \mathbf{z}_{1:k})$ are obtained as

$$\tilde{f}(\underline{\mathbf{x}}_k^{(j)}, 1) = \frac{1}{\underline{C}_k^{(j)}} \alpha_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, 1) \gamma_k^{(j)}(\underline{\mathbf{x}}_k^{(j)}, 1) \quad (2.18)$$

and as $\tilde{f}(\mathbf{x}_k^{(j)}, 0) = \underline{f}_k^{(j)} f_D(\mathbf{x}_k^{(j)})$ with $\underline{f}_k^{(j)} = (1 - \underline{p}_k^{(j)}) \gamma_k^{(j)} / \underline{C}_k^{(j)}$. The constant $\underline{C}_k^{(j)}$ is given by $\underline{C}_k^{(j)} \triangleq \int \alpha_k^{(j)}(\mathbf{x}_k^{(j)}, 1) \gamma_k^{(j)}(\mathbf{x}_k^{(j)}, 1) d\mathbf{x}_k^{(j)} + (1 - \underline{p}_k^{(j)}) \gamma_k^{(j)}$.

Similarly, for new POs we have

$$\tilde{f}(\bar{\mathbf{x}}_k^{(m)}, 1) = \frac{1}{\bar{C}_k^{(m)}} \varsigma_k^{(m)}(\bar{\mathbf{x}}_k^{(m)}, 1) \quad (2.19)$$

and $\tilde{f}(\bar{\mathbf{x}}_k^{(m)}, 0) = \bar{f}_k^{(m)} f_D(\bar{\mathbf{x}}_k^{(m)})$ with $\bar{f}_k^{(m)} = \frac{1}{\bar{C}_k^{(m)}} \varsigma_k^{(m)}$, where $\bar{C}_k^{(m)} \triangleq \int \varsigma_k^{(m)}(\bar{\mathbf{x}}_k^{(m)}, 1) d\bar{\mathbf{x}}_k^{(m)} + \varsigma_k^{(m)}$. Note that the combination of $\tilde{f}(\mathbf{x}_k^{(j)}, r_k^{(j)})$, $j = 1, \dots, J_{k-1}$ and $\tilde{f}(\bar{\mathbf{x}}_k^{(m)}, \bar{r}_k^{(m)})$, $m = 1, \dots, M_k$, i.e. $\tilde{f}(\mathbf{x}_k^{(j)}, r_k^{(j)})$, $j = 1, \dots, J_k$ are the input messages of next recursion.

2.4 Summary

In this chapter, we introduce the background of sequential filtering for Bayesian estimation. We also introduce the message passing algorithms of MOT based on a graphic model. Although representing causal relationship by a factor graph makes the probabilistic inference tractable, the complex integration of equations such as (2.14) (2.15) typically has no closed-form solution. In the next section, we will use nonparametric filtering methods to approximate the messages and beliefs for multisensor MOT problem, and embed PFI for IS to significant improve the sampling efficiency for accurate belief approximation.

Chapter 3

Particle Flow for Importance Sampling

In Chapter 2, we noted that parametric filtering in a general MOT problem becomes intractable when the measurement model is nonlinear and the beliefs in the factor graph are not guaranteed to be Gaussian. Consequently, nonparametric filtering methods based on IS are preferred for such MOT problems. IS is a universal approach, applicable to both traditional Bayesian filtering and belief approximation on graphs, as it provides an asymptotically optimal representation of arbitrarily complex PDFs or functions. However, IS can easily fail in high-dimensional estimation problems [130, 46]. We will use particle flow to enable IS in high-dimensional nonlinear MOT problems¹.

3.1 Curse of Dimensionality and Particle Flow

3.1.1 Curse of Dimensionality Revisited

Modern sensors can provide far more accurate measurements for a given object, while the PDF of the object state is not always Gaussian, either due to the complex shape of the object or due to the processing of multisensor measurements [132]. In statistics, it is often described by a high-dimensional, multimodal distribution. The estimation of such a distribution is a challenging problem.

¹This chapter is based on W. Zhang and F. Meyer, “Graph-based multiobject tracking with embedded particle flow,” in *Proc. IEEE RadarConf-21*, (Atlanta, GA, USA), May 2021 [131] and W. Zhang and F. Meyer, “Multisensor multiobject tracking with improved sampling efficiency,” *IEEE Trans. Signal Process.*, vol. 72, pp. 2036–2053, 2024 [132].

One of the main challenges for nonparametric estimation with the advent of accurate measuring instruments is *particle degeneracy* [46]. The particle degeneracy problem is related to the fact that the predicted PDFs is 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 PDFs. Fig. 3.1 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. In Table 3.1, we use ESS [133] to measure the efficiency of traditional sampling methods. ESS is a measures the number of particles that have significant weights when representing a PDF. We will show that the bootstrap IS and AIS [79] suffers from particle degeneracy badly. For the 2D state dimension, we take the first two dimensions, i.e., $[x_1 \ x_2]^T$ with the same PDF along each axis as in Fig. 3.1. We can see that the bootstrap IS suffers from particle degeneracy with ESS approximates to 1 while the AIS doesn’t with ESS about 1997 from 2000 samples. However, AIS degenerates significantly for 3-D scenario as in Fig. 3.1, of which the ESS is only about 5. What’s more, if we decrease the measurement noise of the TDOA measurement, the ESS is just 1.

Table 3.1. Particle degeneracy of traditional IS methods with TDOA measurements. ESS of IS and AIS using 2000 samples and different measurement standard deviations are shown for the tracking scenario in Fig. 3.1.

Method	Dimension	σ_v (secs.)	ESS
IS	2D	1×10^{-4}	1.0003
AIS	2D	1×10^{-4}	1997.2
AIS	3D	1×10^{-4}	5.2029
AIS	3D	1×10^{-5}	1.0000

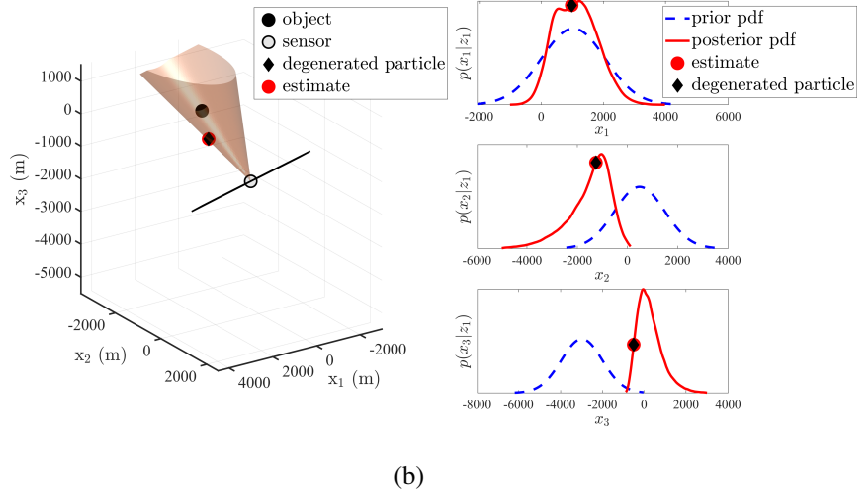
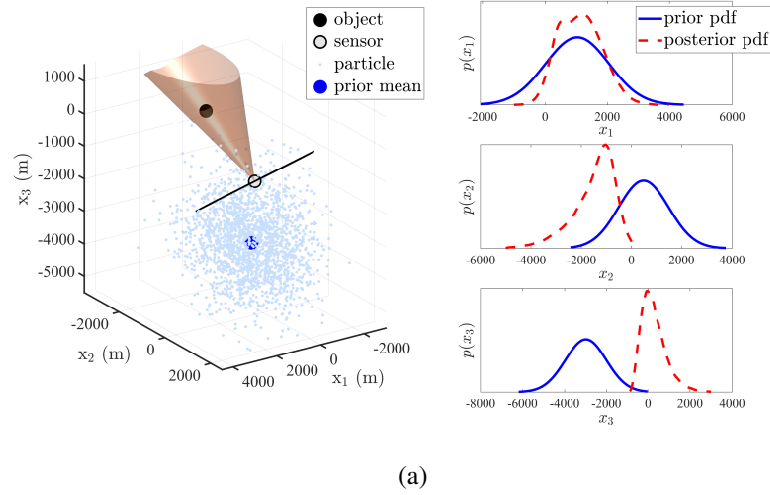


Figure 3.1. Particle degeneracy in a tracking scenario with 3-D object state, $\mathbf{x} = [x_1 \ x_2 \ x_3]^T$, and a single 1-D TDOA measurement z_1 . A single time step is considered. The 1-D TDOA measurement is generated by the sensor shown as gray circle. Assuming no measurement noise, the 1-D 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, $f(\mathbf{x})$, 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 $p(\mathbf{x}|z_1)$ for future processing, e.g., of a measurement, z_2 , provided by a second sensor.

3.1.2 Particle Flow as a Homotopy Transformation

PFI [84, 85, 86, 87] is an approach that aims at avoiding particle degeneracy. Consider a random state to be estimated, $\mathbf{x} \in \mathbb{R}^N$, in a Bayesian setting. PFI [84, 85] establishes a continuous mapping w.r.t. pseudo-time $\lambda \in [0, 1]$, to migrate particles sampled from the prior PDF $\mathbf{x}_0 \sim f(\mathbf{x})$ such that they represent the posterior PDF $\mathbf{x}_1 \sim f(\mathbf{x}|\mathbf{z})$. Let us introduce the homotopy function $\pi_\lambda(\mathbf{x}) = f(\mathbf{x})l^\lambda(\mathbf{x})$ where $l(\mathbf{x}) = f(\mathbf{z}|\mathbf{x})$ is the likelihood function. Note that for $\lambda = 1$, the homotopy function is equal to the unnormalized posterior PDF, i.e., $\pi(\mathbf{x}) \triangleq \pi_1(\mathbf{x}) = f(\mathbf{x})l(\mathbf{x})$. The log-homotopy function is then given by [84, 85],

$$\phi(\mathbf{x}, \lambda) = \log f(\mathbf{x}) + \lambda \log l(\mathbf{x}). \quad (3.1)$$

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

This log-homotopy function is used to constrain the Fokker-Plank equation and the corresponding Langevin SDE of the form

$$d\mathbf{x} = \boldsymbol{\zeta}(\mathbf{x}, \lambda)d\lambda + \mathbf{Q}(\lambda)^{1/2}d\mathbf{w}. \quad (3.2)$$

Here, $\boldsymbol{\zeta}(\mathbf{x}, \lambda) \in \mathbb{R}^N$ is the drift vector, $\mathbf{Q}(\lambda) \in \mathbb{R}^{N \times N}$ is the diffusion matrix, and $\mathbf{w} \in \mathbb{R}^N$ is Brownian motion. The constrained Fokker-Plank equation can be used to derive the functional forms of $\boldsymbol{\zeta}(\mathbf{x}, \lambda)$ and $\mathbf{Q}(\lambda)$ [117]. With the diffusion matrix equal to zero, i.e., $d\mathbf{x} = \boldsymbol{\zeta}(\mathbf{x}, \lambda)d\lambda$, a deterministic flow is obtained by solving following partial differential equation (PDE) [87, 86]

$$\frac{\partial \phi(\mathbf{x}, \lambda)}{\partial \mathbf{x}} \boldsymbol{\zeta}(\mathbf{x}, \lambda) + \log l(\mathbf{x}) = -\text{Tr}\left(\frac{\partial \boldsymbol{\zeta}(\mathbf{x}, \lambda)}{\partial \mathbf{x}}\right) \quad (3.3)$$

The detailed derivation of (3.3) is given in Appendix A.

Since the drift is both time and state-dependent, directly integrating λ from 0 to 1 is analytically infeasible. Thus, the Euler-Maruyama method [134] is commonly used for numerical integration. Here, particle migration is performed by evaluating $\boldsymbol{\zeta}(\mathbf{x}, \lambda)$ at N_λ discrete values of λ , i.e., $0 = \lambda_0 < \lambda_1 < \dots < \lambda_{N_\lambda} = 1$. First, N_s particles $\{\mathbf{x}_0^{(i)}\}_{i=1}^{N_s} = \{\mathbf{x}_{\lambda_0}^{(i)}\}_{i=1}^{N_s}$ are drawn from $f(\mathbf{x})$. Next, each particle $i \in \{1, \dots, N_s\}$ is migrated sequentially across the discrete pseudo time steps $l \in \{1, \dots, N_\lambda\}$, i.e.,

$$\mathbf{x}_{\lambda_l}^{(i)} = \mathbf{x}_{\lambda_{l-1}}^{(i)} + \boldsymbol{\zeta}_s(\mathbf{x}_{\lambda_{l-1}}^{(i)}, \lambda_l) \Delta_l + \sqrt{\Delta_l \mathbf{Q}(\lambda_l)} \mathbf{w}_l^{(i)} \quad (3.4)$$

where $\mathbf{w}_l^{(i)}$ is Gaussian distributed with unit variance and $\Delta_l = \lambda_l - \lambda_{l-1}$. In this way, particles $\{\mathbf{x}_1^{(i)}\}_{i=1}^{N_s} = \{\mathbf{x}_{\lambda_{N_\lambda}}^{(i)}\}_{i=1}^{N_s}$ representing the posterior PDF $f(\mathbf{x}|\mathbf{z}) \propto \exp(\phi(\mathbf{x}, \lambda = 1))$ are finally obtained.

3.1.3 Exact Particle Flow for Gaussian Distribution

For problems in which $\log f(\mathbf{x})$ and $\log l(\mathbf{x})$ are polynomials in the components of vector $\mathbf{x}$ (e.g. Gaussian or exponential families), $\boldsymbol{\zeta}(\mathbf{x}, \lambda)$ in (3.3) can be solved exactly in a closed-form [135, 86], which is known as EDH filter. More precisely, let $f(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{x}_0^*, \mathbf{P})$ and $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$ be a linear measurement model with measurement noise $\mathbf{v} \sim \mathcal{N}(\mathbf{v}; \mathbf{0}, \mathbf{R})$. The exact flow solution [135, 86] now reads $\boldsymbol{\zeta}(\mathbf{x}, \lambda) = \mathbf{A}(\lambda)\mathbf{x} + \mathbf{b}(\lambda)$ where we introduce

$$\mathbf{A}(\lambda) = -\frac{1}{2} \mathbf{P} \mathbf{H}^T (\lambda \mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H} \quad (3.5)$$

and

$$\mathbf{b}(\lambda) = (\mathbf{I} + 2\lambda \mathbf{A}(\lambda)) [(\mathbf{I} + \lambda \mathbf{A}(\lambda)) \mathbf{P} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z} + \mathbf{A}(\lambda) \mathbf{x}_0^*]. \quad (3.6)$$

Here separation of variable is used to model the formula of ζ . the drift is described by the Ornstein-Uhlenbeck process [135]. Note that in (3.6), $\mathbf{z}$ is the observed and thus fixed measurement.

For nonlinear measurement model, i.e. $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$, Taylor series expansions can be employed for a suboptimal linearization step. In particular, in a first-order approximation, a Jacobian matrix is computed, i.e. $\mathbf{H}(\lambda) = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_\lambda^*}$ where $\mathbf{x}_\lambda^*$ is the approximated mean of $\mathbf{x}$ at pseudo time λ .

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

$$\mathbf{x}_{\lambda_l}^{(i)} = \mathbf{x}_{\lambda_{l-1}}^{(i)} + \tilde{\zeta}(\mathbf{x}_{\lambda_{l-1}}^{(i)}, \lambda_l)(\lambda_l - \lambda_{l-1}) \quad (3.7)$$

for all $i \in \{1, \dots, N_p\}$. Here, the linearized flow solution $\tilde{\zeta}(\mathbf{x}_{\lambda_{l-1}}^{(i)}, \lambda_l) = \mathbf{A}_l \mathbf{x}_{\lambda_{l-1}}^{(i)} + \mathbf{b}_l$ is computed based on $\mathbf{A}_l$ and $\mathbf{b}_l$ given by (cf. (3.5) and (3.6))

$$\mathbf{A}_l = -\frac{1}{2} \mathbf{P} \mathbf{H}_l^T (\lambda_l \mathbf{H}_l \mathbf{P} \mathbf{H}_l^T + \mathbf{R})^{-1} \mathbf{H}_l \quad (3.8)$$

$$\mathbf{b}_l = (\mathbf{I} + 2\lambda_l \mathbf{A}_l) [(\mathbf{I} + \lambda_l \mathbf{A}_l) \mathbf{P} \mathbf{H}_l^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{e}_l) + \mathbf{A}_l \mathbf{x}_0^*]. \quad (3.9)$$

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

$$\mathbf{H}_l = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_{\lambda_{l-1}}^*}. \quad (3.10)$$

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

Pseudocode for the PFI update step is provided in **Algorithm 1**. PFI 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 $f(\mathbf{x}|\mathbf{z})$ [84, 85, 86, 87, 88, 89]. In what follows, the PFI related to the measurement $\mathbf{z}$ as defined by (3.7)–(3.9), is denoted as $\mathbf{x}_0 \rightarrow \mathbf{z} \rightarrow \mathbf{x}_1$ or for notational convenience in future derivations as $\mathbf{x}_0 \rightarrow \mathbf{z} \rightarrow \mathbf{x}$.

Algorithm 1: PFI Update Step

- 1 $\left[\{\mathbf{x}_1^{(i)}\}_{i=1}^{N_p}, \{\mathbf{A}_l\}_{l=1}^{N_\lambda} \right] = \text{ParticleFlow}\left(\{\mathbf{x}_0^{(i)}\}_{i=1}^{N_p}, \mathbf{x}_0^*, \mathbf{P}, \mathbf{z}\right)$
 - 2 Define pseudo time steps $0 = \lambda_0 < \lambda_1 < \dots < \lambda_{N_\lambda} = 1$;
 - 3 **for** $l = 1 : N_\lambda$ **do**
 - 4 Calculate the linearized measurement model $\mathbf{H}_l$ according to (3.10);
 - 5 Compute $\mathbf{A}_l$ and $\mathbf{b}_l$ according to (3.8) and (3.9);
 - 6 **for** $i = 1 : N_p$ **do**
 - 7 $\tilde{\boldsymbol{\zeta}}(\mathbf{x}_{\lambda_{l-1}}^{(i)}, \lambda_l) = \mathbf{A}_l \mathbf{x}_{\lambda_{l-1}}^{(i)} + \mathbf{b}_l$;
 - 8 $\mathbf{x}_{\lambda_l}^{(i)} = \mathbf{x}_{\lambda_{l-1}}^{(i)} + \tilde{\boldsymbol{\zeta}}(\mathbf{x}_{\lambda_{l-1}}^{(i)}, \lambda_l)(\lambda_l - \lambda_{l-1})$;
 - 9 $\tilde{\boldsymbol{\zeta}}(\mathbf{x}_{\lambda_{l-1}}^*, \lambda_l) = \mathbf{A}_l \mathbf{x}_{\lambda_{l-1}}^* + \mathbf{b}_l$;
 - 10 $\mathbf{x}_{\lambda_l}^* = \mathbf{x}_{\lambda_{l-1}}^* + \tilde{\boldsymbol{\zeta}}(\mathbf{x}_{\lambda_{l-1}}^*, \lambda_l)(\lambda_l - \lambda_{l-1})$;
 - 11 Output: $\{\mathbf{x}_1^{(i)}\}_{i=1}^{N_p} \triangleq \{\mathbf{x}_{\lambda_{N_\lambda}}^{(i)}\}_{i=1}^{N_p}$ and $\{\mathbf{A}_l\}_{l=1}^{N_\lambda}$
-

As an alternative to EDH PFI as discussed above, LEDH PFI has been introduced in [136, 89]. Here, a linearization is performed for each particle location individually instead of only at the mean particle location, i.e., individual flow parameters $\mathbf{H}_l^{(i)}, \mathbf{A}_l^{(i)}$, and $\mathbf{b}_l^{(i)}$ 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. 3.2, it is depicted how PFI actively move particles representing a prior PDFs to locations of high likelihood for the scenario introduced in Fig. 3.1. Active motion of particles leads to a significantly improved approximation of the posterior PDF compared to conventional importance sampling shown in Fig. 3.1.

The ESS of the case in Fig. 3.2 is 1999.6, which is close to the limit of 2000 and shows a significant improvement compared to results in Table. 3.1.

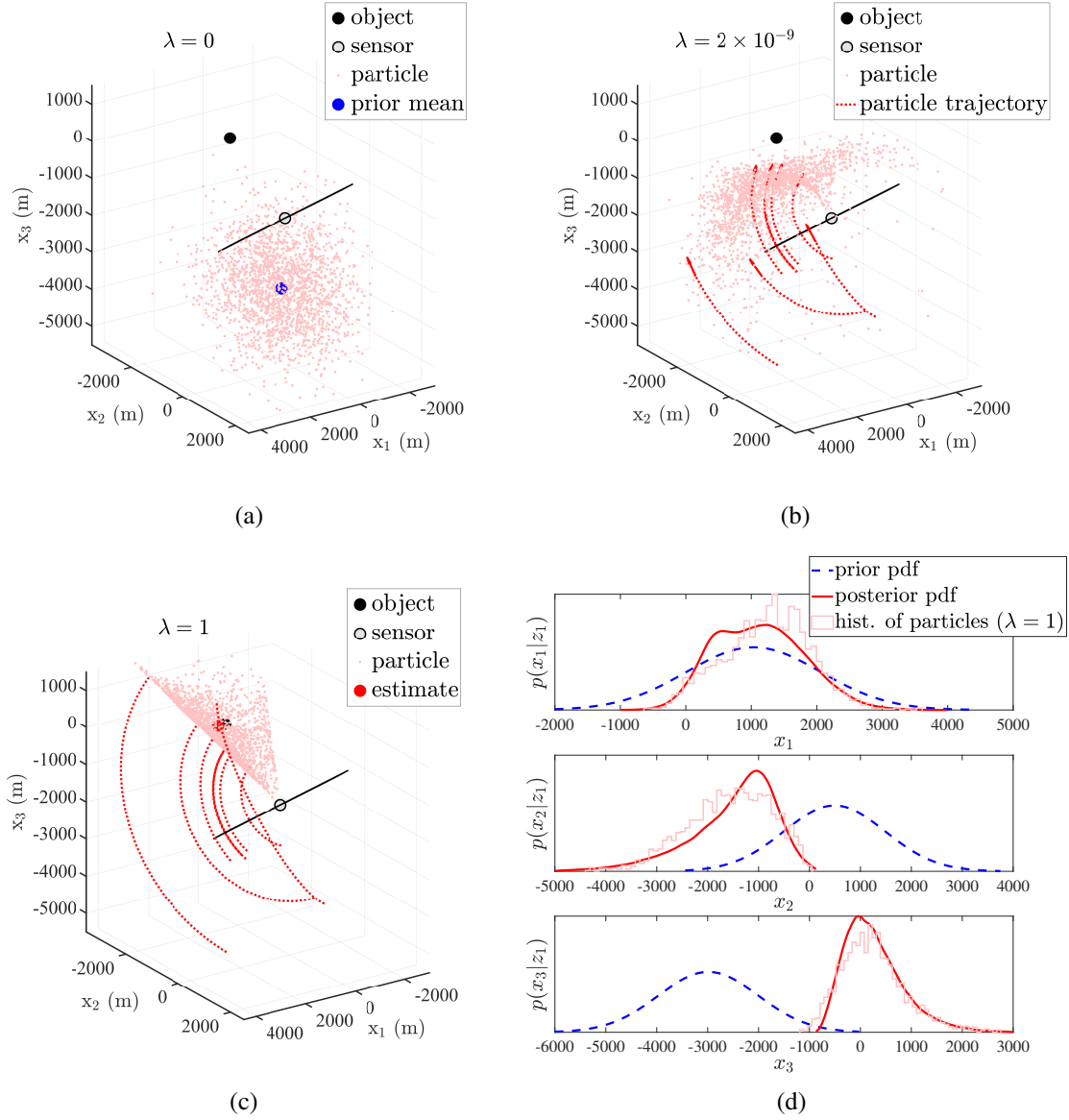


Figure 3.2. Example of PFI in the tracking scenario with 3-D object state and a single 1-D TDOA measurement discussed in Fig. 3.1. (a): 2000 particles represent the prior PDF at the onset of the flow, i.e., at pseudo time $\lambda = 0$, are depicted. (b): An intermediate flow state corresponding to $\lambda = 2 \times 10^{-9}$ 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. 3.1. Due to approximations performed in PFI, 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 PFI 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.

3.2 Importance Sampling with Invertible Flow

Although it has been demonstrated that PFI can overcome particle degeneracy in a variety of nonlinear and high-dimensional problems [84, 85, 86, 117], it has no asymptotic optimality guarantees if directly applied to the particles [89].

For asymptotically optimal estimation², PFI can be used to develop a proposal PDF for IS. In particular, the particle flow particle filter (PFPF)[89] embeds an EDH PFI into IS by introducing a flow-induced proposal PDF. The evaluation of this proposal PDF at the migrated particles requires an invertible mapping from the predicted/prior PDF at particles before the flow and the PDF represented by the particles after the flow.

PFI can be used to compute a measurement-driven proposal PDF $q(\mathbf{x}|\mathbf{z})$ for IS (cf. (2.2)) to perform asymptotically optimal estimation [89]. Here, the mapping as performed by PFI $\mathbf{x}_0 \rightarrow \mathbf{z} \rightarrow \mathbf{x}$ is invertible, i.e., there exists an invertible mapping of the particles after the flow $\{\mathbf{x}^{(i)}\}_{i=1}^{N_p}$ to the particles $\{\mathbf{x}_0^{(i)}\}_{i=1}^{N_p}$ if certain constraints on the differences of consecutive discrete pseudo times $\lambda_l - \lambda_{l-1}$, $l \in \{1, \dots, N_\lambda\}$ are satisfied [89].

By exploiting the invertible mapping, the proposal PDF resulting from PFI can be evaluated at the particles as [89]

$$q_{\text{PFI}}(\mathbf{x}^{(i)}|\mathbf{z}) = \frac{f(\mathbf{x}_0^{(i)})}{\theta}. \quad (3.11)$$

Here, the “mapping factor” θ is defined as

$$\theta = \prod_{l=1}^{N_\lambda} |\det[\mathbf{I} + (\lambda_l - \lambda_{l-1})\mathbf{A}_l]|. \quad (3.12)$$

By plugging (3.11) into (2.2) the weight of the particle $\mathbf{x}^{(i)}$ is obtained as

²IS is asymptotically optimal, assuming weak conditions are satisfied [115, 116], but known to suffer from particle degeneracy in higher dimensional problems [46].

$$w^{(i)} \propto \frac{\theta f(\mathbf{z}|\mathbf{x}^{(i)}) f(\mathbf{x}^{(i)})}{f(\mathbf{x}_0^{(i)})}. \quad (3.13)$$

The resulting particle set $\{\mathbf{x}^{(i)}, w^{(i)}\}_{i=1}^{N_p}$ is an asymptotically optimal sample representation of the posterior PDF $f(\mathbf{x}|\mathbf{z})$ that can often provide accurate estimation results in nonlinear and high-dimensional estimation problems even if the number of particles is moderate [89]. Pseudocode for importance sampling with invertible PFI is provided in **Algorithm 2**. Note that since the PFI used for the measurement-driven proposal PDF is typically based on the EDH filter update step, a Gaussian prior PDF is assumed.

Algorithm 2: Importance Sampling with Invertible Flow

```

1   $\left[\{\mathbf{x}_1^{(i)}, w_1^{(i)}\}_{i=1}^{N_p}\right] = \text{InvertibleFlow}(\mathbf{x}^*, \mathbf{P}, \mathbf{z})$ 
2  for  $i = 1 : N_p$  do
3     $\left[ \text{Draw } \mathbf{x}_0^{(i)} \sim \mathcal{N}(\mathbf{x}; \mathbf{x}^*, \mathbf{P}); \right.$ 
4  Perform PF according  $\left[\{\mathbf{x}_1^{(i)}\}_{i=1}^{N_p}, \{A_l\}_{l=1}^{N_\lambda}\right] = \text{ParticleFlow}\left(\{\mathbf{x}_0^{(i)}\}_{i=1}^{N_p}, \mathbf{x}^*, \mathbf{P}, \mathbf{z}\right);$ 
5     $// \text{ see Alg. 1}$ 
6  Compute the mapping factor  $\theta$  from  $\{A_l\}_{l=1}^{N_\lambda}$  following (3.12);
7  for  $i = 1 : N_p$  do
8     $\left[ \text{Perform weight update according to (3.13), i.e., } w_1^{(i)} = \frac{\theta f(\mathbf{z}|\mathbf{x}_1^{(i)}) \mathcal{N}(\mathbf{x}_1^{(i)}; \mathbf{x}^*, \mathbf{P})}{\mathcal{N}(\mathbf{x}_0^{(i)}; \mathbf{x}^*, \mathbf{P})}; \right.$ 
9  Output:  $\{\mathbf{x}_1^{(i)}, w_1^{(i)}\}_{i=1}^{N_p}$ 

```

An approximate Gaussian representation of this posterior distribution can be subsequently obtained by applying **Algorithm 3** which calculates a mean $\mathbf{x}_1^*$ and a covariance matrix $\mathbf{P}_1$ from the unnormalized weighted particles $\{\mathbf{x}_1^{(i)}, w_1^{(i)}\}_{i=1}^{N_p}$. 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 if multiple flows are performed in parallel, as will be discussed in Section 3.3.

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 [89]. For a large

Algorithm 3: Computation of Gaussian Representation

```
1  $[\mathbf{x}^*, \mathbf{P}] = \text{GaussianRepresentation}\left(\{\mathbf{x}^{(i)}, \omega^{(i)}\}_{i=1}^{N_p}\right)$ 
2 Normalize particles, i.e.,
3 for  $i = 1 : N_p$  do
4    $w^{(i)} = \frac{\omega^{(i)}}{\sum_{i'=1}^{N_p} \omega^{(i')}} ;$ 
5 Compute mean and covariance matrix from particles, i.e.,
6  $\mathbf{x}^* = \sum_{i=1}^{N_p} w^{(i)} \mathbf{x}^{(i)}$ 
7  $\mathbf{P} = \sum_{i=1}^{N_p} w^{(i)} \mathbf{x}^{(i)} \mathbf{x}^{(i)T} - \mathbf{x}^* \mathbf{x}^{*T}$ 
8 Output:  $\mathbf{x}^*, \mathbf{P}$ 
```

number of particles N_p , a particle-based computation is more accurate than a computation based on the extended or unscented Kalman update step.

3.3 Improving Sampling Efficiency for A Single Object with Measurement-Origin Uncertainty

Although PFI has been applied to particle filter with Gaussian approximation [89], improving sampling efficiency for a more challenging problem such as nonlinear MOT is still an open problem. Message passing in multisensor MOT is more complex mainly because

- The state of an object is highly non-Gaussian due to a sequential processing of nonlinear measurements from an underdetermined model in high dimensions.
- The measurement generated by the object is uncertain since multiple measurements are observed, while traditional PFI is formulated for a single measurement per sensor .

In this section, we use GMM [137, 138, 139] to represent multi-modal state PDF of an object for nonlinear estimation in high dimensions. In particular, we developed methods for updating the parameters of GMMs based on PFI. we further develop an algorithm where PFIs are used for graph-based message passing. To consider MOU, we develop a proposal embedding information of multiple measurements driven by parallel PFIs. This approach is suitable for high-dimensional PDFs that are multimodal and thus relevant for estimation problems in MOT and SLAM [140].

3.3.1 GMM Importance Sampling with Invertible Flow

As discussed in Section 3.2, 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. 3.2(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 a “kernel”. Let N_k be the total number of kernels and let $h \in \{1, \dots, N_k\}$ be the kernel index. A multimodal prior PDF that follows a GMM representation can then be written as $f(\mathbf{x}) = \frac{1}{N_k} \sum_{h=1}^{N_k} \mathcal{N}(\mathbf{x}; \mathbf{x}^{*(h)}, \mathbf{P}^{(h)})$. The corresponding multimodal posterior PDF $f(\mathbf{x}|\mathbf{z})$ 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 $\mathcal{N}(\mathbf{x}; \mathbf{x}^{*(h)}, \mathbf{P}^{(h)})$, $h \in \{1, \dots, N_k\}$. 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 $N_k N_p$ 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 PFI is provided in **Algorithm 4**. Importance sampling with invertible PFI makes use of resampling as presented in **Algorithm 5**.

For $N_k = 1$ and $N_p > 1$, this importance sampling approach is equivalent to invertible PFI based on the EDH update step. Furthermore, if $N_p = 1$, $N_k > 1$, this importance sampling approach is equivalent to invertible PFI based on the LEDH update step. Note that for $N_p = 1$, as performed by the LEDH, an additional extended or unscented Kalman update step needs to be used to calculate an approximate covariance matrix [89].

3.3.2 PFI-Based Proposal for Problems with MOU

As discussed in Section 2.3, in a variety of tracking and localization problems, the measurement model suffers from a deficiency beyond measurement noise referred to as MOU

Algorithm 4: GMM Importance Samp. with Invertible Flow

```

1   $\left[ \{\mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)}\}_{h=1}^{N_k} \right] = \text{InvertibleFlowGMM}\left(\{\mathbf{x}^{*(h)}, \mathbf{P}^{(h)}\}_{h=1}^{N_k}, \mathbf{z}\right)$ 
2  for  $h = 1 : N_k$  do
3     $\left[ \{\mathbf{x}_1^{(i,h)}, w_1^{(i,h)}\}_{i=1}^{N_p} \right] = \text{InvertibleFlow}\left(\mathbf{x}^{*(h)}, \mathbf{P}^{(h)}, \mathbf{z}\right)$ 
4     $\quad \quad \quad // \text{ see Alg. 2}$ 
5     $\left[ \sim, \mathbf{P}_1^{(h)} \right] = \text{GaussianRepresentation}\left(\{\mathbf{x}_1^{(i,h)}, w_1^{(i,h)}\}_{i=1}^{N_p}\right)$ 
6     $\quad \quad \quad // \text{ see Alg. 3}$ 
7     $\left[ \{\mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)}\}_{h=1}^{N'_k} \right] = \text{Resampling}\left(\left\{ \mathbf{P}_1^{(h)}, \{\mathbf{x}_1^{(i,h)}, w_1^{(i,h)}\}_{i=1}^{N_p} \right\}_{h=1}^{N_k}\right)$ 
8     $\quad \quad \quad // \text{ see Alg. 5}$ 
9  Output:  $\{\mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)}\}_{h=1}^{N'_k}$ 

```

Algorithm 5: Resampling

```

1   $\left[ \{\mathbf{x}_+^{*(h')}, \mathbf{P}_+^{(h')}\}_{h'=1}^{N'_k} \right] = \text{Resampling}\left(\left\{ \mathbf{P}_1^{(h)}, \{\mathbf{x}_1^{(i,h)}, w_1^{(i,h)}\}_{i=1}^{N_p} \right\}_{h=1}^{N_k}\right)$ 
2  for  $h = 1 : N_k$  do
3    for  $i = 1 : N_p$  do
4       $\left[ \left[ w_1^{(i,h)} = \frac{w_1^{(i,h)}}{\sum_{h'=1}^{N_k} \sum_{i'=1}^{N_p} w_1^{(i',h')}} \right] \right];$ 
5  for  $h' = 1 : N'_k$  do
6     $\left[ \text{Sample index } (i', \bar{h}) \text{ using } \left\{ \{\mathbf{x}_1^{(i,h)}\}_{i=1}^{N_p} \right\}_{h=1}^{N_k} \right];$ 
7     $\left[ \text{Set } \mathbf{x}_+^{*(h')} = \mathbf{x}_1^{(i', \bar{h})} \text{ and } \mathbf{P}_+^{(h')} = \mathbf{P}_1^{(\bar{h})} \right];$ 
8  Output:  $\{\mathbf{x}_+^{*(h')}, \mathbf{P}_+^{(h')}\}_{h'=1}^{N'_k}$ 

```

[48]. To simplify, we first look into the “object-oriented” association problem. Assume there is a single object but multiple measurements and it is not known which measurement was generated by the object. Assume the object does exist 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 $\mathbf{z}^{(m')} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$. Based on this model, the conditional PDF of the object-originated measurement $\mathbf{z}^{(m')}$ reads $f_o(\mathbf{z}^{(m')}|\mathbf{x})$. All the other measurements are FPs that follow the FP PDF $f_{fp}(\mathbf{z}^{(m')})$. Under assumptions (A₂)-(A₄) and following (2.9), for $\mathbf{z}$ fixed, one can use this conditional PDF to directly compute the MOU likelihood function

$$\begin{aligned} f(\mathbf{z}|\mathbf{x}) &= \sum_{a=0}^M f(\mathbf{z}, a|\mathbf{x}) \\ &\propto 1 - p_d + \sum_{a=1}^M \frac{p_d f(\mathbf{z}^{(a)}|\mathbf{x})}{\mu_{fp} f_{fp}(\mathbf{z}^{(a)})} \end{aligned} \quad (3.14)$$

and the marginal probability mass function (PMF)

$$\begin{aligned} p(a|\mathbf{z}) &\propto f(a, \mathbf{z}) \\ &= \int f(\mathbf{z}, a|\mathbf{x}) f(\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (3.15)$$

The values of the PMF $p(a|\mathbf{z})$ are also referred to as marginal association probabilities [48], i.e., they represent the probability of a particular association event $a \in \{0, \dots, M\}$ conditioned on an observed $\mathbf{z}$.

In principle, the MOU likelihood function in (3.14) can be directly used for importance sampling as in (2.2). However, in problems with MOU, importance sampling based on invertible PFI is complicated by the fact that there are multiple measurements, and it is thus not clear which measurement should be used to compute PFI parameters (3.8) and (3.9), i.e., the PFI proposal PDF $q_{PFI}(\mathbf{x}_1^{(i)}|\mathbf{z})$ in (3.11) cannot be directly used. To address this problem, we propose the

combined proposal PDF

$$q(\mathbf{x}|\mathbf{z}) = p(a=0|\mathbf{z})f(\mathbf{x}) + \sum_{m=1}^M p(a=m|\mathbf{z})q_{\text{PFI}}(\mathbf{x}|\mathbf{z}^{(m)}) \quad (3.16)$$

where we used the marginal association probabilities $p(a|\mathbf{z})$ to weight the proposal PDFs $q_{\text{PFI}}(\mathbf{x}|\mathbf{z}^{(m)})$ related to PFI based on measurements $\mathbf{z}^{(m)}$, $m \in \{1, \dots, M\}$ (cf. (3.11)) and the Gaussian prior PDF $f(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{x}_0^*, \mathbf{P})$. In particular, recall that the proposal distribution $q_{\text{PFI}}(\mathbf{x}|\mathbf{z}^{(m)})$ related to the PFI $\mathbf{x}_0 \rightarrow \mathbf{z}^{(m)} \rightarrow \mathbf{x}^{(m)}$, can be evaluated as

$$q_{\text{PFI}}(\mathbf{x}^{(i,m)}|\mathbf{z}^{(m)}) = \frac{\mathcal{N}(\mathbf{x}_0^{(i,m)}; \mathbf{x}_0^*, \mathbf{P})}{\theta^{(m)}} \quad (3.17)$$

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

A total of $(M+1)N_p$ particles representing the proposal PDF in (3.16) is obtained by drawing N_p particles for each of the $M+1$ components in (3.16) and calculating corresponding marginal association probabilities and weights. For the first component related to association event $a=0$, N_p particles $\{\mathbf{x}^{(i,0)}\}_{i=1}^{N_p}$ are directly drawn from $f(\mathbf{x})$, the corresponding marginal association probability is obtained by using (2.9) in (3.15), i.e., $p(a=0|\mathbf{z}) \propto 1 - p_d$, and the corresponding combined proposal weights are set to $\omega^{(i,0)} = p(a=0|\mathbf{z}) \mathcal{N}(\mathbf{x}_0^{(i,0)}; \mathbf{x}_0^*, \mathbf{P})$ (cf. (3.16)). For each other component related to association event $a=m$, $m \in \{1, \dots, M\}$, first N_p particles $\{\mathbf{x}_0^{(i,m)}\}_{i=1}^{N_p}$ are drawn from $f(\mathbf{x})$. Next, the PFI $\mathbf{x}_0 \rightarrow \mathbf{z}^{(m)} \rightarrow \mathbf{x}^{(m)}$ is applied to the particles $\{\mathbf{x}_0^{(i,m)}\}_{i=1}^{N_p}$ and new particles $\{\mathbf{x}^{(i,m)}\}_{i=1}^{N_p}$ are obtained for each $m \in \{1, \dots, M\}$. An approximation of each marginal association probability $\tilde{p}(a=m|\mathbf{z})$, $m \in \{1, \dots, M\}$ is finally calculated from these particles by using (2.9) in (3.15) and performing Monte Carlo integration [76] based on the proposal PDF $q_{\text{PFI}}(\mathbf{x}^{(i,m)}|\mathbf{z}^{(m)})$ in (3.17), i.e.,

$$\tilde{p}(a=m|\mathbf{z}) \propto \sum_{i=1}^{N_p} \frac{p_d \theta^{(m)} f(\mathbf{z}^{(m)}|\mathbf{x}^{(i,m)}) \mathcal{N}(\mathbf{x}^{(i,m)}; \mathbf{x}_0^*, \mathbf{P})}{N_p \mu_{\text{fp}} f_{\text{fp}}(\mathbf{z}^{(m)}) \mathcal{N}(\mathbf{x}_0^{(i,m)}; \mathbf{x}_0^*, \mathbf{P})}. \quad (3.18)$$

The corresponding combined proposal weights are set according to (cf. (3.16) and (3.17))

$$\omega^{(i,m)} = \tilde{p}(a=m|\mathbf{z}) \frac{\mathcal{N}(\mathbf{x}_0^{(i,m)}; \mathbf{x}_0^*, \mathbf{P})}{\theta^{(m)}}. \quad (3.19)$$

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

Following the importance sampling principle, we next aim to compute particles $\{(w^{(l)}, \mathbf{x}^{(l)})\}_{l=1}^L$ that represent the posterior PDF $f(\mathbf{x}|\mathbf{z})$. In particular, by plugging (3.14) into (2.2) and by using $\{(\omega^{(l)}, \mathbf{x}^{(l)})\}$ (cf. (3.19)) to represent $q(\mathbf{x}|\mathbf{z})$ in (2.2), we obtain

$$w^{(l)} \propto \frac{\mathcal{N}(\mathbf{x}^{(l)}; \mathbf{x}_0^*, \mathbf{P}) \left(1 - p_d + \sum_{m=1}^M \frac{p_d f(\mathbf{z}^{(m)}|\mathbf{x}^{(l)}) f(\mathbf{x}^{(l)})}{\mu_{fp} f_{fp}(\mathbf{z}^{(m)})}\right)}{\omega^{(l)}}. \quad (3.20)$$

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

Algorithm 6: GMM Importance Sampling with Invertible Flow and MOU

- 1 $\left[\{\mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)}\}_{h=1}^{N_k}\right] = \text{InvertibleFlowGMMwithMOU}\left(\{\mathbf{x}^{*(h)}, \mathbf{P}^{(h)}\}_{h=1}^{N_k}, \mathbf{z}\right)$
 - 2 $\left[\left\{\left\{\left\{\mathbf{x}^{(i,h,a)}, \omega^{(i,h,a)}\right\}_{i=1}^{N_p}, \theta^{(h,a)}\right\}_{h=1}^{N_k}, \beta^{(a)}\right\}_{a=0}^M\right] =$
 $\text{Evaluation}\left(\{\mathbf{x}^{*(h)}, \mathbf{P}^{(h)}\}_{h=1}^{N_k}, 1, \mathbf{z}\right)$
 - 3 // see Alg. 7
 - 4 $\left[\{\mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)}\}_{h=1}^{N_k}, \sim\right] =$
 $\text{Update}\left(\left\{\left\{\left\{\mathbf{x}^{(i,h,a)}, \omega^{(i,h,a)}\right\}_{i=1}^{N_p}, \theta^{(h,a)}\right\}_{h=1}^{N_k}, \beta^{(a)}, 1\right\}_{a=0}^M\right)$
 - 5 // see Alg. 8
 - 6 Output: $\{\mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)}\}_{h=1}^{N_k}$
-

3.3.3 GMM Importance Sampling with Invertible Flow for Problems with MOU

For problems where the prior distribution is non-Gaussian and potentially multimodal, GMM PFI with invertible flow discussed in Section 3.3.1 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 data association information denoted as $\kappa^{(a)}$, $a \in \{0, \dots, M\}$. In the single object tracking considered here, we have $\kappa^{(a)} = 1$.

Note that for later use in Section 3.4.5, 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 3.3.2. Furthermore, note that for consistency with Section 3.4.5, we introduced the notation $\beta^{(a)} \propto \tilde{p}(a|\mathbf{z})$, $a \in \{0, \dots, M\}$ in Algorithm 6, Algorithm 7, and Algorithm 8.

3.4 Graph-Based Multisensor MOT with Invertible PFI

In previous section, we focus on using PFIs to improve sampling efficiency for the posterior PDF representation of a single object with multiple measurements observed. In this section, we move forward to more complex multisensor MOT problems. Here, the usage of PFI is not limited to represent a PDF, but to more general messages and beliefs in a graph-based message passing framework. We will first introduce nonparametric approximation of a general function with complicated integration so that no closed-form of the function can be achieved. Then we propose a PFI-based multisensor MOT algorithm with improved sampling efficiency for graph-based message passing algorithms.

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

```

1   $\left[ \left\{ \left\{ \left\{ \mathbf{x}^{(i,h,a)}, \omega^{(i,h,a)} \right\}_{i=1}^{N_p}, \boldsymbol{\theta}^{(h,a)} \right\}_{h=1}^{N_k}, \boldsymbol{\beta}^{(a)} \right\}_{a=0}^M \right] =$ 
   Evaluation $\left( \left\{ \mathbf{x}^{*(h)}, \mathbf{P}^{(h)} \right\}_{h=1}^{N_k}, p, \mathbf{z} \right)$ 
2  Initialize association variables as  $\boldsymbol{\beta}^{(0)} = (1 - p_d)p + (1 - p)$ ;
3  for  $h = 1 : N_k$  do
4      for  $i = 1 : N_p$  do
5          Draw  $\mathbf{x}_0^{(i,h)} \sim \mathcal{N}(\mathbf{x}^{(h)}, \mathbf{x}^{*(h)}, \mathbf{P}^{(h)})$ ;
6          Initialize particles as  $\{\mathbf{x}^{(i,h,0)}\}_{i=1}^{N_p} = \{\mathbf{x}_0^{(i,h)}\}_{i=1}^{N_p}$ ;
7          Initialize proposal weights according to
8           $\omega^{(i,h,0)} = \mathcal{N}(\mathbf{x}^{(i,h,0)}; \mathbf{x}^{*(h)}, \mathbf{P}^{(h)}), i = 1, \dots, N_p$ ;
9  Initialize mapping factor  $\boldsymbol{\theta}^{(h,0)} = 1$ ;
10 for  $m = 1 : M$  do
11     for  $h = 1 : N_k$  do
12         Perform PFI, i.e.,
13          $\left[ \{\mathbf{x}^{(i,h,m)}\}_{i=1}^{N_p}, \{A_l^{(h,m)}\}_{l=1}^{N_\lambda} \right]$ 
14          $= \text{ParticleFlow}\left( \{\mathbf{x}_0^{(i,h)}\}_{i=1}^{N_p}, \mathbf{x}^{*(h)}, \mathbf{P}^{(h)}, \mathbf{z}^{(m)} \right)$ 
15         // see Alg. 1
16         Compute mapping factor  $\boldsymbol{\theta}^{(h,m)}$  from  $\{A_l^{(h,m)}\}_{l=1}^{N_\lambda}$  as in (3.12);
17     Precompute weights (cf. (3.19) and (3.20)), i.e.,
18      $\omega^{(i,h,m)} = \frac{\mathcal{N}(\mathbf{x}^{(i,h,0)}; \mathbf{x}^{*(h)}, \mathbf{P}^{(h)})}{\mathcal{N}(\mathbf{x}^{(i,h,m)}; \mathbf{x}^{*(h)}, \mathbf{P}^{(h)}) \boldsymbol{\theta}^{(h,m)}}, i = 1, \dots, N_p$ ;
19     Compute approximate association variables following (3.18), i.e.,
        
$$\boldsymbol{\beta}^{(m)} = \frac{p p_d}{N_p N_k \mu_{fp}} \sum_{h=1}^{N_k} \sum_{i=1}^{N_p} \frac{f(\mathbf{z}^{(m)} | \mathbf{x}^{(i,h,m)})}{f_{fp}(\mathbf{z}^{(m)}) \omega^{(i,h,m)}};$$

20 Output:  $\left\{ \left\{ \left\{ \mathbf{x}^{(i,h,a)}, \omega^{(i,h,a)} \right\}_{i=1}^{N_p}, \boldsymbol{\theta}^{(h,a)} \right\}_{h=1}^{N_k}, \boldsymbol{\beta}^{(a)} \right\}_{a=0}^M$ 

```

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

```

1   $\left[ \left\{ \mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)} \right\}_{h=1}^{N_k}, p_+ \right] =$ 
    Update  $\left( \left\{ \left\{ \mathbf{x}^{(i,h,a)}, \omega^{(i,h,a)} \right\}_{i=1}^{N_p}, \boldsymbol{\theta}^{(h,a)} \right\}_{h=1}^{N_k}, \boldsymbol{\beta}^{(a)}, \boldsymbol{\kappa}^{(a)} \right\}_{a=0}^M, p \right)$ 
2  Compute association probabilities  $\tilde{p}(a|\mathbf{z}) = \frac{\beta^{(a)} \kappa^{(a)}}{\sum_{a=0}^M \beta^{(a)} \kappa^{(a)}}, a=0, \dots, M$ 
3  for  $h = 1 : N_k$  do
4      for  $a = 0 : M$  do
5          Update proposal weights (cf. (3.16)–(3.20)), i.e.,
6           $\omega'^{(i,h,a)} = \tilde{p}(a|\mathbf{z}) \omega^{(i,h,a)}, i=1, \dots, N_p;$ 
7          Reindex particles and proposal weights using  $l = i(a+1)$  to obtain
             $\left\{ (\omega^{(l,h)}, \mathbf{x}^{(l,h)}) \right\}_{l=1}^L$  from  $\left\{ (\omega'^{(i,h,a)}, \mathbf{x}^{(i,h,a)}) \right\}_{i=1}^{N_p} a \in \{0, \dots, M\};$ 
8          Compute final particle weights according to (cf. (3.20))
9          for  $l = 1 : L$  do
10              $w^{(l,h)} = \frac{\left( \kappa^{(0)}(1-p_d) + \sum_{m=1}^M \frac{\kappa^{(m)} p_d f(\mathbf{z}^{(m)} | \mathbf{x}^{(l,h)})}{\mu_{\text{fp}} f_{\text{fp}}(\mathbf{z}^{(m)})} \right)}{\omega^{(l,h)}};$ 
11              $[\sim, \mathbf{P}^{(h)}] = \text{GaussianRepresentation} \left( \left\{ \mathbf{x}^{(l,h)}, w^{(l,h)} \right\}_{l=1}^{L_s} \right)$ 
12              $// \text{ see Alg. 3}$ 
13   $w = \frac{p}{N_k L_s} \sum_{h=1}^{N_k} \sum_{l=1}^{L_s} w^{(l,h)}$ 
14   $p_+ = \frac{w}{(1-p) \tilde{p}(a=0|\mathbf{z}) + w}$ 
15   $\left[ \left\{ \mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)} \right\}_{h=1}^{N_k} \right] = \text{Resampling} \left( \left\{ \mathbf{P}^{(h)}, \left\{ \mathbf{x}^{(l,h)}, w^{(l,h)} \right\}_{l=1}^{L_s} \right\}_{h=1}^{N_k} \right)$ 
16   $// \text{ see Alg. 5}$ 
17  Output:  $\left\{ \mathbf{x}_+^{*(h)}, \mathbf{P}_+^{(h)} \right\}_{h=1}^{N_k}, p_+$ 

```

3.4.1 Nonparametric Estimation of General Integral Functions

Assume we want approximate a complex integral function with no closed-form solution as

$$E(\mathbf{x}) = \int \alpha(\mathbf{x})\beta(\mathbf{x})d\mathbf{x} \quad (3.21)$$

where α, β are arbitrary functions of object state $\mathbf{x}$. Without generality, assume $\beta(\mathbf{x})$ is more informative (less heavy-tailed) than $\alpha(\mathbf{x})$. If we sample from a proposal $\{\mathbf{x}^{(i)}\}_{i=1}^N \sim q(\mathbf{x})$, we can approximate the estimate as $E(\mathbf{x}) \approx \sum_{i=1}^N \alpha(\mathbf{x}^{(i)})w^{(i)}$, where $w^{(i)} = \frac{\beta(\mathbf{x}^{(i)})}{Nq(\mathbf{x}^{(i)})}$. Note that this is the nonparametric representation of a general function, and it yields to (2.2) if we take $\alpha(\mathbf{x}) = \mathbf{x}$ and $\beta(\mathbf{x}) = f(\mathbf{x}|\mathbf{z})$ the posterior PDF of x , hence $E(\mathbf{x}) = \mathbb{E}_{f(\mathbf{x}|\mathbf{z})}(\mathbf{x})$. For efficient as well as accurate sampling in the case of (3.21), $q(\mathbf{x})$ should be informative enough close to $\beta(\mathbf{x})$ but more heavy-tailed than $\beta(\mathbf{x})$, similar to PFI as proposal compared with the posterior PDF.

Following the discussion above, in nonlinear MOT scenarios, calculation of integral functions $\beta_k^{(j)}(a_k^{(j)})$ in (2.14) and $\tilde{f}(\underline{\mathbf{x}}_k^{(j)}, r_k^{(j)})$ in (2.18) related to legacy PO states as well as their counterparts $\xi_k^{(m)}(b_k^{(m)})$ and $\tilde{f}(\underline{\mathbf{x}}_k^{(m)}, \bar{r}_k^{(m)})$, related to new PO states cannot be performed in closed form. We propose a nonparametric implementation of the message passing algorithm where a proposal PDF is established using invertible PFI. 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 $k-1$. The following Section 3.4.2-3.4.4 correspond to Step (S_1) -(S_5) in Section 2.3.4. In particular, Section 3.4.2 corresponds to (S_1) , 3.4.3 to (S_2) , 3.4.4 to (S_4) and (S_5) . For Section 3.4.3 and 3.4.4, we add the index of sensor s for loop processing over multiple sensors.

3.4.2 Prediction

It is assumed that the beliefs of legacy POs at time $k-1$ are represented by a single Gaussian distribution, i.e., $\tilde{f}(\mathbf{x}_-^{(j)}, r_-^{(j)}=1) = p_-^{(j)} \mathcal{N}(\mathbf{x}_-^{(j)}; \mathbf{x}_-^{*(j)}, \mathbf{P}_-^{(j)})$, $j \in \{1, \dots, J_-\}$. In MOT problems, the state transition function underlying the state transition model $f(\mathbf{x}_1^{(j)} | \mathbf{x}_-^{(j)})$ discussed in [65, Section VIII-C] is typically linear with additive Gaussian noise, i.e., $\underline{\mathbf{x}}_1^{(j)} = \mathbf{G}\mathbf{x}_-^{(j)} + \mathbf{u}^{(j)}$ where $\mathbf{G}$ is the state transition matrix and $\mathbf{u}^{(j)}$ is an additive Gaussian noise vector with mean $\mathbf{u}^*$ and covariance matrix $\mathbf{P}_u$. Consequently, the messages computed in the prediction step are also represented by a Gaussian distribution, i.e., $\alpha_1^{(j)}(\underline{\mathbf{x}}_1^{(j)}, r_1^{(j)}) = \underline{p}_1^{(j)} \mathcal{N}(\underline{\mathbf{x}}_1^{(j)}; \underline{\mathbf{x}}_1^{*(j)}, \underline{\mathbf{P}}_1^{(j)})$, $j \in \{1, \dots, J_-\}$ with mean, covariance matrix, and existence probability given by $\underline{\mathbf{x}}_1^{*(j)} = \mathbf{G}\mathbf{x}_-^{*(j)} + \mathbf{u}^*$, $\underline{\mathbf{P}}_1^{(j)} = \mathbf{G}\mathbf{P}_-^{(j)}\mathbf{G}^T + \mathbf{P}_u$, and $\underline{p}_1^{(j)} = p_{\text{su}}p_-^{(j)}$, 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 . If the state transition function is not linear with additive Gaussian noise, for each $j \in \{1, \dots, J_-\}$, N_p particles are drawn from $\mathcal{N}(\mathbf{x}_-^{(j)}; \mathbf{x}_-^{*(j)}, \mathbf{P}_-^{(j)})$, the prediction step of a conventional particle filter is performed [60], and a predicted Gaussian representation $\mathcal{N}(\underline{\mathbf{x}}_1^{(j)}; \underline{\mathbf{x}}_1^{*(j)}, \underline{\mathbf{P}}_1^{(j)})$ is computed from the resulting particles using Algorithm 3.

3.4.3 Measurement Evaluation

The following steps are performed sequentially for each sensor $s = 1, \dots, S$. First, a particle representation $\{(\underline{\mathbf{x}}_{0,s}^{(i,j)}, \underline{\omega}_{0,s}^{(i,j)})\}_{i=1}^{N_p}$ is obtained by drawing particles $\underline{\mathbf{x}}_{0,s}^{(i,j)}$, $i \in \{1, \dots, N_p\}$ from $\mathcal{N}(\mathbf{x}_s^{(j)}; \mathbf{x}_s^{*(j)}, \mathbf{P}_s^{(j)})$ and setting the corresponding weights to $\underline{\omega}_{0,s}^{(i,j)} = \underline{p}_s^{(j)}/N_p$. Next, we compute an extended set $\{(\underline{\mathbf{x}}_s^{(i,a,j)}, \underline{\omega}_s^{(i,a,j)})\}_{i=1}^{N_p}\}_{a=0}^{M_s}$, that consists of N_p particles and weights for each value of $a_s^{(j)} \in \{0, \dots, M_s\}$ and $j \in \{1, \dots, J_{s-1}\}$. (Note that measurement gating [48] can be employed to reduce the number of measurements used for PFI.) For $a_s^{(j)} = 0$, we perform no flow, i.e., we set $\{(\underline{\mathbf{x}}_s^{(i,0,j)}, \underline{\omega}_s^{(i,0,j)})\}_{i=1}^{N_p} = \{(\underline{\mathbf{x}}_{0,s}^{(i,j)}, \underline{\omega}_{0,s}^{(i,j)})\}_{i=1}^{N_p}$. For $a_s^{(j)} = m \in \{1, \dots, M_s\}$, the PFI $\underline{\mathbf{x}}_{0,s}^{(j)} \rightarrow \mathbf{z}_s^{(m)} \rightarrow \underline{\mathbf{x}}_s^{(j,m)}$ is applied to obtain new particles $\{\underline{\mathbf{x}}_s^{(i,m,j)}\}_{i=1}^{N_p}$ by migrating the particles $\{\underline{\mathbf{x}}_{0,s}^{(i,m,j)}\}_{i=1}^{N_p}$. By making use of the invertible PFI principle (cf. (3.11)) [89], the weights $\underline{\omega}_s^{(i,m,j)}$

corresponding to the migrated particles $\underline{\mathbf{x}}_s^{(i,m,j)}$ are obtained as

$$\underline{\omega}_s^{(i,m,j)} = \frac{\mathcal{N}(\underline{\mathbf{x}}_s^{(i,m,j)}; \underline{\mathbf{x}}_s^{(j)*}, \underline{\mathbf{P}}_s^{(j)}) \underline{\theta}_{m,s}^{(j)}}{\mathcal{N}(\underline{\mathbf{x}}_{0,s}^{(i,j)}; \underline{\mathbf{x}}_s^{(j)*}, \underline{\mathbf{P}}_s^{(j)})} \underline{\omega}_{0,s}^{(i,j)}, i \in \{1, \dots, N_p\}$$

with mapping factor $\underline{\theta}_{m,s}^{(j)}$ (cf. (3.12)). Note that the sets of weighted particles $\{\underline{\mathbf{x}}_s^{(i,m,j)}, \underline{\omega}_s^{(i,m,j)}\}_{i=1}^{N_p}$, $m \in \{1, \dots, M_s\}$, despite all being based on a different proposal PDF $q_{\text{PFI}}(\underline{\mathbf{x}}_s^{(j)} | \mathbf{z}_s^{(m)})$, $m \in \{1, \dots, M_s\}$, still represent $\alpha_s^{(j)}(\underline{\mathbf{x}}_s^{(j)}, 1)$. The result of this particle migration along the flow defined by a measurement $\mathbf{z}_s^{(m)}$, $m \in \{1, \dots, M_s\}$, is that the particles are now at locations where the evaluation of the corresponding likelihood function $f(\mathbf{z}_s^{(m)} | \underline{\mathbf{x}}_s^{(j)})$ will produce a significant particle weight. Particle degeneracy and thus approximate message-passing operations accurately even if the dimension of the state is high [64].

The measurement evaluation step can now be performed on weighted particles $\{\{\underline{\mathbf{x}}_s^{(i,a,j)}, \underline{\omega}_s^{(i,a,j)}\}_{i=1}^{N_p}\}_{a=0}^{M_s}$ by calculating an approximation $\tilde{\beta}_s^{(j)}(a)$ of the messages $\beta_s^{(j)}(a)$ in (2.14) for all $j \in \{1, \dots, J_{s-1}\}$, $a \in \{0, \dots, M_s\}$ as

$$\tilde{\beta}_s^{(j)}(a_s^{(j)} = a) = \sum_{i=1}^{N_p} q(\underline{\mathbf{x}}_s^{(i,a,j)}, 1, a; \mathbf{z}_s) \underline{\omega}_s^{(i,a,j)} + 1(a) (1 - \tilde{p}_s^{(j)}).$$

For the computation of a particle representation of new PO states, we first draw particles $\bar{\mathbf{x}}_{0,s}^{(i)}$, $i \in \{1, \dots, N_p\}$ from $f_b(\bar{\mathbf{x}})$ as introduced in [Sec. 1.1][127]. Next, for each new PO $j = J_{s-1} + m$, $m \in \{1, \dots, M_s\}$, new particles and corresponding weights $\{\bar{\mathbf{x}}_s^{(m,i)}, \bar{w}_s^{(m,i)}\}_{i=1}^{N_p}$ are obtained from $\{\bar{\mathbf{x}}_{0,s}^{(i)}\}_{i=1}^{N_p}$ by performing the invertible PFI $\bar{\mathbf{x}}_{0,s} \rightarrow \mathbf{z}_s^{(m)} \rightarrow \bar{\mathbf{x}}_s^{(m)}$. Note that this flow relies on the mean $\bar{\mathbf{x}}_b$ and covariance matrix $\bar{\mathbf{P}}_b$ of $f_b(\bar{\mathbf{x}})$. Finally, for each $m \in \{1, \dots, M_s\}$ approximate messages $\tilde{\xi}_s^{(m)}(b_s^{(m)})$ are calculated from $\{\bar{\mathbf{x}}_s^{(m,i)}, \bar{w}_s^{(m,i)}\}_{i=1}^{N_p}$ by performing the same steps as described above for the calculation of $\tilde{\beta}_s^{(j)}(a_s^{(j)})$. These messages are used as an input for the iterative BP for data association [128, 64, 65] performed next (see [65, Sec. VI] for details).

3.4.4 Measurement Update and Belief Calculation

After the iterative loopy BP for data association has been converged, the messages $\tilde{\kappa}_s^{(j)}(a_s^{(j)})$, $j \in \{1, \dots, J_{s-1}\}$ and $\tilde{\iota}_s^{(m)}(b_s^{(m)})$, $m \in \{1, \dots, M_s\}$ are available. These messages are used to obtain an approximation $\tilde{\gamma}_s^{(j)}(\mathbf{x}_s^{(j)}, 1)$ of the messages $\gamma_s^{(j)}(\mathbf{x}_s^{(j)}, 1)$, $j \in \{1, \dots, J_{s-1}\}$ in (2.16) as well as an approximation $\tilde{\zeta}_s^{(m)}(\mathbf{x}_s^{(m)}, 1)$ of the messages $\zeta_s^{(m)}(\mathbf{x}_s^{(m)}, 1)$, $m \in \{1, \dots, M_s\}$ in [65, Section IX].

Beliefs approximating the posterior PDF of POs are now computed by means of importance sampling. As in Section 3.3.2, we use the marginal association probabilities $p(a_s^{(j)}|\mathbf{z}_s)$ to weight the proposal PDFs $q_{\text{PFI}}(\mathbf{x}_s^{(j)}|\mathbf{z}_s^{(m)})$ related to PFI based on measurements $\mathbf{z}_s^{(m)}$, $m \in \{1, \dots, M\}$. Note that in MOT scenarios, accurate approximations of $p(a_s^{(j)}|\mathbf{z}_s)$ can be obtained as $\tilde{p}(a_s^{(j)}|\mathbf{z}_s) \propto \tilde{\beta}_s^{(j)}(a_s^{(j)}) \tilde{\kappa}_s^{(j)}(a_s^{(j)})$ (see [65, Section IX] for details). Consequently, we obtain a new set of reindex particles and weights $\{(\mathbf{x}_s^{(l,j)}, \underline{\omega}_s^{(l,j)})\}_{l=1}^{L_s}$ from $\{(\mathbf{x}^{(i,a,j)}, \underline{\omega}^{(i,a,j)})\}_{i=1}^{N_p}$, $a \in \{0, \dots, M_s\}$ by using $l = i(a+1)$, $\underline{\omega}_s^{(l,j)} = \underline{\omega}^{(i,a,j)} / \tilde{p}(a_s^{(j)} = a|\mathbf{z}_s)$, and $L_s = N_p(M_s + 1)$.

Next, based on (2.18), we update the particle weights of the legacy POs $j \in \{1, \dots, J_{s-1}\}$ by first computing

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

and then calculating normalized weights as

$$\underline{w}_s^{(l,j)} = \frac{\underline{w}_s'^{(l,j)}}{\sum_{l=1}^{L_s} \underline{w}_s'^{(l,j)} + (1 - \tilde{p}_s^{(j)}) \tilde{\gamma}_s^{(j)}}, \quad l \in \{1, \dots, L_s\}. \quad (3.22)$$

Note that denominator of (3.22) is a particle-based approximation of $\underline{C}_s^{(j)}$ in (2.18). The resulting particles and weights $\{(\mathbf{x}_s^{(l,j)}, \underline{w}_s^{(l,j)})\}_{l=1}^{L_s}$ represent the belief $\tilde{f}(\mathbf{x}_s^{(j)}, 1)$ of legacy PO $j \in \{1, \dots, J_{s-1}\}$. 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 $\tilde{f}(\mathbf{x}_s^{(j)}, r_s^{(j)}) \triangleq \tilde{f}(\mathbf{x}_s^{(j)}, \underline{r}_s^{(j)})$ is furthermore obtained by applying Algorithm 3, which calculates a mean $\mathbf{x}_s^{(j)*}$ and a covariance

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

```

1   $\left[ \left\{ \left\{ \mathbf{x}_+^{*(h,j)}, \mathbf{P}_+^{(h,j)} \right\}_{h=1}^{N_k}, p_+^{(j)} \right\}_{j=1}^{J_s} \right] =$ 
    SingleSensorUpdate  $\left( \left\{ \left\{ \mathbf{x}_+^{*(h,j)}, \mathbf{P}_+^{(h,j)} \right\}_{h=1}^{N_k}, p_+^{(j)} \right\}_{j=1}^{J_{s-1}}, \mathbf{z}_s \right)$ 
2   $\left[ \left\{ \left\{ \bar{\mathbf{x}}_+^{*(h,m)}, \bar{\mathbf{P}}_+^{(h,m)} \right\}_{h=1}^{N_k}, \xi^{(m)} \right\}_{m=1}^{M_s} \right] = \text{NewObjects}(\mathbf{z}_s)$ 
3  // see Alg. 10
4  for  $j = 1 : J_{s-1}$  do
5   $\left[ \left\{ \left\{ \left\{ \mathbf{x}_+^{(i,h,a,j)}, \mathbf{w}_+^{(i,h,a,j)} \right\}_{i=1}^{N_p}, \boldsymbol{\theta}^{(h,a,j)} \right\}_{h=1}^{N_k}, \boldsymbol{\beta}^{(a,j)} \right\}_{a=0}^{M_s} \right] =$ 
    Evaluation  $\left( \left\{ \mathbf{x}_+^{*(h,j)}, \mathbf{P}_+^{(h,j)} \right\}_{h=1}^{N_k}, p_+^{(j)}, \mathbf{z}_s \right)$ 
6  // see Alg. 7
7   $\left[ \left\{ \left\{ \boldsymbol{\kappa}^{(a,j)} \right\}_{a=0}^{M_s} \right\}_{j=1}^{J_{s-1}}, \left\{ \mathbf{l}^{(a)} \right\}_{a=0}^{M_s} \right] =$ 
    DataAssociation  $\left( \left\{ \left\{ \boldsymbol{\beta}^{(a,j)} \right\}_{a=0}^{M_s} \right\}_{j=1}^{J_{s-1}}, \left\{ \xi^{(a)} \right\}_{a=0}^{M_s} \right)$ 
8  // see [65, Sec. VI]
9  Compute  $\bar{p}_+^{(m)} = \frac{(\xi^{(m)} - 1)\mathbf{l}^{(m)}}{(\xi^{(m)} - 1)\mathbf{l}^{(m)} + 1}, m \in \{1, \dots, M_s\}$ 
11 for  $j = 1 : J_{s-1}$  do
12  $\left[ \left\{ \mathbf{x}_+^{*(h,j)}, \mathbf{P}_+^{(h,j)} \right\}_{h=1}^{N_k}, p_+^{(j)} \right] = \text{Update} \left( \left\{ \left\{ \mathbf{x}_+^{(i,h,a,j)}, \right. \right. \right.$ 
13  $\left. \left. \mathbf{w}_+^{(i,h,a,j)} \right\}_{i=1}^{N_p}, \boldsymbol{\theta}^{(h,a,j)} \right\}_{h=1}^{N_k}, \boldsymbol{\beta}^{(a,j)}, \boldsymbol{\kappa}^{(a,j)} \right\}_{a=0}^M, p_+^{(j)} \right)$ 
14 // see Alg. 8
15 For  $j = 1, \dots, J_{s-1}$ , reindex legacy object state information according to
     $\left\{ \mathbf{x}_+^{*(h,j)}, \mathbf{P}_+^{(h,j)} \right\}_{h=1}^{N_k} = \left\{ \mathbf{x}_+^{*(h,j)}, \mathbf{P}_+^{(h,j)} \right\}_{h=1}^{N_k}$  and  $p_+^{(j)} = p_+^{(j)}$ .
16 For  $j = J_{s-1} + 1, \dots, J_s$ , reindex new object state information following
     $\left\{ \mathbf{x}_+^{*(h,j)}, \mathbf{P}_+^{(h,j)} \right\}_{h=1}^{N_k} = \left\{ \bar{\mathbf{x}}_+^{*(h,m)}, \bar{\mathbf{P}}_+^{(h,m)} \right\}_{h=1}^{N_k}$  and  $p_+^{(j)} = \bar{p}_+^{(m)}$  using  $m = j - J_{s-1}$  and
     $J_s = J_{s-1} + M_s$ .
17 Output:  $\left\{ \left\{ \mathbf{x}_+^{*(h,j)}, \mathbf{P}_+^{(h,j)} \right\}_{h=1}^{N_k}, p_+^{(j)} \right\}_{j=1}^{J_s}$ 

```

matrix $\mathbf{P}_s^{(j)}$ from $\{(\mathbf{x}_s^{(l,j)}, \mathbf{w}_s^{(l,j)} / \tilde{p}_s^{(j)})\}_{l=1}^{L_s}$.

For new POs $j \in \{J_{s-1} + 1, \dots, J_s\}$, approximate existence probabilities $\tilde{p}_s^{(j)}$ and a Gaussian representation of the beliefs $\tilde{f}(\mathbf{x}_s^{(j)}, r_s^{(j)}) \triangleq \tilde{f}(\bar{\mathbf{x}}_s^{(m)}, \bar{r}_s^{(m)})$, 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., $\hat{\mathbf{x}}^{(j)} \approx \mathbf{x}_s^{*(j)}$.

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

```

1   $\left[ \left\{ \left\{ \bar{\mathbf{x}}_+^{*(h,m)}, \bar{\mathbf{P}}_+^{(h,m)} \right\}_{h=1}^{N_k}, \xi^{(m)} \right\}_{m=1}^M \right] = \text{NewObjects}(\mathbf{z}_s)$ 
2  Draw  $\bar{\mathbf{x}}_b^{*(h)}, h = 1, \dots, N_k$  from  $f_b(\bar{\mathbf{x}})$ ;
3  Compute covariance matrix  $\bar{\mathbf{P}}_b$  from  $f_b(\bar{\mathbf{x}})$ ;
4  for  $m = 1 : M_s$  do
5      for  $h = 1 : N_k$  do
6           $\left[ \left\{ \bar{\mathbf{x}}^{(i,h,m)}, \bar{\mathbf{w}}^{(i,h,m)} \right\}_{i=1}^{N_p} \right] =$ 
7               $\text{InvertibleFlow}(\bar{\mathbf{x}}_b^{*(h)}, \bar{\mathbf{P}}_b, \mathbf{z}_s^{(m)});$ 
8               $// \text{ see Alg. 4}$ 
9           $\left[ \sim, \bar{\mathbf{P}}^{(h,m)} \right] = \text{GaussianRepresentation}(\left\{ \bar{\mathbf{x}}^{(i,h,m)}, \bar{\mathbf{w}}^{(i,h,m)} \right\}_{i=1}^{N_p});$ 
10              $// \text{ see Alg. 3}$ 
11       $\xi^{(m)} = 1 + \frac{\mu_b}{N_k N_p \mu_{fp} f_{fp}(\mathbf{z}_s^{(m)})} \sum_{h=1}^{N_k} \sum_{i=1}^{N_p} \bar{\mathbf{w}}^{(i,h,m)};$ 
12       $\left[ \left\{ \bar{\mathbf{x}}_+^{*(h,m)}, \bar{\mathbf{P}}_+^{(h,m)} \right\}_{h=1}^{N_k} \right] =$ 
13           $\text{Resampling}(\left\{ \bar{\mathbf{P}}^{(h,m)}, \left\{ \bar{\mathbf{x}}^{(i,h,m)}, \bar{\mathbf{w}}^{(i,h,m)} \right\}_{i=1}^{N_p} \right\}_{h=1}^{N_k});$ 
14           $// \text{ see Alg. 5}$ 
15  Output:  $\left\{ \left\{ \bar{\mathbf{x}}_+^{*(h,m)}, \bar{\mathbf{P}}_+^{(h,m)} \right\}_{h=1}^{N_k}, \xi^{(m)} \right\}_{m=1}^M$ 

```

3.4.5 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 PFI has to be combined with a GMM discussed in Section 3.3.1. 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 $\left\{ \left\{ \mathbf{x}_{-}^{*(h,j)}, \mathbf{P}_{-}^{(h,j)} \right\}_{h=1}^{N_k}, p_{-}^{(j)} \right\}_{j=1}^{J_-}$. Note that in a GMM, an approximate MMSE estimate can be obtained as $\hat{\mathbf{x}}^{(j)} \approx \frac{1}{N_k} \sum_{h=1}^{N_k} \mathbf{x}_S^{*(h,j)}$.

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 $\mathcal{O}(N_\lambda N_k N_p (\sum_{s=1}^S M_s))$. The complexity of a conventional bootstrap implementation per object and time step is $\mathcal{O}(N_b (\sum_{s=1}^S M_s))$, 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 Sections 3.4, 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. And as we will show in the next section, traditional particle filter typically requires $N_b \gg N_\lambda N_k N_p$ to achieve an acceptable estimation accuracy in 3D multisensor MOT.

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

```

1   $\left[ \left\{ \left\{ \mathbf{x}^{*(h,j)}, \mathbf{P}^{(h,j)} \right\}_{h=1}^{N_k}, p^{(j)} \right\}_{j=1}^J \right] =$ 
   MultisensorMOT  $\left[ \left\{ \left\{ \mathbf{x}_{-}^{*(h,j)}, \mathbf{P}_{-}^{(h,j)} \right\}_{h=1}^{N_k}, p_{-}^{(j)} \right\}_{j=1}^{J_-}, \mathbf{z} \right]$ 
2  Perform prediction step according to
3  for  $j = 1 : J_-$  do
4      for  $h = 1 : N_k$  do
5           $\underline{\mathbf{x}}_1^{*(h,j)} = \mathbf{G} \mathbf{x}_{-}^{*(h,j)} + \mathbf{u}^*$ 
6           $\underline{\mathbf{P}}_1^{(h,j)} = \mathbf{G} \mathbf{P}_{-}^{(h,j)} \mathbf{G}^T + \mathbf{P}_u$ 
7       $\underline{p}_1^{(j)} = p_{su} p_{-}^{(j)}$ 
8  Perform single sensor update steps sequentially, i.e.,
9  for  $s = 1 : S$  do
10      $\left[ \left\{ \left\{ \mathbf{x}_s^{*(h,j)}, \mathbf{P}_s^{(h,j)} \right\}_{h=1}^{N_k}, p_s^{(j)} \right\}_{j=1}^{J_s} \right] =$ 
        SingleSensorUpdate  $\left[ \left\{ \left\{ \underline{\mathbf{x}}_s^{*(h,j)}, \underline{\mathbf{P}}_s^{(h,j)} \right\}_{h=1}^{N_k}, \underline{p}_s^{(j)} \right\}_{j=1}^{J_{s-1}}, \mathbf{z}_s \right]$ 
11      $// \text{ see Alg. 9}$ 
12 Set  $\left\{ \left\{ \mathbf{x}^{*(h,j)}, \mathbf{P}^{(h,j)} \right\}_{h=1}^{N_k}, p^{(j)} \right\}_{j=1}^J = \left\{ \left\{ \mathbf{x}_S^{*(h,j)}, \mathbf{P}_S^{(h,j)} \right\}_{h=1}^{N_k}, p_S^{(j)} \right\}_{j=1}^{J_S}$  where  $J = J_S$ 
13 Output:  $\left\{ \left\{ \mathbf{x}^{*(h,j)}, \mathbf{P}^{(h,j)} \right\}_{h=1}^{N_k}, p^{(j)} \right\}_{j=1}^J$ 

```

3.5 Experiments and Numerical Results

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

3.5.1 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., $\mathbf{x}_k^{(j)} = [x_{1,k}^{(j)} x_{2,k}^{(j)} x_{3,k}^{(j)} \dot{x}_{1,k}^{(j)} \dot{x}_{2,k}^{(j)} \dot{x}_{3,k}^{(j)}]^T$, $j = 1, \dots, 8$ and evolve according to a constant-velocity model [1, 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 $[-1000\text{m}, 1000\text{m}] \times [-1000\text{m}, 1000\text{m}] \times [-1500\text{m}, -500\text{m}]$. 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 data association, 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 $\sqrt{x_{1,0}^{(1)2} + x_{2,0}^{(1)2}} = 50$ m and $x_{3,0}^{(1)} = -1000$ m. The velocity is obtained by setting the vertical speed to zero and the horizontal velocity vector pointing to the circle center with $\sqrt{\dot{x}_{1,0}^{(1)2} + \dot{x}_{2,0}^{(1)2}} = 0.3\text{m/s}$. For the appearance of each further object $j = 2, \dots, 8$, 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 data association 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 $\mu_b = 0.05$. The prior PDF for newborn object, $f_b(\bar{\mathbf{x}}_k)$, is uniform on the ROI for the 3-D position and Gaussian with zero mean and covariance matrix $5\text{m}^2/\text{s}^2 \mathbf{I}_3$ for the 3-D velocity. The survival probability is $p_{\text{su}} = 0.95$. The object declaration threshold is set to $P_{\text{th}} = 0.5$ and the pruning threshold to $P_{\text{pr}} = 10^{-4}$.

The two hydrophone arrays have the geometry of the array described in [141] and are located at $[519\text{m } 137\text{m } -1300\text{m}]^T$ and $[-519\text{m } -137\text{m } -1300\text{m}]^T$, 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 $\mathbf{z}_{k,s}^{(m)}$, $m \in \{1, \dots, M_{k,s}\}$ and $s \in \{1, \dots, 12\}$ are obtained by the two arrays at time k . At each sensor s , a random number of $M_{k,s}$ measurements are generated. In particular, the TDOA measurement $\mathbf{z}_{k,s}^{(m)}$ of a detected object with state $\mathbf{x}_k^{(j)}$ is modeled as $\mathbf{z}_{k,s}^{(m)} = \frac{1}{c} \left(\left\| [\mathbf{x}_{1,k}^{(j)} \mathbf{x}_{2,k}^{(j)} \mathbf{x}_{3,k}^{(j)}]^T - \mathbf{p}_{s_L} \right\| - \left\| [\mathbf{x}_{1,k}^{(j)} \mathbf{x}_{2,k}^{(j)} \mathbf{x}_{3,k}^{(j)}]^T - \mathbf{p}_{s_R} \right\| \right) + \mathbf{v}_{k,s}^{(m)}$ where $\mathbf{p}_{s_L}$ and $\mathbf{p}_{s_R}$ are the paired receiver positions of sensor s , $c = 1500\text{m/s}$ is the propagation speed, and $\mathbf{v}_{k,s}^{(m)}$ 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, $f_{\text{fp}}(\mathbf{z}_{k,s}^{(m)})$, is uniform on $\frac{1}{c} [-\|\mathbf{p}_{s_L} - \mathbf{p}_{s_R}\|, \|\mathbf{p}_{s_L} - \mathbf{p}_{s_R}\|]$.

We compare the proposed BP-based MOT with the embedded particle flow sampling strategy (“BP-PF”) with two reference sampling strategies. The first (“BP-PM”) follows the sampling strategy of the bootstrap particle filter [42, 60] and uses predicted beliefs as proposal PDF [64]. The second (yet unpublished) (“BP-UT”) follows the sampling strategy of the unscented particle filter [83], i.e., it uses a Gaussian mixture representation and the unscented transformation to calculate an informative proposal PDFs. For BP-PM, we use $N_b = 10^6$ particles for newborn POs and $N_b = 6 \cdot 10^4$ particles for legacy POs. For all other simulated methods, we use $N_k = 100$ kernels. For each kernel representing a newborn PO, we set $N_p = 500$ for BP-UT and BP-PF and for each kernel representing a legacy PO, we set $N_p = 30$ for BP-UT and BP-PF.

Since $N_b/N_p N_k = 20$ the memory requirements of BP-PM are 20 times higher compared to BP-PF and BP-UT.

We also simulate two variants of the second reference method to obtain a similar runtime for BP-PF. In particular, for “BP-UT-1”, we set $N_p = 4000$ for kernels representing newborn POs and $N_p = 250$ for kernels representing legacy POs. Furthermore, for “BP-UT-2”, we set $N_p = 6000$ for kernels representing newborn PO and $N_p = 30$ for kernels representing legacy PO. Note that the memory requirements of BP-UT-1 and BP-UT-2 are higher than the ones of BP-UT and BP-PF due to their higher values of N_p . Finally, we also simulate a method (“BP-PF-H”) that uses the sampling strategy of BP-PF for kernels representing newborn POs and the sampling strategy of BP-PM for kernels representing legacy POs. Using fewer samples for the kernels representing legacy PO, or even using the strategy of BP-PF, is motivated by the fact that the beliefs of legacy POs are typically unimodal and quite informative. We set $N_\lambda = 20$ for BP-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 σ_w , (ii) mean number of FPs μ_{fp} , (iii) detection probability p_d and (iv) measurement noise standard deviation σ_v . Note that for the setting $\sigma_v = 5 \times 10^{-7}$ 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 $C = 50$ [142].

3.5.2 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 $\sigma_w = 1 \text{ m/s}^2$. In addition, to obtain a scenario with an informative likelihood

function, we consider the case $\sigma_v = 5 \times 10^{-7}$ s. Fig. 3.3 and Fig. 3.4 show the 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 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., BP-PF and BP-PF-H, have lower error peaks at the time steps when objects appear, i.e., BP-PF and BP-PF-H can often declare the existence of an object faster. In addition, BP-PF and BP-PF-H can outperform the other reference methods at almost all time steps. BP-PM, in particular performs very poorly due to particle degeneracy. It can be noted that BP-UT-1 and BP-UT-2 have improved performance compared to BP-UT but are still outperformed by BP-PF and BP-PF-H despite their more extensive memory requirements.

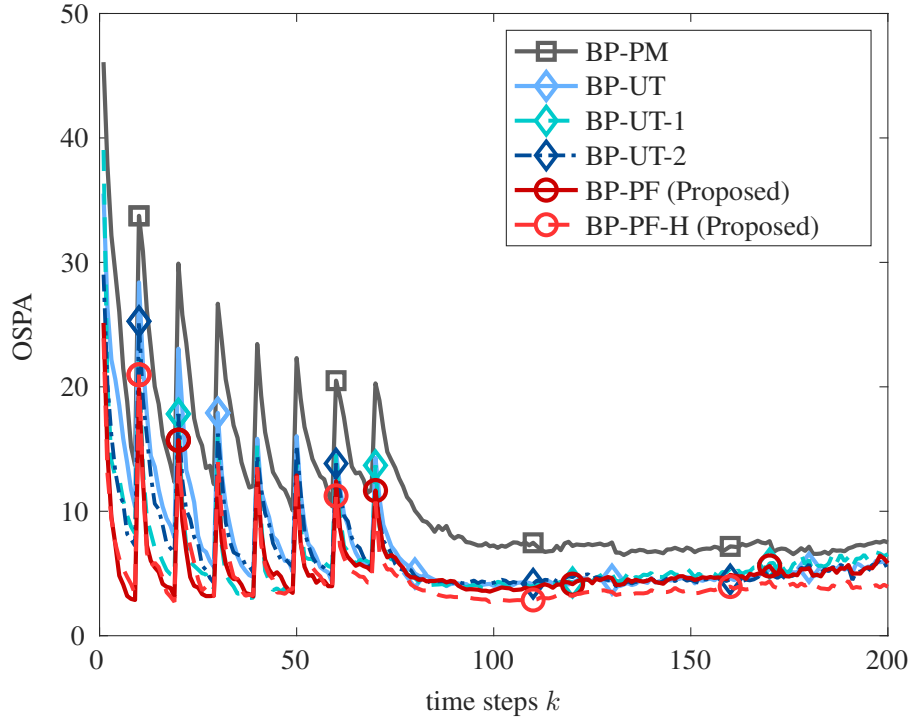


Figure 3.3. OSPA performance for high uncertainty of prior with $\sigma_w = 1$ m/s². Other parameters are set as $\mu_{fp} = 5$, $\sigma_v = 10^{-6}$ s, $p_d = 0.9$.

Furthermore, the MOSPA error and the runtime per time step of the six methods for different system parameter values are shown in Table 3.2. The default value of the four parameters are $\sigma_w = 0.1 \text{ m/s}^2$, $\sigma_v = 1 \times 10^{-6} \text{ s}$, $\mu_{fp} = 5$ and $p_d = 0.9$. 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 BP-PM significantly degrades when measurement noise standard deviation is reduced to $\sigma_v = 5 \times 10^{-7} \text{ s}$. This unwanted and counterintuitive behavior is a clear indicator of particle degeneracy in BP-PM. Only with the proposed method is it possible to yield improved tracking performance as measurement noise variance is reduced.

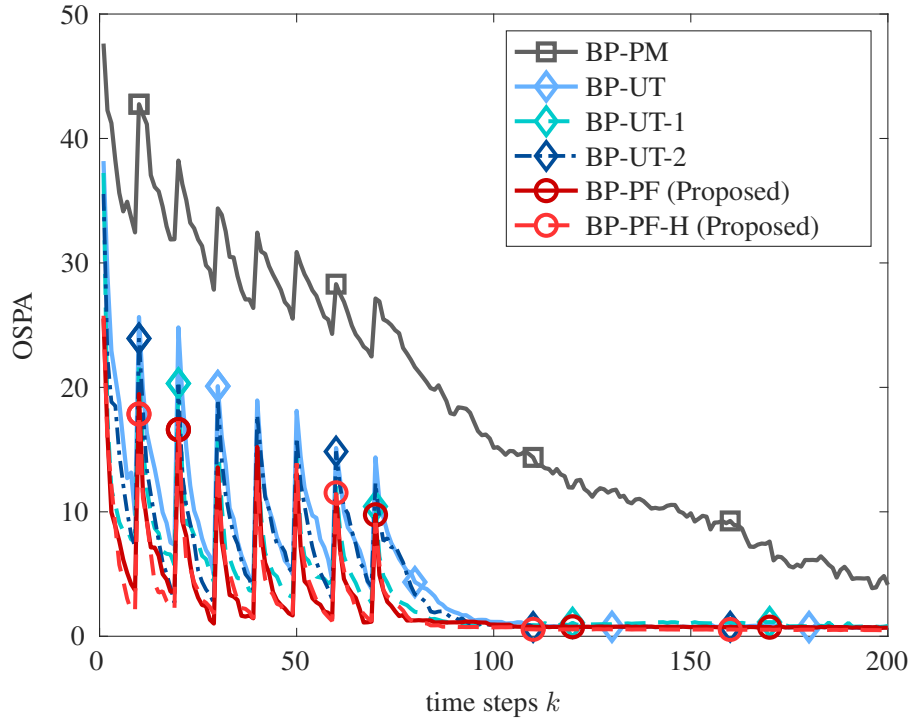


Figure 3.4. OSPA performance for informative measurement model with $\sigma_v = 5 \times 10^{-7} \text{ s}$. Other parameters are set as $\mu_{fp} = 5$, $\sigma_w = 0.1 \text{ m/s}^2$, $p_d = 0.9$.

It can be seen that for almost all system parameter values, the proposed BP-PF and

BP-PF-H outperform the reference methods. At the same time, their runtime is comparable with BP-UT-1 and BP-UT-2, which yield higher memory requirements. The only scenario where BP-PF and BP-PF-H do not result in the lowest MOSPA is when $\sigma_v = 2 \times 10^{-6}$ s. In this case, since measurements are not very informative, BP-PM does not suffer from particle degeneracy. It can also be seen that BP-PF-H outperforms BP-PF both in terms of MOSPA and runtime, while both BP-PF and BP-PF-H typically outperform BP-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 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 graphical processing units (GPUs), can strongly reduce runtimes by exploiting the highly parallelizable nature of PFI.

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 $N_k \in \{1, 5, 10, 50, 100\}$ in BP-PF-H and compare BP-PF-H for these different values of N_k with BP-PM. The parameters $\sigma_v, \mu_{fp}, \sigma_w$, and p_d are set as discussed above. For comparable runtimes of all BP-PF-H variants, we set $N_p \in \{100000, 20000, 8000, 1300, 500\}$ for newborn object and $N_p \in \{15000, 1000, 400, 70, 30\}$ for legacy objects. As expected, it can be seen in Fig. 3.5 that the accuracy of BP-PF-H improves with increasing N_k . While BP-PF-H performs very poorly for $N_k = 1$, notably, it can already outperform BP-PM for $N_k = 5$. The average runtime per time step of BP-PF-H is 27.14s, 15.06s, 10.85s, 11.35s, and 12.72s for $N_k = 1, N_k = 5, N_k = 10, N_k = 50$, and $N_k = 100$, respectively, as well as 22.66s for BP-PM.

3.5.3 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)[143], each of which is equipped with

Table 3.2. MOSPA and runtime per time step of different algorithms for the considered tracking scenario w.r.t related parameters.

Parameter	Value	BP-bootstrap		BP-UT		BP-UT-1		BP-UT-2		BP-PF		BP-PF-H	
		OSPA	Runtime (s)	OSPA	Runtime (s)	OSPA	Runtime (s)	OSPA	Runtime (s)	OSPA	Runtime (s)	OSPA	Runtime (s)
σ_w	1 m/s ²	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/s ²	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/s ²	7.61	23.99	4.82	3.35	3.45	21.59	4.35	29.10	2.96	22.48	2.67	13.68
σ_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

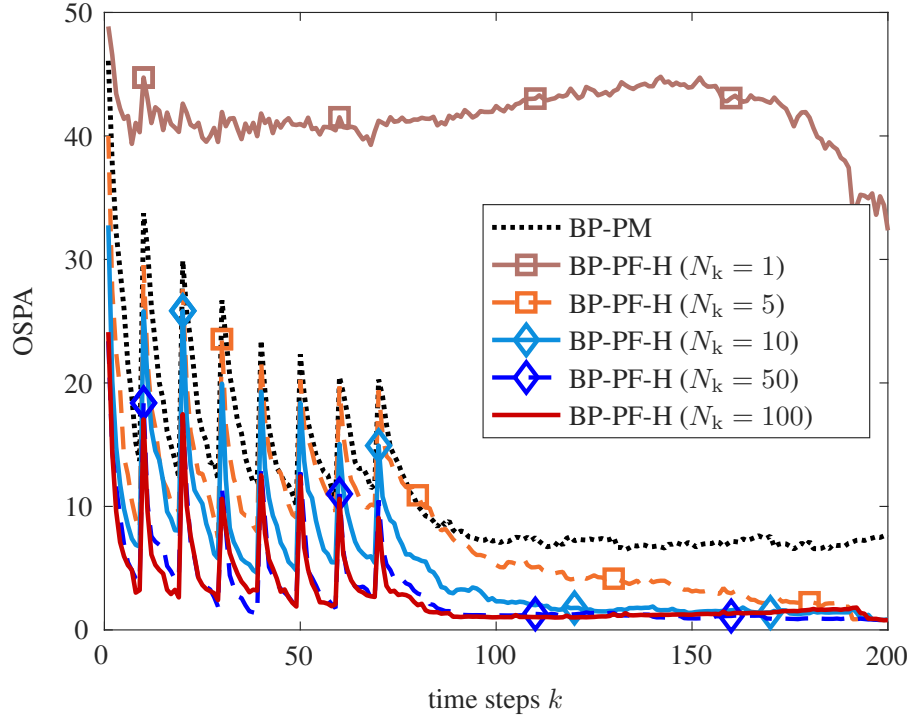


Figure 3.5. OSPA performance of BP-PM and BP-PF-H for different number number of kernels, N_k .

four hydrophones. HARPs are deployed at a depth of 1330 m and approximately 1 km apart. Preprocessing of acoustic data is described in [4]. Each pair of hydrophones on each HARP acts as a sensor that provides TDOA measurements every 7s. 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. 3.6. The red solid lines show the estimated tracks of two whales provided by BP-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 Figure 3.7, we show the estimated tracks of BP-PF-H compared with BP-PM in 2-D. BP-PM does break and merge tracks, which makes it very difficult to determine how many

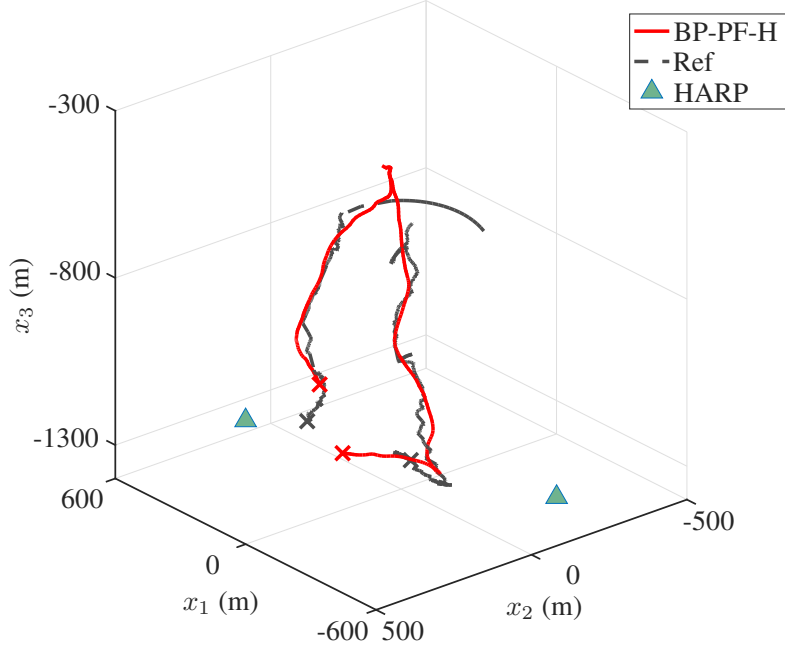


Figure 3.6. Underwater acoustic tracking scenario with two Cuvier’s beaked whales.

whales are actually there. Most importantly, BP-PF-H can potentially replace the human operator, while BP-PM cannot. The overall number of particles of BP-PM is 20 times that of BP-PF-H. The runtime per time step of BP-PM is 4.65 s while 2.35 s of BP-PF-H, i.e., BP-PF-H is faster than BP-PM. Since the measurement interval is 7 s, both BP-PF-H and BP-PM can be used in real-time. For larger scenarios with more than two whales, a GPU implementation is required for real-time processing.

3.6 Summary

In this chapter, we propose a sampling efficient nonparametric method for multisensor MOT that can achieve a superior runtime–estimation accuracy tradeoff in nonlinear and high-dimensional problems. The improved tradeoff is obtained by carefully embedding invertible PFI. 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

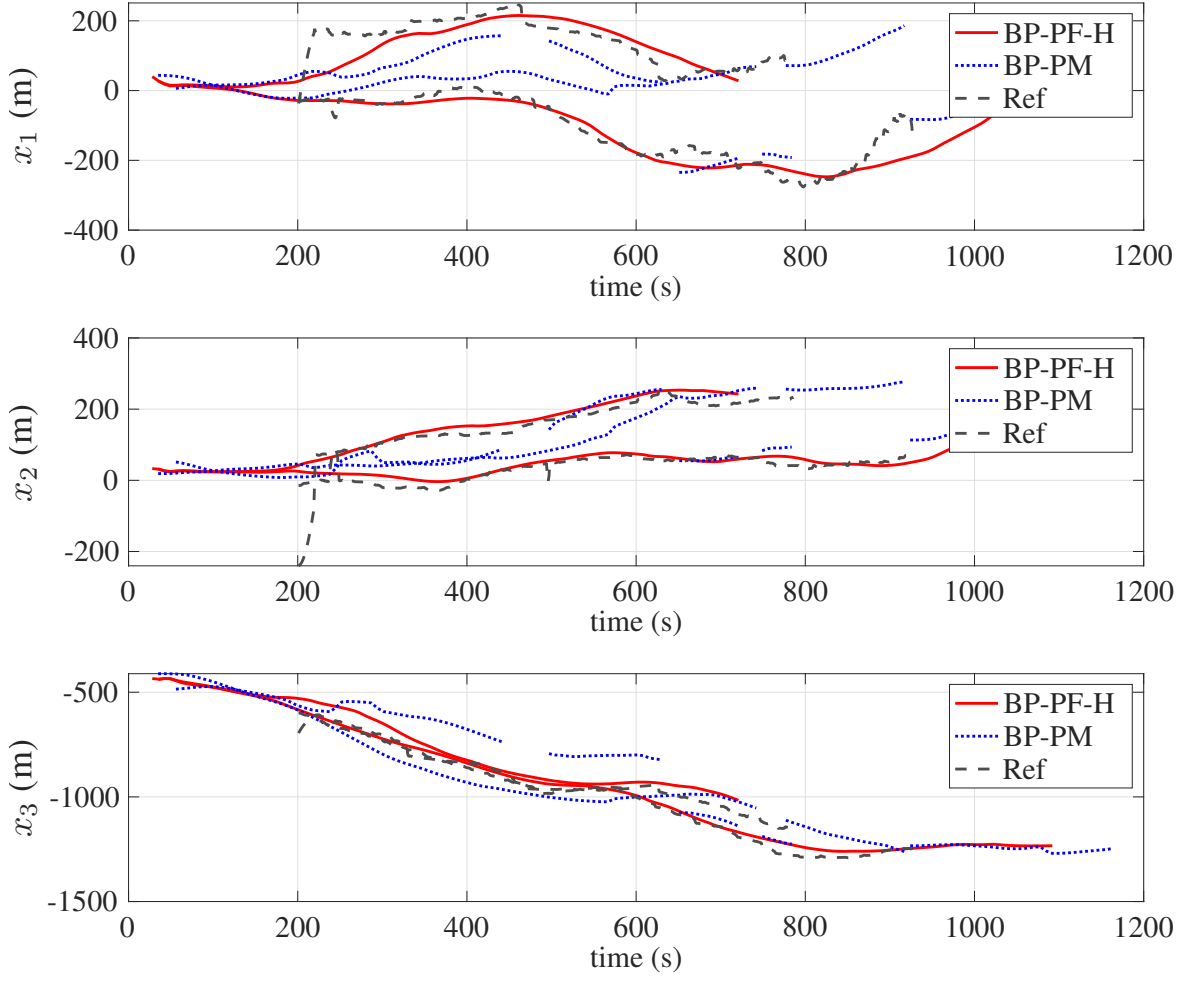


Figure 3.7. A comparison between the estimated tracks provided by BP-PF-H and BP-PM. The hand-annotated tracks of the two whales are marked for reference (gray dashed line). Each axis of the 3-D domain is shown individually.

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 BP-based computation of beliefs is asymptotically optimal in the sense that the resulting particle representation of the beliefs provided by the BP 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 based on PFI.

Chapter 4

Stochastic Particle Flow

In Chapter 3, we propose to use PFI to improve sampling efficiency for multisensor MOT problem. In this chapter, we focus on the design of PFIs with improved transient dynamics. More specifically, we aim to reduce the stiffness of PDE or SDE defined by the homotopy function¹.

Existing work on applying PFI for IS is mainly limited to a specific closed-form solution, while designing a general stochastic PFI for IS is still an open problem. We start by introducing the general form of stochastic PFI. Then we propose an invertible method to evaluate the proposal based on the general stochastic PFI. Furthermore, we discuss methods to improve the transient dynamics of PFI by optimizing the diffusion term of the stochastic PFI. The resulting IS framework is combined with a GMM and used within a BP framework for the detection and localization of an unknown number of sources in 3-D. The main contributions of our work are summarized as follows.

- We develop efficient IS where the parameters of a GMM are migrated by stochastic PFIs that rely on an SDE with an optimizable diffusion term.
- We evaluate our method in a challenging 3-D multi-source localization problem and demonstrate significant improvements compared to state-of-the-art methods.

¹This chapter is based on W. Zhang, M. J. Khojasteh, N. A. Atanasov, and F. Meyer, “Importance sampling with stochastic particle flow and diffusion optimization,” *IEEE Signal Process. Lett.*, 2025 [144] and W. Zhang, M. J. Khojasteh, and F. Meyer, “Particle flows for source localization in 3-D using TDOA measurements,” in *Proc. FUSION-24*, (Venice, Italy), 2024 [145].

4.1 Stochastic PFI

Let's rewrite the Langevin SDE of the form in (3.2)

$$d\mathbf{x} = \boldsymbol{\zeta}(\mathbf{x}, \lambda) d\lambda + \mathbf{Q}(\lambda)^{1/2} d\mathbf{w}.$$

Here, $\boldsymbol{\zeta}(\mathbf{x}, \lambda) \in \mathbb{R}^N$ is the drift vector, $\mathbf{Q}(\lambda) \in \mathbb{R}^{N \times N}$ is the diffusion matrix, and $\mathbf{w} \in \mathbb{R}^N$ is Brownian motion. The constrained Fokker-Plank equation can be used to derive the functional forms of $\boldsymbol{\zeta}(\mathbf{x}, \lambda)$ and $\mathbf{Q}(\lambda)$ [117]. In Chapter 3, we use EDH PFI, which is a well-known solution of the deterministic PFI by setting diffusion matrix as zero.

Recent results demonstrate that, based on an appropriate choice of the diffusion term $\mathbf{Q}(\lambda)$, stochastic PFI can provide a strongly reduced number of integration steps and computational complexity compared to deterministic PFI [146]. A popular stochastic flow is Gromov's flow [117], given by the drift

$$\boldsymbol{\zeta}_g(\mathbf{x}, \lambda) = -(\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \phi)^{-1} \nabla_{\mathbf{x}} \log l \quad (4.1)$$

and the diffusion matrix

$$\mathbf{Q}_g(\lambda) = -(\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \phi)^{-1} (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log l) (\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \phi)^{-1}. \quad (4.2)$$

Here, we used the short notation $\phi \triangleq \phi(\mathbf{x}, \lambda)$ and $l \triangleq l(\mathbf{x})$ as originally defined in (3.1).

For a linear measurement model $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$ with zero-mean additive Gaussian noise $\mathbf{v}$ and covariance matrix $\mathbf{R}$, we have $\nabla_{\mathbf{x}} \log l = \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x})$, $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \log l = -\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$, and $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \phi = -\mathbf{P}^{-1} - \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$.

We can now rewrite drift (4.1) and diffusion (4.2) of Gromov's flow as

$$\boldsymbol{\zeta}_g(\mathbf{x}, \lambda) = (\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x})$$

$$\begin{aligned} \mathbf{Q}_g(\lambda) &= (\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}) \\ &\quad \times (\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1}. \end{aligned} \quad (4.3)$$

For a nonlinear measurement model $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$, one can use the solution of Gromov's flow in (4.3) based on a local linearization of the measurement function $\mathbf{h}(\cdot)$. Gromov's flow applied to linearized measurement models has been demonstrated to outperform deterministic flows [147, 145, 148].

4.2 IS With Stochastic PFI

To get asymptotic optimal solution, we want to use PFI to develop a “flow-induced” proposal PDF similar to what we did in Section 3.2. However, existing work to use PFI for IS typically limited to deterministic flow, or more specifically EDH flow since it satisfies an invertible mapping between PDF before and after the flow. However, for stochastic PFI it's intractable to get an invertible mapping due to the non-zero diffusion term. In [88] particles are drawn at each flow step from a different proposal PDF, which leads to a computationally expensive weights computation for IS. The work in [149, 147] utilizes auxiliary variables and their filtering for IS with embedded PFI. In particular, [147] extended this framework to the stochastic Gromov's flow. Although the method in [147] is computationally less expensive than [88], it has twice computational complexity of PFI due to an additional auxiliary variable for each particle. Furthermore, it relies on heuristic and suboptimal diffusion matrix selections without optimization. Hence it's still an open question for stochastic PFI to be applied to IS to improve sampling efficiency. In the remaining of this section, we propose a more computationally affordable method to evaluate the proposal using a general form of stochastic PFI.

Let us first transform the drift of Gromov's flow to an affine function, i.e., $\boldsymbol{\zeta}_g(\mathbf{x}, \lambda) = \mathbf{A}_g(\lambda)\mathbf{x} + \mathbf{b}_g(\lambda)$, with

$$\begin{aligned}\mathbf{A}_g(\lambda) &= -\left(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \\ \mathbf{b}_g(\lambda) &= \left(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z}.\end{aligned}$$

Next, consider a single Gaussian $f(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_0, \mathbf{P}_0)$ with mean $\boldsymbol{\mu}_0$ and covariance matrix $\mathbf{P}_0$ as predicted/prior PDF. Based on the affine form introduced above, we can migrate the mean and covariance of this Gaussian predicted/prior PDF based on PFI, i.e.,

$$\boldsymbol{\mu}_l = \boldsymbol{\mu}_{l-1} + \boldsymbol{\zeta}_s(\boldsymbol{\mu}_{\lambda_{l-1}}, \lambda_l) \Delta_l \quad (4.4)$$

$$\mathbf{P}_l = [\mathbf{I} + \Delta_l \mathbf{A}_s(\lambda_l)] \mathbf{P}_{l-1} [\mathbf{I} + \Delta_l \mathbf{A}_s(\lambda_l)]^T + \Delta_l \mathbf{Q}(\lambda_l) \quad (4.5)$$

for $l = 1, \dots, N_\lambda$ and by setting $\boldsymbol{\zeta}_s(\boldsymbol{\mu}_{\lambda_{l-1}}, \lambda_l) = \boldsymbol{\zeta}_g(\boldsymbol{\mu}_{\lambda_{l-1}}, \lambda_l)$ and $\mathbf{A}_s = \mathbf{A}_g$. The resulting $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \mathbf{P}_1)$ can be used as a “flow-induced” proposal PDF for asymptotically optimal IS. This principle can be extended to GMMs in a straightforward way. Assuming a prior/predicted PDF represented by a GMM, i.e., $f(\mathbf{x}) = \sum_{n=1}^{N_g} w^{(n)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_0^{(n)}, \mathbf{P}_0^{(n)})$, one can use (4.4) and (4.5) to perform N_g PFIs, one for each GMM component. The resulting proposal still follows a GMM as $q(\mathbf{x}) = \sum_{n=1}^{N_g} w^{(n)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1^{(n)}, \mathbf{P}_1^{(n)})$.

For an accurate numerical implementation of the PFI as (3.4), the step sizes, Δ_l , need to be adapted to the stiffness of the flow. A flow with reduced stiffness can be implemented with larger step sizes, i.e., fewer steps, and thus yields reduced computational complexity. Next, we investigate how to develop stochastic PFIs with reduced stiffness. Recently, it has been shown that a solution to (3.2) can be obtained by choosing $\mathbf{Q}(\lambda)$ arbitrarily and computing $\boldsymbol{\zeta}_s(\mathbf{x}, \lambda)$ according to [146]

$$\boldsymbol{\zeta}_s(\mathbf{x}, \lambda) = \boldsymbol{\zeta}_d(\mathbf{x}, \lambda) + \frac{1}{2} \mathbf{Q}(\lambda) \nabla_{\mathbf{x}} \phi(\mathbf{x}, \lambda) \quad (4.6)$$

where $\boldsymbol{\zeta}_d(\mathbf{x}, \lambda)$ is the drift of a deterministic flow, i.e., $d\mathbf{x} = \boldsymbol{\zeta}(\mathbf{x}, \lambda) d\lambda$. Based on this result, one can choose a deterministic PFI and then design a diffusion matrix to get a stochastic PFI with reduced stiffness.

Consider Gromov's flow with stochastic drift and diffusion as in (4.1) and (4.2). Based on (4.6), the corresponding deterministic drift can be obtained as

$$\boldsymbol{\zeta}_{\text{d-g}}(\mathbf{x}, \lambda) = \boldsymbol{\zeta}_{\text{g}}(\mathbf{x}, \lambda) - \frac{1}{2} \mathbf{Q}_{\text{g}}(\lambda) \nabla_{\mathbf{x}} \phi(\mathbf{x}, \lambda). \quad (4.7)$$

By using an arbitrary diffusion matrix $\mathbf{Q}(\lambda)$ and by substituting $\boldsymbol{\zeta}_{\text{d}}(\mathbf{x}, \lambda)$ in (4.6) by $\boldsymbol{\zeta}_{\text{d-g}}(\mathbf{x}, \lambda)$ in (4.7), a new stochastic drift can be developed as

$$\boldsymbol{\zeta}_{\text{s}}(\mathbf{x}, \lambda) = \boldsymbol{\zeta}_{\text{g}}(\mathbf{x}, \lambda) + \frac{1}{2} (\mathbf{Q}(\lambda) - \mathbf{Q}_{\text{g}}(\lambda)) \nabla_{\mathbf{x}} \phi(\mathbf{x}, \lambda). \quad (4.8)$$

By considering a linearized measurement model, which results in $\nabla_{\mathbf{x}} \phi(\mathbf{x}, \lambda) = \mathbf{P}^{-1}(\boldsymbol{\mu}_0 - \mathbf{x}) + \lambda \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x})$, we can transform (4.8) to an affine function, i.e.,

$$\begin{aligned} \mathbf{A}_{\text{s}}(\lambda) = & -\frac{1}{2} \left(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \\ & - \frac{1}{2} \mathbf{Q}(\lambda) (\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}) \end{aligned} \quad (4.9)$$

$$\begin{aligned} \mathbf{b}_{\text{s}}(\lambda) = & \left(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z} \\ & + \frac{1}{2} (\mathbf{Q}(\lambda) - \mathbf{Q}_{\text{g}}(\lambda)) (\mathbf{P}^{-1} \boldsymbol{\mu}_0 + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z}). \end{aligned}$$

Based on (4.4) and (4.5), this affine function can again be used to migrate the means and covariances of a GMM representing a predicted/prior PDF based on PFI.

The arbitrariness in the choice of the diffusion matrix in (4.8) enables us to design a diffusion matrix such that stiffness is reduced and numerical efficiency is improved. In particular, the transient dynamics of the PFI are measured by the condition number $\kappa(\cdot)$ of the nonsingular matrix $\mathbf{A}_{\text{s}}(\lambda)$ in (4.9), i.e., the ratio of the largest singular value to the smallest singular value of nonsingular $\mathbf{A}_{\text{s}}(\lambda)$. A small condition number close to one usually implies

a reduced stiffness of the SDE [134]. Consider the following function form of $\mathbf{Q}(\lambda)$ in (4.9), i.e., $\mathbf{Q} = c(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1}$ where c is a constant. We can get $\lim_{c \rightarrow \infty} \kappa(\mathbf{A}_s) = 1$ for $\lim_{c \rightarrow \infty} \|\mathbf{Q}\| = \infty$. However, we cannot make $\|\mathbf{Q}\|$ too large, since it defines an upper bound of the numerical integration error using the Euler-Maruyama method [146].

To balance the stiffness reduction and the error of numerical evaluation of the SDE, we adopt the objective function [146] $J(\mathbf{Q}) = \kappa(\mathbf{A}) + \alpha c$ where α is a hyperparameter. The optimal solution $\mathbf{Q}^*$ obtained by minimizing $J(\mathbf{Q})$ is $\mathbf{Q}^* = c^*(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1}$, with $c^* = \max \{ \sqrt{k_{\max} - k_{\min}} / \sqrt{\alpha} - k_{\min}, 0 \}$, where $k_{\max}$ and $k_{\min}$ are the absolute values of the largest and smallest eigenvalue of the negative Jacobian matrix of $\boldsymbol{\zeta}_{\text{d-g}}(\mathbf{x}, \lambda)$ in (4.7), i.e., $-\mathbf{A}_{\text{d-g}} = \frac{1}{2}(\mathbf{P}^{-1} + \lambda \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$. For a detailed proof, see [146]. The resulting flow induced proposal PDF can be used for IS in a traditional particle filter [60], for Monte Carlo integration [76], or for the computation of messages in BP-based estimation framework [150].

4.3 Numerical Experiments and Results

In this section, we evaluate our flow-induced proposal PDF in a 3-D source localization scenario where a volumetric array of receivers provides TDOA measurements [33]. Note that this problem can be solved similarly using the nonparametric message passing algorithm of multisensor MOT as proposed in Section 3.4. The main difference is that in this problem we have to detect and localize sources with measurements collected in only one time step while the multisensor MOT keep estimating the objects' state in a series of time steps. Since it is hard to detect the new birth of objects in the very first time step (c.f. Figure 3.3), the source localization scenario is challenging.

Similarly to the multisensor MOT problem, this scenario is complicated by (i) the highly nonlinear TDOA measurement model, (ii) MOU, and (iii) an unknown number of sources to be localized [131, 132]. For the algorithm development of this problem, we follows Algorithm 9 in Section 3.4, but use the proposal induced by a stochastic PFI and evaluate it as introduced

in this chapter. As expected from the discussion in Section 4.2, we will demonstrate that the GMM stochastic flow outperforms the other reference methods. In Section 4.3.1, the parametric setting of our 3-D source localization problem is specified. For performance analysis, we will first use Gromov's flow as a canonical closed-form solution of stochastic PFI to compare with deterministic EDH and LEDH flow in Section 4.3.2. We will further show the improvement of numerical performance by optimizing the diffusion term in Section 4.3.3.

4.3.1 Simulation Parameters of Source Localization Scenario

We consider the localization of an unknown number of static sources in a 3-D ROI. There are V receivers. Pairs of receivers provide TDOA measurements, e.g., obtain by cross-correlation. In particular, the source-originated TDOA measurement model related to measurement with index m provided by receivers with indexes a and b , can be expressed as $z_{ab}^{(m)} = h_{ab}(\mathbf{x}^{(j)}) + v_{ab}^{(m)}$. Here, $h_{ab}(\mathbf{x}^{(j)}) = 1/c(\|\mathbf{x}^{(j)} - \mathbf{p}^{(a)}\| - \|\mathbf{x}^{(j)} - \mathbf{p}^{(b)}\|)$, $\mathbf{p}^{(a)}$ and $\mathbf{p}^{(b)}$ are the 3-D positions of the receivers, c is the propagation speed in the considered medium, and $v_{ab}^{(m)}$ is the additive white noise with variance σ_v^2 . The noise $v_{ab}^{(m)}$ is statistically independent across m and across all receiver pairs (a, b) . The dependence of a measurement $z_{ab}^{(m)}$ on the source-location $\mathbf{x}^{(j)}$ is described by the likelihood function $f(z_{ab}^{(m)}|\mathbf{x}^{(j)})$ that can be directly obtained from the TDOA measurement model. This likelihood function has the shape of a hyperboloid (cf. Fig. 4.1). For unambiguous source localization, the measurements of multiple receiver pairs have to be used. Our scenario is further complicated by (ii) and (iii) discussed above.

Each receiver pair is considered one of S sensors indexed by $s \in \{1, \dots, S\}$. The receivers of sensor s are indexed (s_a, s_b) and the number of measurements at sensor s is M_s . We consider a topology with $V = 6$ receivers and $S = 9$ sensors is shown in Fig. 4.1-b. Furthermore, we set $\sigma_v = 0.5\text{ms}$ and $c = 1500\text{m/s}$. The clutter measurements at sensor s follow a uniform PDF on $\|\mathbf{q}^{(s_a)} - \mathbf{q}^{(s_b)}\|/c$. The mean number of clutter measurements is $\mu_c = 1$ and the probability of source detection, p_d , is set to 0.95. The ROI is defined as $[-1000\text{m}, 1000\text{m}] \times [-1000\text{m}, 1000\text{m}] \times [-1000\text{m}, 1000\text{m}]$.

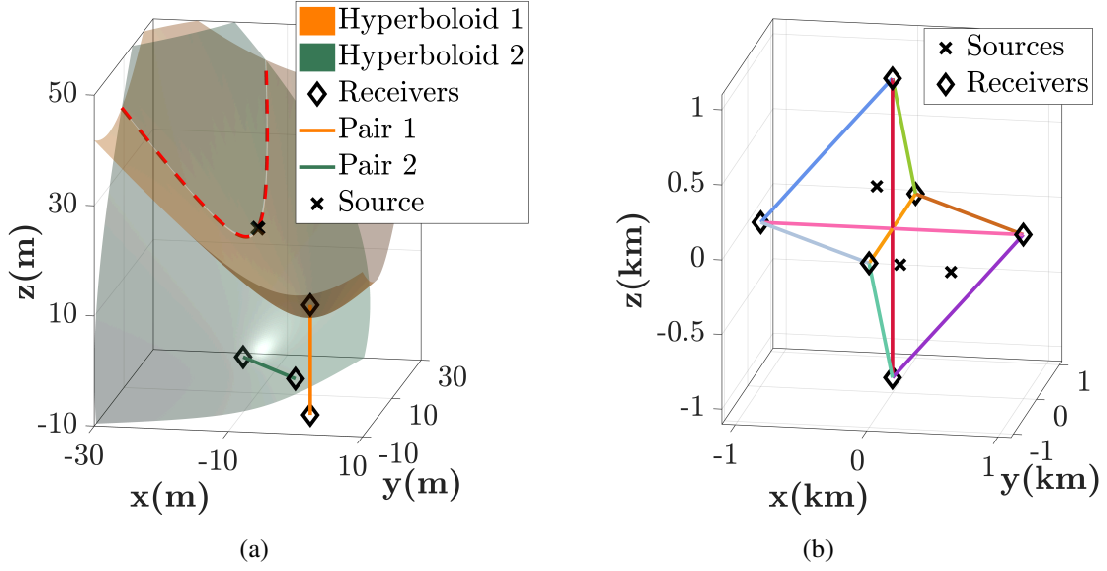


Figure 4.1. (a). Source position and hyperboloids resulting from the TDOA measurements of two sensors. Each sensor consists of two receiver pairs. A dashed red line indicates the intersection of the two hyperboloids. (b). Receiver and sensor topology used in our simulation. There are six receivers located at the center of each face of the ROI cube. Three sources are randomly placed in the ROI.

Following [65, 132], TDOA measurements are processed sequentially across sensors. More precisely, let $f(\mathbf{x}|\mathbf{z}_{1:s-1})$ be the multimodal posterior PDF after sensor update $s - 1$. This PDF is represented by N_k Gaussian mixture components. For each component k and each TDOA measurement, $z_m^{(s)}$, of sensor s , we perform PFI and update each kernel mean and covariance matrix based on (4.4) and (4.5). The resulting $N_k \times M_s$ Gaussian components will be used for IS and Monte Carlo integration within BP-based message passing [65]. The result is an approximate posterior PDFs $f(\mathbf{x}|\mathbf{z}_{1:s})$ represented by N_k Gaussian mixture components.

For a single realization, Fig. 4.2 shows simulation results for a scenario with three static sources. For each source, the GMM consists of $N_k = 100$ kernels; each is importance sampled using a EDH flow. The particle size is $\bar{N}_p = 1000$ for previously detected sources and $\bar{N}_p = 1000$ for newly detected sources at the current sensor. In Fig. 4.2, particles are shown in different colors to represent the spatial PDFs of the potential sources (PSs) j with the existence probabilities $f(r_s^{(j)} = 1|\mathbf{z}_{1:s})$ larger than $P_{th} = 0.5$ (hence regarded as existence) by sensor s , i.e.,

$f(\mathbf{x}_s^{(j)} | r_s^{(j)} = 1, \mathbf{z}_{1:s})$ for $s = 1, 2, 4, 9$ in Fig. 4.2(a)-(d) respectively. Some observations regarding the evolution of particles between sensor pairs are as follows.

- The first receiver pair ($s = 1$) has one missed detection and one clutter measurement. Two sets of particles (pink and orange) meet the true locations of two sources, while the other set (grey) corresponds to the clutter. Spatial PDFs are approximated hyperbolas in Y-Z and X-Z planes. The particles in the X-Y plane have not been updated effectively since the first sensor pair is at unfavorable locations for localization in that plane.
- The second receiver pair ($s = 2$) has no missed detection. The two components with the highest existence probabilities are the legacy components generated from the measurements of the first receiver pair (pink and orange). The third existing component (blue) is regarded as the third source, which is newly detected. Note that the clutter source shown as a grey particle in the first receiver pair no longer exists since there is no measurement matching that clutter in the current receiver pair.
- The number of sources has been correctly determined after the first two sensors. After the fourth sensor had been updated, spatial PDFs of the three sources were still multimodal but more concentrated around the true source locations than the $s = 2$ result. Their spatial PDFs have modes that correspond to the intersection points of the hyperbolas related to previous measurements and current measurements.
- After the update step has been performed for all receiver pairs, all three existing components have a single mode well localized around the actual locations of the three sources.

4.3.2 Performance of Gromov’s Flow

We first compare Gromov’s flow with some reference methods. The baseline method under comparison is the bootstrap particle filter, wherein the prior message is utilized for particle sampling (abbreviated as “PM” for later notation). Apart from the Gromov’s flow, we also

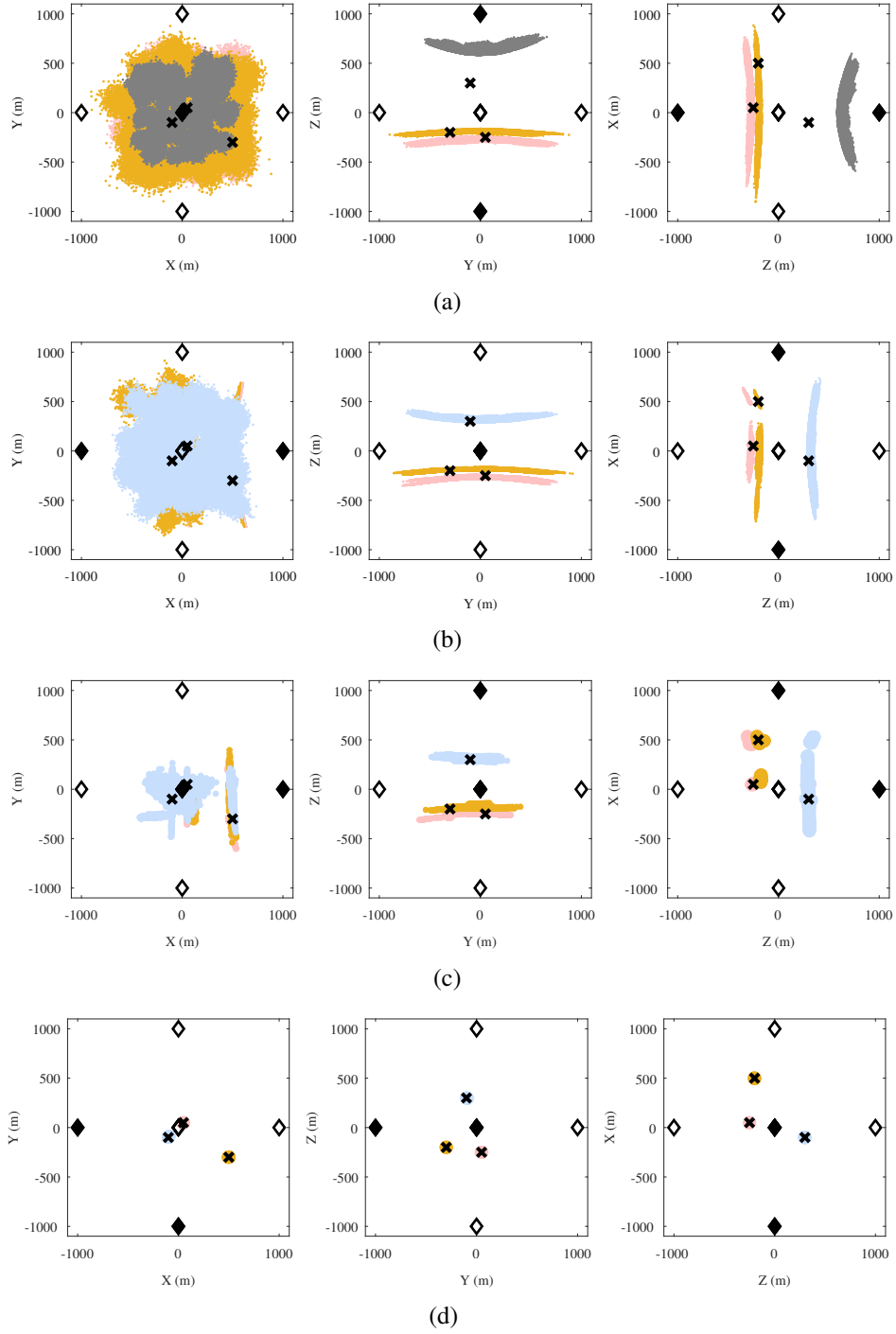


Figure 4.2. Particle-based representation after the EDH flow of the spatial PDFs by GMM model. There exist three signal sources in a 3-D environment with clutters, and the state is sequentially filtered by sensor: (a) $s = 1$, (b) $s = 2$, (c) $s = 4$, and (d) $s = 9 = n_s$. Filled diamonds indicate the locations of receivers involved in the current update step; the locations of other receivers are indicated by empty diamonds. X-marks indicate the sources' ground truth positions.

implement GMM with deterministic flows embedded as a reference, such as the EDH flow [86] (abbreviated as “EDH”) and the LEDH flow (abbreviated as “LEDH”).

To investigate how the performance of our method depends on GMM parameters and sampling size, we simulated scenarios with different numbers of kernels and particles per kernel.

Table 4.1 shows the mean OSPA error [142] (with a cutoff threshold at 50) and runtime per run for different algorithms and system parameters. Note that for PM, we have $N_p = N_s$. All methods are simulated on a single core of an Intel Xeon Gold 5222 CPU. The runtime is averaged over 100 Monte Carlo runs, each with five static sources randomly placed in ROI. It can be seen that the PM performed the worst in terms of OSPA error. This is expected since the PM is based on bootstrap sampling, which suffers particle degeneracy in the considered scenario. In contrast, the implementation based on deterministic flow, such as EDH, can achieve a much smaller OSPA error with much fewer particles and reasonable runtime. The implementation based on the LEDH has a slightly better performance than EDH, but the runtime has been increased significantly (see ID 3 vs. ID 4). Further increasing particle size N_p and $\bar{N}_p$ can improve the localization accuracy (see ID 5 and 6 vs. ID 3) but limited to a bottleneck with OSPA around 20. The implementation based on stochastic flow such as Gromov’s flow can significantly reduce the OSPA error compared with EDH flow using same particle size (see ID 7 vs. ID 3, and ID 8 vs. ID 5), whereas three times in runtime. Overall, Gromov’s flow with parameter setting as ID 7 is best among all tested methods with a trade-off between localization accuracy and runtime.

The statistical box plot of the 100 OSPA samples for each algorithm is shown in Fig. 4.3 for further comparison. Here, the box with a blue boundary has its lower and upper bound as the first and third quartile of the 100 samples, respectively. The vertical range of the box is the interquartile range (IQR). The red line in the box is the median. The black lines that extend from the box are the expected variation as either the minimum/maximum value of the samples or the whiskers (i.e., $1.5 \times \text{IQR}$ extension from the box boundary) if there are outliers. If the second case, the outliers are marked as red ‘+’s. Notably, the median of Gromov’s flow is much lower than that of EDH flow, which means it has a much higher chance of detecting all five sources.

Table 4.1. Simulated mean OSPA error and runtime per run for different algorithms and system parameters.

ID	Method	$(N_k, \underline{N}_p, \overline{N}_p)$	OSPA	Runtime(s)
1	Bootstrap	$(-, 2e6, 2e6)$	43.90	75.4
2	Bootstrap	$(-, 1e7, 1e7)$	32.07	443.3
3	Det. PFI	$(100, 500, 30)$	25.17	196.9
4	Det. PFI	$(100, 3e3, 500)$	20.57	379.6
5	Det. PFI	$(100, 1e4, 1e4)$	19.58	2586.8
6	Sto. PFI	$(100, 500, 30)$	10.43	568.8
7	Sto. PFI	$(100, 3e3, 500)$	8.75	1356.1

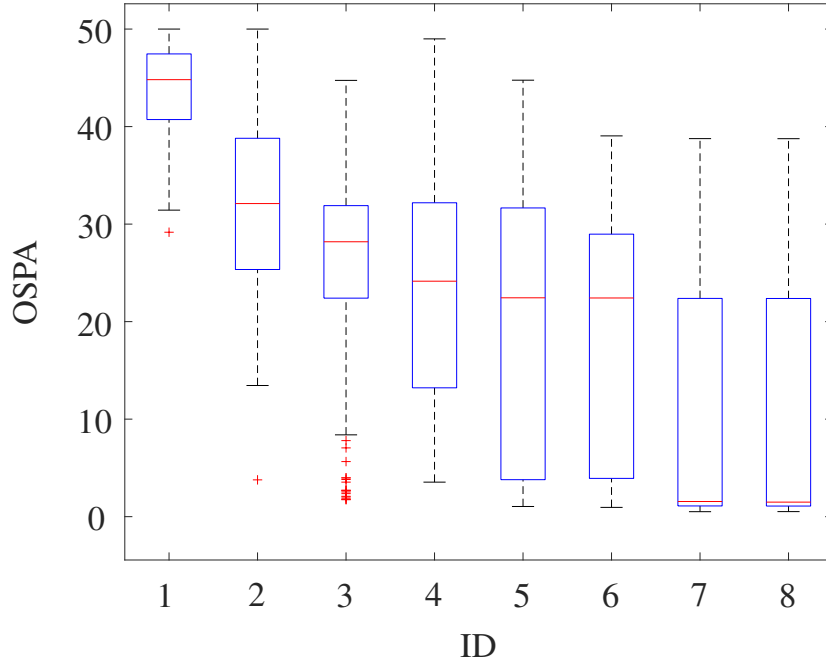


Figure 4.3. Statistics of the OSPA error for different algorithms. Each column corresponds to the algorithm with the same ID as in Table 4.1.

4.3.3 Performance of General Stochastic PFI with Diffusion Optimization

We further look into the performance of a general stochastic PFI with diffusion optimization. As a reference methods for the proposed IS with *optimized stochastic* PFI (“PFI-OS”), we consider bootstrap IS (“PM”), the “flow-induced” proposal PDFs using the deterministic EDH PFI (“EDH”) and the stochastic Gromov’s PFI (“Gromov”). Since every method yields different stiffness, for numerical integration, every method requires a different resolution of the pseudo-time. This resolution is defined as the inverse of the time-interval Δ_l . In addition, for every method, a higher resolution is typically needed at the first few steps of the numerical integration. We use an exponentially increasing ratio of the time-interval, i.e., $\Delta_l = \beta \Delta_{l-1}$, where β is a fixed ratio for $l = 2, \dots, N_\lambda - 1$. Then, we control the interval by two parameters, the initial difference Δ_1 and the increasing ratio β . Note that a larger Δ_1 and β result in fewer discrete steps N_λ and thus to a reduced runtime. For each method, we choose Δ_1 and β to obtain a good runtime-accuracy tradeoff.

Table 4.2 shows the mean OSPA error (with a cutoff threshold at 30) and runtime per run for the different methods and different parameters settings. OSPA and runtime are averaged over 100 Monte Carlo runs, for each run three sources are randomly place in the ROI. We also list the number of Gaussian components, N_k , as well as the number of samples per component, N_p .

Table 4.2. Simulated mean OSPA error and runtime per run for different methods and system parameters.

ID	Method	(N_k, N_p)	(β, Δ_1)	OSPA	Runtime(s)
1	bootstrap	$(-, 2e6)$	$(-, -)$	25.89	47.0
2	bootstrap	$(-, 4e7)$	$(-, -)$	14.73	718.8
3	EDH	$(100, 5e3)$	$(1.5, 1e-7)$	11.1	202.2
4	Gromov	$(100, 5e3)$	$(1.5, 1e-7)$	8.36	612.7
5	PFI-OS	$(100, 5e3)$	$(1.5, 1e-5)$	6.33	490.9
6	PFI-OS	$(100, 5e3)$	$(2, 1e-4)$	6.70	295.2

PM suffers from particle degeneracy and thus yields the highest OSPA. In addition, the large number of $4e7$ particles results in the largest memory requirements. In contrast, PFI-based methods required much fewer particles. It can be seen that, for the same N_λ value, Gromov has a smaller OSPA error compared to EDH while its runtime is three times larger. PFI-OS relies on a diffusion matrix $\mathbf{Q}$ optimized as discussed in Section 4.2, by setting $\alpha = 0.1$. Notably, PFI-OS yields the lowest OSPA error and, at the same time, has a low runtime.

To better understand the influence of α on the estimation error for different step sizes, we compare the proposed methods for different values of α and different resolutions of pseudo time. Results are shown in Table 4.3. EDH and Gromov results are also listed. When the step size is small, PFI can lead to accurate results despite the stiffness of the underlying differential equation. In this case, traditional methods such as EDH and Gromov's flow also perform well. However, the small step size comes at the cost of a strongly increased runtime. As the step size increases, only PFI-OS with carefully chosen parameter α results in a good estimation accuracy. Importantly, our results indicate that PFI-OS has a significantly improved complexity accuracy tradeoff compared to EDH and Gromov.

Table 4.3. Mean OSPA error of different PFI-based IS method and different integration step sizes.

(β, Δ_1) Method	$(1.3, 1e-13)$	$(1.5, 1e-5)$	$(2, 1e-4)$
EDH	6.29	30	30
Gromov	8.70	24.51	28.4
PFI-OS ($\alpha = 0.01$)	4.79	30	30
PFI-OS ($\alpha = 0.1$)	10.28	6.33	6.70
PFI-OS ($\alpha = 0.5$)	10.32	9.36	12.85

The tradeoff between the condition number of the Jacobian matrix $\mathbf{A}$ and the norm of the diffusion matrix $\mathbf{Q}$ is analyzed numerically for different values of α . Fig. 4.4 shows the condition number of the Jacobian matrix $\mathbf{A}$ and the norm of the diffusion matrix $\mathbf{Q}$ for the stochastic PFIs. Different values of α are compared. The time resolution is set to $(1.5, 1e-5)$. For efficient

particle transition and small process noise, $\kappa(\mathbf{A})$ and $\|\mathbf{Q}\|$ have to be small. However, there is an inherent traded off between these two parameters [146]. As can be seen in Fig. 4.4, compared to IS with Gromov's flow, the proposed IS with diffusion optimization can significantly reduce the condition number while at the same time avoiding large process noise peaks. A smaller value of α results in a stronger reduction of the condition number (i.e., a stronger effort to reduce stiffness), which, in turn, leads to an increased process noise variance (i.e., a larger norm of $\mathbf{Q}$). Since a too-large process noise variance also reduces the accuracy of the flow, carefully selection of α is crucial for the success of the proposed method.

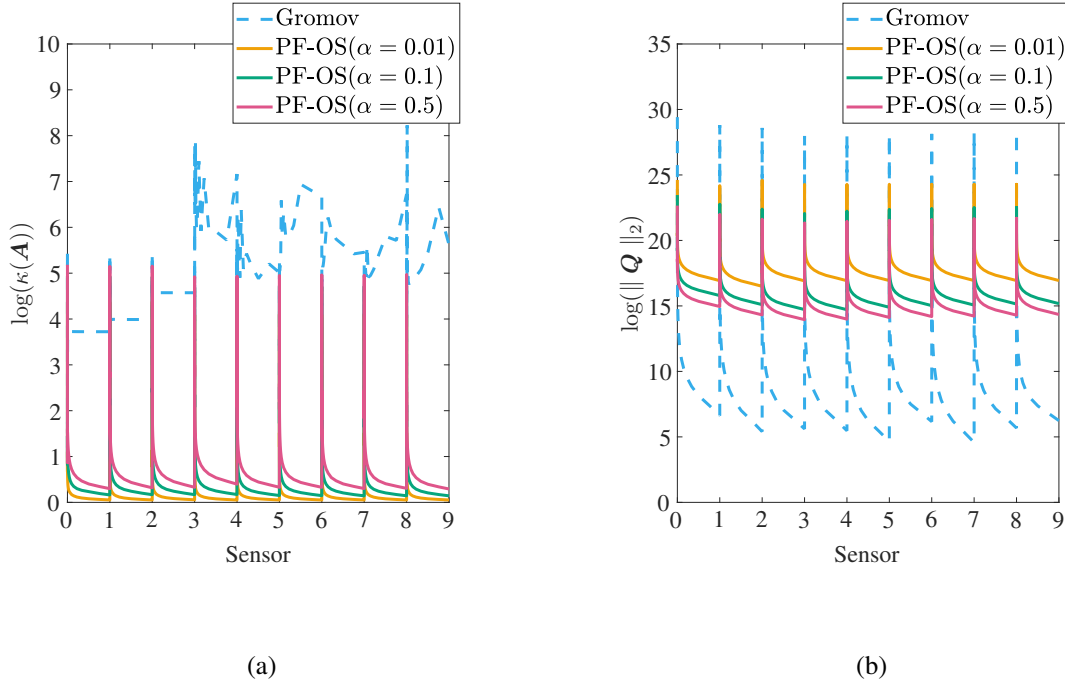


Figure 4.4. Log condition number of the Jacobian matrix (a) and log 2-norm of the diffusion matrix (b) of the drift term of different stochastic particle flows. The x-axis of the plot consists of a concatenation of pseudotimes across different sensors. The numerical integration parameters $(\beta, \Delta_1) = (1.5, 1e-5)$ were used across different sensor update steps.

4.4 Summary

In this chapter, we propose a IS method with a general stochastic PFI. The flow-induced proposal can be evaluated in a form of GMM. The method is general for arbitrary choice of diffusion matrix, which can be designed to improve the transient dynamics of a PFI. The numerical results shows significant improvement of estimation performance in a 3-D multisource localization problem compared to deterministic EDH flow. Further research includes applying our stochastic PFI IS methods for real tracking problems and algorithms to automatically optimize parameters for the diffusion model.

Chapter 5

FIM-Based Active Planning

In Chapter 3 and 4, we focus on improving sampling efficiency for nonparametric estimation of object states with application to 3-D tracking and localizing problems. We denote the sensing scenario “passive” since the sensors are assumed to be static during the tracking process. As introduced in Section 1.2.4, smart control of sensors is crucial for cooperative localization in certain types of mobile networks.

In this chapter, we introduce “active” planning for cooperative localization networks. Active planning aims at increasing the information gain of future measurements¹. Contrary to [110], which develops a control policy based on negative conditional differential entropy, we propose a control policy that minimizes traces of approximate inverse Bayesian FIMs [120]. In this way, costly Monte Carlo integration can be avoided, and the computational complexity related to a myopic control, i.e., a control strategy with a planning horizon of one timestep, can be strongly reduced.

Due to this reduced computational complexity, we can develop a RH control, i.e., a control strategy with a planning horizon that consists of multiple future timesteps. In particular, we develop a cost function that accumulates Bayesian FIM contributions and state the tree search problem that results from cost function minimization. Approximations for solving the resulting tree-search problem efficiently are also discussed. The key contributions of this work are as

¹This chapter is based on W. Zhang, B. Teague, and F. Meyer, “Active planning for cooperative localization: A Fisher information approach,” in *Proc. Asilomar-22*, Pacific Grove, CA, USA, 2022, pp. 795–800. [151]

follows.

- We introduce the trace of the inverse Bayesian FIM as the cost function for active planning in cooperative localization.
- We develop a cost function that accumulates Bayesian FIM contributions for RH control.
- We demonstrate that the proposed method can improve localization performance in a numerical case study.

5.1 System Model and Estimation Problem

We consider a wireless network for cooperative localization in 3-D. The network consists of agents with unknown positions and anchors with known positions. An example network is shown in Fig. 5.1. The agents and some anchors are mobile. The neighborhood of an agent is defined by its communication range. Mobile agents communicate with other mobile agents and anchors in their neighborhood to exchange position information and perform pairwise measurements of the distance. The goal of each mobile agent is to sequentially estimate its location by using all available pairwise distance measurements and the position information of anchors and mobile agents in its neighborhood. While the motion of agents is arbitrary, it is assumed that the motion of the mobile anchors can be controlled and planned to improve localization accuracy in the network. In what follows, we introduce the system model, formulate the joint estimation and control problem, and review the estimation layer of the proposed estimation and control method.

5.1.1 System Model

We consider a wireless network that consists of M anchors and N mobile agents. There are M_1 mobile anchors indexed $j \in \{1, \dots, M_1\}$ and $M_2 = M - M_1$ static anchors indexed by $j \in \{M_1 + 1, \dots, M\}$. The state of agent $i \in \{1, \dots, N\}$ at discrete timestep k is denoted as $\mathbf{x}_k^{(i)}$

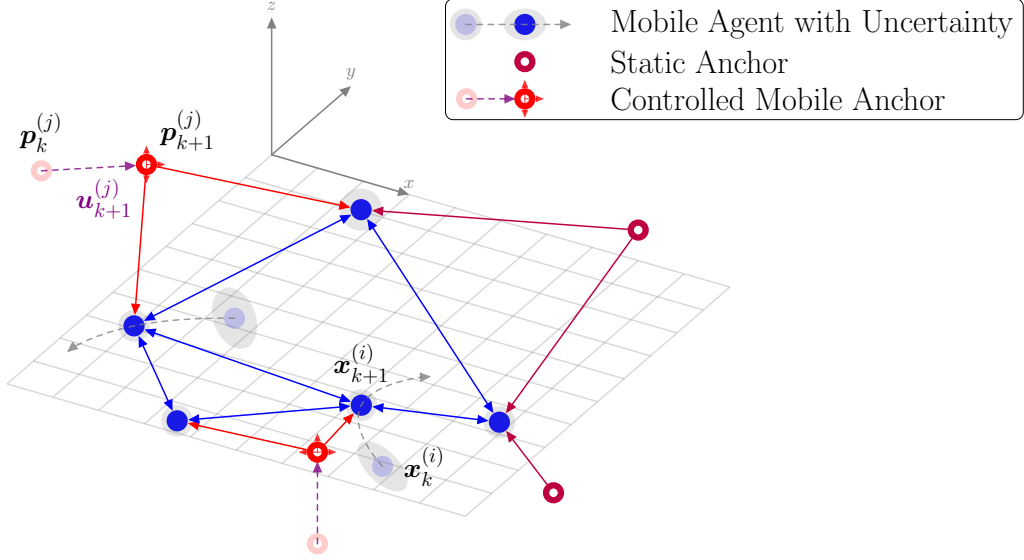


Figure 5.1. Cooperative localization scenario with mobile agents, mobile anchors, and static anchors.

and consists of the agent's 3-D position as well as further motion-related parameters. The 3-D position of anchor $j \in \{1, \dots, M\}$ at discrete timestep k is denoted as $\mathbf{p}_k^{(j)} = [p_{1,k}^{(j)} \ p_{2,k}^{(j)} \ p_{3,k}^{(j)}]^T$.

Agent motion is described by a potentially nonlinear state-transition model, i.e.,

$$\mathbf{x}_{k+1}^{(i)} = f(\mathbf{x}_k^{(i)}) + \mathbf{v}_k^{(i)}, \quad i \in \{1, \dots, N\} \quad (5.1)$$

where $f(\mathbf{x}_k^{(i)})$ is the state transition function and $\mathbf{v}_k^{(i)}$ is zero-mean Gaussian driving noise with standard deviation σ_v .

Anchor motion is described by the velocity control vector $\mathbf{u}_{k+1}^{(j)} \in \mathcal{U}$, i.e.,

$$\mathbf{p}_{k+1}^{(j)} = \mathbf{p}_k^{(j)} + T_\Delta \mathbf{u}_{k+1}^{(j)}, \quad j \in \{1, \dots, M_1\} \quad (5.2)$$

where T_Δ is the duration of a discrete timestep. Note that $\mathcal{U}$ is a dictionary that consists of a finite number of possible control vectors. The position of static anchors remains fixed, i.e., $\mathbf{p}_{k+1}^{(j)} = \mathbf{p}_k^{(j)}$, $j \in \{M_1 + 1, \dots, M\}$, $\forall k$.

The set of all anchors that are in the neighborhood of agents i is denoted as $\mathcal{M}_i \subset$

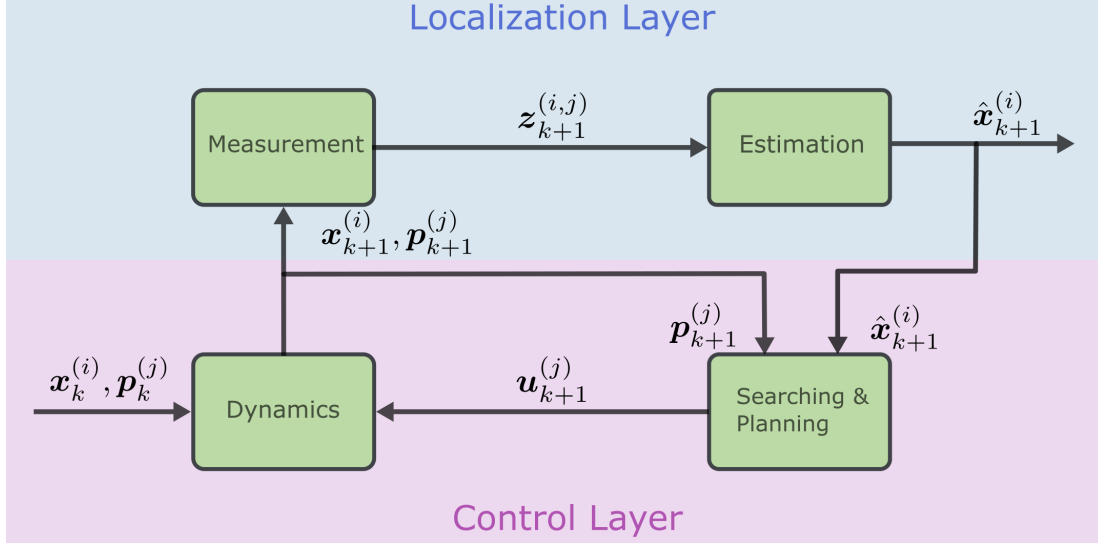


Figure 5.2. Block diagram of the proposed estimation and control framework for cooperative localization.

$\{1, \dots, M\}$. Agent i performs pairwise distance measurements with all anchors $j \in \mathcal{M}_i$, i.e.,

$$z_k^{(i,j)} = \|\chi_k^{(i)} - p_k^{(j)}\| + \omega_k^{(i,j)}, \quad j \in \mathcal{M}_i \quad (5.3)$$

where $\chi_k^{(i)} = [x_{1,k}^{(i)} \ x_{2,k}^{(i)} \ x_{3,k}^{(i)}]^T$ denotes the position of agent i , and $\omega_k^{(i,j)}$ is zero-mean Gaussian measurement noise with standard deviation $\sigma_\omega^{(i,j)}$. We also introduce the set of all agents that are in the neighborhood of anchor $j \in \{1, \dots, M\}$, i.e., $\mathcal{A}_j = \{i | j \in \mathcal{M}_i\}$.

The set of all agents that are in the neighborhood of agents i is denoted as $\mathcal{N}_i \subset \{1, \dots, N\} \setminus \{i\}$. Agent i performs pairwise distance measurements with all agents $i' \in \mathcal{N}_i$, i.e.,

$$z_k^{(i,i')} = \|\chi_k^{(i)} - \chi_k^{(i')}\| + \eta_k^{(i,i')}, \quad i' \in \mathcal{N}_i \quad (5.4)$$

where $\eta_k^{(i,i')}$ is zero-mean Gaussian measurement noise with standard deviation $\sigma_\eta^{(i,i')}$. Typically, neighborhood sets can be assumed symmetric, i.e., $i' \in \mathcal{N}_i \iff i \in \mathcal{N}_{i'}$. At time $k = 0$, it is assumed that prior PDFs of all agent states $f(x_0^{(i)})$, $i \in \{1, \dots, N\}$ are known.

5.1.2 Estimation for Cooperative Localization

Fig. 5.2 shows a block diagram of the proposed estimation and control framework for cooperative localization. The “dynamics” block represents the state transition model of the agents in (5.1) and the control update model of the mobile anchors in (5.2). The “measurement” block represents the measurement models in (5.3) and (5.4).

The “measurement” and “estimation” blocks constitute the localization layer. The main goal of the localization layer is to estimate the state of each agent based on all measurements collect up to the current time k . Let $\mathbf{x}_k = [\mathbf{x}_k^{(1)\top} \cdots \mathbf{x}_k^{(N)\top}]^\top$ be the joint vector that consists of all agent states at time k and let $\mathbf{z}_k$ be the joint vector that consists of all measurements at time k , as defined in (5.3) and (5.4). From the statistical model discussed in Section 5.1.1, the joint posterior PDF $f(\mathbf{x}_k | \mathbf{z}_{1:k}; \mathbf{u}_{1:k})$ can be derived (see [152] and [111] for details). Note that here we do not explicitly denote the known anchor positions.

State estimation of the agent i can be performed by calculating the MMSE estimate [129] according to

$$\hat{\mathbf{x}}_k^{(i)} \triangleq \int \mathbf{x}_k^{(i)} f(\mathbf{x}_k^{(i)} | \mathbf{z}_{1:k}; \mathbf{u}_{1:k}) d\mathbf{x}_k \quad (5.5)$$

where $f(\mathbf{x}_k^{(i)} | \mathbf{z}_{1:k}; \mathbf{u}_{1:k})$ is the marginal posterior PDF of agent state i . Direct computation of this marginal posterior PDF from the joint posterior PDF $f(\mathbf{x}_k | \mathbf{z}_{1:k}; \mathbf{u}_{1:k})$ by marginalization is typically unfeasible. However, using common assumptions [152, 111], by exploiting the factorization structure of $f(\mathbf{x}_k | \mathbf{z}_{1:k}; \mathbf{u}_{1:k})$, it is possible to compute an accurate approximation of $f(\mathbf{x}_k^{(i)} | \mathbf{z}_{1:k}; \mathbf{u}_{1:k})$ in a fully distributed way. In particular, for distributed estimation of $\hat{\mathbf{x}}_k^{(i)}$, agent i only needs measurements that involve its state as well as position information from agents and anchors in its neighbor sets [152, 111].

Distributed estimation is typically performed using loopy BP [152, 111]. There is a wide range of BP variants, which can be roughly classified into methods that represent marginal posterior PDFs by Gaussians and methods that represent marginal posterior PDFs by particles. Particle representations are typically more accurate compared to Gaussian representations but

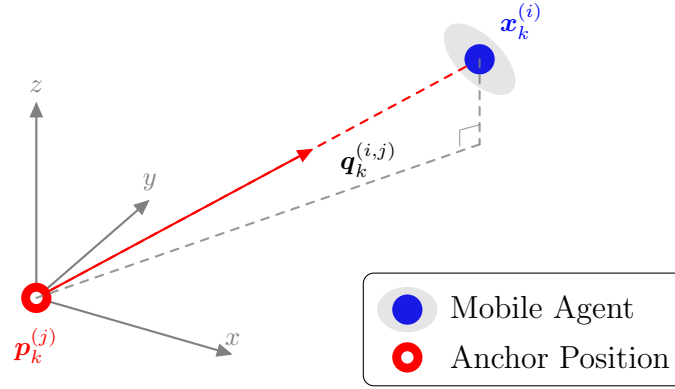


Figure 5.3. 3-D range direction vector between position of anchor j and position of agent i .

suffer from the curse of dimensionality [130]. Using a particle representation in the considered 3-D cooperative localization scenario would lead to an unfeasible computational complexity. For this reason, we make use of sigma point BP [153], which computes a Gaussian representation of marginal posterior PDFs based on the unscented transformation [47].

5.2 Active Planning

In this section, we will introduce the main contribution of this chapter, namely, active planning based on the FIM. The active planning method is represented by the “searching & planning” block in Fig. 5.2. For active planning, a control policy is designed to move anchors such that future measurements increase the trace of the FIM also known as the Cramér-Rao bound (CRB). In particular, the trace of the FIM for the considered problem is reviewed and then used to develop an accumulated cost function for RH control. Finally, a near-optimal control vector is computed for each anchor by performing a tree search.

5.2.1 Equivalent FIM and Myopic Control Policy

At time k , for each mobile anchor $j = 1, \dots, M_1$, we want to determine the control vector $\mathbf{u}_{k+1}^{(j)}$ in (5.2), that maximizes the overall information provided by the anchor measurements performed at the next timestep, i.e., $\mathbf{z}_{k+1}^{(i,j)}, i \in \mathcal{A}_j$. As a measure of information, we use the equivalent Bayesian FIM as derived in [120] that will be reviewed next.

Let us consider the range vector $\mathbf{q}_k^{(i,j)} = [x_{1,k}^{(j)} - p_{1,k}^{(j)} \ x_{2,k}^{(j)} - p_{2,k}^{(j)} \ x_{3,k}^{(j)} - p_{3,k}^{(j)}]^\top$ and the corresponding range direction vector $\mathbf{q}_k^{(i,j)} = \mathbf{q}_k^{(i,j)} / \|\mathbf{q}_k^{(i,j)}\|$ from the position $\mathbf{p}_k^{(j)}$ of anchor j to the position $\chi_k^{(i)}$ of agent i , as shown in Fig. 5.3. The ranging direction matrix (RDM) [120] of agent i with anchor j can now be obtained as

$$\mathbf{J}(\chi_k^{(i)}, \mathbf{p}_k^{(j)}) \triangleq \mathbf{q}_k^{(i,j)} (\mathbf{q}_k^{(i,j)})^\top. \quad (5.6)$$

Based on the RDMs of agent i with all anchors $j \in \mathcal{M}_i$, an equivalent FIM [120] for the position of agent i is given by

$$\mathbf{I}^{(i)}(\chi_k^{(i)}, \mathbf{p}_k) = \sum_{j \in \mathcal{M}_i} \frac{1}{(\sigma_\omega^{(i,j)})^2} \mathbf{J}(\chi_k^{(i)}, \mathbf{p}_k^{(j)}) \quad (5.7)$$

where we recall that $\sigma_\omega^{(i,j)}$ is the measurement standard deviation of the measurement noise $\omega_k^{(i,j)}$ in (5.3).

Let us assume that a Gaussian representation of the predicted posterior PDF for the agent state position $\chi_{k+1}^{(i)}$ is available, i.e., $f(\chi_{k+1}^{(i)} | \mathbf{z}_{1:k}; \mathbf{u}_{1:k}) = \mathcal{N}(\boldsymbol{\mu}_{k+1}^{(i)}, \boldsymbol{\Sigma}_{k+1}^{(i)})$. This Gaussian representation can be provided by sigma point BP [153] performed in the estimation layer. Now, we can obtain an approximate Bayesian FIM [120] based on the predicted mean $\boldsymbol{\mu}_{k+1}^{(i)}$ and covariance $\boldsymbol{\Sigma}_{k+1}^{(i)}$ of the agent position $\chi_{k+1}^{(i)}$, as

$$\mathbf{I}_k^{(i)}(\mathbf{p}_{k+1}) = \mathbf{I}^{(i)}(\boldsymbol{\mu}_{k+1}^{(i)}, \mathbf{p}_{k+1}) + \boldsymbol{\Sigma}_{k+1}^{(i)-1}. \quad (5.8)$$

Note that in this preliminary work, we use an approximation of the Bayesian FIM in (5.8), which does not take the measurements among agents into account.

Next, let $\mathbf{u}_{k+1} = [\mathbf{u}_{k+1}^{(1)} \cdots \mathbf{u}_{k+1}^{(M_1)}]^\top$ be the joint control vector and let $\mathbf{u}_{k+1} = [\mathbf{u}_k^\top \ \mathbf{0}_{3M_2}^\top]^\top$ be the augmented joint control vector. By using (5.2) in (5.8), we can now rewrite the approximate Bayesian FIM as a function as of $\mathbf{u}_{k+1}$, i.e.,

$$\mathbf{C}_k^{(i)}(\mathbf{u}_{k+1}) = \mathbf{I}_k^{(i)}(\mathbf{p}_k + T_\Delta \mathbf{u}_{k+1}). \quad (5.9)$$

Here, we make use of the fact that at time k , all anchor positions $\mathbf{p}_k$ are known and only the first M_1 anchors are mobile, i.e., the anchor topology only depends on the joint control vector $\mathbf{u}_{k+1}$.

To obtain a cost function for all agents states, we compute the sum of all traces of the individual approximate Bayesian FIMs, i.e.,

$$C_k(\mathbf{u}_{k+1}) = \sum_{i=1}^N \text{Tr}(\mathbf{C}_k^{(i)}(\mathbf{u}_{k+1})^{-1}). \quad (5.10)$$

This sum is an approximation of the Bayesian CRB of the joint agent position $\chi_{k+1} = [\chi_{k+1}^{(1)\text{T}} \cdots \chi_{k+1}^{(N)\text{T}}]^\text{T}$. The myopic control policy finally reads

$$\hat{\mathbf{u}}_{k+1} = \arg \min_{\mathbf{u}_{k+1} \in \mathcal{U}^{M_1}} C_k(\mathbf{u}_{k+1}). \quad (5.11)$$

5.2.2 RH Control Policy and Tree-Search

Due to uncertain agent states, a myopic control policy as introduced in (5.11), can lead to highly suboptimal control decisions that also negatively influence estimation and control at future timesteps. This “getting stuck” at a local minimum often happens at initial timesteps where the localization uncertainty of the network is significant. To avoid local minima, we make use of RH control [112] which uses model-based prediction of future costs within a sliding window that consists of T timesteps.

For RH control, at the current time k , we aim to develop an accumulated cost function that considers $t \in \{1, \dots, T\}$ future timesteps. First, for all T future timesteps and all agents, Gaussian representations for the predicted posterior PDFs of the agent state positions $\chi_{k+t}^{(i)}$ are obtained, i.e., we calculate $f(\chi_{k+t}^{(i)} | \mathbf{z}_{1:k}; \mathbf{u}_{1:k}) = \mathcal{N}(\boldsymbol{\mu}_{k,t}^{(i)}, \boldsymbol{\Sigma}_{k,t}^{(i)})$, $t \in \{1, \dots, T\}$, $i \in \{1, \dots, N\}$. These Gaussian representations can be computed by the prediction step of a Kalman filter or a sigma point filter [47]. Note that $\boldsymbol{\mu}_{k,1}^{(i)}$ and $\boldsymbol{\Sigma}_{k,1}^{(i)}$ are equal to $\boldsymbol{\mu}_{k+1}^{(i)}$ and $\boldsymbol{\Sigma}_{k+1}^{(i)}$ in (5.8). Next, based on (5.8), we

introduce the approximate Bayesian FIM of agent state i for future timestep $k+t$, predicted at current timestep k , i.e.,

$$\mathbf{I}_{k,t}^{(i)}(\mathbf{p}_{k,t}) = \mathbf{I}^{(i)}(\boldsymbol{\mu}_{k,t}^{(i)}, \mathbf{p}_{k,t}) + \boldsymbol{\Sigma}_{k,t}^{(i)-1} \quad (5.12)$$

where $\mathbf{p}_{k,t}$ is given by $\mathbf{p}_{k,t} = \mathbf{p}_k + T_\Delta \sum_{t'=1}^t \mathbf{u}_{k+t'}$. We can now rewrite the predicted approximate Bayesian FIM in (5.12) as a function of $\mathbf{u}_{k+1:k+t}$, i.e.,

$$\mathbf{C}_{k,t}^{(i)}(\mathbf{u}_{k+1:k+t}) = \mathbf{I}_{k,t}^{(i)}\left(\mathbf{p}_k + T_\Delta \sum_{t'=1}^t \mathbf{u}_{k+t'}\right). \quad (5.13)$$

To obtain a cost function that consists for all agents states, as in (5.10), we again compute the sum of all traces of the individual predicted approximate Bayesian FIMs, i.e.,

$$C_{k,t}(\mathbf{u}_{k+1:k+t}) = \sum_{i=1}^N \text{Tr}\left(\mathbf{C}_{k,t}^{(i)}(\mathbf{u}_{k+1:k+t})^{-1}\right). \quad (5.14)$$

Note that for $t = 1$, (5.14) is equal to (5.10).

To simplify the notation, we now drop the time index k . In particular, we write $\mathbf{u}_{1:t}$ for $\mathbf{u}_{k+1:k+t}$ and $C_t(\mathbf{u}_t; \mathbf{u}_{1:t-1})$ for $C_{k,t}(\mathbf{u}_{k+1:k+t})$. The accumulated cost function for a horizon $T \geq 1$ based on the control sequence $\mathbf{u}_1, \dots, \mathbf{u}_T$, is now expressed as

$$C(\mathbf{u}_{1:T}) = \sum_{t=1}^T \gamma^{t-1} C_t(\mathbf{u}_t; \mathbf{u}_{1:t-1}) \quad (5.15)$$

where $0 < \gamma \leq 1$ is the discount factor [154]. The discount factor is motivated by the fact that at timesteps that are far in the future, predicted agent states are less certain, and thus the cost contributions related to these timesteps should have lower weights in the accumulated cost function. The discount factor also helps to motivate a choice for the length of the horizon T [154]. The optimal control sequence $\hat{\mathbf{u}}_{1:T}$ can finally be obtained based on the RH control policy

$$\hat{\mathbf{u}}_{1:T} = \underset{\mathbf{u}_1, \dots, \mathbf{u}_T \in \mathcal{U}^{M_1}}{\operatorname{argmin}} C(\mathbf{u}_{1:T}). \quad (5.16)$$

Note that for $T = 1$, the RH control policy is equal to the myopic control policy in (5.11).

The so-called forward propagation defined by the accumulated cost function in (5.15) naturally leads to a tree search, where a new level of the tree is introduced for every t . Thus, every possible sequence of control vectors $\mathbf{u}_{1:t-1}$ up to time t , leads to $|\mathcal{U}|^{M_1}$ possible sequences of control vectors at time t . This implies that the computational complexity of the tree search scales as $\mathcal{O}(|\mathcal{U}|^{TM_1})$, i.e., exponentially in T and M_1 . To reduce the complexity of the search, a suboptimal control sequence can be obtained by determining, at each timestep t , the control sequence of each mobile anchor $j \in \{1, \dots, M_1\}$ individually, assuming $\mathbf{u}_{k+t}^{(j')} = 0$ for all the other mobile anchors $j' \in \{1, \dots, M_1\} \setminus \{j\}$. In particular, there are M_1 parallel searches and for every search, every possible sequence of control vectors $\mathbf{u}_{1:t-1}^{(j)}$ up to time t , leads to $|\mathcal{U}|$ possible sequences of control vectors at time t . The computational complexity of this approximate search thus only scales as $\mathcal{O}(M_1|\mathcal{U}|^T)$. To further reduce computational complexity, a pruning strategy can be used. Here, after processing at every tree level $t \in \{1, \dots, T\}$ the number of possible sequences of control vectors in every parallel search is limited to V . In particular, we only keep the V sequences with different final anchor positions, which yield the largest accumulated cost $C(\mathbf{u}_{1:t})$. It can be easily verified that the computational complexity of the approximated search with the additional pruning strategy only scales as $\mathcal{O}(VM_1|\mathcal{U}|^T)$.

5.3 Results

Next, we will present a numerical case study to demonstrate that active planning can significantly improve the localization accuracy in a 3-D scenario at a reasonably computational cost.

5.3.1 Simulation Setup and Example Realization

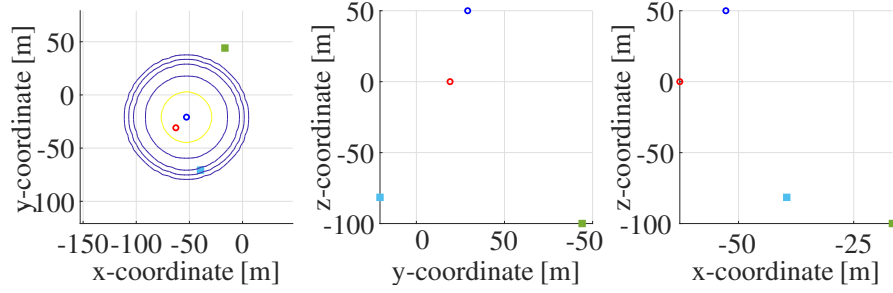
We consider a network that consists of one static anchor, one mobile anchor, and one mobile agent, i.e., $M = 2$, $M_1 = 1$, and $N = 1$. All initial positions are random and uniformly distributed on the ROI. The mobile agent represents a pedestrian that moves in the X-Y plane and has a constant Z-coordinate equal to 0. Hence, the state of the agent at time k can be represented by the vector $\mathbf{x}_k^{(i)} = [x_{1,k}^{(i)} x_{2,k}^{(i)} x_{3,k}^{(i)} \dot{x}_{1,k}^{(i)} \dot{x}_{2,k}^{(i)}]^T$. At time $k = 0$, the prior covariance of $\mathbf{x}_0^{(i)}$ is set to $\Sigma_0^{(i)} = \text{diag}(10^2, 10^2, 50^2, 0.1, 0.1)$, and the prior mean $\mu_0^{(i)}$ is sampled from a Gaussian distribution that has the true agent state as the mean and $\Sigma_0^{(i)}$ as the covariance matrix. The state transition function in (5.1) is defined as follows. The substate $[x_{1,k}^{(i)} x_{2,k}^{(i)} \dot{x}_{1,k}^{(i)} \dot{x}_{2,k}^{(i)}]^T$ follows a constant velocity motion model [48] with driving noise variance $\sigma_v^2 = 0.001 \text{m}^2/\text{s}^4$ while the Z-coordinate $x_{3,k}^{(i)}$ remains constant.

The initial positions of the anchors are determined by drawing samples from the PDF that is uniform on the 3-D ROI defined as $[-100\text{m}, 100\text{m}] \times [-100\text{m}, 100\text{m}] \times [-100\text{m}, 100\text{m}]$. The mobile anchor moves in 3-D according to (5.2). The dictionary $\mathcal{U}$ consists of $|\mathcal{U}| = 19$ possible control vectors that all have a constant norm $\|\mathbf{u}\| = 2 \text{m/s}$. The control vectors are chosen such that their angles form a uniform grid of spherical coordinates. The duration of a discrete timestep is $T_\Delta = 1 \text{s}$.

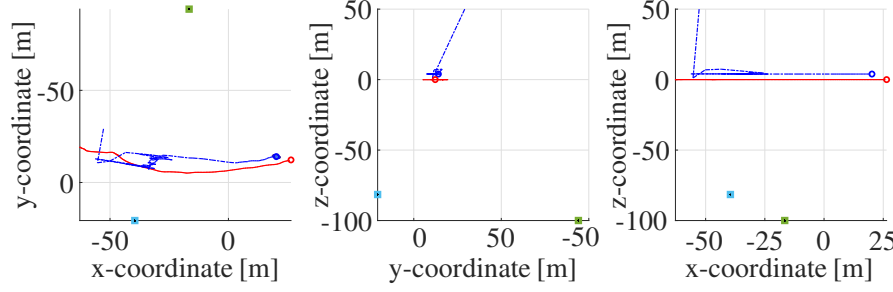
Range measurements are performed between each anchor and the agent following (5.3). The standard deviation of the measurement noise is $\sigma_\omega^{(1,1)} = \sigma_\omega^{(1,2)} = 0.1 \text{m}$.

In addition to the proposed RH control in (5.16), denoted “FIM-RH”, we simulate the proposed myopic control in (5.11), denoted “FIM-Myopic”, and two additional reference approaches. In “static”, both anchors do not move. In “random”, the mobile anchor moves randomly, i.e., at each timestep k , a control vector $\mathbf{u}_{k+1}^{(1)}$ is randomly chosen from $\mathcal{U}$. For “FIM-RH”, planning horizon is set as $T = 20$, the pruning threshold to $V = 1000$, and the discount factor to $\gamma = 0.9$. For estimation, all methods rely on sigma point BP [153].

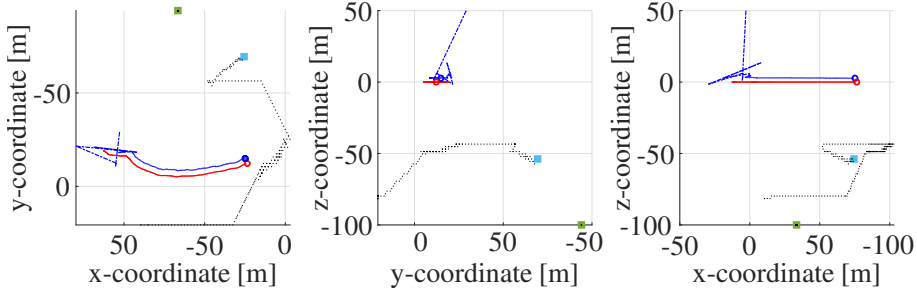
Fig. 5.4 shows an example realization that consists of 200 timesteps. Fig. 5.4(a) shows



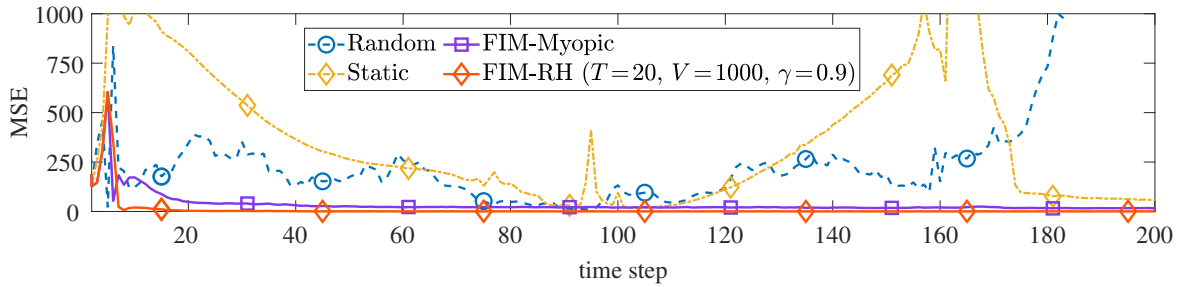
(a) Initial agent and anchor positions.



(b) Two static anchors: True and estimated agent track.



(c) Myopic anchor control: True and estimated agent track.



(d) MSE of the agent state estimate versus time.

Figure 5.4. Example realization of the simulated 3-D localization scenario. The 3-D environment is mapped onto three planes. The two anchor positions are shown as green and blue squares, respectively. The red dot denotes the true position of the agent. The blue dot denotes the estimated position of the agent. The true and estimated agents tracks are shown as red and blue solid lines, respectively. The agent track is shown as a black dotted line. Error ellipses corresponding to the prior covariance matrix are also shown in the X-Y plane of (a).

initial agent and anchor positions. In Fig. 5.4(b), where “static” has been used, we can see that the localization accuracy is low and time-varying. In Fig. 5.4(c), FIM-Myopic” has been used to control the mobile anchor. Here, we can see that the agent position estimate converges quickly to a position close to the true agent position and that the mobile agent can be tracked accurately. Fig. 5.4(d) shows the MSE of the agent state estimates versus time for this example realization that results from the different control methods. We can see that the proposed methods, i.e., “FIM-Myopic” and “FIM-RH”, significantly outperform the two reference approaches. More specifically, we can see that the estimation accuracy is poor and time-varying for both the “random” and the “static” approach. On the contrary, the “FIM-Myopic” and “FIM-RH” methods, after a quick initial convergence, can accurately track the agents at all times. It can be noted that the “FIM-RH” method converges quicker and has a higher estimation accuracy compared to “FIM-Myopic”. This is because “FIM-RH” can compute more effective control vectors by performing optimization over a receding planning horizon.

5.3.2 Numerical Results and Computational Complexity

To statistically validate the localization accuracy, we performed 100 simulations that consist of 300 timesteps in the scenario discussed in Section 5.3.1.

Estimation Performance

Fig. 5.5 shows the average base-10 logarithm of the MSE versus timestep for the different control strategies and the different RH parameters. We can see that active planning significantly reduces the MSE. More specifically, neither the “static” nor the “random” approach works in this scenario, in the sense that the average logarithm of the MSE resulting from these two approaches is huge. On the contrary, “FIM-Myopic” and “FIM-RH” can accurately localize and track the agent. As expected, “FIM-RH” yields an improved accuracy compared to “FIM-Myopic”.

To better understand the influence of the different RH parameters in this scenario, we compare three different RH parameter sets, i.e., $T = 2$; $T = 5, V = 100$; and $T = 20, V = 1000$.

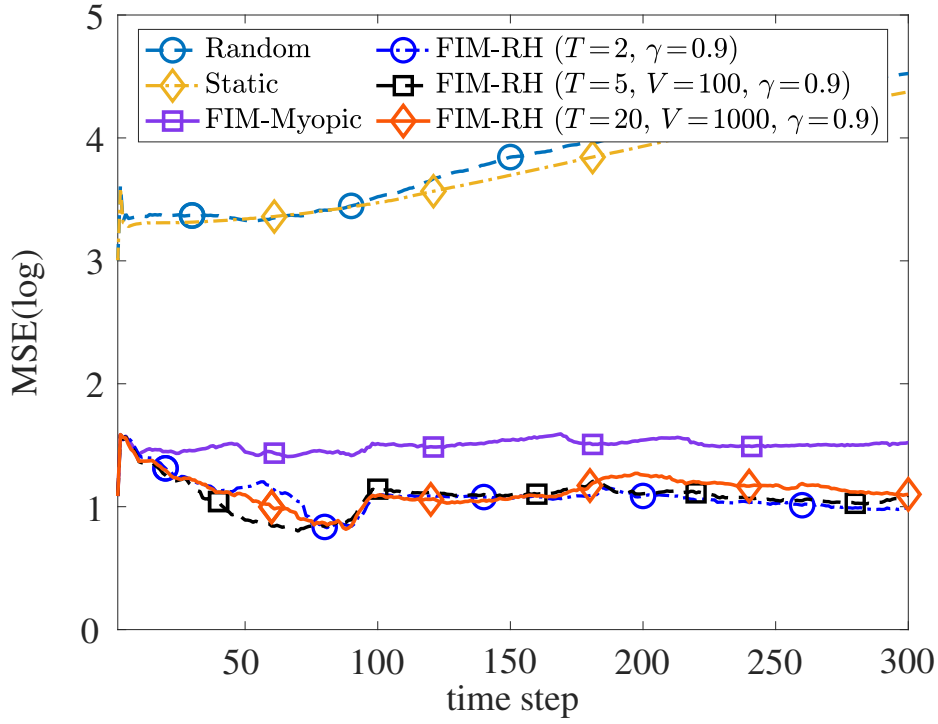


Figure 5.5. Logarithmic MSE of agent position versus time. The proposed active planning methods result in a significant reduction of the MSE.

The resulting estimation performance curves are shown in Fig. 5.5. These results indicate that in the considered scenario, high localization accuracy can already be achieved for $T = 2$, while $T = 5, V = 100$ and $T = 20, V = 1000$ perform very similarly. In this rather simple scenario, a planning horizon of $T = 2$ is enough to leverage the full estimation performance, and thus increasing T does not lead to improved accuracy. Longer planning horizons T are expected to help in larger scenarios with more anchors and more agents.

Lost Tracks

In the simulated scenario, we sometimes observe lost tracks, i.e., the agent position estimate deviates from the true agent position and thus the MSE grows with time. These track losses are related to the fact that we initialize the agent and anchor positions randomly, which can lead to geometries that are not favorable for estimating the position of the agent. This results

in a large MSE at initial time steps and the possibility that the estimation error grows with time. We declare a track loss if the MSE exceeds a value of 1000 at any timestep between timestep 3 to 50.

To determine the track loss percentage, we simulated 10000 random trails with 5 different methods, i.e., “static”, “random”, “FIM-Myopic”, “FIM-RH ($T = 2, \gamma = 0.9$)” and “FIM-RH ($T = 5, V = 100, \gamma = 0.9$)”. Note that we have not simulated “FIM-RH ($T = 20, V = 1000, \gamma = 0.9$)” since this would result in an excessive runtime as discussed below. Table 5.1 shows the track loss ratio of all methods. While the reference methods “Static” and “Random” result in more than 40% of lost tracks, the proposed methods can reduce the lost tracks to less than 1%. This demonstrates that the use of active planning can counteract initial network topologies that are unfavorable for agent localization

Table 5.1. Percentage of lost tracks for all simulated methods.

Setting	Lost Tracks (%)
Static	44.24
Random	46.02
FIM-Myopic	13.72
FIM-RH ($T = 2, \gamma = 0.9$)	0.66
FIM-RH ($T = 5, V = 100, \gamma = 0.9$)	0.62

Computational Complexity

The simulation is implemented using MATLAB on a 2017 MacBook Pro with an Intel Core-i5 processor. The runtime for a single run in the considered simulation scenario with 200 timesteps is 1.1 seconds for “FIM-Myopic”, 4.8 seconds for “FIM-RH ($T = 2, \gamma = 0.9$)”, 23 seconds for “FIM-RH ($T = 5, V = 100, \gamma = 0.9$)”, and 1964 seconds for “FIM-RH ($T = 20, V = 1000, \gamma = 0.9$)”. This demonstrates the low computational complexity of our method for short planning horizons T . However, as also analyzed in Section 5.2.2, the complexity increases rapidly with increasing T , which makes the current approach unsuitable for large networks.

Chapter 6

Conclusion

In this dissertation, we address two significant challenges in multiobject tracking (MOT) and localization in complex environments.

In Chapters 3 and 4, we tackle particle degeneracy in multisensor MOT and localization with high-dimensional object states. We improve the efficiency of IS by performing operations on the graph using PFI, while ensuring asymptotic optimality. Our contributions make it possible to evaluate the proposal PDF of IS using either deterministic or stochastic PFI, yielding accurate approximations of target beliefs in BP. Compared to deterministic PFI, stochastic PFI introduces a diffusion term into the PDE to model randomness in particle dynamics. A carefully chosen diffusion term reduces stiffness in this PDE, allowing PFI to operate with fewer numerical integration steps than traditional deterministic PFI. Our method, in particular, enables optimization of the diffusion matrix to improve the trade-off between accuracy and computational complexity.

Our numerical results show that:

- The proposed PFI-embedded IS and BP method achieves favorable tracking performance—both in estimation accuracy and computational complexity—compared to state-of-the-art methods in a highly nonlinear 3-D tracking and localization scenario using TDOA measurements.
- The main challenge for IS in 3-D MOT lies in representing the posterior distribution at

the initial time step after a new object appears, when the prior state PDF is much less informative than the high-accuracy, sharply peaked likelihood PDF.

- Using stochastic PFI (Chapter 4) reduces stiffness and improves estimation accuracy compared to deterministic PFI (Chapter 3). However, the trade-off between stiffness and accuracy depends on a heuristic hyperparameter that must be carefully tuned for optimal performance.

In Chapter 5, we present an active planning method for controlling mobile anchors in a localization network to maximize the information gain of future measurements. We use the trace of approximate inverse Bayesian FIMs as an efficient cost function for active planning suitable for RH control. Minimizing this cost function can be formulated as a tree-search problem. We also propose simple approximation strategies to solve the resulting tree-search efficiently. A numerical case study demonstrates that active planning with RH control can outperform myopic control.

For the proposed IS and BP methods with improved sampling efficiency, future research directions include flow-induced IS for various measurement models, such as extended object tracking [155, 69], simultaneous localization and object tracking [75], and information-seeking control. We also aim to demonstrate real-time processing capabilities on graphical processing units (GPUs), exploiting the high parallelizability of PFI [156]. Improving sampling efficiency is a fundamental problem not limited to MOT. Incorporating measurement information into sampling has broad implications for contextual robotics and cyber-physical systems, particularly when DA uncertainty must be quantified.

Further research opportunities include developing PFI methods both forward and backward in time with multimodal uncertainty quantification to enhance scalability. Probabilistic data association methods [157] may also prove useful for fusing information from multiple measurements into a flow process. Another promising direction involves innovative stochastic differential methods with reversibility for continuous-time state estimation [158], especially in

autonomous dynamic systems with multiple agents. In addition, learning-enhanced probabilistic inference with heterogeneous sensor fusion is an exciting frontier. With sufficient data about open-world environments, it is worth unifying reasoning-based and learning-based inference methods in complex cyber-physical systems, particularly for multi-sensor fusion. The integration of LiDAR, radar, and camera measurements can provide rich environmental information, and it is crucial to incorporate such heterogeneous data into a probabilistic inference framework using both signal/image processing and multimodal learning.

For the active planning method, the main limitation is that the current Bayesian FIM approximation does not incorporate measurements from other agents. Moreover, the present search strategies are expected to be highly suboptimal in large networks. Future work will address these limitations. In particular, we anticipate that advanced learning methods could enable near-optimal RH control for autonomous collaboration and distributed information collection in large sensing networks. Designing control policies that adapt to both the network structure and the sensing environment remains an open challenge. To be effective, smart robot controllers must be derived from global objective functions and implemented in a distributed fashion. We also plan to extend active planning to estimation problems with DA uncertainty. Another interesting direction is the integration of planning and learning [114] to improve scalability. Reinforcement learning is already widely used for search and planning problems, and future research should explore policy learning over infinite horizons using deep neural networks.

From a broader perspective, my research focuses on establishing scalable methodologies for high-dimensional, highly nonlinear machine perception and decision-making problems, especially for tracking and navigation in multi-agent dynamic systems with unknown interrelationships. By integrating statistical signal processing with probabilistic graphical models, I aim to develop inference frameworks that estimate the spatio-temporal state of multiple agents from unstructured sensor measurements. With prior knowledge of the system state, robot control methods can be designed for active sensing, thereby collecting informative measurements to refine estimation. Ultimately, my goal is to unify estimation and control optimization in order to

approach the theoretical performance bounds of autonomous systems.

With advances in semiconductor technology and precision instrumentation, and in the era of the IoT, tracking and localization are trending toward:

- Large-scale network localization and tracking [8, 159], where scalability and adaptability in complex data association are critical.
- Migration of tracking tasks from ground to low-altitude air/space intelligent transportation. This extends state dimensionality from 2-D to 3-D with nonlinear rotation ($SE(3)$), introducing additional degree of freedom (DoF) for perception and control in robotics.
- Multifunctional sensing devices capable of joint communication & perception, joint estimation & control, and joint planning & learning.
- Data-driven AI techniques that enhance parametric tracking/localization models, enabling seamless fusion of multimodal, heterogeneous sensor measurements with close coordination between physical and tracking layers.

Key thrusts for future research include:

1. developing fundamental signal and channel models for efficient spatial perception;
2. designing multi-robot cooperative localization algorithms with network-structure awareness; and
3. advancing actor–critic algorithms with explicit exploration–exploitation balance.

In summary, future tracking and localization in challenging environments will require a deep understanding of underlying statistical–physical models combined with flexible environmental adaptivity, creating new opportunities for iterative algorithm optimization. A good estimation method should advance the fundamental theory of autonomous systems operating in fast-evolving environments, with potential model enhancement through machine learning. Novel

methods for statistical estimation and active sensing will substantially improve perception and decision-making capabilities of autonomous systems, leading to tangible benefits across multiple sectors that deploy robots in unstructured environments. The nonlinear, high-dimensional estimation and control problems arising in dynamic, uncertain environments are closely aligned with emerging trends in autonomous transportation and scientific exploration.

Appendix A

Derivation PDE of PFI

A.1 Review of Fokker-Planck equation

A.1.1 Kramers-Moyal Expansion

The original aim of the Fokker-Planck and Langevin equations is to describe the Brownian motion. But in general, they could also explain the behavior of a system in presence of a random noise and its evolution toward a stationary state. We introduce a statistical method to describe the continuous evolution of a time-dependent probability distribution of a random variable x at time t , denoted as $p(x;t)$. This evolution is governed in terms of the moments w.r.t. the transition probability of particles $p(x_{t+\tau}|x_t;t,\tau)$. This describes the forward transition of a particle with position x_t at time t drawn from density $p(x;t)$ to a new position $x_{t+\tau}$ at time $t + \tau$, with τ a minor time interval.

The probability density of the particle $x_{t+\tau}$ is given by

$$p(x_{t+\tau};t+\tau) = \int p(x_t;t)p(x_{t+\tau}|x_t;t,\tau)dx_t. \quad (\text{A.1})$$

where the integral is carried over the whole space of $f(x;t)$. Note that (A.1) describes the forward evolution of this time dependent probability density.

Similarly, the n th-ordered moments around x_t at time t with minor τ can also be defined

as

$$M_n(x_t; t, \tau) = \int (x_{t+\tau} - x_t)^n p(x_{t+\tau} | x_t; t, \tau) dx_{t+\tau}. \quad (\text{A.2})$$

Theorem 1 (Kramers-Moyal Expansion) *In stochastic processes of a scaled random variable x , under general assumptions that the system density exists and is continuously differentiable with respect to time t and the state x ,*

$$p(x; t + \tau) - p(x; t) = \frac{\partial p(x; t)}{\partial t} \tau + \mathcal{O}(\tau^2) \quad (\text{A.3})$$

Kramers-Moyal expansion refers to a Taylor series expansion of the master equation¹, i.e.

$$\frac{\partial p(x; t)}{\partial t} = \lim_{\tau \rightarrow 0} \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \frac{\partial^n}{\partial x^n} \left[\frac{M_n(x; t, \tau)}{\tau} f(x; t) \right] = \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \frac{\partial^n}{\partial x^n} [D^n(x; t) p(x; t)]. \quad (\text{A.4})$$

where $M_n(x; t, \tau)$ is the n th-order moments of $p(x; t)$ and $D^n(x; t) = \lim_{\tau \rightarrow 0} \frac{M_n(x; t, \tau)}{\tau}$ is the Kramers-Moyal coefficients.

Note that the Kramers-Moyal expansion approximates the time evolution of a distribution by differential series of the state moments. *Fokker-Planck* equation or *Kolmogorov's forward* equation is introduced by keeping only the first two terms of the series in which $D^1(x; t)$ is the drift and $D^2(x; t)$ is the diffusion coefficient.

A.1.2 Fokker-Planck Equation

Note that the K-M expansion can be extended to vectorized $\mathbf{x}$. Now let's revisit the drift's physical meaning in 3D Euclidean space $\mathbb{R}^3$, which is exactly the migrating velocity of particle at position $\mathbf{x}$ and time t , i.e. $\lim_{\tau \rightarrow 0} \frac{M_1(\mathbf{x}; t, \tau)}{\tau} = \dot{\mathbf{x}}(t) = \boldsymbol{\zeta}(\mathbf{x}; t)$. Now the deterministic

¹A master equation is a phenomenological set of first-order differential equations describing the time evolution of (usually) the probability of a system to occupy each one of a discrete set of states with regard to a continuous time variable. Here it refers to $\frac{\partial f(x; t)}{\partial t}$.

(zero-diffusion) vectorized *Fokker-Planck* equation can be written as

$$\begin{aligned}
\frac{\partial p(\mathbf{x};t)}{\partial t} &= -\nabla_{\mathbf{x}} \cdot [\boldsymbol{\zeta}(\mathbf{x};t)p(\mathbf{x};t)] \\
&= -\left\langle \frac{\partial p(\mathbf{x};t)}{\partial \mathbf{x}}, \boldsymbol{\zeta}(\mathbf{x};t) \right\rangle - p(\mathbf{x};t) \nabla_{\mathbf{x}} \cdot \boldsymbol{\zeta} \\
&= -\left\langle \frac{\partial p(\mathbf{x};t)}{\partial \mathbf{x}}, \boldsymbol{\zeta}(\mathbf{x};t) \right\rangle - p(\mathbf{x};t) \text{Tr}\left(\frac{\partial \boldsymbol{\zeta}}{\partial \mathbf{x}}\right).
\end{aligned} \tag{A.5}$$

Here, the scalar derivative is replaced by the divergence on a vector field.

With chain rule we get

$$\frac{dp(\mathbf{x};t)}{dt} = \frac{\partial p(\mathbf{x};t)}{\partial t} + \frac{\partial p(\mathbf{x};t)}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} \tag{A.6}$$

Since $\frac{d\mathbf{x}}{dt} = \boldsymbol{\zeta}(\mathbf{x};t)$ the drift vector and subsequently combining (A.5) and (A.6), we get the total derivative of the PDF as

$$\frac{dp(\mathbf{x};t)}{dt} = -p(\mathbf{x};t) \text{Tr}\left(\frac{\partial \boldsymbol{\zeta}}{\partial \mathbf{x}}\right) \tag{A.7}$$

A.2 PDE of Bayesian Particle Flow

Denote the state of an object of interest as $\mathbf{x}$, the environmental observation w.r.t. $\mathbf{x}$ as $\mathbf{z}$, denote $f(\cdot)$ the PDF, e.g., prior as $f(\mathbf{x})$, the conditional likelihood as $f(\mathbf{z}|\mathbf{x})$, the posterior as $f(\mathbf{x}|\mathbf{z})$. Given an observation $\mathbf{z}$, the Bayes' rule shows the posterior PDF as

$$f(\mathbf{x}|\mathbf{z}) = \frac{f(\mathbf{z}|\mathbf{x})f(\mathbf{x})}{f(\mathbf{z})}. \tag{A.8}$$

Since $\mathbf{z}$ is given and fixed, the denominator $f(\mathbf{z})$ is a constant. We have $f(\mathbf{x}|\mathbf{z}) \propto f(\mathbf{z}|\mathbf{x})f(\mathbf{x})$.

Bayes' rule in a Markovian process can be used to identify an unnormalized marginal posterior density as

$$\tilde{f}(\mathbf{x}_k|\mathbf{z}_{1:k}) = f(\mathbf{x}_k|\mathbf{z}_{1:k-1})f(\mathbf{z}_k|\mathbf{x}_k) \tag{A.9}$$

In particle based filter algorithms, particles are usually drawn from a density call proposal. The BPF simply sampling from the prior distribution. However, it suffers particle degeneracy in high dimensional state space of $\mathbf{x}$. A good sampling strategy should be approximately posterior, whose analytical solution is typically hard to achieve. Inspired by A.1, we try to describe the evolution of a sampling density from prior to posterior as a stochastic process driven by the likelihood function w.r.t. a pseudotime t . This method is introduced as particle flow [88, 84, 85, 86, 87]. We dismiss the time index k by denoting $\pi(\mathbf{x}) = \tilde{f}(\mathbf{x}_k|\mathbf{z}_{1:k})$ the posterior, $g(\mathbf{x}) = f(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ the prior/prediction and $l(\mathbf{x}) = f(\mathbf{z}_k|\mathbf{x}_k)$ the likelihood, Duam and Huang introduce a log-homotopy function [84, 85]

$$\phi(\mathbf{x};t) = \log g(\mathbf{x}) + t \log l(\mathbf{x}) \quad (\text{A.10})$$

where $t \in [0, 1]$ is the pseudo-time to describe the flow process. The homotopy defines a continuous and smooth deformation of the topological space of vector $\mathbf{x}$, from log-predictive distribution $\phi(\mathbf{x};0) = \log g(\mathbf{x})$ to log-posterior (unnormalized) distribution $\phi(\mathbf{x};1) = \log \pi(\mathbf{x})$. With $f(\mathbf{x};t) = \exp^{\phi(\mathbf{x};t)}$ the probability density in the stochastic process, the zero-diffusion total derivative of the log-homotopy using (A.7) is derived as

$$\begin{aligned} \frac{d\phi(\mathbf{x};t)}{dt} &= \frac{d \log(f(\mathbf{x};t))}{dt} \\ &= \frac{1}{f(\mathbf{x};t)} \frac{df(\mathbf{x};t)}{dt} \\ &= -\text{Tr}\left(\frac{\partial \boldsymbol{\zeta}}{\partial \mathbf{x}}\right) \end{aligned} \quad (\text{A.11})$$

With (A.10), using chain rule on $\phi(\mathbf{x};t)$, we get

$$\frac{d\phi(\mathbf{x};t)}{dt} = \frac{\partial \phi}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} + \frac{\partial \phi}{\partial t} = \frac{\partial \phi}{\partial \mathbf{x}} \boldsymbol{\zeta}(\mathbf{x};t) + \log l(\mathbf{x}) \quad (\text{A.12})$$

with ϕ in abbreviation of $\phi(\mathbf{x};t)$. Combining (A.11) and (A.12), the exact particle flow [86] is

derived in form of PDE as

$$\frac{\partial \phi}{\partial \mathbf{x}} \boldsymbol{\zeta}(\mathbf{x}; t) + \log l(\mathbf{x}) = -\text{Tr}\left(\frac{\partial \boldsymbol{\zeta}}{\partial \mathbf{x}}\right) \quad (\text{A.13})$$

The solution of $\boldsymbol{\zeta}$ according to (A.13) specifies the exact flow of the probability density. In [135], several methods to get solutions of (A.13) are given.

Bibliography

- [1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY: Wiley, 2001.
- [2] O. Hlinka, F. Hlawatsch, and P. M. Djuric, “Distributed particle filtering in agent networks: A survey, classification, and comparison,” *IEEE Signal Process. Mag.*, vol. 30, pp. 61–81, Jan. 2013.
- [3] F. Meyer and K. L. Gemba, “Probabilistic focalization for shallow water localization,” *J. Acoust. Soc. Am.*, vol. 150, pp. 1057–1066, 08 2021.
- [4] 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. Am.*, vol. 153, p. 2690–2705, May 2023.
- [5] L. Watkins, P. Stinco, A. Tesei, and F. Meyer, “Sequential source localization in shallow water by single-receiver probabilistic focalization,” *J. Acoust. Soc. Am.*, vol. 157, no. 4, pp. 2819–2834, 2025.
- [6] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [7] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Z. Win, “A scalable belief propagation algorithm for radio signal based SLAM,” *IEEE Trans. Wireless Commun.*, vol. 18, Nov. 2019.
- [8] M. Z. Win, F. Meyer, Z. Liu, W. Dai, S. Bartoletti, and A. Conti, “Efficient multi-sensor localization for the Internet-of-Things,” *IEEE Signal Process. Mag.*, vol. 35, pp. 153–167, Sept. 2018.
- [9] B. Teague, Z. Liu, F. Meyer, A. Conti, and M. Z. Win, “Network localization and navigation with scalable inference and efficient operation,” *IEEE Trans. Mob. Comput.*, vol. 21, no. 6, pp. 2072–2087, 2022.
- [10] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. USA: Princeton University Press, 2009.

- [11] T. Shima and S. Rasmussen, *UAV Cooperative Decision and Control*. Society for Industrial and Applied Mathematics, 2009.
- [12] 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, Nov. 2015.
- [13] S. Godsill and P. Rayner, *Digital Audio Restoration*. Springer London, 1998.
- [14] P. Wolfe and S. Godsill, “Efficient alternatives to the ephraim and malah suppression rule for audio signal enhancement,” *EURASIP J. Adv. Signal Process.*, 2003.
- [15] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, “Wireless sensor networks for healthcare,” *Proc. IEEE*, vol. 98, pp. 1947–1960, Nov. 2010.
- [16] T. Zhao and A. Nehorai, “Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 55, pp. 1511–1524, Apr. 2007.
- [17] A. Nayak and I. Stojmenovic, *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*. Wiley, 2010.
- [18] K. Cuthbertson and D. Nitzsche, *Quantitative Financial Economics: Stocks, Bonds and Foreign Exchange*. Wiley, 2005.
- [19] M. Crocker, *Handbook of Acoustics*. Wiley, 1998.
- [20] S. Silver, *Microwave Antenna Theory and Design*. P. Peregrinus, 1984.
- [21] P. McManamon, *LiDAR Technologies and Systems*. SPIE Press, 2019.
- [22] B. Jähne, *Digital Image Processing*. Springer Berlin Heidelberg, 2005.
- [23] M. Skolnik, *Radar Handbook*. McGraw-Hill, 1990.
- [24] A. Waite, *Sonar for Practising Engineers*. Wiley, 2002.
- [25] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*. Norwood, MA, USA: Artech House, 1996.
- [26] D. Dardari, M. Luise, and E. Falletti, *Satellite and Terrestrial Radio Positioning Techniques: A Signal Processing Perspective*. Academic Press, Nov. 2011.
- [27] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008.
- [28] J. Gibson, *The Perception of the Visual World*. Houghton Mifflin, 1950.

- [29] Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1403–1410 vol.2, 2003.
- [30] M. Taj and A. Cavallaro, “Distributed and decentralized multicamera tracking,” *IEEE Signal Process. Mag.*, vol. 28, pp. 46–58, May 2011.
- [31] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, New Jersey 07632: Prentice Hall, first ed., 1989.
- [32] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing*. New York, NY, 10020: McGraw-Hill, Inc., first ed., 2000.
- [33] F. Gustafsson and F. Gunnarsson, “Positioning using time-difference of arrival measurements,” in *Proc. IEEE ICASSP-03*, vol. 6, (Hong Kong, China), pp. 553–556, Apr. 2003.
- [34] M. Vanderveen, C. Papadias, and A. Paulraj, “Joint angle and delay estimation (jade) for multipath signals arriving at an antenna array,” *IEEE Commun. Lett.*, vol. 1, no. 1, pp. 12–14, 1997.
- [35] S. Mazuelas, F. A. Lago, D. Gonzalez, A. Bahillo, J. Blas, P. Fernandez, R. Lorenzo, and E. Abril, “Dynamic estimation of optimum path loss model in a RSS positioning system,” in *IEEE/ION Position, Location and Navigation Symposium*, pp. 679–684, May 2008.
- [36] D. Dardari, “Pseudo-random active UWB reflectors for accurate ranging,” *IEEE Commun. Lett.*, vol. 8, pp. 608–610, Oct. 2004.
- [37] G. E. Garcia, L. S. Muppirisetty, and H. Wymeersch, “On the trade-off between accuracy and delay in cooperative UWB navigation,” in *Proc. IEEE Wireless Commun. and Networking Conf.*, (Shanghai, China), pp. 1603–1608, Apr. 2013.
- [38] E. Leitinger, M. Fröhle, P. Meissner, and K. Witrisal, “Multipath-assisted maximum-likelihood indoor positioning using uwb signals,” in *Proc. ICC-14*, pp. 170–175, 2014.
- [39] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *ASME J. Basic Eng.*, vol. 82, pp. 35–45, 03 1960.
- [40] S. Haykin, “Kalman filters,” in *Kalman Filtering and Neural Networks* (S. Haykin, ed.), ch. 1, pp. 1–21, New York, NY, USA: Wiley, 2001.
- [41] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [42] 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.

- [43] M. 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, pp. 174–188, Feb. 2002.
- [44] O. Hlinka, F. Hlawatsch, and P. M. Djuric, “Consensus-based distributed particle filtering with distributed proposal adaptation,” *IEEE Trans. Signal Process.*, vol. 62, pp. 3029–3041, Jun. 2014.
- [45] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, “Set-membership constrained particle filter: Distributed adaptation for sensor networks,” *IEEE Trans. Signal Process.*, vol. 59, pp. 4122–4138, Sep. 2011.
- [46] 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), pp. 318–329, Inst. Math. Statist., 2008.
- [47] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proc. IEEE*, vol. 92, pp. 401–422, Mar. 2004.
- [48] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT: Yaakov Bar-Shalom, 2011.
- [49] D. B. Reid, “An algorithm for tracking multiple targets,” *IEEE Trans. Autom. Control*, vol. 24, pp. 843–854, Dec. 1979.
- [50] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2007.
- [51] J. L. Williams, “Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA and association-based MeMBer,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, pp. 1664–1687, Jul. 2015.
- [52] A. Saucan, Y. Li, and M. Coates, “Particle flow SMC delta-GLMB filter,” in *Proc. IEEE ICASSP-17*, (New Orleans, LA, USA), pp. 4381–4385, Mar. 2017.
- [53] B.-N. Vo, B.-T. Vo, and D. Phung, “Labeled random finite sets and the Bayes multi-target tracking filter,” *IEEE Trans. Signal Process.*, vol. 62, pp. 6554–6567, Dec. 2014.
- [54] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.
- [55] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Found. Trends Mach. Learn.*, vol. 1, Jan. 2008.
- [56] H.-A. Loeliger, “An introduction to factor graphs,” *IEEE Signal Process. Mag.*, vol. 21, pp. 28–41, Jan. 2004.

- [57] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.
- [58] Y. Weiss and W. T. Freeman, “Correctness of belief propagation in Gaussian graphical models of arbitrary topology,” *Neural Computation*, vol. 13, pp. 2173–2200, Oct. 2001.
- [59] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, “Nonparametric belief propagation for self-localization of sensor networks,” *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 809–819, Apr. 2005.
- [60] 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, pp. 174–188, Feb. 2002.
- [61] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative localization in wireless networks,” *Proc. IEEE*, vol. 97, pp. 427–450, Feb. 2009.
- [62] C. Pedersen, T. Pedersen, and B. H. Fleury, “A variational message passing algorithm for sensor self-localization in wireless networks,” in *Proc. IEEE ISIT-11*, pp. 2158–2162, 2011.
- [63] B. Cakmak, D. N. Urup, F. Meyer, T. Pedersen, B. H. Fleury, and F. Hlawatsch, “Cooperative localization for mobile networks: A distributed belief propagation-mean field message passing algorithm,” *IEEE Signal Process. Lett.*, vol. 23, pp. 828–832, 2016.
- [64] 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, pp. 3478–3493, July 2017.
- [65] F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, “Message passing algorithms for scalable multitarget tracking,” *Proc. IEEE*, vol. 106, pp. 221–259, Feb. 2018.
- [66] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge, UK: Cambridge University Press, 2011.
- [67] W. Koch, *Tracking and Sensor Data Fusion: Methodological Framework and Selected Applications*. Berlin, Germany: Springer, 2014.
- [68] L. Zhang, Y. Li, and R. Nevatia, “Global data association for multi-object tracking using network flows,” in *Proc. CVPR-08*, pp. 1–8, 2008.
- [69] 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.
- [70] J. W. Koch, “Bayesian approach to extended object and cluster tracking using random matrices,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, pp. 1042–1059, Jul. 2008.

- [71] K. Granstrom, C. Lundquist, and O. Orguner, "Extended target tracking using a Gaussian-mixture PHD filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 3268–3286, 2012.
- [72] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proc. FUSION-03*, vol. 2, pp. 792–799, 2003.
- [73] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, 2006.
- [74] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularised particle filters," in *Sequential Monte Carlo Methods in Practice*, pp. 247–271, Springer, New York, NY, 2001.
- [75] 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, pp. 57–71, Mar. 2016.
- [76] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [77] 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, 2017.
- [78] 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, 2019.
- [79] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Am. Statist. Assoc.*, vol. 94, pp. 590–599, June 1999.
- [80] P. M. Djuric, T. Lu, and M. F. Bugallo, "Multiple particle filtering," in *Proc. ICASSP-07*, vol. 3, pp. 1181–1184, 2007.
- [81] C. Berzuini, N. G. Best, W. R. Gilks, and C. Larizza, "Dynamic conditional independence models and Markov chain Monte Carlo methods," *Journal of the American Statistical Association*, vol. 92, no. 440, pp. 1403–1412, 1997.
- [82] W. R. Gilks and C. Berzuini, "Following a moving target-Monte Carlo inference for dynamic Bayesian models," *J. R. Stat. Soc. Series B Stat. Methodol.*, vol. 63, no. 1, pp. 127–146, 2001.
- [83] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," in *Proc. NIPS-00*, (Denver, CO, USA), pp. 584–590, Dec. 2000.
- [84] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proc. SPIE-07*, pp. 423–437, Aug. 2007.

- [85] F. Daum and J. Huang, “Nonlinear filters with particle flow induced by log-homotopy,” in *Proc. SPIE-09*, pp. 76–87, May 2009.
- [86] F. Daum, J. Huang, and A. Noushin, “Exact particle flow for nonlinear filters,” in *Proc. SPIE-10*, pp. 92–110, Apr. 2010.
- [87] F. Daum and J. Huang, “Particle flow with non-zero diffusion for nonlinear filters,” in *Proc. SPIE-13*, pp. 226–238, May 2013.
- [88] P. Bunch and S. Godsill, “Approximations of the optimal importance density using Gaussian particle flow importance sampling,” *J. Amer. Statist. Assoc.*, vol. 111, pp. 748–762, Aug. 2016.
- [89] Y. Li and M. Coates, “Particle filtering with invertible particle flow,” *IEEE Trans. Signal Process.*, vol. 65, pp. 4102–4116, Aug. 2017.
- [90] Y. Yi, J. Cortés, and N. Atanasov, “Variational formulation of the particle flow particle filter,” 2025. Arxiv preprint stat.ML/2505.04007.
- [91] Q. Liu and D. Wang, “Stein variational gradient descent: a general purpose bayesian inference algorithm,” in *Proc. NIPS-16*, p. 2378–2386, 2016.
- [92] A. Doucet, W. S. Grathwohl, A. G. D. G. Matthews, and H. Strathmann, “Score-based diffusion meets annealed importance sampling,” in *Proc. NIPS-22*, 2022.
- [93] T. Geffner and J. Domke, “Langevin diffusion variational inference,” in *Proc. AISTATS-23*, pp. 576–593, 2023.
- [94] S. Pal and M. Coates, “Gaussian sum particle flow filter,” in *Proc. IEEE CAMSAP-18*, (Curaçao, Dutch Antilles), Dec. 2017.
- [95] S. Pal and M. Coates, “Particle flow particle filter for Gaussian mixture noise models,” in *Proc. IEEE ICASSP-18*, (Calgary, Canada), pp. 4249–4253, 2018.
- [96] 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, 2019.
- [97] 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.
- [98] L. Wielandner, E. Leitinger, F. Meyer, B. Teague, and K. Witrisal, “Message passing-based cooperative localization with embedded particle flow,” in *Proc. IEEE ICASSP-22*, pp. 5652–5656, 2022.
- [99] 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, 2017.

- [100] 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, 2022.
- [101] G. Ferri, A. Munafò, A. Tesei, P. Braca, F. Meyer, K. Pelekanakis, R. Petroccia, J. Alves, C. Strode, and K. LePage, “Cooperative robotic networks for underwater surveillance: an overview,” *IET Radar Sonar Navig.*, vol. 11, pp. 1740–1761, Dec. 2017.
- [102] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, “Distributed robotic sensor networks: An information-theoretic approach,” *Int. J. Robot. Res.*, vol. 31, no. 10, 2012.
- [103] W. Burgard, D. Fox, and S. Thrun, “Active mobile robot localization by entropy minimization,” in *Proc. EUROBOT-97*, pp. 155–162, 1997.
- [104] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, “Information-theoretic coordinated control of multiple sensor platforms,” in *Proc. IEEE ICRA-02*, pp. 1521–1526 vol.1, 2003.
- [105] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 2006.
- [106] F. Morbidi and G. L. Mariottini, “Active target tracking and cooperative localization for teams of aerial vehicles,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1694–1707, 2013.
- [107] G. M. Hoffmann and C. J. Tomlin, “Mobile sensor network control using mutual information methods and particle filters,” *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 32–47, 2010.
- [108] M. Schwager, P. M. Dames, D. Rus, and V. Kumar, “A multi-robot control policy for information gathering in the presence of unknown hazards,” in *Proc. ISRR-11*, pp. 455–472, 2011.
- [109] N. A. Atanasov, J. Le Ny, and G. J. Pappas, “Distributed algorithms for stochastic source seeking with mobile robot networks,” *J. Dyn. Syst. Meas. Control Trans.*, vol. 137, 10 2014.
- [110] F. Meyer, H. Wymeersch, and F. Hlawatsch, “Cooperative localization with information-seeking control,” in *Proc. ICASSP’15*, pp. 2854–2858, 2015.
- [111] 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, Nov. 2015.
- [112] J. Mattingley, Y. Wang, and S. Boyd, “Receding horizon control,” *IEEE Control Syst. Mag.*, vol. 31, no. 3, pp. 52–65, 2011.
- [113] J. Schrittwieser, T. Hubert, A. Mandhane, M. Barekatin, I. Antonoglou, and D. Silver, “Online and offline reinforcement learning by planning with a learned model,” in *Proc. NeurIPS-21*, vol. 34, pp. 27580–27591, 2021.

- [114] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, “Plan online, learn offline: Efficient learning and exploration via model-based control,” 2019. Arxiv preprint cs.LG/1811.01848, 2019.
- [115] D. Crisan and A. Doucet, “A survey of convergence results on particle filtering methods for practitioners,” *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 736–746, 2002.
- [116] S. Maskell and S. Julier, “Optimised proposals for improved propagation of multi-modal distributions in particle filters,” in *Proc. FUSION-13*, pp. 296–303, 2013.
- [117] F. Daum, J. Huang, and A. Noushin, “New theory and numerical results for Gromov’s method for stochastic particle flow filters,” in *Proc. FUSION-18*, pp. 108–115, 2018.
- [118] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.
- [119] 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, 2023.
- [120] Y. Shen and M. Z. Win, “Fundamental limits of wideband localization— part i: A general framework,” *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4956–4980, 2010.
- [121] B.-N. Vo, M. Mallick, Y. Bar-Shalom, S. Coraluppi, R. Osborne, R. Mahler, and B.-T. Vo, “Multitarget tracking,” in *Wiley Encyclopedia of Electrical and Electronics Engineering* (M. Peterca, ed.), Hoboken, NJ, USA: Wiley, 2015.
- [122] K. Granström, M. Baum, and S. Reuter, “Extended object tracking: Introduction, overview and applications,” 2017.
- [123] T. Bayes and n. Price, “Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s,” *Philosophical Transactions of the Royal Society of London*, vol. 53, pp. 370–418, 1763.
- [124] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, 1960.
- [125] E. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proc. of the IEEE 2000 AS-SPCC*, pp. 153–158, 2000.
- [126] B. Schutz, B. Tapley, and G. H. Born, *Statistical Orbit Determination*. Elsevier, 2004.
- [127] W. Zhang and F. Meyer, “Multisensor multiobject tracking with improved sampling efficiency: Supplementary material,” 2024. <https://fmeyer.ucsd.edu/TSP-ZhaMey-2024-SM.pdf>.
- [128] J. L. Williams and R. Lau, “Approximate evaluation of marginal association probabilities with belief propagation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, pp. 2942–2959, Oct. 2014.

- [129] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, second ed., 1994.
- [130] F. Daum and J. Huang, “Curse of dimensionality and particle filters,” in *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, vol. 4, pp. 1979 – 1993, Mar. 2003.
- [131] W. Zhang and F. Meyer, “Graph-based multiobject tracking with embedded particle flow,” in *Proc. IEEE RadarConf-21*, (Atlanta, GA, USA), May 2021.
- [132] W. Zhang and F. Meyer, “Multisensor multiobject tracking with improved sampling efficiency,” *IEEE Trans. Signal Process.*, vol. 72, pp. 2036–2053, 2024.
- [133] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin, *Bayesian Data Analysis, Third Edition*. Taylor & Francis, 2013.
- [134] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Berlin, Germany: Springer, 1992.
- [135] F. Daum and J. Huang, “Exact particle flow for nonlinear filters: Seventeen dubious solutions to a first order linear underdetermined PDE,” in *Proc. Asilomar-10*, (Pacific Grove, CA, USA), pp. 64–71, Nov. 2010.
- [136] T. Ding and M. J. Coates, “Implementation of the daum-huang exact-flow particle filter,” in *Proc. IEEE SSP-12*, pp. 257–260, 2012.
- [137] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Taylor & Francis, 1986.
- [138] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, “Finite mixture models,” *Annu. Rev. Stat. Appl.*, vol. 6, no. 1, pp. 355–378, 2019.
- [139] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*. USA: Cambridge University Press, 2007.
- [140] 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, 2019.
- [141] S. M. Wiggins and J. A. Hildebrand, “High-frequency acoustic recording package (HARP) for broad-band, long-term marine mammal monitoring,” in *Proc. UT-07*, pp. 551–557, Apr. 2007.
- [142] 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, 2008.
- [143] 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. Underw. Technol.*, pp. 551–557, 2007.

- [144] W. Zhang, M. J. Khojasteh, N. A. Atanasov, and F. Meyer, “Importance sampling with stochastic particle flow and diffusion optimization,” *IEEE Signal Process. Lett.*, vol. 32, pp. 2304–2308, 2025.
- [145] W. Zhang, M. J. Khojasteh, and F. Meyer, “Particle flows for source localization in 3-D using TDOA measurements,” in *Proc. FUSION-24*, (Venice, Italy), 2024.
- [146] 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, 2023.
- [147] S. Pal and M. Coates, “Particle flow particle filter using Gromov’s method,” in *Proc. IEEE CAMSAP-19*, (Guadeloupe, France), pp. 634–638, Dec. 2019.
- [148] D. F. Crouse, “Particle flow filters: Biases and bias avoidance,” in *Proc. FUSION-22*, (Ottawa, Canada), pp. 1–8, 2019.
- [149] Y. Li, L. Zhao, and M. Coates, “Particle flow auxiliary particle filter,” in *Proc. IEEE CAMSAP-15*, (Cancun, Mexico), pp. 157–160, Dec. 2015.
- [150] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.
- [151] W. Zhang, B. Teague, and F. Meyer, “Active planning for cooperative localization: A Fisher information approach,” in *Proc. Asilomar-22*, (Pacific Grove, CA, USA), pp. 795–800, 2022.
- [152] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative localization in wireless networks,” *Proc. IEEE*, vol. 97, pp. 427–450, Feb. 2009.
- [153] F. Meyer, O. Hlinka, and F. Hlawatsch, “Sigma point belief propagation,” *IEEE Signal Process. Lett.*, vol. 21, pp. 145–149, Feb. 2014.
- [154] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: Wiley-Interscience, 1994.
- [155] F. Meyer and J. L. Williams, “Scalable detection and tracking of geometric extended objects,” *IEEE Trans. Signal Process.*, vol. 69, pp. 6283–6298, Oct. 2021.
- [156] V. P. Jilkov, J. Wu, and H. Chen, “Performance comparison of GPU-accelerated particle flow and particle filters,” in *Proc. FUSION-13*, pp. 1095–1102, 2013.
- [157] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 82–100, 2009.
- [158] W. Talbot, J. Nubert, T. Tuna, C. Cadena, F. Dümbgen, J. Tordesillas, T. D. Barfoot, and M. Hutter, “Continuous-time state estimation methods in robotics: A survey,” *IEEE Trans. Robot.*, vol. 41, pp. 4975–4999, 2025.

- [159] B. Teague, Z. Liu, F. Meyer, and M. Z. Win, “Peregrine: 3-D network localization and navigation,” in *Proc. IEEE LATINCOM-17*, (Guatemala City, Guatemala), pp. 1–6, Nov. 2017.