

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Scalable Inference in Probabilistic Graphical Models for Bayesian Machine Perception

Permalink

<https://escholarship.org/uc/item/1qm5c7pb>

ISBN

9798293888276

Author

Liang, Mingchao

Publication Date

2025-09-27

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Scalable Inference in Probabilistic Graphical Models for Bayesian Machine Perception

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

in

Electrical Engineering
(Machine Learning and Data Science)

by

Mingchao Liang

Committee in charge:

Professor Florian Meyer, Chair
Professor Nicholas A. Antipa
Professor Sicun Gao
Professor Bhaskar D. Rao

2025

Copyright
Mingchao Liang, 2025
All rights reserved.

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

University of California San Diego

2025

DEDICATION

To all that has been, the welcome and the unwelcome, for making me who I am.

TABLE OF CONTENTS

Dissertation Approval Page	iii
Dedication	iv
Table of Contents	v
List of Figures	viii
List of Tables	xi
Acknowledgements	xii
Vita	xiv
Abstract of the Dissertation	xvi
Chapter 1 Introduction	1
1.1 Bayesian Estimation and Graph-Based Processing	2
1.1.1 The Bayesian Approach to Parameter Estimation	2
1.1.2 Factor Graphs and Belief Propagation	3
1.1.3 Sequential Bayesian Estimation	4
1.2 Multi-Object Tracking	6
1.2.1 Detect-Then-Track (DTT) Methods	7
1.2.2 Track-Before-Detect (TBD) Methods	8
1.3 Multipath-Based SLAM	10
1.4 Contributions and Outline	11
Chapter 2 Neural Enhanced Belief Propagation for Multiobject Tracking	14
2.1 Chapter Overview	15
2.1.1 Model-Based and Data-Driven MOT	15
2.1.2 Contribution and Chapter Organization	17
2.2 Review of Belief Propagation (BP) and Neural Enhanced Belief Propagation (NEBP)	19
2.2.1 Factor Graph and BP	19
2.2.2 NEBP	21
2.3 System Model and Statistical Formulation	24
2.3.1 Potential Objects and Object States	24
2.3.2 Data Association Vector and Measurement Model	25
2.3.3 Object Dynamics	26
2.3.4 Declaration, Estimation, Initialization, and Termination	27
2.4 Conventional BP-based MOT Algorithm	28
2.4.1 Alternative MOT Factor Graph	28
2.4.2 BP for MOT	30

2.5	Proposed NEBP-based MOT Algorithm	34
2.5.1	Feature Extraction	34
2.5.2	GNN Topology and BP Message Enhancement	35
2.5.3	Statement of the NEBP for MOT Algorithm	38
2.5.4	Complexity Analysis	40
2.6	Loss Function and Training	41
2.7	Numerical Results	43
2.7.1	Experimental Setup	43
2.7.2	Implementation Details	44
2.7.3	Calibration	46
2.7.4	Performance Evaluation	47
2.7.5	Ablation Study	50
2.8	Acknowledgement	53
Chapter 3	Message Passing for Track-Before-Detect	54
3.1	Chapter Overview	55
3.1.1	State-of-the-Art MOT	55
3.1.2	Contributions and Chapter Organization	57
3.2	Signal Model and Problem Formulation	59
3.2.1	Measurement Model	60
3.2.2	State-Transition Models	62
3.2.3	Object Birth Models	62
3.2.4	Joint Posterior PDF and Factor Graph	64
3.3	Proposed Method	65
3.3.1	Prediction and Birth Messages	66
3.3.2	Iterative Measurement Update Messages	68
3.3.3	Message Approximation and Belief Calculation	70
3.4	Experimental Results	72
3.4.1	Synthetic Radar Tracking Scenario	73
3.4.2	High-SNR Passive Acoustic Scenario	77
3.4.3	Low-SNR Passive Acoustic Scenario	81
3.5	Acknowledgment	84
Chapter 4	Direct Multipath-Based SLAM	85
4.1	Chapter Overview	86
4.1.1	Multipath-Based SLAM and State of the Art	86
4.1.2	Contributions and Chapter Organization	88
4.2	System Model and Problem Formulation	90
4.2.1	Radio Signal and Specular MPCs	90
4.2.2	State Vectors and Measurement Model	92
4.2.3	State-Transition Models and Prior PDFs	94
4.2.4	Declaration and Estimation	97
4.3	The Proposed BP Algorithm	99
4.3.1	Prediction and Birth Messages	100

4.3.2	Measurement Update Messages	102
4.3.3	Approximate Update and Belief Calculation	103
4.4	Particle-Based Implementation	105
4.4.1	Prediction and Birth	106
4.4.2	Measurement Update and Belief Calculation	106
4.4.3	Complexity Analysis	109
4.5	Numerical Results	110
4.5.1	Experimental Setup	111
4.5.2	Results of Synthetic Data	113
4.5.3	Results of Real Data	120
4.6	Appendices	121
4.6.1	Computation of Matrices $\mathbf{C}_{\iota,k}^{(j)}(\mathbf{x}_k)$, $\mathbf{C}_{\kappa,k,n}^{(j)}(\phi_{k,n}^{(j)})$, and $\mathbf{C}_{\nu,k}^{(j)}$	121
4.6.2	Monte Carlo Integration of $\mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_k)$, $\mathbf{C}_{2,k,n}^{(j)}(\phi_{k,n}^{(j)})$, and $\mathbf{C}_{3,k,n}^{(j)}$	124
4.6.3	Alternative Approximation of Measurement Update Messages	125
4.6.4	Qualitative Real Data Results	127
4.7	Acknowledgement	129
Chapter 5	Conclusions	130
	Bibliography	133

LIST OF FIGURES

Figure 2.1.	Flow diagram of one time step of conventional BP and the proposed NEBP method for MOT. Black boxes show the computation modules performed by both BP and NEBP. The red dotted boxes show additional modules only performed by the proposed NEBP method. The goal of both MOT methods	18
Figure 2.2.	Factor graph (a) and its corresponding GNN (b) of a simple NEBP example.	20
Figure 2.3.	Factor graph (a) and corresponding bipartite GNN (b) for a single time step k of the considered NEBP approach for MOT. BP and GNN messages are shown. The time index k is omitted. A GNN node was introduced for each legacy PO and each new PO. The topology of the GNN, which only	31
Figure 2.4.	Top-down view of the considered autonomous driving scene. For each ground truth and estimated vehicle, we plot tracks as well as positions and bounding boxes at the last time step. This scene is part of the <i>nuScenes</i> autonomous driving dataset.	45
Figure 2.5.	Top-down view and single time step for an example autonomous driving scene. Ground truth objects (a) as well as estimated object provided by CenterPointT (b), belief propagation (BP) (c), and neural enhanced belief propagation (NEBP) (d) are shown. Note that BP and NEBP do not provide	49
Figure 3.1.	Factor graph representing the joint posterior probability density function (PDF) in (3.4) and the BP messages for a single time step k . The time index k is omitted. BP messages are shown as blue arrows. The following shorthand notations are used: $\underline{N} = N_{k-1}$, $N = N_k$, $\mathbf{x}_n = \mathbf{x}_{k,n}$, $\boldsymbol{\phi}_n = \boldsymbol{\phi}_{k,n}$...	65
Figure 3.2.	Visualization of the synthetic radar tracking scenario in an example simulation run: (a) Trajectories of the objects with their initial positions marked as red crosses and sensor positions marked as blue stars. (b) Tracking results of different methods at time step $k = 41$. An SNR of -3 dB was ..	75
Figure 3.3.	GOSPA error of different tracking methods for synthetic radar tracking averaged over 100 simulation runs at (a) 0 dB, (b) -3 dB, and (c) -6 dB. .	76
Figure 3.4.	Estimated and true noise variance $\eta_k^{(i)}$ averaged over 100 simulations runs for (a) the sensor at $[0\ 0]^T$ and (b) the sensor at $[50\ 50]^T$. An SNR of -3 dB was considered.	78
Figure 3.5.	Bartlett spectrum of high tonal set in dB at time $k = 246$. The red circles are the true positions of the two sources.....	79

Figure 3.6.	GOSPA error in the high-SNR passive acoustic scenario. Different tracking methods based on the high tonal set are considered.	79
Figure 3.7.	Range estimates of different methods in the high-SNR passive acoustic scenario. The true range of the two sources is indicated by black lines. ...	81
Figure 3.8.	Bartlett spectrum in dB at time $k = 1$ in the low-SNR passive acoustic scenario with virtual array processing. The red circle is the true position of the source.	83
Figure 3.9.	GOSPA error of different tracking methods in the low-SNR passive acoustic scenario.	83
Figure 4.1.	Flow diagram of the proposed Direct-SLAM compared to conventional multipath-based SLAM. Discrete time is indexed by k . Conventional methods process received radio signals in two stages. In stage one, a channel estimator extracts the parameters of MPCs, e.g., the delays, from J	88
Figure 4.2.	Scenario with one mobile agent, one physical anchor (PA), and two virtual anchors (VAs). The radio signal transmitted by the agent at position $\mathbf{p}_k$ propagates to the receiver at position $\mathbf{p}_1^{(1)}$ on the LOS path and on the reflected paths. The path length of the reflected paths are equal to the length	91
Figure 4.3.	Factor graph representing the joint posterior distribution in (4.20) and the BP messages for a single time step k . Each box with light blue background indicates a different physical anchors (PAs). BP messages are shown as blue arrows. The time index k is omitted. The.....	99
Figure 4.4.	The considered floor plan for performance evaluation. Stars indicate the PAs and the black line indicates the agent trajectory. The black triangle is the position of the agent at the current time step $k = 1$. The crosses are the first-order VAs and their colors indicate the PAs they associate with.	110
Figure 4.5.	The magnitude $ \tilde{z}_{k,m}^{(j)} $ of synthetic data with 400MHz bandwidth normalized to $[0, 1]$ from PA (a) $j = 1$ and (b) $j = 2$. The red dashed lines are the delay between the agent and ground truth features (PAs plus VAs). VA positions are provided by ray tracing.	111

Figure 4.6.	Example of a single simulation run using synthetic data with 400MHz bandwidth at times (a) $k = 18$, (b) $k = 282$, (c) $k = 476$, and (d) $k = 655$. Blue and red indicate features associated with PA 1 and PA 2, respectively. Crosses indicate the true positions of features. Bullets indicate the position	114
Figure 4.7.	Performance of agent localization on synthetic data: (a) Agent position RMSEs averaged over 100 simulation runs and (b) empirical CDFs of the RMSEs with 300 MHz bandwidth. (c)\(d) and (e)\(f) are the counterparts of (a)\(b) for bandwidth 400 MHz and 600 MHz, respectively.	115
Figure 4.8.	Visualization of the distance between the agent position and the true and estimated feature position. The first column with (a)/(e) shows the true distances for PA 1 and PA 2, respectively. The following columns show the averaged distance between the true agent position and the particles of .	116
Figure 4.9.	Visualization of the range measurements (blue dots) provided by the sparse Bayesian learning estimator for (a) PA 1 and (b) PA 2 with bandwidth 300 MHz. (c)\(d) and (e)\(f) are the counterparts of (a)\(b) for bandwidth 400 MHz and 600 MHz, respectively.	117
Figure 4.10.	Performance of mapping on synthetic data with 400MHz bandwidth: GOSPA error of estimated VAs associated with PA 1 (a) and PA 2 (b), averaged over 100 simulation runs.	117
Figure 4.11.	True and estimated measurement noise variance $\eta_k^{(j)}$ averaged over 100 simulation runs for PA 1 (a) and PA 2 (b) at 400MHz bandwidth.	118
Figure 4.12.	Results using real data for $k = 102$. Blue and red indicate features associated with the two PAs. Crosses depict the positions of PAs. Bullets are particles of declared PF positions, and circles with black edges are estimated positions of declared PFs. The yellow line is the expected agent	119
Figure 4.13.	Performance of agent localization on real data: (a) Agent position RMSEs averaged over 30 simulation runs and (b) empirical CDFs of the RMSEs. .	120
Figure 4.14.	Performance of agent localization on synthetic data: The first row shows the agent position RMSEs averaged over 100 simulation runs with (a) 300 MHz, (b) 400 MHz, and (c) 600 MHz bandwidth, respectively. The second row shows the empirical CDFs of the RMSEs with (d) 300 MHz, (e)	128
Figure 4.15.	The magnitude spectrum $ \tilde{z}_{k,m}^{(j)} $ of real data with 400MHz bandwidth and normalized to $[0, 1]$ of PA (a) $j = 1$ and (b) $j = 2$	129

LIST OF TABLES

Table 2.1.	Performance results: nuScenes test set	48
Table 2.2.	Performance results: nuScenes validation set	48
Table 2.3.	GOSPA results: nuScenes validation set	49
Table 2.4.	Results for different detectors: nuScenes validation set	50
Table 2.5.	Ablation studies: nuScenes validation set	51
Table 2.6.	AMOTA results for different sigmoid temperatures: nuScenes validation set	52
Table 4.1.	Average runtime of different methods per time step	119
Table 4.2.	Average runtime of fast implementation per time step	127

ACKNOWLEDGEMENTS

First, I would like to express my deepest gratitude to my advisor Prof. Florian Meyer for his mentorship. He was always available to listen and offer support, creating an environment where I could fully immerse myself in my research. His patience and his constant encouragement gave me the confidence to overcome the hurdles throughout my PhD journey. His guidance has been invaluable.

I would also like to thank Prof. Bhaskar D. Rao, Prof. Nicholas A. Antipa, and Prof. Sicun Gao for being my doctoral committee. Their constructive feedback has significantly strengthened this dissertation. I am especially grateful to Prof. Rao; his lectures were a great source of inspiration and knowledge.

My PhD journey would not have been the same without the intellectual exchange with my labmates and collaborators, both past and present. I am grateful to Kuan-Lin Chen, Harinath Garudadri, Ish Jain, Junsu Jang, Thomas Kropfreiter, Erik Leitingner, Xuhong Li, Clair Ma, Khoa Nguyen, Luisa Watkins, Shaoxiu Wei, and Wenyu Zhang. Our insightful discussions have sharpened my ideas, challenged my assumptions, and made the entire process more enjoyable. A note of special thank is due to Ching-Hua Lee, who generously shared his experience and guidance during the early stages of my research.

Finally, this journey would not have been possible without the unwavering love and support from my parents, family and friends. Thank you for always believing in me.

The work presented in this thesis is supported in part by the University of California San Diego, National Science Foundation (NSF) under CAREER Award 2146261, the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001, and the Qualcomm Innovation Fellowship No. 492866.

Chapter 2, in part, is a reprint of the material as it appears in M. Liang and F. Meyer, “Neural enhanced belief propagation for multiobject tracking,” *IEEE Transactions on Signal Processing*, 2023 [1], and M. Liang and F. Meyer, “Neural enhanced belief propagation for data association in multiobject tracking,” in *International Conference on Information Fusion*

(*FUSION*), 2022 [2]. The dissertation author was the primary investigator and author of this material.

Chapter 3, in part, is a reprint of the material as it appears in M. Liang and F. Meyer, “Message passing for track-before-detect,” which was submitted to *IEEE Transactions on Signal Processing* in 2025, the preprint of which is available on arXiv [3], and M. Liang and F. Meyer, “An approach of directly tracking of multiple objects,” in *Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2024 [4]. The dissertation author was the primary investigator and author of this material.

Chapter 4, in part, is a reprint of the material as it appears in M. Liang, E. Leitingner, and F. Meyer, “Direct multipath-based SLAM,” *IEEE Transactions on Signal Processing*, 2025 [5], and M. Liang, E. Leitingner, and F. Meyer, “A belief propagation approach for direct multipath-based SLAM,” in *Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2023 [6]. The dissertation author was the primary investigator and author of this material.

VITA

2018	Bachelor of Engineering, The Chinese University of Hong Kong
2021	Master of Science, University of California San Diego
2025	Doctor of Philosophy, University of California San Diego

PUBLICATIONS

M. Liang, and F. Meyer, “Message passing for track-before-detect,” submitted to *IEEE Transactions on Signal Processing*, 2025.

S. Wei, **M. Liang**, and F. Meyer, “Bayesian multiobject tracking with neural-enhanced motion and measurement models,” submitted to *IEEE Transactions on Signal Processing*, 2025.

M. Liang, E. Leitingner, and F. Meyer, “Direct multipath-based SLAM,” *IEEE Transactions on Signal Processing*, vol. 73, pp.2336-2352, 2025.

S. Wei, **M. Liang**, B. D. Rao, and F. Meyer, “Bernoulli-Gaussian scale mixture model and BP method for multi-snapshot sparse signal recovery,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1-5, 2025.

M. Liang and F. Meyer, “An approach of directly tracking of multiple objects,” in *Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, pp. 995-999, 2024.

E. Leitingner, L. Wielandner, A. Venus, **M. Liang**, F. Meyer, and K. Witrisal “Multipath-based SLAM with cooperation and map fusion in MIMO systems,” in *Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, pp. 1316-1322, 2024.

S. Wei, **M. Liang**, and F. Meyer, “A new architecture for neural enhanced multiobject tracking,” in *International Conference on Information Fusion (FUSION)*, pp. 1-8, 2024.

M. Liang and F. Meyer, “Neural enhanced belief propagation for multiobject tracking,” *IEEE Transactions on Signal Processing*, vol. 72, pp. 15-30, 2023.

M. Liang, E. Leitingner, and F. Meyer, “A belief propagation approach for direct multipath-based SLAM,” in *Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, pp. 588-593, 2023.

M. Liang, Thomas Kropfreiter, and F. Meyer, “A BP method for track-before-detect,” *IEEE Signal Processing Letters*, vol. 30, pp. 1137-1141, 2023.

M. Liang and F. Meyer, “Neural enhanced belief propagation for data association in multiobject tracking,” in *International Conference on Information Fusion (FUSION)*, pp. 1-7, 2022 (**Best student paper award runner up**).

E. Leitingner, B. Teague, W. Zhang, **M. Liang**, and F. Meyer, “Data fusion for radio frequency SLAM with robust sampling,” in *International Conference on Information Fusion (FUSION)*, pp. 1-6, 2022

M. Liang and F. Meyer, “Neural enhanced belief propagation for cooperative localization,” in *IEEE Statistical Signal Processing Workshop (SSP)*, pp. 326-330, 2021.

ABSTRACT OF THE DISSERTATION

Scalable Inference in Probabilistic Graphical Models for Bayesian Machine Perception

by

Mingchao Liang

Doctor of Philosophy in Electrical Engineering
(Machine Learning and Data Science)

University of California San Diego, 2025

Professor Florian Meyer, Chair

Machine perception is essential for the functionality and safety of modern autonomous system, yet achieving accurate and robust estimations of the state of the environment from sensor data remains challenging. Core perception tasks like multi-object tracking (MOT) and simultaneous localization and mapping (SLAM) are frequently addressed using a two-stage paradigm, where sensor data is preprocessed into an intermediate representation before being used for inference. This paradigm, however, suffers from a critical weakness that the initial processing stage can discard valuable information, leading to suboptimal performance in challenging conditions.

This dissertation posits that a graph-based Bayesian framework provides a powerful

and flexible platform for overcoming these limitations. This framework leverages the explicit structure of factor graphs to pursue two powerful, complementary strategies: first, by designing more physically-grounded statistical models for direct inference on raw sensor data, and second, by seamlessly integrating data-driven learning components to correct and enhance simpler, hand-designed models.

We demonstrate the potential of this approach through three primary contributions. We first address the dominant detect-then-track (DTT) paradigm by introducing a hybrid method that mitigates model mismatch, using a graph neural network (GNN) to learn corrections for a BP algorithm. This approach significantly enhances tracking performance in complex scenes by learning to reject false alarms and improve data association, as demonstrated on a large-scale autonomous driving dataset. Next, to overcome the fundamental information loss of DTT, we propose a principled track-before-detect (TBD) method for MOT, built upon a comprehensive signal model that operates directly on raw sensor data. Finally, we extend this direct-processing philosophy to a new domain, introducing the first multipath-based SLAM algorithm that functions on raw radio signals without a separate, lossy channel estimation stage. This enables precise geometric mapping in challenging indoor environments by resolving multipath components (MPCs) that are unresolvable by traditional two-stage methods.

In each case, a factor graph is constructed and a scalable BP algorithm is developed to enable efficient inference. Through extensive validation on both synthetic and real data, our contributions collectively demonstrate that a graph-based approach leads to significant improvements in accuracy. This work provides a versatile foundation for the next generation of intelligent perception systems.

Chapter 1

Introduction

Machine perception is foundational to the capabilities of modern intelligent systems, providing the essential interface for perceiving, interpreting, and navigating the physical world. This fundamental capability underpins critical advancements across various domains, enabling autonomous vehicles to safely tackle complex traffic scenarios, facilitating the monitoring of intricate underwater acoustic environments for ocean science, and allowing wireless networks to sense and adapt to dynamic propagation conditions for optimized communication. In these safety-critical applications, the reliability of the entire system highly depends on the quality of its perception; a single failure can have catastrophic consequences for safety.

Achieving robust situational awareness in these demanding settings relies on solving complex perception tasks. Prominent tasks are multi-object tracking (MOT), which involves estimating the time-varying states of an unknown number of objects in the environment, and simultaneous localization and mapping (SLAM), which is particularly challenging in scenarios where signals propagate via multiple paths. Despite decades of progress, the gap between algorithm performance and the demands of real-world deployment remains significant. Consequently, developing machine perception algorithms that are simultaneously accurate, reliable, and computationally efficient remains a central and ongoing research challenge.

Key difficulties arise from dynamic nature of targets, complex environments, and the inherent limitations and diverse characteristics of sensing modalities. Addressing these chal-

lenges requires fundamental advancements in how sensor data is modeled and processed. This dissertation provides advancements by focusing on the design of more accurate and physically grounded statistical models to represent the underlying data-generating processes, coupled with the development of scalable and robust inference algorithms tailored to these models.

1.1 Bayesian Estimation and Graph-Based Processing

Effectively addressing aforementioned challenges in MOT and SLAM requires robust methods for inferring the state of the environment from sensor data. This inference task involves estimating unknown quantities, which we will collectively denote by a state vector $\mathbf{x}$, based on available observations typically represented by a data vector $\mathbf{z}$. Formally, this is the domain of parameter estimation [7], which seeks to establish a principled mapping or function that transforms the observed data $\mathbf{z}$ into an estimate of the underlying state $\mathbf{x}$. The choice of an estimation framework shapes how we model the problem.

1.1.1 The Bayesian Approach to Parameter Estimation

The field of parameter estimation is broadly divided into two principal philosophies: traditional (or frequentist) and Bayesian estimation, each having distinct historical roots and fundamental ideas regarding the nature of unknown parameters. Traditional parameter estimation, at its core, treats unknown parameters as fixed, deterministic constants. Consequently, inference in this framework relies primarily on the observed data to derive point estimates, making it difficult to make use of any prior knowledge.

In contrast, Bayesian estimation models the unknown parameters as random variables. This probabilistic formulation facilitates the incorporation of prior knowledge by specifying a prior probability density function (PDF) $f(\mathbf{x})$, which summarizes the information about the unknown parameter before any data is observed. When data becomes available, Bayesian inference uses Bayes' theorem to combine the prior PDF with the likelihood function $f(\mathbf{z}|\mathbf{x})$, which is a conditional PDF summarizing the knowledge provided by the data conditioned on

specific values of the parameter. The result is the posterior PDF $f(\mathbf{x}|\mathbf{z})$, which represents our updated knowledge about the parameter after observing the data. From this posterior distribution, specific point estimates can be derived by minimizing the expected value of a loss function. Common examples include the minimum mean-square error (MMSE) estimator, which computes the posterior mean, and the maximum a posterior error (MAP) estimator, which finds the posterior mode. In addition, the posterior PDF provides a comprehensive description of the estimate, inherently quantifying its uncertainty. For instance, in robot localization, a Bayesian estimator provides not just the most likely location of the robot, but also the uncertainty of that location estimate, which is critical for safe navigation and decision-making in autonomous systems. These characteristics make the Bayesian framework powerful for developing advanced sensing and perception methods.

1.1.2 Factor Graphs and Belief Propagation

Bayesian estimator requires the knowledge of the posterior distribution, which in principle can be obtained via Bayes' theorem,

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

However, for continuous random variables, the normalizing constant, also known as the partition function, $f(\mathbf{z}) = \int f(\mathbf{x})f(\mathbf{z}|\mathbf{x})d\mathbf{x}$ may not have closed-form analytical solutions. The dimensionality of the space and the complexity of the integral may prohibit numerical integration. To deal with the intractability of the posterior computation, various approximate inference techniques have been developed [8]. Prominent examples include variational inference, which frames the approximation as an optimization problem that minimizes the Kullback-Leibler (KL) divergence between the approximated distribution $q(\mathbf{x})$ and the true posterior $f(\mathbf{x}|\mathbf{z})$ [9]. Expectation propagation offers another iterative approach, which refines approximations to the factors of the posterior distribution by matching the moments of the true posterior [10].

Alongside these methods, especially when the probabilistic model exhibits a known conditional independence structure, graphical models provide a powerful framework for representing joint distributions. In particular, factor graphs offer bipartite undirected graphical representation for joint PDFs that encapsulate statistical dependencies among random variables. It consists of a set of variable nodes $\mathcal{Q}$, a set of factor nodes $\mathcal{F}$, and edges connecting them. A variable node $q \in \mathcal{Q}$ represents a random variable $\mathbf{x}_q$ and a factor node $s \in \mathcal{F}$ represents a factor $\psi_s(\mathbf{x}^{(s)})$. The joint PDF represented by the factor graph reads

$$f(\mathbf{x}|\mathbf{z}) \propto \prod_{s \in \mathcal{F}} \psi_s(\mathbf{x}^{(s)})$$

which is decomposed into a product of factors. This explicit factorization is not only a representational convenience, but also forms the foundation for belief propagation (BP), also known as the sum-product algorithm [11, 12, 13, 14]. BP is an efficient message-passing algorithm that operates on the factor graph to compute the beliefs $\tilde{f}(\mathbf{x}_q)$ for each variable $\mathbf{x}_q, q \in \mathcal{Q}$. When the factor graph is tree-structured, the beliefs are exact, i.e., $\tilde{f}(\mathbf{x}_q) = f(\mathbf{x}_q|\mathbf{z})$. For graphs with loops, BP is conducted iteratively, known as the loopy BP algorithm [15]. Loopy BP is not guaranteed to converge, and the resulting beliefs $\tilde{f}(\mathbf{x}_q)$ are approximations of the true posteriors. However, its empirical performance is often remarkably accurate and has been applied in many applications [16, 17, 18, 19]. While the messages themselves can be computationally intractable, they are often approximated effectively using parametric forms like Gaussians [20] or non-parametric representations like particles [21].

1.1.3 Sequential Bayesian Estimation

The principles of factor graphs and BP are not limited to static estimation. They are exceptionally well-suited for parameter estimation problems in dynamic systems where the state $\mathbf{x}_k$ evolves over time k , and measurements $\mathbf{z}_k$ arrive sequentially. While one could collect all measurements and then estimate the entire state history, known as smoothing, many applications

like navigation and control demand real-time operation. This necessitates filtering, where the goal is to estimate the current state $\mathbf{x}_k$ using only the measurements $\mathbf{z}_{1:k} = [\mathbf{z}_1^T \cdots \mathbf{z}_k^T]^T$ received up to the current time step k . In the Bayesian context, this boils down to computing the posterior $f(\mathbf{x}_k | \mathbf{z}_{1:k})$.

In such dynamic systems, the true state is typically assumed to be an unobserved Markov process. Given the immediately previous state $\mathbf{x}_{k-1}$, the current state $\mathbf{x}_k$ is conditionally independent of the other earlier states $\mathbf{x}_{k'}, k' < k - 1$, i.e., $f(\mathbf{x}_k | \mathbf{x}_{0:k-1}) = f(\mathbf{x}_k | \mathbf{x}_{k-1})$ where $\mathbf{x}_{0:k} = [\mathbf{x}_0^T \cdots \mathbf{x}_k^T]^T$, known as the state-transition model. Similarly, the measurement $\mathbf{z}_k$ is only dependent on the current state $\mathbf{x}_k$, i.e., $f(\mathbf{z}_k | \mathbf{x}_{0:k}) = f(\mathbf{z}_k | \mathbf{x}_k)$, known as the measurement model. This leads to a joint posterior distribution over the entire state trajectory $\mathbf{x}_{0:k}$ factorizes

$$f(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}) = f(\mathbf{x}_0) \prod_{k'=1}^k f(\mathbf{z}_{k'} | \mathbf{x}_{k'}) f(\mathbf{x}_{k'} | \mathbf{x}_{k'-1})$$

which can be perfectly represented by a factor graph with a simple chain structure. This graph-based perspective provides a unifying view of sequential estimation. The classic recursive Bayesian estimation (or Bayesian filtering) framework [22], which seeks to compute the posterior $f(\mathbf{x}_k | \mathbf{z}_{1:k})$ recursively, emerges as a specific message-passing schedule on this chain-structured graph. The famous “predict-update” recursion can be seen as running BP with messages passed only forward in time. When the state-transition and measurement models are linear-Gaussian, this recursion has an exact closed-form solution, known as Kalman filter (KF) [23]. For nonlinear or non-Gaussian models, the integrals become intractable, and approximation methods are required. Prominent examples include extended KF (EKF) [24], unscented KF (UKF) [25], which use Gaussian approximations, and the particle filter [26], which uses a non-parametric sample-based representation.

The true power of this graphical model perspective, however, is revealed when we move beyond this simple chain structure to address complex perception problems like MOT and SLAM. While these problems retain the principle of sequential estimation, their underlying statistical

models are more complicated and induce a more complex factorization. For instance, the measurement models can introduce cumbersome dependencies, such as measurements depending on multiple objects. Such dependency is difficult to capture with traditional recursive filters. On the other hand, the factor graph provides a unified and principled language to model the entire problem, including all temporal and cross-sectional dependencies. BP then serves as a general-purpose inference engine on this complete graph, enabling the development of scalable algorithms that can exploit the full model structure, far beyond the capabilities of classic filters. The specific design of these advanced graph-based methods in different perception problems will be detailed in the following chapters.

1.2 Multi-Object Tracking

Tracking is the estimation of the states of moving objects over time, inferring the value of interest from indirect, inaccurate and uncertain observations [27]. The state of an object usually includes motion-related parameter like its position and velocity, while observations are obtained from sensors such as LiDAR, radar or sonar. Contrary to the detection task that finds the objects for a single time step, tracking is about maintaining object identities and estimating their evolutions over time, using not only the observations at the current time but also past estimates. As a result, the trajectories of the objects can be used for downstream tasks such as trajectory prediction and path planning.

The sequential Bayesian estimation framework discussed Section 1.1.3 can be seen as an ideal blueprint for tracking problems. However, directly applying this blueprint to MOT is not straightforward. One fundamental challenge of MOT is that the number of objects is unknown and time-varying due to object birth and death. This necessitates robust track management to handle the appearance and disappearance of objects. Within the Bayesian framework, a common approach to address this is representing the existence of objects by a Bernoulli random variable [28, 29, 30, 31, 32]. By inferring the posterior probability of this variable, also known as

existence probability, the algorithm can make principled decisions about track initialization and termination.

The other challenge lies in formulating the measurement model, which describes the statistical relationship between the object states and the sensor observations. Depending on how the sensor data is preprocessed, the state-of-the-art tracking approaches can be divided into detect-then-track (DTT) and track-before-detect (TBD) algorithms.

1.2.1 Detect-Then-Track (DTT) Methods

The DTT paradigm addresses the MOT problem by adopting a two-stage approach. In the first stage, an object detector processes the raw sensor data to generate measurements, often referred to as “point measurements”. For instance, these can be bounding boxes from a LiDAR point cloud, or range-bearing estimates from a radar system. The second stage, the tracker, then operates exclusively on these “point measurements” to estimate and maintain object trajectories over time.

The major challenge of DTT-based methods is data association: the task of determining the origin of each measurement. Uncertainty arises because it is unclear which object generated which measurement; some measurements may not correspond to any real object at all, i.e., false alarms, and objects may not be detected, i.e., missed detections. In DTT-based MOT, a common assumption for the “point measurements” is that an object can generate at most one measurement and a measurement can originate from at most one object. This distinguishes from the extended object tracking [33], where a single large object generates multiple measurements, and cases of merged measurements [34], where a single measurement originates from multiple objects. A robust data association mechanism is essential for any DTT tracker. Note that if the association is known, MOT can be split into multiple single object tracking (SOT) problems and solved individually.

Early and straightforward approaches treat data association as a “hard” assignment problem. They seek a single association between measurements and objects at each time

step, often using greedy matching or the Hungarian algorithm [35]. The multi-hypothesis tracking (MHT) method [36] extends this idea to find the most probable assignments over a sliding time window and compute the MAP estimate. However, by committing to a single assignment, an early association error can lead to irrecoverable tracking failures. Furthermore, they separate the association and estimation steps, which is suboptimal from a Bayesian perspective.

To address these limitations, probabilistic data association is developed. Instead of making a hard decision, it considers all possible association hypotheses by modeling the data association event as a nuisance random variable, weighting each by its probability. Probabilistic data association was first applied in SOT [28, 37, 38] and was later extended to MOT [27, 29, 38], aiming to compute the MMSE state estimate by marginalizing over all association events. The primary drawback of this principled approach is its computational complexity, which grows exponentially with the number of objects and states. Consequently, much research has focused on developing efficient approximation techniques. Common strategies include using Murty’s algorithm [39] to consider only the k -best linear assignment for approximation [32, 40], or leveraging graphical model structures, as in the BP-based approach proposed in [41]. This latter line of work, which integrates data association into a unified Bayesian framework via message passing [18], has proven particularly powerful and forms a key point of reference for the methods developed in this dissertation.

In summary, the DTT paradigm offers a computationally pragmatic two-stage solution that has become the mainstream tracking approach. However, it comes at the cost of a fundamental limitation that valuable information may lose during the initial detection stage, leading to suboptimal performance.

1.2.2 Track-Before-Detect (TBD) Methods

TBD methods were developed to overcome the irreversible loss of information inherent in the DTT paradigm’s detection stage. The idea of TBD is to operate directly on the raw or minimally processed sensor data, before any detection decisions are made. Data is typically

structured as a grid of resolution cells (e.g., range-bearing bins) containing signal amplitude and phase. It preserves the full information content available for tracking, which is particularly advantageous for detecting weak or closely-spaced objects.

This fundamental shift in data processing strategy directly translates to a different formulation of the measurement model. In TBD, the explicit data association problem of DTT vanishes, and is replaced by a signal separation problem, i.e., the measurement model describes how multiple objects contribute to the raw sensor data.

The histogram probabilistic multi-hypothesis tracking (H-PMHT) [42, 43] quantizes the raw sensor data within each resolution cell, e.g., range-bearing cells in radar. The intensity value in each cell is then treated as if it were a count of independent pseudo-measurements originating from that cell. This effectively transforms the signal separation problem to a data association framework, where a single object can be associated with multiple pseudo-measurements from different cells.

A more physically-grounded superpositional signal model is introduced [44, 45]. In this framework, the value in each sensor data cell is modeled directly as the summation of signal contributions from all objects, plus an additive noise. Here, the measurement likelihood not only depends on the object states, but also their unknown amplitudes which represent their signal strength. This model avoids artificial quantization and captures the physics of wave-based sensors like radar and sonar. While this provides a more principled foundation, it also introduces significant computational challenges for inference and the complexity grows exponentially with the number of objects. To make the problem tractable, existing works typically make simplifying assumptions that constrain the model. Some approaches impose structural limitations, such as assuming that each data cell can be affected by at most one object [46, 47, 48]. Others simplify the problem by assuming key parameters, like the signal amplitudes, are known [49, 50]. A third, widely used strategy is to assume conditional independence between the data cells given the object states and amplitudes [51, 52, 53].

1.3 Multipath-Based SLAM

Alongside tracking dynamic objects, a cornerstone of autonomous systems is the ability to simultaneously build a map of the environment while determining the position of the map agent, a problem known as SLAM. The predominant line of research in SLAM has historically focused on sensors that perceive direct geometric features of the environment, such as cameras and LiDAR. These feature-based SLAM methods operate by identifying and matching landmarks (or anchors), like corners and edges, to build a map and triangulate the agent’s position. This feature-based SLAM paradigm has achieved remarkable success and is foundational to modern robotics and autonomous driving [54, 55].

A parallel and increasingly important line of research approaches SLAM not by observing direct geometric features, but by leveraging the rich information embedded in signal propagation itself, particularly through multipath components (MPCs). Many conventional radio-based localization techniques, e.g., global navigation satellite system (GNSS), rely on line-of-sight (LOS) ranging and treat non-line-of-sight (NLOS) signals as harmful interference. Consequently, GNSS is often ineffective in indoor or dense urban environments. Multipath-based SLAM, conversely, repurposes these reflections as valuable sources of geometric information

The core principle is that each NLOS path encodes the geometry of the reflecting surfaces it interacted with. By measuring properties of these paths, such as their delay and angle-of-arrival, one can infer the location and orientation of the reflectors. This concept was first established in multipath-assisted localization, where, given a known map of reflectors and signal anchors, the geometry of signal reflections is exploited for precise positioning [56]. The map consists of two types of features: physical anchors (PAs), which are signal transceivers at known locations, and virtual anchors (VAs), which are the geometric mirror images of the PAs with respect to the reflecting surfaces. An NLOS path from a PA can then be interpreted as a direct LOS path from its corresponding VA, with the path’s properties providing geometric constraints.

The multipath-based SLAM takes one step further that while the PAs locations are known, the location and visibility of VAs are unknown and must be estimated alongside the agent’s state. Therefore, multipath-based SLAM can be understood as a specific form of feature-based SLAM where the “features” to be mapped are these VAs, thereby simultaneously localizing the agent and mapping the reflective geometry of the environment. There are also methods that consider features beyond specular reflections, e.g., scatterers [57].

Multipath-based SLAM shares a lot of similarity with MOT. The VAs are functionally equivalent to objects in the MOT problem, i.e., their number is unknown and their existence (i.e., visibility to the agent) must be jointly inferred with their state. Reflecting this similarity, many existing methods adopt a DTT-like, two-stage approach: a channel estimator first extracts the MPCs from the raw signal, which are then fed to a tracker that performs data association and state estimation [58]. Even more integrated approaches, which handle data association and estimation jointly within a unified Bayesian framework [19], still operate on these pre-processed MPC estimates rather than the raw signal itself. Consequently, these methods inherit the potential for information loss characteristic of the DTT paradigm, creating a clear motivation for exploring alternative approaches.

1.4 Contributions and Outline

This dissertation addresses fundamental challenges in machine perception by developing unified and principled Bayesian methods for MOT and multipath-based SLAM. The central thesis of this work is that by first designing accurate, physically-grounded statistical models that capture the fine-grained conditional independence structure of a problem, then representing these models explicitly using factor graphs, and finally developing scalable message-passing algorithms for inference, it is possible to create perception systems that are simultaneously robust, accurate, and computationally efficient. The structure of this dissertation is introduced below.

Chapter 2 addresses the critical challenge of model mismatch in traditional DTT methods. These methods often rely on hand-crafted statistical models that often struggle to capture the true data generation process, leading to performance degradation due to model mismatch. To overcome this limitation, we introduce a novel hybrid inference framework for MOT that integrates traditional model-based BP with data-driven components learned by neural networks. Specifically, the proposed neural enhanced belief propagation (NEBP) method augments the factor graph by a graph neural network (GNN), which learns information from raw sensor data and corrects the inaccurate BP messages in the factor graph. This hybrid architecture retains the core benefits of the graphical model approach, such as scalability and principled uncertainty quantification, while harnessing the expressive power of neural networks to adapt to training data. Evaluated on nuScenes tracking challenge, a large autonomous driving dataset, NEBP demonstrates state-of-the-art performance. Furthermore, by achieving lower identity switches (IDS) and track fragments (Frag), the proposed method proves its superior robustness compared to non-BP-based approaches.

Chapter 3 shifts focus to the alternative paradigm of TBD-based MOT. Here, we propose a novel TBD method built upon a physically-grounded, comprehensive signal model that avoids the information loss inherent in DTT. Unlike prior TBD work that often relies on simplifying assumptions, our model explicitly accounts for sensor data correlations and unknown amplitude fluctuations, enabling a more faithful representation of the data generation process. This general-purpose model is not tied to a specific sensor but is grounded in the physics of wave propagation, making it broadly applicable to active and passive radar, active and passive sonar, as well as integrated sensing and communication systems. Based on a factor graph representation of the new measurement model, we develop a scalable BP method for efficient Bayesian inference. Experimental results, on both synthetic and real-world data, demonstrate that the proposed method outperforms state-of-the-art DTT methods, particularly in challenging scenarios with low signal-to-noise ratios or closely-spaced objects where the information loss of DTT is most detrimental.

Chapter 4 extends the physically-grounded, comprehensive signal model to multipath-based SLAM. Existing methods largely follow a DTT-like pipeline, relying on a channel estimator to extract MPCs before performing SLAM. This preprocessing step is a potential bottleneck and source of information loss. Chapter 4 introduces a direct multipath-based SLAM method that, for the first time, operates on raw radio signal without a separate channel estimation stage. A novel statistical model is developed that links the raw sensor data directly to the agent’s location, and the existence and location of VAs. Representing this model as a factor graph, we derive an efficient BP algorithm that performs SLAM. The resulting BP algorithm demonstrates significantly improved localization and mapping accuracy in both synthetic and real-data experiments, showcasing the power of the direct approach for robust geometric inference in challenging multipath environments.

Chapter 2

Neural Enhanced Belief Propagation for Multiobject Tracking

Algorithmic solutions for MOT are a key enabler for applications in autonomous navigation and applied ocean sciences. State-of-the-art MOT methods fully rely on a statistical model and typically use preprocessed sensor data as measurements. State-of-the-art Bayesian MOT methods that are based on BP systematically exploit graph structures of the statistical model to reduce computational complexity and improve scalability. However, as a fully model-based approach, BP can provide highly suboptimal estimates when there is a mismatch between the statistical model and the true data-generating process. Existing BP-based MOT methods can further only make use of preprocessed measurements. In this chapter, we introduce a variant of BP that combines model-based with data-driven MOT. The proposed NEBP method complements the statistical model of BP by information learned from raw sensor data. This approach conjectures that the learned information can reduce model mismatch and thus improve data association and false alarm rejection. Our NEBP method improves tracking performance compared to model-based methods. At the same time, it inherits the advantages of BP-based MOT, i.e., it scales only quadratically in the number of objects, and it can thus generate and maintain a large number of object tracks. We evaluate the performance of our NEBP approach for MOT on the nuScenes autonomous driving dataset and demonstrate that it can achieve state-of-the-art performance.

2.1 Chapter Overview

Multi-object tracking (MOT) [18, 30, 32, 40, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80] enables emerging applications including autonomous driving, applied ocean sciences, and indoor localization. MOT aims at estimating the states (e.g., positions and possibly other parameters) of moving objects over time, based on measurements provided by sensing technologies such as Light Detection and Ranging (LiDAR), radar, or sonar [18, 30, 32, 40, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80]. An inherent problem in MOT is measurement-origin uncertainty, i.e., the unknown association between measurements and objects. MOT is further complicated by the fact that the number of objects is unknown, i.e., for the initialization and termination of object tracks, track management schemes need to be employed.

2.1.1 Model-Based and Data-Driven MOT

Typically MOT methods rely on measurements that have been extracted from raw sensor data in a detection process. For example, an object detector [81, 82, 83, 84, 85, 86] can be applied to LiDAR scans or images at each time step independently, and the detected objects are then used as measurements for MOT [87, 88]. This common strategy is referred to as “detect-then-track”. Based on the assumption that, at each time step, an object can generate at most one measurement and each measurement can be originated by at most one object, data association can be cast as a bipartite matching problem.

The first class of MOT methods follows a global nearest neighbor association approach [59]. Here, a Hungarian [35] or a greedy matching algorithm is used to perform “hard” measurement-to-object associations [71, 72, 73, 74, 75, 76, 77, 78]. To improve the reliability of hard associations, these methods often rely on discriminative shape information of objects and measurements. Shape information is extracted from raw sensor data based on deep neural networks [77, 78] and used to compute pairwise distances between objects and measurements

more accurately. The methods in this class typically rely on heuristics for track management.

A second class of methods formulates and solves the MOT problem in the Bayesian estimation framework [18, 30, 32, 40, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]. Methods in this class rely on a statistical model for object birth, object motion, and measurement generation [18, 30, 32, 40, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]. The statistical model makes it possible to perform a more robust probabilistic “soft” data association and to avoid heuristics for track initialization and termination. Methods in this class include the joint probabilistic data association (JPDA) filter [59], the MHT [89], random finite sets (RFS) filters [32, 40, 65, 66, 67, 68], and BP-based MOT [18, 32, 60, 61, 62, 63, 64].

BP, also known as the sum-product algorithm, [12, 14, 90] provides an efficient and scalable solution to high-dimensional inference problems. BP operates by “passing messages” along the edges of the factor graph [12] that represents the statistical model of the inference problem. Important algorithms such as sequential Bayesian estimation [26] and the JPDA filter [59] are instances of BP. By exploiting the structure of the graph, BP-based MOT methods [18, 32, 60, 61, 62, 63, 64] are highly scalable. In particular, using BP, “soft” probabilistic data association can be performed for hundreds of objects. This makes it possible to generate and maintain a very large number of potential object tracks and, in turn, achieve state-of-the-art MOT performance [18, 32, 60, 61, 62, 63, 64].

Existing BP-based methods entirely rely on “hand-designed” statistical models. However, the statistical models are often unable to accurately represent all the intricate details of the true data-generating process. This mismatch leads to suboptimal object state estimates. In addition, since BP methods rely on the detect-then-track strategy, important object-related information might be discarded by the object detector. On the other hand, learning-based methods are fully data-driven, i.e., they do not make use of any statistical model. Typically, learning-based methods rely on deep neural networks, which facilitate the extraction of all relevant information from raw sensor data [77, 78]. However, learning-based MOT typically makes use of potentially unreliable heuristics for track management and only performs well in “big data” problems.

A GNN [91, 92] is a graphical model formed by neural networks. The neural network “passes messages”, i.e., exchange processing results, along the edges of the GNN. This mechanism is similar to the message passing performed by BP. It has been demonstrated that in Bayesian estimation problems, a GNN can outperform loopy BP [93] if sufficient training data is available. Recently, NEBP [94] was introduced. In NEBP, a GNN [91, 92] that matches the topology of the factor graph is established. After training the GNN, the GNN messages can complement the corresponding BP messages to correct errors introduced by cycles and model mismatch. The resulting hybrid message passing method combines the benefits of model-based and data-driven inference. In particular, NEBP can leverage the performance advantages of GNNs in big data problems. The benefits of NEBP have been demonstrated in decoding [94] and cooperative localization [95] problems.

2.1.2 Contribution and Chapter Organization

In this chapter, we address the fundamental question of how model-based and data-driven approaches can be combined in a hybrid inference method. In particular, we aim to develop a BP-based MOT method that augments its “hand-designed” statistical model with information learned from raw sensor data. In this context, we propose NEBP for MOT. Here, BP messages calculated for probabilistic data association are combined with the output of a GNN. The GNN uses object detections and features learned from raw sensor information as inputs. It can improve the MOT performance of BP by introducing data-driven *false alarm rejection* and *object shape association*.

False alarm rejection aims at identifying which measurements are likely false alarms. For measurements identified as a potential false alarm, the false alarm distribution in the statistical model used by BP is locally increased. This reduces the probability that the measurement is associated with an existing object track or initializes a new object track. *Object shape association* computes improved association probabilities by taking features of existing object tracks and measurements into account. The features are learned from raw sensor data. Compared to BP

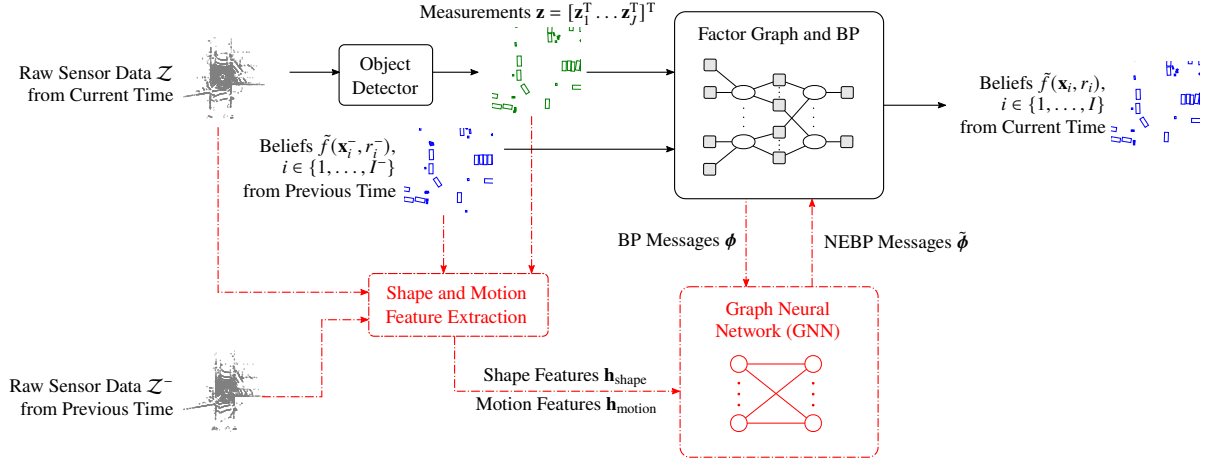


Figure 2.1. Flow diagram of one time step of conventional BP and the proposed NEBP method for MOT. Black boxes show the computation modules performed by both BP and NEBP. The red dotted boxes show additional modules only performed by the proposed NEBP method. The goal of both MOT methods is to obtain estimates $\hat{\mathbf{x}}_i$ and $\hat{r}_i$ of object state $\mathbf{x}_i$ and existence variable $r_i \in \{0, 1\}$ for all objects $i \in \{1, \dots, I\}$. First, raw sensor data $\mathcal{Z}$ is preprocessed by an object detector and the resulting object detection vector $\mathbf{z}$ consists of the measurements used for MOT. In addition to the measurements $\mathbf{z}$, approximate marginal posterior distributions (“beliefs”) $\tilde{f}(\mathbf{x}_i^-, r_i^-), i \in \{1, \dots, I\}$ that have been computed at the previous time step are used as input for MOT. The BP method reviewed in Section 2.2.1 performs operations on the factor graph discussed in Section 2.4 to compute updated beliefs $\tilde{f}(\mathbf{x}_i, r_i), i \in \{1, \dots, I\}$. Estimates $\hat{\mathbf{x}}_i$ and $\hat{r}_i$ can be computed from the beliefs $\tilde{f}(\mathbf{x}_i, r_i)$ as discussed in Section 2.3.4. Compared to conventional BP for MOT, the proposed NEBP approach introduces a shape and motion feature extraction module and a GNN, discussed in Section 2.5.1 and 2.5.2, respectively. The shape and motion feature extraction module computes shape and motion features $\mathbf{h}_{\text{shape}}$ and $\mathbf{h}_{\text{motion}}$ from raw sensor data $\mathcal{Z}^-$ and $\mathcal{Z}$ of the previous and current time. The GNN computes NEBP messages $\tilde{\phi}$ based on the conventional BP messages ϕ and the features $\mathbf{h}_{\text{shape}}$ and $\mathbf{h}_{\text{motion}}$, to obtain more accurate beliefs $\tilde{f}(\mathbf{x}_i, r_i), i \in \{1, \dots, I\}$ and thus more accurate estimates $\hat{\mathbf{x}}_i$ and $\hat{r}_i$ as discussed in Section 2.5.3.

for MOT, the resulting NEBP method for MOT can improve object declaration and estimation performance if annotated data is available and consequently provide state-of-the-art performance in big data MOT problems.

The key contributions of this chapter are summarized as follows.

- We introduce NEBP for MOT where probabilistic data association is enhanced by learned information provided by a GNN.
- We present the procedure and the loss function, used for the training of the GNN, that

enable false alarm rejection and object shape association.

- We apply the proposed method to an autonomous driving dataset and demonstrate state-of-the-art object tracking performance.

An overview of the proposed NEBP method for MOT is presented as a flow diagram in Fig. 2.1. Here, black boxes show the computation modules performed by both conventional BP and NEBP. The red dotted boxes show the additional modules only performed by the proposed NEBP method. A detailed description of each module will be provided in Sections 2.4 and 2.5.

In modern MOT scenarios with high-resolution sensors, it is often challenging to capture object shapes and the corresponding data-generating process by a statistical model. Thus, in contrast to the extended object tracking strategy [61, 96, 97], the influence of object shapes on data generation is best learned directly from data.

This chapter is organized as follows. Section 2.2 reviews the general BP and NEBP algorithm. Section 2.3 describes the system model and statistical formulation. Section 2.4 reviews the factor graph and the BP for MOT algorithm. Section 2.5 develops the proposed NEBP for MOT algorithm. Section 2.6 introduces the loss function used for NEBP training and Section 2.7 discusses experimental results.

2.2 Review of Belief Propagation (BP) and Neural Enhanced Belief Propagation (NEBP)

In what follows, we review the general framework of BP and NEBP.

2.2.1 Factor Graph and BP

A factor graph [12, 13] is a bipartite undirected graph $\mathcal{G}_f = (\mathcal{V}_f, \mathcal{E}_f)$ that consists of a set of edges $\mathcal{E}_f$ and a set of vertices or nodes $\mathcal{V}_f = \mathcal{Q} \cup \mathcal{F}$. A variable node $q \in \mathcal{Q}$ represents a random variable $\mathbf{x}_q$ and a factor node $s \in \mathcal{F}$ represents a factor $\psi_s(\mathbf{x}^{(s)})$. The argument $\mathbf{x}^{(s)}$ of a factor, consists of certain random variables $\mathbf{x}_q$ (each $\mathbf{x}_q$ can appear in several $\mathbf{x}^{(s)}$). Variable

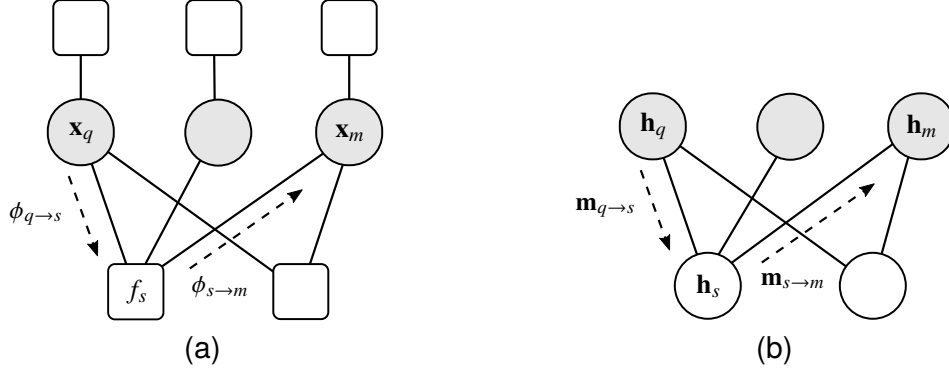


Figure 2.2. Factor graph (a) and its corresponding GNN (b) of a simple NEBP example.

nodes and factor nodes are typically depicted by circles and boxes, respectively. An example is shown in Fig. 2.2a. The joint PDF represented by the factor graph reads

$$f(\mathbf{x}) \propto \prod_{s \in \mathcal{F}} \psi_s(\mathbf{x}^{(s)})$$

where $\propto$ denotes equality up to a multiplicative constant.

BP [12], also known as the sum-product algorithm, can compute marginal PDFs $f(\mathbf{x}_q)$, $q \in Q$ efficiently. BP performs local operations on the factor graph. The local operations can be interpreted as “messages” that are passed over the edges of the graph. There are two types of messages.

At message passing iteration $\ell \in \{1, \dots, L\}$, the messages passed from variable nodes to factor nodes are defined as

$$\phi_{q \rightarrow s}^{(\ell)}(\mathbf{x}_q) = \prod_{a \in \mathcal{N}_{\mathcal{F}}(q) \setminus s} \phi_{a \rightarrow q}^{(\ell-1)}(\mathbf{x}_q). \quad (2.1)$$

In addition, the messages passed from factor nodes to variable nodes are given by

$$\phi_{s \rightarrow q}^{(\ell)}(\mathbf{x}_q) = \sum_{\mathbf{x}^{(s)} \setminus \mathbf{x}_q} \psi_s(\mathbf{x}^{(s)}) \prod_{m \in \mathcal{N}_Q(s) \setminus q} \phi_{m \rightarrow s}^{(\ell)}(\mathbf{x}_m) \quad (2.2)$$

where $\mathcal{N}_Q(\cdot) \subseteq \mathcal{Q}$ and $\mathcal{N}_{\mathcal{F}}(\cdot) \subseteq \mathcal{F}$ denote the set of neighboring variable and factor nodes, respectively. Here, $\sum_{\mathbf{x}^{(s)} \setminus \mathbf{x}_q}$ denotes marginalization of all elements in $\mathbf{x}^{(s)}$ except the element $\mathbf{x}_q$. If $\psi_s(\mathbf{x}^{(s)})$ is a singleton factor node in the sense that it is connected to a single variable node $\mathbf{x}_q$, i.e., $\mathbf{x}^{(s)} = \mathbf{x}_q$, then the message from the factor node to the variable node is equal to the factor node itself, i.e., $\phi_q(\mathbf{x}_q) \triangleq \phi_{s \rightarrow q}(\mathbf{x}_q) = \psi_s(\mathbf{x}_q)$. For future use, we introduce the joint set of messages $\boldsymbol{\phi}^{(\ell)} = \{\phi_q^{(\ell)}, \phi_{q \rightarrow s}^{(\ell)}, \phi_{s \rightarrow q}^{(\ell)}\}_{q \in \mathcal{Q}, s \in \mathcal{F}}$ and $\boldsymbol{\phi}_{\mathcal{F} \rightarrow \mathcal{Q}}^{(\ell)} = \{\phi_q^{(\ell)}, \phi_{s \rightarrow q}^{(\ell)}\}_{q \in \mathcal{Q}, s \in \mathcal{F}}$ as well as the function

$$\boldsymbol{\phi}^{(\ell)} = \text{BP}\left(\boldsymbol{\phi}_{\mathcal{F} \rightarrow \mathcal{Q}}^{(\ell-1)}\right)$$

that summarizes all message computations (2.1) and (2.2) related to one iteration ℓ .

After message passing is completed, one can subsequently obtain “beliefs” $\tilde{f}(\mathbf{x}_q)$, for each variable node $\mathbf{x}_q$, computed as the product of all incoming messages, i.e.,

$$\tilde{f}(\mathbf{x}_q) \propto \prod_{s \in \mathcal{N}_{\mathcal{F}}(q)} \phi_{s \rightarrow q}^{(L)}(\mathbf{x}_q).$$

If the factor graph is a tree, then the beliefs are exactly equal to the marginal PDF, i.e., $\tilde{f}(\mathbf{x}_q) = f(\mathbf{x}_q)$. In factor graphs with loops, BP is applied in an iterative manner and the message passing order is not unique. Different message passing orders may lead to different beliefs. The beliefs $\tilde{f}(\mathbf{x}_q)$ provided by this “loopy BP” scheme are only approximations of marginal posterior PDFs $f(\mathbf{x}_q)$. However, the beliefs $\tilde{f}(\mathbf{x}_q)$ have been observed to be very accurate in many applications[18, 98, 99].

2.2.2 NEBP

NEBP [94] is a hybrid message passing method that combines the benefits of model-based and data-driven inference. In particular, NEBP aims at improving the BP solution by augmenting the factor graph by a GNN. While BP messages are calculated based on the statistical model represented by the factor graph, GNN messages are computed based on information learned from data. In NEBP, a GNN that matches the network topology of the factor graph is introduced

as shown in Fig. 2.2. To avoid redundancies, singleton factor nodes are not represented by additional nodes in the GNN [94]. In particular, singleton factor nodes are already represented by GNN nodes introduced for variable nodes. An iterative message passing procedure is performed on the GNN. In one GNN iteration, nodes send messages to their neighboring nodes (cf. (2.3) and (2.4)), receive messages from neighboring nodes, and aggregate received messages to update their node embeddings (cf. (2.5) and (2.6)). In particular, at message passing iteration $\ell \in \{1, \dots, L\}$, the equations that describe message passing of the GNN are given as follows [94]. The messages exchanged between variable nodes $q \in \mathcal{Q}$ to factor nodes $s \in \mathcal{F}$ along the edges of the GNN are given by the vectors

$$\mathbf{m}_{q \rightarrow s}^{(\ell)} = g_{\mathcal{Q} \rightarrow \mathcal{F}}(\mathbf{h}_q^{(\ell-1)}, \mathbf{h}_s^{(\ell-1)}, \phi_{q \rightarrow s}^{(\ell)}) \quad (2.3)$$

$$\mathbf{m}_{s \rightarrow q}^{(\ell)} = g_{\mathcal{F} \rightarrow \mathcal{Q}}(\mathbf{h}_s^{(\ell-1)}, \mathbf{h}_q^{(\ell-1)}, \phi_{s \rightarrow q}^{(\ell)}) \quad (2.4)$$

where $\phi_{q \rightarrow s}^{(\ell)}$ and $\phi_{s \rightarrow q}^{(\ell)}$ are the BP messages (cf. (2.1)-(2.2)) and $g_{\mathcal{Q} \rightarrow \mathcal{F}}(\cdot)$ as well as $g_{\mathcal{F} \rightarrow \mathcal{Q}}(\cdot)$ are referred to as edge functions [94]. In addition, after GNN messages have been exchanged, so-called node embedding vectors $\mathbf{h}_s^{(\ell)}$ and $\mathbf{h}_q^{(\ell)}$ for factor node $s \in \mathcal{F}$ and variable node $q \in \mathcal{Q}$, are computed as

$$\mathbf{h}_s^{(\ell)} = g_{\mathcal{F}}\left(\mathbf{h}_s^{(\ell-1)}, \sum_{q \in \mathcal{N}_{\mathcal{Q}}(s)} \mathbf{m}_{q \rightarrow s}^{(\ell)}\right) \quad (2.5)$$

$$\mathbf{h}_q^{(\ell)} = g_{\mathcal{Q}}\left(\mathbf{h}_q^{(\ell-1)}, \sum_{s \in \mathcal{N}_{\mathcal{F}}(q)} \mathbf{m}_{s \rightarrow q}^{(\ell)}, \phi_q^{(\ell)}\right) \quad (2.6)$$

where $\phi_q^{(\ell)}$ are the singleton BP messages discussed in Sec. 2.2.1 and $g_{\mathcal{F}}(\cdot)$ as well as $g_{\mathcal{Q}}(\cdot)$ are referred to as node functions [94]. Edge and node functions are neural networks. For future use, we introduce the joint set of GNN messages $\mathbf{m}^{(\ell)} = \{\mathbf{m}_{s \rightarrow q}^{(\ell)}, \mathbf{m}_{q \rightarrow s}^{(\ell)}\}_{q \in \mathcal{Q}, s \in \mathcal{F}}$, node embeddings $\mathbf{h}^{(\ell)} = \{\mathbf{h}_s^{(\ell)}, \mathbf{h}_q^{(\ell)}\}_{q \in \mathcal{Q}, s \in \mathcal{F}}$, and the function $\{\mathbf{h}^{(\ell)}, \mathbf{m}^{(\ell)}\} = \text{GNN}(\mathbf{h}^{(\ell-1)}, \boldsymbol{\phi}^{(\ell)})$ that summarizes all GNN computations (2.3)–(2.6) at iteration ℓ . Note that singleton factor nodes are not included

in the GNN.

Based on the BP message passing procedure discussed in Section 2.2.1 and the GNN message passing procedure discussed above, the hybrid NEBP method can be summarized as follows. In particular, at iteration ℓ

$$\begin{aligned}\boldsymbol{\phi}^{(\ell)} &= \text{BP}\left(\tilde{\boldsymbol{\phi}}_{\mathcal{F} \rightarrow \mathcal{Q}}^{(\ell-1)}\right) \\ \{\mathbf{h}^{(\ell)}, \mathbf{m}^{(\ell)}\} &= \text{GNN}\left(\mathbf{h}^{(\ell-1)}, \boldsymbol{\phi}^{(\ell)}\right)\end{aligned}$$

where $\tilde{\boldsymbol{\phi}}_{\mathcal{F} \rightarrow \mathcal{Q}}^{(\ell-1)} = \{\tilde{\phi}_{s \rightarrow q}^{(\ell-1)}\}_{q \in \mathcal{Q}, s \in \mathcal{F}}$ is the set of NEBP messages from the last iteration ℓ that are passed from factor nodes to variables nodes. In the MOT context, the BP messages $\boldsymbol{\phi}_{\mathcal{F} \rightarrow \mathcal{Q}}^{(\ell-1)}$ provide a preliminary data association solution, which does not make use of the object shape information, to the GNN. The GNN then aims at refining this preliminary solution by also taking the object shape information into account. Providing a preliminary solution to the GNN can make training and inference more efficient and accurate [94].

Finally, the NEBP messages at the current iteration, are calculated as

$$\tilde{\phi}_{s \rightarrow q}^{(\ell)} = g_{\text{nebp},1}\left(\mathbf{h}^{(\ell)}, \mathbf{m}^{(\ell)}\right) \odot \phi_{s \rightarrow q}^{(\ell)} + g_{\text{nebp},2}\left(\mathbf{h}^{(\ell)}, \mathbf{m}^{(\ell)}\right) \quad (2.7)$$

where $g_{\text{nebp},1}(\mathbf{h}^{(\ell)}, \mathbf{m}^{(\ell)})$ and $g_{\text{nebp},2}(\mathbf{h}^{(\ell)}, \mathbf{m}^{(\ell)})$ are neural networks that, in general, output a nonnegative vector with the same dimension as $\phi_{s \rightarrow q}^{(\ell)}$ and $\odot$ is element-wise multiplication. The BP messages passed from variable nodes to factor nodes are not “neural enhanced”.

After the last message passing iteration $\ell = L$, the beliefs for each variable node $\mathbf{x}_q$ are calculated as

$$\tilde{f}(\mathbf{x}_q) = \prod_{s \in \mathcal{N}_{\mathcal{F}}(q)} \tilde{\phi}_{s \rightarrow q}^{(L)}(\mathbf{x}_q).$$

2.3 System Model and Statistical Formulation

In this section, we review the system model of BP-based MOT and the multiobject declaration and state estimation problem BP-based MOT aims to solve.

2.3.1 Potential Objects and Object States

The number of objects is unknown and time-varying. We describe this scenario by introducing N_k potential objects (POs) [18, 60] where N_k is the maximum possible number of objects¹. At time k , the existence of a PO $n \in \{1, \dots, N_k\}$ is modeled by a binary random variable $r_{k,n} \in \{0, 1\}$. PO n exists, in the sense that it represents an actual object, if and only if $r_{k,n} = 1$. The kinematic state of PO n is modeled by a random vector $\mathbf{x}_{k,n}$ that consists of the object's position and possibly motion information. The augmented PO state is defined as $\mathbf{y}_{k,n} \triangleq [\mathbf{x}_{k,n}^T r_{k,n}]^T$. In what follows, we will refer to augmented PO states simply as PO states. We also introduce the joint PO state vector at time k as $\mathbf{y}_k \triangleq [\mathbf{y}_{k,1}^T \cdots \mathbf{y}_{k,N_k}^T]^T$.

At time k , a detector $g_{\text{det}}(\cdot)$ produces a vector $\mathbf{z}_k \triangleq [\mathbf{z}_{k,1}^T \cdots \mathbf{z}_{k,J_k}^T]^T$ of preprocessed measurements from raw sensor data $\mathcal{Z}_k$, i.e., $\mathbf{z}_k = g_{\text{det}}(\mathcal{Z}_k)$. The joint measurement vector that consists of all preprocessed measurements up to time k is denoted as $\mathbf{z}_{1:k} \triangleq [\mathbf{z}_1^T \cdots \mathbf{z}_k^T]^T$.

There are two types of POs:

- *New POs* represent objects that, for the first time, have generated a measurement at the current time step k . Their states are denoted as $\bar{\mathbf{y}}_{k,j} = [\bar{\mathbf{x}}_{k,j}^T \bar{r}_{k,j}]^T$. At time k , a new PO is introduced² for each measurement $\mathbf{z}_{k,j}$, $j \in \{1, \dots, J_k\}$.
- *Legacy POs* represent objects that already have generated at least one measurement at previous time steps $k' < k$. Their states are denoted by $\underline{\mathbf{y}}_{k,i} = [\underline{\mathbf{x}}_{k,i}^T \underline{r}_{k,i}]^T$, $i \in \{1, \dots, I_k\}$, where I_k is the total number of legacy POs.

¹The number of POs N_k is the maximum possible number of actual objects that have produced a measurement so far [18].

²Introducing a new PO is equal to initializing a new potential object track [18].

All new POs that have been introduced at time $k - 1$ become legacy POs at time k . Thus, the number of legacy POs at time k is $I_k = I_{k-1} + J_{k-1} = N_{k-1}$ and the total number of POs is $N_k = I_k + J_k$. A pruning step that limits the growth of the number of PO states will be discussed in Section 2.3.4. For future reference, we further define the joint PO states $\bar{\mathbf{y}}_k \triangleq [\bar{\mathbf{y}}_{k,1}^T \cdots \bar{\mathbf{y}}_{k,J_k}^T]^T$, $\underline{\mathbf{y}}_k \triangleq [\underline{\mathbf{y}}_{k,1}^T \cdots \underline{\mathbf{y}}_{k,I_k}^T]^T$, and $\mathbf{y}_k \triangleq [\underline{\mathbf{y}}_k^T \bar{\mathbf{y}}_k^T]^T$.

POs represent actual objects that already have generated at least one measurement. In addition, there may also be actual objects that have not generated any measurements yet. These objects are referred to as “unknown” objects. Unknown objects are independent and identically distributed according to $f_u(\cdot)$. The number of unknown objects is modeled by a Poisson distribution with mean μ_u . The statistical model for unknown objects induces a statistical model for new POs [18] as further discussed in Section 2.4.

2.3.2 Data Association Vector and Measurement Model

MOT is subject to measurement origin uncertainty, i.e., it is unknown which actual object generates which measurement $\mathbf{z}_{k,j}$. It is also possible that a measurement is not originated from any actual object. Such a measurement is referred to as a false alarm. Furthermore, an actual object may also not generate any measurements. This is referred to as missed detection. We assume that an object can generate at most one measurement and a measurement can originate from at most one object; this is known as the “data association assumption.”

Since every actual object that has generated a measurement is represented by a PO, we can model measurement origin uncertainty by PO-to-measurement associations. These associations are represented by multinoulli random variables. In particular, the PO-to-measurement association at time k can be described by the “object-oriented” data association (DA) vector $\mathbf{a}_k = [a_{k,1} \cdots a_{k,I_k}]^T$. The case where legacy PO i generates measurement j at time k , is represented by $a_{k,i} = j \in \{1, \dots, J_k\}$. On the other hand, the case where legacy PO i does not generate any measurement at time k is represented by $a_{k,i} = 0$.

The computation complexity of MOT can be reduced by also introducing the “measure-

ment-oriented" DA vector [100, 101] $\mathbf{b}_k = [b_{k,1} \cdots b_{k,J_k}]^T$. Here, $b_{k,j} = i \in \{1, \dots, I_k\}$ represents the case where measurement j is originated by legacy PO i . In addition, $b_{k,j} = 0$ represents the case where measurement j is not originated by any legacy PO. Modeling PO-to-measurement associations in terms of both $\mathbf{a}_k$ and $\mathbf{b}_k$ is redundant in that $\mathbf{b}_k$ can be determined from $\mathbf{a}_k$ and vice versa. However, the resulting hybrid representation makes it possible to check the consistency of the data association assumption based on indicators that are only a function of two scalar association variables. In particular, we introduce the indicator function $\Psi_{i,j}(a_{k,i}, b_{k,j})$ [100, 101] which is equal to 0 if $a_{k,i} = j, b_{k,j} \neq i$ or $a_{k,i} \neq j, b_{k,j} = i$, and is equal to 1 otherwise. If and only if a data association event can be expressed by both an object-oriented $\mathbf{a}_k$ and a measurement-oriented association vector $\mathbf{b}_k$, then the event does not violate the data association assumption and all indicator functions are equal to one. Finally, we also introduce the joint DA vectors $\mathbf{a}_{1:k} \triangleq [\mathbf{a}_1^T \cdots \mathbf{a}_k^T]^T$ and $\mathbf{b}_{1:k} \triangleq [\mathbf{b}_1^T \cdots \mathbf{b}_k^T]^T$. It is assumed that an actual object generates a measurement with probability of detection p_d . If and only if a PO n represents an actual object, i.e., $r_{k,n} = 1$, it can generate a measurement. If measurement $\mathbf{z}_{k,j}$ has been generated by (legacy or new) PO $n \in \{1, \dots, N_k\}$, its conditional PDF given PO state $\mathbf{x}_{k,n}$ is $f(\mathbf{z}_{k,j}|\mathbf{x}_{k,n})$. The functional form of $f(\mathbf{z}_{k,j}|\mathbf{x}_{k,n})$ is determined by the statistical measurement model chosen for the considered tracking problem. For example, if we have a linear measurement with respect to PO state $\mathbf{x}_{k,n}$ with zero-mean, additive Gaussian noise, i.e. $\mathbf{z}_{k,j} = \mathbf{H}_k \mathbf{x}_{k,n} + \mathbf{v}_{k,j}$, then $f(\mathbf{z}_{k,j}|\mathbf{x}_{k,n})$ is given by $\mathcal{N}(\mathbf{H}_k \mathbf{x}_{k,n}, \mathbf{R}_k)$, where $\mathbf{R}_k$ is the covariance of the measurement noise $\mathbf{v}_{k,j}$.

If measurement $\mathbf{z}_{k,j}$ has not been generated any PO, it is a false alarm measurement. False alarm measurements are independent and identically distributed according to $f_{\text{fa}}(\mathbf{z}_{k,j})$. The number of false alarm measurements is modeled by a Poisson distribution with mean μ_{fa} .

2.3.3 Object Dynamics

The PO states $\mathbf{y}_{k-1,i}$ are assumed to evolve independently and identically according to a Markovian dynamic model [59]. In addition, for each PO at time $k - 1$, there is a legacy PO

at time k . The state transition function of the joint PO state $\mathbf{y}_{k-1}$ at time $k-1$, can thus be expressed as

$$f(\underline{\mathbf{y}}_k | \mathbf{y}_{k-1}) = \prod_{i=1}^{N_{k-1}} f(\underline{\mathbf{y}}_{k,i} | \mathbf{y}_{k-1,i})$$

where the state-transition PDF $f(\underline{\mathbf{y}}_{k,i} | \mathbf{y}_{k-1,i}) = f(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i} | \mathbf{x}_{k-1,i}, r_{k-1,i})$ models the dynamics of individual POs and is given as follows. If PO i does not exist at time $k-1$, i.e., $r_{k-1,i} = 0$, then it cannot exist at time k either. The state-transition PDF for $r_{k-1,i} = 0$ is thus given by

$$f(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i} | \mathbf{x}_{k-1,i}, 0) = \begin{cases} f_{\text{du}}(\underline{\mathbf{x}}_{k,i}), & \underline{r}_{k,i} = 0 \\ 0, & \underline{r}_{k,i} = 1 \end{cases}$$

where $f_{\text{du}}(\underline{\mathbf{x}}_{k,i})$ is an arbitrary “dummy” PDF since states of nonexisting POs are irrelevant. If PO i exists at time $k-1$, i.e., $r_{k-1,i} = 1$, then, the probability that it stills exists at time k is given by the survival probability p_s . If PO i still exists at time k , its state $\underline{\mathbf{x}}_{k,i}$ is distributed according to $f(\underline{\mathbf{x}}_{k,i} | \mathbf{x}_{k-1,i})$. The state-transition PDF for $r_{k-1,i} = 1$, is thus given by

$$f(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i} | \mathbf{x}_{k-1,i}, 1) = \begin{cases} (1 - p_s) f_{\text{du}}(\underline{\mathbf{x}}_{k,i}), & \underline{r}_{k,i} = 0 \\ p_s f(\underline{\mathbf{x}}_{k,i} | \mathbf{x}_{k-1,i}), & \underline{r}_{k,i} = 1. \end{cases}$$

2.3.4 Declaration, Estimation, Initialization, and Termination

At each time step k , our goal is to declare whether a PO $n \in \{1, \dots, N_k\}$ exists and to estimate the PO states $\mathbf{x}_{k,n}$ of all existing POs, based on all measurements $\mathbf{z}_{1:k} = [\mathbf{z}_1^T \cdots \mathbf{z}_k^T]^T$. In the Bayesian setting, object declaration and state estimation essentially amount to, respectively, calculating the marginal posterior existence probabilities $p(r_{k,n} = 1 | \mathbf{z}_{1:k})$ and the marginal posterior state PDFs $f(\mathbf{x}_{k,n} | r_{k,n} = 1, \mathbf{z}_{1:k})$. Then, a PO n is declared to exist if $p(r_{k,n} = 1 | \mathbf{z}_{1:k})$ is larger than a suitably chosen threshold T_{dec} [102, Ch. 2]. Furthermore, for each declared PO

n , an estimate of $\mathbf{x}_{k,n}$ is provided by the MMSE estimator [102, Ch. 4]

$$\hat{\mathbf{x}}_{k,n} = \int \mathbf{x}_{k,n} f(\mathbf{x}_{k,n} | r_{k,n} = 1, \mathbf{z}_{1:k}) d\mathbf{x}_{k,n}.$$

Both $p(r_{k,n} = 1 | \mathbf{z}_{1:k})$ and $f(\mathbf{x}_{k,n} | r_{k,n} = 1, \mathbf{z}_{1:k})$ can be obtained from the marginal posterior PDFs of augmented states $f(\mathbf{y}_{k,n} | \mathbf{z}_{1:k}) = f(\mathbf{x}_{k,n}, r_{k,n} | \mathbf{z}_{1:k})$. Thus, the problem to be solved is finding an efficient computation of $f(\mathbf{y}_{k,n} | \mathbf{z}_{1:k})$. For future reference, we introduce the notation $\hat{r}_{k,n} = p(r_{k,n} = 1 | \mathbf{z}_{1:k})$.

Track initialization and termination can be summarized as follows. We initialize a new potential object track for each measurement. The initial existence probability of each potential object track is determined by the statistical model for unknown objects discussed in Section 2.3.2. With this initialization approach, the number of object tracks grows linearly with time k . Therefore, we terminate (“prune”) potential object tracks by removing legacy and new POs with existence probabilities below a threshold T_{pru} from the state space.

2.4 Conventional BP-based MOT Algorithm

In this section, we review the BP-based MOT approach. Contrary to the original BP-based MOT approach, we introduce an alternative factor graph which makes it easier to describe the proposed NEBP method presented in Section 2.5.

2.4.1 Alternative MOT Factor Graph

By using common assumptions, the factorization structure of the joint posterior PDF $f(\mathbf{y}_{0:k}, \mathbf{a}_{1:k}, \mathbf{b}_{1:k} | \mathbf{z}_{1:k})$ is given as follows (cf. [18, Sec. VIII-G])

$$\begin{aligned}
f(\mathbf{y}_{0:k}, \mathbf{a}_{1:k}, \mathbf{b}_{1:k} | \mathbf{z}_{1:k}) &\propto \left(\prod_{n'=1}^{N_0} f(\mathbf{y}_{0,n'}) \right) \prod_{k'=1}^k \left(\prod_{n=1}^{N_{k'-1}} f(\underline{\mathbf{y}}_{k',n} | \mathbf{y}_{k'-1,n}) \right) \\
&\times \left(\prod_{i=1}^{I_{k'}} q(\underline{\mathbf{y}}_{k',i}, a_{k',i}; \mathbf{z}_{k'}) \prod_{j'=1}^{J_{k'}} \Psi_{i,j'}(a_{k',i}, b_{k',j'}) \right) \\
&\times \prod_{j=1}^{J_{k'}} v(\bar{\mathbf{y}}_{k',j}, b_{k',j}; \mathbf{z}_{k',j}). \tag{2.8}
\end{aligned}$$

Note that often there are no POs at time $k = 0$, i.e., $N_0 = 0$.

The factor $q(\underline{\mathbf{y}}_{k,i}, a_{k,i}; \mathbf{z}_k) \triangleq q(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i}, a_{k,i}; \mathbf{z}_k)$, describing the measurement model of the sensor for legacy POs, is defined as

$$\begin{aligned}
q(\underline{\mathbf{x}}_{k,i}, 1, a_{k,i}; \mathbf{z}_k) &\triangleq \begin{cases} \frac{p_d f(\mathbf{z}_{k,j} | \underline{\mathbf{x}}_{k,i})}{\mu_{fa} f_{fa}(\mathbf{z}_{k,j})}, & a_{k,i} = j \in \{1, \dots, J_k\} \\ 1 - p_d, & a_{k,i} = 0 \end{cases} \\
q(\underline{\mathbf{x}}_{k,i}, 0, a_{k,i}; \mathbf{z}_k) &\triangleq 1(a_{k,i}) \tag{2.9}
\end{aligned}$$

where $1(a) \in \{0, 1\}$ is the indicator function of the event $a = 0$, i.e., $1(a) = 1$ if $a = 0$ and 0 otherwise.

The factor $v(\bar{\mathbf{y}}_{k,j}, b_{k,j}; \mathbf{z}_{k,j}) \triangleq v(\bar{\mathbf{x}}_{k,j}, \bar{r}_{k,j}, b_{k,j}; \mathbf{z}_{k,j})$, describing the measurement model of the sensor as well as prior information for new POs, is given by

$$\begin{aligned}
v(\bar{\mathbf{x}}_{k,j}, 1, b_{k,j}; \mathbf{z}_{k,j}) &\triangleq \begin{cases} 0, & b_{k,j} = i \in \{1, \dots, I_k\} \\ \frac{p_d \mu_u f_u(\bar{\mathbf{x}}_{k,j}) f(\mathbf{z}_{k,j} | \bar{\mathbf{x}}_{k,j})}{\mu_{fa} f_{fa}(\mathbf{z}_{k,j})}, & b_{k,j} = 0 \end{cases} \\
v(\bar{\mathbf{x}}_{k,j}, 0, b_{k,j}; \mathbf{z}_{k,j}) &\triangleq f_{du}(\bar{\mathbf{x}}_{k,j}) \tag{2.10}
\end{aligned}$$

where $f_{du}(\bar{\mathbf{x}}_{k,j})$ is an arbitrary “dummy” PDF. Here, the distribution $f_u(\bar{\mathbf{x}}_{k,j})$ and mean number μ_u of unknown objects are used as prior information for new POs. Note that a detailed derivation

of the factors $q(\underline{\mathbf{y}}_{k,i}, a_{k,i}; \mathbf{z}_k)$ and $v(\bar{\mathbf{y}}_{k,j}, b_{k,j}; \mathbf{z}_{k,j})$ is provided in [18].

The factorization in (2.8) provides the basis for a factor graph representation. Contrary to [18], in this work, we consider an alternative factor graph where PO states and association variables are combined in joint variable nodes. In particular, legacy PO states $\underline{\mathbf{y}}_{k,i}$ and object-oriented association variables $a_{k,i}$ form joint nodes “ $\underline{\mathbf{y}}_{k,i}, a_{k,i}$ ”. In addition, new PO states $\bar{\mathbf{y}}_{k,j}$ and measurement-oriented association variables $b_{k,j}$ form joint nodes “ $\bar{\mathbf{y}}_{k,j}, b_{k,j}$ ”. This combination of variable nodes is motivated by the fact that in the original factor graph there is exactly one $a_{k,i}$ connected to the corresponding $\underline{\mathbf{y}}_{k,i}$ and exactly one $b_{k,j}$ connected to the corresponding $\bar{\mathbf{y}}_{k,j}$. The resulting alternative factor graph leads to a presentation of the proposed method in Section 2.5 that is consistent with BP message passing rules [12] as well as the original work on NEBP [94]. A single time step of the considered factor graph is shown in Fig. 2.3a.

Next, BP is applied to efficiently compute the beliefs $\tilde{f}(\mathbf{y}_{k,n})$ that approximate the marginal posterior PDFs $f(\mathbf{y}_{k,n} | \mathbf{z}_{1:k})$. Since the considered factor graph in Fig. 2.3a has loops, a specific message-passing order has to be chosen. As in [18, 60], we choose an order that is based on the following rules: (i) BP messages are only sent forward in time, and (ii) iterative message passing is only performed for data association and at each time step individually.

2.4.2 BP for MOT

In what follows, we will briefly discuss the calculation of BP messages on the considered factor graph shown in Fig. 2.3a. Note that messages sent from the singleton factor nodes “ $q(\underline{\mathbf{y}}_{k,i}, a_{k,i}; \mathbf{z}_k)$ ” to the joint variable nodes “ $\underline{\mathbf{y}}_{k,i}, a_{k,i}$ ”, and messages sent from the singleton factor nodes “ $v(\bar{\mathbf{y}}_{k,j}, b_{k,j}; \mathbf{z}_{k,j})$ ” to the joint variable nodes “ $\bar{\mathbf{y}}_{k,j}, b_{k,j}$ ” are equal to the singleton factor nodes “ $q(\underline{\mathbf{y}}_{k,i}, a_{k,i}; \mathbf{z}_k)$ ” and “ $v(\bar{\mathbf{y}}_{k,j}, b_{k,j}; \mathbf{z}_{k,j})$ ” themselves. Thus, we reuse the same notation for factor nodes and corresponding messages.

The messages $\alpha_{k,i}(\underline{\mathbf{y}}_{k,i}) = \alpha_{k,i}(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i})$, $i \in \{1, \dots, I_k\}$ passed from the factor nodes

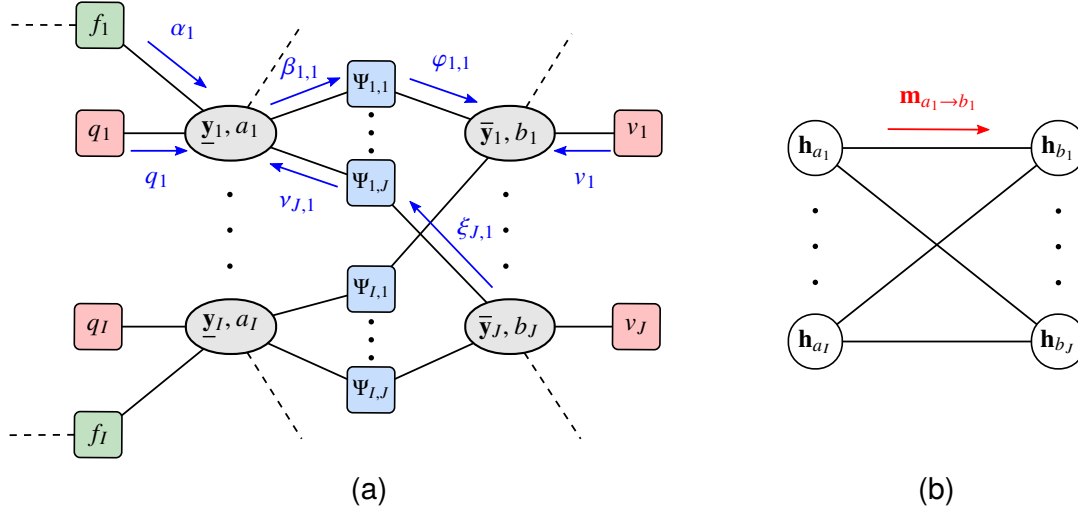


Figure 2.3. Factor graph (a) and corresponding bipartite GNN (b) for a single time step k of the considered NEBP approach for MOT. BP and GNN messages are shown. The time index k is omitted. A GNN node was introduced for each legacy PO and each new PO. The topology of the GNN, which only matches the part of the factor graph that models the data generating process, will be discussed in Section 2.5.2. Following the topology of the data association part of the factor graph in (a), GNN edges were introduced such that the bipartite GNN shown in (b) is obtained. The following shorthand notation is used: $\underline{y}_i = \underline{y}_{k,i}$, $\bar{y}_j = \bar{y}_{k,j}$, $a_i = a_{k,i}$, $b_j = b_{k,j}$, $\Psi_{i,j} = \Psi_{i,j}(a_{k,i}, b_{k,j})$, $q_i = q(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i}, a_{k,i}; \mathbf{z}_k)$, $v_j = v(\bar{\mathbf{x}}_{k,j}, \bar{r}_{k,j}, b_{k,j}; \mathbf{z}_k)$, $f_i = f(\underline{\mathbf{y}}_{k,i} | \mathbf{y}_{k-1,i})$, $\alpha_i = \alpha_{k,i}(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i})$, $\beta_{i,j} = \beta_{k,i,j}^{(\ell)}(\underline{\mathbf{y}}_{k,i}, a_{k,i})$, $\xi_{j,i} = \xi_{k,j,i}^{(\ell)}(\bar{\mathbf{y}}_{k,j}, b_{k,j})$.

“ $f(\underline{\mathbf{y}}_{k,i} | \mathbf{y}_{k-1,i})$ ” to the joint variable nodes “ $\underline{\mathbf{y}}_{k,i}, a_{k,i}$ ” are calculated as (cf. [18, Sec. IX-A1])

$$\alpha_{k,i}(\underline{\mathbf{x}}_{k,i}, 1) = \int p_s f(\underline{\mathbf{x}}_{k,i} | \mathbf{x}_{k-1,i}) \tilde{f}(\mathbf{x}_{k-1,i}, 1) d\mathbf{x}_{k-1,i}$$

and $\alpha_{k,i}(\underline{\mathbf{x}}_{k,i}, 0) = \alpha_{k,i} f_{du}(\underline{\mathbf{x}}_{k,i})$ where $\alpha_{k,i}$ is a scalar that can be computed by making use of $\tilde{f}(\mathbf{y}_{k-1,i}) = \tilde{f}(\mathbf{x}_{k-1,i}, r_{k-1,i})$ and p_s (cf. [18, eq. (77)]). Note the belief $\tilde{f}(\mathbf{y}_{k-1,i})$ and the message $\alpha_{k,i}(\underline{\mathbf{y}}_{k,i})$ are normalized in the sense that they sum to unity, e.g.,

$$\sum_{\underline{r}_{k,i} \in \{0,1\}} \int \alpha_{k,i}(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i}) d\underline{\mathbf{x}}_{k,i} = 1.$$

At message passing iteration $\ell \in \{1, \dots, L\}$, the messages $\beta_{k,i,j}^{(\ell)}(\underline{\mathbf{y}}_{k,i}, a_{k,i})$ and $\xi_{k,j,i}^{(\ell)}(\bar{\mathbf{y}}_{k,j}, b_{k,j})$, $i \in \{1, \dots, I_k\}$, $j \in \{1, \dots, J_k\}$ passed from variable nodes “ $\underline{\mathbf{y}}_{k,i}, a_{k,i}$ ” and “ $\bar{\mathbf{y}}_{k,j}, b_{k,j}$ ” to

the indicator nodes “ $\Psi_{i,j}(a_{k,i}, b_{k,j})$ ” are given by (cf. (2.1))

$$\xi_{k,j,i}^{(\ell)}(\bar{\mathbf{y}}_{k,j}, b_{k,j}) = v(\bar{\mathbf{y}}_{k,j}, b_{k,j}; \mathbf{z}_{k,j}) \prod_{\substack{i'=1 \\ i' \neq i}}^{I_k} \varphi_{\Psi_{i',j} \rightarrow b_{k,j}}^{(\ell-1)}(b_{k,j}) \quad (2.11)$$

$$\beta_{k,i,j}^{(\ell)}(\mathbf{y}_{k,i}, a_{k,i}) = q(\mathbf{y}_{k,i}, a_{k,i}; \mathbf{z}_k) \alpha_{k,i}(\mathbf{y}_{k,i}) \prod_{\substack{j'=1 \\ j' \neq j}}^{J_k} v_{\Psi_{i,j'} \rightarrow a_{k,i}}^{(\ell)}(a_{k,i}). \quad (2.12)$$

In addition, the messages $\varphi_{\Psi_{i,j} \rightarrow b_{k,j}}^{(\ell)}(b_{k,j})$ and $v_{\Psi_{i,j} \rightarrow a_{k,i}}^{(\ell)}(a_{k,i})$ passed from indicator nodes “ $\Psi_{i,j}(a_{k,i}, b_{k,j})$ ” to variables nodes “ $\bar{\mathbf{y}}_{k,j}, b_{k,j}$ ” and “ $\mathbf{y}_{k,i}, a_{k,i}$ ” are obtained as (cf. (2.2)),

$$\varphi_{\Psi_{i,j} \rightarrow b_{k,j}}^{(\ell)}(b_{k,j}) = \sum_{a_{k,i}=0}^{J_k} \Psi_{i,j}(a_{k,i}, b_{k,j}) \sum_{\underline{r}_{k,i} \in \{0,1\}} \int \beta_{k,i,j}^{(\ell)}(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i}, a_{k,i}) d\underline{\mathbf{x}}_{k,i} \quad (2.13)$$

$$v_{\Psi_{i,j} \rightarrow a_{k,i}}^{(\ell)}(a_{k,i}) = \sum_{b_{k,j}=0}^{I_k} \Psi_{i,j}(a_{k,i}, b_{k,j}) \sum_{\bar{r}_{k,j} \in \{0,1\}} \int \xi_{k,j,i}^{(\ell)}(\bar{\mathbf{x}}_{k,j}, \bar{r}_{k,j}, b_{k,j}) d\bar{\mathbf{x}}_{k,j}. \quad (2.14)$$

By plugging (2.11) into (2.14) and (2.12) into (2.13), we finally obtain the following combined messages for $i \in \{1, \dots, I_k\}$, and $j \in \{1, \dots, J_k\}$

$$\varphi_{\Psi_{i,j} \rightarrow b_{k,j}}^{(\ell)}(b_{k,j}) = \sum_{a_{k,i}=0}^{J_k} \beta_{k,i}(a_{k,i}) \Psi_{i,j}(a_{k,i}, b_{k,j}) \prod_{\substack{j'=1 \\ j' \neq j}}^{J_k} v_{\Psi_{i,j'} \rightarrow a_{k,i}}^{(\ell)}(a_{k,i}) \quad (2.15)$$

$$v_{\Psi_{i,j} \rightarrow a_{k,i}}^{(\ell)}(a_{k,i}) = \sum_{b_{k,j}=0}^{I_k} \xi_{k,j}(b_{k,j}) \Psi_{i,j}(a_{k,i}, b_{k,j}) \prod_{\substack{i'=1 \\ i' \neq i}}^{I_k} \varphi_{\Psi_{i',j} \rightarrow b_{k,j}}^{(\ell-1)}(b_{k,j}). \quad (2.16)$$

Here, we have introduced the short notation

$$\beta_{k,i}(a_{k,i}) \triangleq \sum_{\underline{r}_{k,i} \in \{0,1\}} \int q(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i}, a_{k,i}; \mathbf{z}_k) \alpha_{k,i}(\underline{\mathbf{x}}_{k,i}, \underline{r}_{k,i}) d\underline{\mathbf{x}}_{k,i} \quad (2.17)$$

and

$$\xi_{k,j}(b_{k,j}) \triangleq \sum_{\bar{r}_{k,j} \in \{0,1\}} \int v(\bar{\mathbf{x}}_{k,j}, \bar{r}_{k,j}, b_{k,j}; \mathbf{z}_{k,j}) d\bar{\mathbf{x}}_{k,j}. \quad (2.18)$$

Note that with a small abuse of notation, one message passing iteration, indexed by ℓ , in (2.15) and (2.16) corresponds to two message passing iterations in (2.1) and (2.2). Iterative message passing in (2.15) and (2.16) is initialized at $\ell = 1$, by setting in (2.16) $\varphi_{\Psi_{i,j} \rightarrow b_{k,j}}^{(0)}(b_{k,j}) = 1$ for all $j \in \{1, \dots, J_k\}$ and $i \in \{1, \dots, I_k\}$.

These combined messages can be further simplified [41] as follows. Because of the binary consistency constraints expressed by $\Psi_{i,j}(a_{k,i}, b_{k,j})$, each message comprises only two different values. In particular, $\varphi_{\Psi_{i,j} \rightarrow b_{k,j}}^{(\ell)}(b_{k,j})$ in (2.15) takes on one value for $b_{k,j} = i$ and another for all $b_{k,j} \neq i$. Furthermore, $v_{\Psi_{i,j} \rightarrow a_{k,i}}^{(\ell)}(a_{k,i})$ in (2.16) takes on one value for $a_{k,i} = j$ and another for all $a_{k,i} \neq j$. Thus, each message can be represented (up to an irrelevant constant factor) by the ratio of the first value and the second value, hereafter denoted as $\varphi_{i,j}^{(\ell)}$ for $\varphi_{\Psi_{i,j} \rightarrow b_{k,j}}^{(\ell)}(b_{k,j})$ and $v_{j,i}^{(\ell)}$ for $v_{\Psi_{i,j} \rightarrow a_{k,i}}^{(\ell)}(a_{k,i})$. By exchanging simplified messages the computational complexity of each message passing iteration only scales as $O(I_k J_k)$ (see [18, 41] for details). Furthermore, it has been shown in [41] that iterative probabilistic data association following (2.15)–(2.16) and its simplified version discussed above are guaranteed to converge.

Finally, after the last message passing iteration $\ell = L$, the beliefs $\tilde{f}(\underline{\mathbf{y}}_{k,i}, a_{k,i})$, $i \in \{1, \dots, I_k\}$ and $\tilde{f}(\bar{\mathbf{y}}_{k,j}, b_{k,j})$, $j \in \{1, \dots, J_k\}$ are computed according to

$$\begin{aligned} \tilde{f}(\underline{\mathbf{y}}_{k,i}, a_{k,i}) &= \frac{1}{\underline{C}_{k,i}} q(\underline{\mathbf{y}}_{k,i}, a_{k,i}; \mathbf{z}_k) \alpha_{k,i}(\underline{\mathbf{y}}_{k,i}) \prod_{j=1}^{J_k} v_{\Psi_{i,j} \rightarrow a_{k,i}}^{(L)}(a_{k,i}) \\ \tilde{f}(\bar{\mathbf{y}}_{k,j}, b_{k,j}) &= \frac{1}{\bar{C}_{k,j}} v(\bar{\mathbf{y}}_{k,j}, b_{k,j}; \mathbf{z}_{k,j}) \prod_{i=1}^{I_k} \varphi_{\Psi_{i,j} \rightarrow b_{k,j}}^{(L)}(b_{k,j}) \end{aligned}$$

where $\underline{C}_{k,i}$, and $\bar{C}_{k,j}$ are normalizing constants that make sure that $\tilde{f}(\underline{\mathbf{y}}_{k,i}, a_{k,i})$ and $\tilde{f}(\bar{\mathbf{y}}_{k,j}, b_{k,j})$ sum and integrate to unity. The marginal beliefs $\tilde{f}(\underline{\mathbf{y}}_{k,i})$, $\tilde{f}(\bar{\mathbf{y}}_{k,j})$, $\tilde{p}(a_{k,i})$, and $\tilde{p}(b_{k,j})$ can then be obtained from $\tilde{f}(\underline{\mathbf{y}}_{k,i}, a_{k,i})$ and $\tilde{f}(\bar{\mathbf{y}}_{k,j}, b_{k,j})$ by marginalization. In particular, the

approximate marginal posterior PDFs of augmented states $\tilde{f}(\underline{\mathbf{y}}_{k,i}) \approx f(\underline{\mathbf{y}}_{k,i}|\mathbf{z}_{1:k})$, $i \in \{1, \dots, I_k\}$ and $\tilde{f}(\underline{\mathbf{y}}_{k,j}) \approx f(\underline{\mathbf{y}}_{k,j}|\mathbf{z}_{1:k})$, $j \in \{1, \dots, J_k\}$ are used for object declaration and state estimation as discussed in Section 2.3.4. Furthermore, the approximate marginal association probabilities $\tilde{p}(a_{k,i}) \approx p(a_{k,i}|\mathbf{z}_{1:k})$, $i \in \{1, \dots, I_k\}$ and $\tilde{p}(b_{k,j}) \approx p(b_{k,j}|\mathbf{z}_{1:k})$, $j \in \{1, \dots, J_k\}$ are used in a preprocessing step for performance evaluation discussed in Sections 2.7.1 and 2.7.2.

2.5 Proposed NEBP-based MOT Algorithm

In this section, we start with a discussion on how neural networks can extract features from raw sensor data by using previous state estimates and preprocessed measurements. We then introduce the proposed NEBP framework for MOT, which compared to BP for MOT uses features as an additional input. Since we limit our discussion to a single time step, we will omit the time index k in what follows.

2.5.1 Feature Extraction

We consider two types of features: (i) features that represent motion information (e.g., position and velocity) and (ii) features that represent shape information.

For each legacy PO $i \in \{1, \dots, I\}$, the shape feature $\mathbf{h}_{a_i, \text{shape}}$ is extracted as

$$\mathbf{h}_{a_i, \text{shape}} = g_{\text{shape},1}(\mathcal{Z}^-; \hat{\underline{\mathbf{x}}}_i^-) \quad (2.19)$$

where $\hat{\underline{\mathbf{x}}}_i^-$ is the approximate MMSE state estimate of legacy PO i at the previous time step. Furthermore, $\mathcal{Z}^-$ is the raw sensor data at the previous time step and $g_{\text{shape},1}(\mathcal{Z}^-; \hat{\underline{\mathbf{x}}}_i^-)$ is a neural network.

Similarly, for each preprocessed measurement $\mathbf{z}_j$, $j \in \{1, \dots, J\}$ the shape feature $\mathbf{h}_{b_j, \text{shape}}$ is obtained as

$$\mathbf{h}_{b_j, \text{shape}} = g_{\text{shape},2}(\mathcal{Z}; \mathbf{z}_j) \quad (2.20)$$

where $g_{\text{shape},2}(\mathcal{Z}; \mathbf{z}_j)$ is again a neural network and $\mathcal{Z}$ is the raw sensor data collected at the

current time step.

Finally, for each legacy PO $i \in \{1, \dots, I\}$ and each measurement $j \in \{1, \dots, J\}$, a motion feature is computed according to

$$\begin{aligned} \mathbf{h}_{a_i, \text{motion}} &= g_{\text{motion},1}(\hat{\mathbf{x}}_i^-, \hat{r}_i^-) \\ \mathbf{h}_{b_j, \text{motion}} &= g_{\text{motion},2}(\mathbf{z}_j) \end{aligned} \quad (2.21)$$

where $\hat{r}_i^-$ is the approximate existence probability of legacy PO i at the previous time step. Furthermore, $g_{\text{motion},1}(\hat{\mathbf{x}}_i^-, \hat{r}_i^-)$ and $g_{\text{motion},2}(\mathbf{z}_j)$ are again neural networks. We will discuss one particular instance of shape feature extraction in Section 2.7.2.

2.5.2 GNN Topology and BP Message Enhancement

The conjecture of this work is that in many MOT applications (i) object dynamics and existence can be described accurately by a statistical model represented by the PDFs $f(\mathbf{x}_{k,i}|\mathbf{x}_{k-1,i})$, $f_u(\bar{\mathbf{x}}_{k,j})$ and parameters p_s, μ_u ; (ii) measurement detection and the resulting measurements of the object's position can also be described well by a statistical model represented by PDFs $f(\mathbf{z}_{k,j}|\mathbf{x}_{k,n})$, $f_{fa}(\mathbf{z}_{k,j})$ and parameters p_d, μ_{fa} ; but (iii) object shape information are difficult to represent accurately by a statistical model. Thus, we can make use of models available for (i) and (ii), while for (iii), we best learn the influence of object shape information on measurement detection directly from the data itself. Thus, contrary to the original NEBP approach, in our NEBP method, only the parts of the MOT factor graph that model the data generating process are matched by the GNN. We also use a message passing order that is tailored to the considered MOT problem. The matched parts of the factor graph are highlighted in Fig. 2.3. All factor nodes in this part of the factor graph are either singleton or pairwise factor nodes. As discussed in Section 2.2.2, in NEBP singleton factor nodes are not matched by GNN nodes. In addition, in our considered factor graph in Fig. 2.3(a), the pairwise factor nodes “ $\Psi_{i,j}(a_i, b_j)$ ”, $j \in \{1, \dots, J\}$ and $i \in \{1, \dots, I\}$ represent simple binary consistency constraints. Thus, we do not explicitly

model factor nodes by GNN nodes. The node embeddings of GNNs nodes introduced for the variable nodes “ $\underline{\mathbf{y}}_i, a_i$ ”, $i \in \{1, \dots, I\}$ are denoted as $\mathbf{h}_{a_i}$ and the node embeddings introduced for variables nodes “ $\bar{\mathbf{y}}_j, b_j$ ”, $j \in \{1, \dots, J\}$ are denoted as $\mathbf{h}_{b_j}$. Finally, following the topology of the factor graph for data association in Fig. 2.3a, GNN edges are introduced such that the bipartite GNN shown in Fig. 2.3b is obtained.

We recall from Section 2.2.1 that for singleton factor nodes, the message passed to the adjacent variable node is equal to the factor node itself. As a result, $q(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $v(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$ not only describe factor nodes but also the messages that are enhanced. There are two challenges related to directly enhancing these messages based on the GNN according to (2.3)–(2.7), i.e., (i) the codomain of $q(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $v(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$ can be very large, which complicates the training of the GNN [103] (see also Sections 2.5.3 and 2.6) and (ii) the messages $q(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $v(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$ involve the continuous random variables $\underline{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$ which makes it impossible to enhance them by the output of a GNN for every possible value of $\underline{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$ individually.

To address the first challenge, we introduce normalized versions³ of the original BP messages as

$$q_s(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z}) = \frac{1}{C_q} q(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z}) \quad (2.22)$$

$$v_s(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j) = \frac{1}{C_v} v(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j) \quad (2.23)$$

where $C_q = \sum_{a_i=0}^J \sum_{\underline{r}_i \in \{0,1\}} \int q(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z}) d\underline{\mathbf{x}}_i$ and $C_v = \sum_{b_j=0}^I \sum_{\bar{r}_j \in \{0,1\}} \int v(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j) d\bar{\mathbf{x}}_j$ are the normalization constants. Note that after normalization the codomain of $q_s(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $v_s(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$ is limited to the interval $[0, 1]$.

The second challenge is addressed by enhancing BP messages $q_s(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$, $i \in$

³Multiplying BP messages by a constant factor does not alter the resulting beliefs [12].

$\{1, \dots, I\}$ and $v_s(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$, $j \in \{1, \dots, J\}$ as follows (cf. (2.7))

$$\tilde{q}_s(\underline{\mathbf{x}}_i, 1, a_i = j; \mathbf{z}) = \omega_j \cdot q_s(\underline{\mathbf{x}}_i, 1, a_i = j; \mathbf{z}) + \mu_i(j) \quad (2.24)$$

$$\tilde{v}_s(\bar{\mathbf{x}}_j, 1, b_j = 0; \mathbf{z}_j) = \omega_j \cdot v_s(\bar{\mathbf{x}}_j, 1, b_j = 0; \mathbf{z}_j). \quad (2.25)$$

Here, $\omega_j \in (0, 1)$ and $\mu_i(j) \in \mathbb{R}^+$ are computed from information provided by the GNN as discussed in the following Section 2.5.3. The other entries of the messages $\tilde{q}_s(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$, $i \in \{1, \dots, I\}$ and $\tilde{v}_s(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$, $j \in \{1, \dots, J\}$ remain unenhanced, i.e., $\tilde{q}_s(\underline{\mathbf{x}}_i, 0, a_i; \mathbf{z}) = q_s(\underline{\mathbf{x}}_i, 0, a_i; \mathbf{z})$, $\tilde{q}_s(\underline{\mathbf{x}}_i, 1, a_i = 0; \mathbf{z}) = q_s(\underline{\mathbf{x}}_i, 1, a_i = 0; \mathbf{z})$, $\tilde{v}_s(\bar{\mathbf{x}}_j, 0, b_j; \mathbf{z}_j) = v_s(\bar{\mathbf{x}}_j, 0, b_j; \mathbf{z}_j)$, and $\tilde{v}_s(\bar{\mathbf{x}}_j, 1, b_j = i; \mathbf{z}_j) = v_s(\bar{\mathbf{x}}_j, 1, b_j = i; \mathbf{z}_j)$, $i \in \{1, \dots, I\}$. Note that calculating NEBP messages according to (2.24) and (2.25) avoids enhancing $q(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $v(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$ for every possible value of $\underline{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$. All other BP messages are not neural enhanced.

Finally, neural enhanced data association can be performed by replacing the functions $\beta_i(a_i)$ and $\xi_j(b_j)$ in (2.15) and (2.16) with their neural enhanced counterparts $\tilde{\beta}_{s,i}(a_i)$ and $\tilde{\xi}_{s,j}(b_j)$. These neural enhanced counterparts are obtained by replacing in (2.17) and (2.18) the BP messages $q(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $v(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$ with the NEBP messages $\tilde{q}_s(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $\tilde{v}_s(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$, respectively. In particular, for $i \in \{1, \dots, I\}$ we obtain

$$\tilde{\beta}_{s,i}(a_i = j) = \omega_j \frac{\beta_i(a_i = j)}{C_q} + \mu_i(j) \quad j \in \{1, \dots, J\} \quad (2.26)$$

and $\tilde{\beta}_{s,i}(a_i = 0) = \beta_{s,i}(a_i = 0) = \beta_i(a_i = 0)/C_q$. Similarly, for $j \in \{1, \dots, J\}$ we get

$$\tilde{\xi}_{s,j}(b_j = 0) = \omega_j \frac{\xi_j(b_j = 0)}{C_v} \quad (2.27)$$

and $\tilde{\xi}_{s,j}(b_j = i) = \xi_{s,j}(b_j = i) = \xi_j(b_j = i)/C_v$, $i \in \{1, \dots, I\}$.

The value $\beta_i(a_i = j)$, $j \in \{1, \dots, J\}$ provides a likelihood ratio for the measurement with index j being associated to the legacy PO with index i [18]. In addition, $\xi_j(b_j = 0)$

provides a likelihood ratio for the measurement with index j being generated by a new PO. The shape association term $\mu_i(j) \geq 0$ in (2.26), calculated by the GNN implements object shape association, which can be interpreted as follows. The GNN compares the shape feature extracted for legacy PO i with the shape feature extracted for measurement j and, if there is a good match, outputs a large $\mu_i(j) > 0$. According to (2.26), this effectively increases the likelihood ratio that the legacy PO i is associated with the measurement j . Note that there is no shape association term in (2.27). Since the shape feature extracted for new PO j would be the same as the shape feature for measurement j , comparing shape features as performed for legacy POs and measurements is not possible.

The scalar $\omega_j \in (0, 1)$ in (2.26) and (2.27) provided by the GNN implements false alarm rejection. In particular, $\omega_j < 1$ is equal to the local increase of the false alarm distribution according to $\tilde{f}_{\text{fa}}(\mathbf{z}_j) = \frac{1}{\omega_j} f_{\text{fa}}(\mathbf{z}_j)$ (cf. (2.22), (2.23), (2.9), and (2.10)). In (2.26), this local increase of the false alarm distribution makes it less likely that the measurement $\mathbf{z}_j$ is associated to a legacy PO. In (2.27), this local increase reduces the existence probability of the new PO introduced for the measurement $\mathbf{z}_j$.

2.5.3 Statement of the NEBP for MOT Algorithm

NEBP for MOT consists of the following steps:

First, the conventional BP-based MOT algorithm is run until convergence, from which we obtain $\beta_{s,i} = [\beta_{s,i}(0) \cdots \beta_{s,i}(J)]^T \in \mathbb{R}^{J+1}$, $\xi_{s,j} = [\xi_{s,j}(0) \cdots \xi_{s,j}(I)]^T \in \mathbb{R}^{I+1}$, $\varphi_{i,j} \in \mathbb{R}$, and $\nu_{i,j} \in \mathbb{R}$ (cf. Section 2.4.2), where $\beta_{s,i}(j) = \beta_i(a_i = j)/C_q$ and $\xi_{s,j}(i) = \xi_j(b_j = i)/C_v$.

Next, GNN message passing is executed iteratively. In particular, at iteration $p \in \{1, \dots, P\}$ the messages passed along the edges of GNN are computed as

$$\mathbf{m}_{a_i \rightarrow b_j}^{(p)} = g_e(\mathbf{h}_{a_i}^{(p)}, \mathbf{h}_{b_j}^{(p)}, \beta_{s,i}(j), \varphi_{i,j}) \quad (2.28)$$

$$\mathbf{m}_{b_j \rightarrow a_i}^{(p)} = g_e(\mathbf{h}_{a_i}^{(p)}, \mathbf{h}_{b_j}^{(p)}, \beta_{s,i}(j), \nu_{i,j}) \quad (2.29)$$

where $g_e(\cdot)$ is the edge neural network. Furthermore, node embedding vectors of each node are obtained as

$$\mathbf{h}_{a_i}^{(p+1)} = g_n\left(\mathbf{h}_{a_i}^{(p)}, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{b_j \rightarrow a_i}^{(p)}, \beta_{s,i}(0)\right) \quad (2.30)$$

$$\mathbf{h}_{b_j}^{(p+1)} = g_n\left(\mathbf{h}_{b_j}^{(p)}, \sum_{i \in \mathcal{N}(j)} \mathbf{m}_{a_i \rightarrow b_j}^{(p)}, \xi_{s,j}(0)\right). \quad (2.31)$$

where $g_n(\cdot)$ is the node neural network. The iterative processing scheme is initialized by setting node embeddings equal to motion and shape features, i.e., $\mathbf{h}_{a_i}^{(1)} = [\mathbf{h}_{a_i, \text{motion}}^T \ \mathbf{h}_{a_i, \text{shape}}^T]^T$ and $\mathbf{h}_{b_j}^{(1)} = [\mathbf{h}_{b_j, \text{motion}}^T \ \mathbf{h}_{b_j, \text{shape}}^T]^T$.

After computing (2.28)-(2.31) for P iterations, the refinement ω_j used in (2.24) and (2.25) is calculated as

$$\omega_j = \sigma(T(\omega_j^* - \delta)) \text{ with } \omega_j^* = g_s(\mathbf{h}_{b_j}^{(P)}). \quad (2.32)$$

Here, $g_s(\cdot)$ is a neural network and $\sigma(x) = 1/(1 + e^{-x}) \in (0, 1)$ is the sigmoid function. Furthermore, the temperature T and the bias δ are hyperparameters [104] that make it possible to calibrate the transition of the sigmoid.

Finally, the refinement $\mu_i(j)$ used in (2.24) is obtained as

$$\mu_i(j) = \text{ReLU}(\mu_i^*(j)) \text{ with } \mu_i^*(j) = g_d(\mathbf{m}_{b_j \rightarrow a_i}^{(P)}) \quad (2.33)$$

where $g_d(\cdot)$ is another neural network and $\text{ReLU}(\cdot)$ is the rectified linear unit defined as $\text{ReLU}(x) = (x + |x|)/2$.

Finally, iterative probabilistic DA is again run until convergence by replacing $q_s(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $v_s(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$ in (2.22) and (2.23) by its neural enhanced counterparts $\tilde{q}_s(\underline{\mathbf{x}}_i, \underline{r}_i, a_i; \mathbf{z})$ and $\tilde{v}_s(\bar{\mathbf{x}}_j, \bar{r}_j, b_j; \mathbf{z}_j)$ in (2.24) and (2.25), respectively. This results in the enhanced messages $\tilde{\varphi}_{\Psi_{i,j} \rightarrow b_j}^{(L)}(b_j)$ and $\tilde{\nu}_{\Psi_{i,j} \rightarrow a_i}^{(L)}(a_i)$ (cf. Section 2.4.2), which are then used for the calculation of legacy PO beliefs $\tilde{f}(\underline{\mathbf{y}}_i)$, $i \in \{1, \dots, I\}$ and new PO beliefs $\tilde{f}(\bar{\mathbf{y}}_j)$, $j \in \{1, \dots, J\}$ as discussed

in Section 2.4.2.

2.5.4 Complexity Analysis

In this section, we analyze and compare the computational complexity of the conventional BP and the proposed NEBP methods for MOT. Since both BP and NEBP for MOT follow the detect-then-track paradigm, the complexity of the detector has to be taken into account. We denote by $|\mathcal{Z}|$ the number of raw sensor data points. For example, if a LiDAR sensor is considered, this is the number of points of the LiDAR point cloud, and if a camera sensor is used, this is the number of pixels of the camera image. Then the number of operations needed for detection is $c_{\text{det}}|\mathcal{Z}|$, where c_{det} is a constant that depends on the size and type of neural network used as the detector $g_{\text{det}}(\cdot)$. As discussed in [18, 60], the number of operations needed for the conventional BP method for MOT algorithm is $c_{\text{bp}}IJ$, where c_{bp} is a constant that depends on the number of message passing iterations for DA, the number of particles, and further parameters. In total, the number of operations for BP is $c_{\text{det}}|\mathcal{Z}| + c_{\text{bp}}IJ$. Thus, the computational complexity of BP scales as $O(|\mathcal{Z}| + IJ)$.

The additional operations of NEBP compared to conventional BP are related to feature extraction and the GNN. Feature extraction as discussed in (2.19)–(2.21) requires $c_{\text{shape}}|\mathcal{Z}| + c_{\text{motion}}(I + J)$ operations, where $c_{\text{shape}}, c_{\text{motion}}$ are constants that depend on the size and type of the neural networks $g_{\text{shape}}(\cdot), g_{\text{motion}}(\cdot)$, respectively. The GNN is a fully connected bipartite graph, i.e., it consists of two sets of nodes, and each node in the first set is connected via an edge to each node in the second set. The number of nodes in each set is equal to I and J , respectively. GNN messages are exchanged on the IJ edges of the network according to (2.28) and (2.29). This is followed by GNN messages aggregation in (2.30) and (2.31), as well as BP message refinement in (2.32) and (2.33). The total number of operations is hence equal to $c_{\text{gnn},1}IJ + c_{\text{gnn},2}I + c_{\text{gnn},3}J$, where $c_{\text{gnn},\cdot}$ depends on the size and type of neural networks $g_e(\cdot), g_n(\cdot), g_s(\cdot)$, and $g_d(\cdot)$. It can thus be seen, that the computational complexity of NEBP also scales as $O(|\mathcal{Z}| + IJ)$. Note that due to the additional operations performed by NEBP, the

runtime of NEBP is longer compared to BP. Runtimes of BP and NEBP are further analyzed in Section 2.7.4.

2.6 Loss Function and Training

Training of the proposed NEBP method is performed in a supervised manner. It is assumed that a training set consisting of ground truth object tracks is available. A ground truth object track is a sequence of object positions. Every sequence is characterized by an object identity (ID). During the training phase, the parameters of all neural networks are updated through back-propagation, which computes the gradient of the loss function. The loss function has the form $\mathcal{L} = \mathcal{L}_r + \mathcal{L}_a$, where the two contributions $\mathcal{L}_r$ and $\mathcal{L}_a$ are related to false alarm rejection and object shape association, respectively.

We consider the following binary cross-entropy loss [8, Chapter 4.3] for false alarm rejection, i.e.,

$$\mathcal{L}_r = \frac{-1}{J} \sum_{j=1}^J \omega_j^{\text{gt}} \ln(\omega_j) + \epsilon(1 - \omega_j^{\text{gt}}) \ln(1 - \omega_j) \quad (2.34)$$

where $\omega_j^{\text{gt}} \in \{0, 1\}$ is the pseudo ground truth label for each measurement and $\epsilon \in \mathbb{R}^+$ is a tuning parameter. The pseudo ground truth label ω_j^{gt} is equal to 1 if the distance between the measurement and any ground truth position is smaller or equal to T_{dist} , and 0 otherwise. The tuning parameter ϵ addresses the imbalance problem in learning-based binary classification (see [105] for details). This problem is caused by the fact that object detectors for MOT typically produce more false alarm measurements than true measurements⁴. Consequently, the weight of individual false alarm measurements in the loss function has to be reduced by setting $\epsilon < 1$.

Since $\tilde{\beta}_{s,i}(a_i = j)$ in (2.26) represents the likelihood that the legacy PO i is associated to the measurement j , ideally $\mu_i(j)$ is large if PO i is associated to the measurement j , and is equal to zero if they are not associated. Thus, we consider the following binary cross-entropy loss for object shape association, i.e.,

⁴Missing an object is typically more severe than producing a false alarm.

$$\mathcal{L}_a = \frac{-1}{IJ} \sum_{i=1}^I \sum_{j=1}^J \mu_i^{\text{gt}}(j) \ln(\sigma(\mu_i^*(j))) + (1 - \mu_i^{\text{gt}}(j)) \ln(1 - \sigma(\mu_i^*(j))) \quad (2.35)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function and $\mu_i^{\text{gt}} = [\mu_i^{\text{gt}}(1) \cdots \mu_i^{\text{gt}}(J)]^T \in \{0, 1\}^J$ is the pseudo ground truth association vector of legacy PO $i \in \{1, \dots, I\}$. In each pseudo ground truth association vector μ_i^{gt} , at most one element is equal to one and all the other elements are equal to zero. In the binary entropy loss (2.35), we use $\mu_i^*(j)$ from (2.33) instead of $\mu_i(j)$. This is because otherwise the ReLU operation “blocks” certain gradients that are beneficial for the training of the GNN. Note that gradients $\partial \mathcal{L}_a / \partial \mu_i^*(j)$ are zero for negative values of $\mu_i^*(j)$ if $\mu_i(j)$ is used in (2.35). It has been observed that by performing backpropagation by also making use of the gradients related to the negative values of $\mu_i^*(j)$, the GNN can be trained more efficiently.

At each time step, pseudo ground truth association vectors are constructed from measurements and ground truth object tracks based on the following rules:

- *Get Measurement IDs:* First, the Euclidean distance between all ground truth positions and measurements is computed. Next, the Hungarian algorithm [35] is performed to find the best association between ground truth positions and measurements. Finally, all measurements that have been associated with a ground truth position and have a distance to that ground truth position that is smaller than T_{dist} , inherit the ID of the ground truth position. All other measurements do not have an ID.
- *Update Legacy PO IDs:* Legacy POs inherit the ID from the previous time step. If a legacy PO with ID has a distance not larger than T_{dist} to a ground truth position with the same ID, it keeps its ID. If a legacy PO $i \in \{1, \dots, I\}$ has the same ID as measurement $j \in \{1, \dots, J\}$, the entry $\mu_i^{\text{gt}}(j)$ is set to one. All other entries of μ_i^{gt} are set to zero.
- *Introduce New PO IDs:* A new PO $j \in \{1, \dots, J\}$ inherits the ID from the corresponding measurement if the measurement has an ID that is different from the ID of any legacy PO.

All other new POs do not have an ID.

2.7 Numerical Results

To validate the performance of our method, we present results in an autonomous driving scenario.

2.7.1 Experimental Setup

Our numerical evaluation is based on the *nuScenes* autonomous driving dataset [106], which contains 1000 autonomous driving scenes. We use the official predefined dataset split, where 700 scenes are considered for training, 150 for validation, and 150 for testing. Each scene has a length of roughly 20 seconds and contains keyframes (frames with ground truth annotations) sampled at 2Hz. There are seven object classes. The proposed MOT method and reference techniques are performed for each class individually. If not stated otherwise, all the operations described next are performed for each class in parallel. We only consider the LiDAR data provided by the nuScenes dataset. A scene of the considered autonomous driving application is shown in Fig. 2.4.

The state of a PO $\mathbf{x}_{k,n} \in \mathbb{R}^4$ consists of its 2-D position and 2-D velocity. Preprocessed measurements are extracted from the LiDAR data. For the extraction of preprocessed measurements, we employed the *CenterPoint* [69] detector which is based on deep learning⁵. Any preprocessed measurement $\mathbf{z}_{k,j}$ consists of 2-D position, 2-D velocity, and a confidence score $0 < s_{k,j} \leq 1$.

Object dynamics are modeled by a constant-velocity motion model [108]. Object tracking is performed in a global reference frame that is predefined for each scene [106]. The considered region of interest (ROI) is defined by $[x_{e,k} - 54, x_{e,k} + 54] \times [y_{e,k} - 54, y_{e,k} + 54]$, where $(x_{e,k}, y_{e,k})$ is the 2-D position of the “ego vehicle” that is equipped with the LiDAR sensor. The PDFs that

⁵The measurements provided by the CenterPoint detector are further preprocessed using non-maximum suppression (NMS) with 3-D intersection over union (IoU) [75, 107] where the threshold is set to 0.1.

describe false alarms $f_{\text{fa}}(\cdot)$ and unknown objects $f_{\text{u}}(\cdot)$ are uniformly distributed over the ROI. The measurement model that defines the likelihood function $f(\mathbf{z}_{k,j}|\mathbf{x}_{k,n})$ is linear with additive Gaussian noise, i.e., $\mathbf{z}_{k,j} = \mathbf{H}_k\mathbf{x}_{k,n} + \mathbf{v}_{k,j}$, where $\mathbf{v}_{k,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ with $\mathbf{R}$ being the diagonal covariance matrix. The probability of survival is set to $p_s = 0.999$. The threshold for target declaration is $T_{\text{dec}} = 0.5$.

The pruning threshold discussed in Section 2.3.4 is set to $T_{\text{pru}} = 10^{-3}$. In addition, we also prune new POs with $\tilde{p}(b_{k,j} = 0) < 0.8$ to further reduce the number of false objects and computational complexity. All other parameters used in the system model are extracted from training data. For the BP part of the proposed NEBP method, we use the particle-based implementation introduced in [60].

We consider using the average multi-object tracking accuracy (AMOTA) metric proposed in [72] to evaluate the performance of our algorithm. In addition, we also use the widely used CLEAR metrics [109] and track quality measures [110] that include the number of IDS and Frag. The number of IDSs is increased if a ground truth object is matched to an estimated object with index i at the current time step, while it was matched to an estimated object with index $j \neq i$ at a previous time. The number of Frags is increased if a ground truth object is matched to an estimated object at the previous time step, but it is not matched to any estimated objects at the current time step. Note that AMOTA is the primary metric used by the nuScenes tracking challenge [106].

2.7.2 Implementation Details

For shape features extraction as discussed in Section 2.5.1, a neural network $g_{\text{shape}}(\cdot) \triangleq g_{\text{shape},1}(\cdot) = g_{\text{shape},2}(\cdot)$ that consists of two stages is introduced. The first stage is a *VoxelNet* [81], a neural network architecture that is used as the backbone for a variety of object detectors [69, 83]. The VoxelNet takes the LiDAR scan, $\mathcal{Z}_k$, as input, and outputs a 3D tensor of size $180 \times 180 \times 512$. This tensor is typically referred to as feature map. The first two dimensions of the feature map form a grid with 180×180 elements that cover the ROI. For each grid

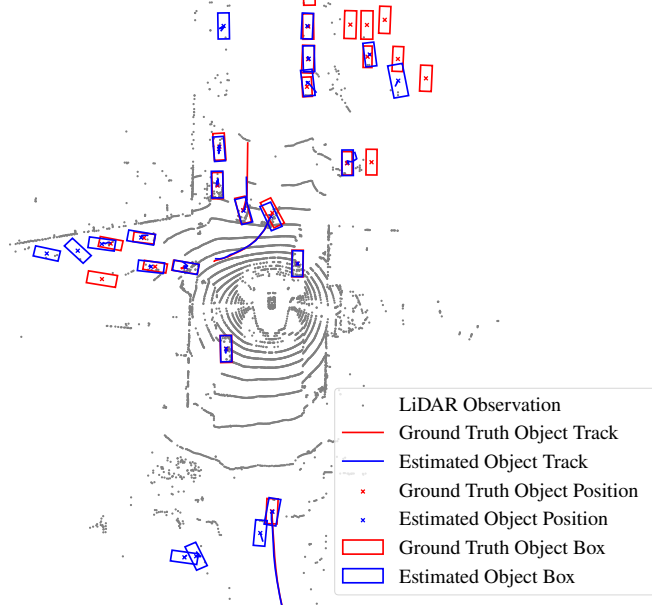


Figure 2.4. Top-down view of the considered autonomous driving scene. For each ground truth and estimated vehicle, we plot tracks as well as positions and bounding boxes at the last time step. This scene is part of the *nuScenes* autonomous driving dataset.

point, there is a feature vector with 512 elements. The second stage is a convolutional neural network (CNN) that consists of two convolutional layers and a single-hidden-layer multi-layer perceptron (MLP). Here, we use a CNN since it has fewer trainable parameters compared to an MLP and is thus easier to train. Note that CNNs have been widely used for feature extraction [81, 111]. The CNN extracts shape features from a reduced feature map, as discussed next.

For the extraction of shape features in the second stage, at first, the grid point of the feature map that corresponds to the considered POs or measurements is located. Then, the feature vector at this grid point and the 8 feature vectors at adjacent grid points are extracted. As a result, for each PO and each measurement, a reduced feature map of size $3 \times 3 \times 512$ is extracted. This reduced feature map is then used as the input of a CNN. Finally, the CNN computes the shape feature $\mathbf{h}_{a_i, \text{shape}}$ or $\mathbf{h}_{b_j, \text{shape}}$. The considered feature map of dimension $180 \times 180 \times 512$ has been precomputed by the CenterPoint [69] method. The same VoxelNet is shared across all seven object classes. Its parameters remain fixed during the training of the proposed method.

The other neural networks $g_e(\cdot)$, $g_n(\cdot)$, $g_d(\cdot)$, $g_s(\cdot)$, and $g_{\text{motion}}(\cdot) \triangleq g_{\text{motion},1}(\cdot) =$

$g_{\text{motion},2}(\cdot)$ are MLPs with a single hidden layer and a leaky ReLU activation function. All feature vectors, i.e., $\mathbf{h}_{a_i, \text{motion}}$ and $\mathbf{h}_{a_i, \text{shape}}$, $i \in \{1, \dots, I\}$ as well as $\mathbf{h}_{b_j, \text{motion}}$ and $\mathbf{h}_{b_j, \text{shape}}$, $j \in \{1, \dots, J\}$, consist of 128 elements. The number of GNN iterations is $P = 3$. Training of the proposed method is performed based on the *Adam optimizer* [112]. The batch size, learning rate, and the number of “epochs”, i.e., the number of times the Adam optimizer processes the entire training dataset, are set to 1, 10^{-4} , and 8, respectively. The hyperparameter ϵ in (2.34) is set to 0.1 and the threshold T_{dist} for the pseudo ground truth extraction discussed in Section 2.6 is set to 2 meters.

Evaluation of AMOTA requires a score for each estimated object. It was observed that a high AMOTA performance is obtained by calculating the estimated object score as a combination of existence probability and measurement score. In particular, for legacy PO $i \in \{1, \dots, I\}$ we calculate an estimated object score as

$$\underline{s}_i = \tilde{p}(\underline{r}_i = 1) + \sum_{j=1}^J \tilde{p}(a_i = j) s_j. \quad (2.36)$$

For new PO $j \in \{1, \dots, J\}$ the estimated object score is given by $\bar{s}_j = \tilde{p}(\bar{r}_j = 1) + s_j$.

2.7.3 Calibration

The calibration of the sigmoid introduced in (2.32) is performed as follows. For training, we set $T = 1$ and $\delta = 0$. However, for inference we set $T > 0$ and $\delta > 0$ such that the sigmoid in (2.32) transitions to zero quicker and for smaller values of ω_j^* . The different calibration values for inference are necessary because the loss function used for training and the AMOTA metric used for performance evaluation behave differently. In particular, for performance evaluation based on AMOTA, missing an object is significantly more severe than a false object. (Note that the AMOTA metric can not directly be used for training because it is not differentiable [8].) The calibration values T and δ used for inference are selected based on a grid search over possible values $T \in \mathcal{T} = \{0.5, 1, 2, 4, 8, 16, 32, +\infty\}$ and $\delta \in \mathcal{A} = \{\delta \mid \sigma(\delta) \in \{0.01, 0.02, 0.05, 0.1, 0.2\}\}$.

The values for δ and T that achieved the highest AMOTA value on a subset of the training set were obtained as $\sigma^{-1}(0.2, 0.02, 0.2, 0.02, 0.2, 0.01, 0.01)$ and $(\infty, 8, 1, \infty, 1, 8, 0.5)$ for the classes bicycle, bus, car, motorcycle, pedestrian, trailer, and truck, respectively.

2.7.4 Performance Evaluation

For performance evaluation, we use state-of-the-art reference methods that all use measurements provided by the CenterPoint detector [69], which was the best LiDAR-only object detector for the nuScenes dataset at the time of the submission of this work. In particular, BP refers to the conventional BP-based MOT method [18]. CenterPointT refers to the tracking method proposed in [69]. It uses a heuristic to create new tracks and a greedy matching algorithm based on the Euclidean distance to associate measurements provided by the CenterPoint detector. The methods in [75, 78, 113] all follow a similar strategy. The CBMOT method [113] adopts a score update function for estimated object scores. Chiu et al. [78] make use of a hybrid distance that combines the Mahalanobis distance with a proposed deep feature distance. SimpleTrack [75] uses the generalized IoU (GIoU) as the distance for measurement association. In SimpleTrack [75], the object detector is also applied to non-keyframes and has a measurement rate of 10Hz. The Immortal tracker [76] has a measurement rate of 2Hz. It follows the tracking approach of SimpleTrack, except that it never terminates tracks. PMB [67] implements a Poisson multi-Bernoulli filter for MOT that relies on a global nearest neighbor approach for data association. Finally, OGR3MOT [73] utilizes a network flow formulation and transforms the data association problem into a classification problem.

In Table 2.1 and 2.2, we present the tracking performance of the considered methods on the nuScenes validation and test sets based on measurements provided by the CenterPoint detector. The symbol “-” in Table 2.2 indicates that the metric is not reported. It can be seen that the proposed NEBP approach outperforms all reference methods in terms of AMOTA performance. Furthermore, it can be observed, that BP and NEBP achieve a much lower IDS and Frag metric compared to the reference methods. This is because both BP and NEBP make

Table 2.1. Performance results: nuScenes test set

Method	Modalities	AMOTA $\uparrow$	IDS $\downarrow$	Frag $\downarrow$
CenterPointT [69]	LiDAR	0.638	760	529
CBMOT [113]	LiDAR	0.649	557	450
Chiu et al. [78]	LiDAR+Camera	0.655	1043	717
OGR3MOT [73]	LiDAR	0.656	288	371
SimpleTrack [75]	LiDAR	0.668	575	591
Immortal [76]	LiDAR	0.677	320	477
PMB [67]	LiDAR	0.678	770	431
BP	LiDAR	0.666	182	245
NEBP (proposed)	LiDAR	0.683	227	299

Table 2.2. Performance results: nuScenes validation set

Method	Modalities	AMOTA $\uparrow$	IDS $\downarrow$	Frag $\downarrow$
CenterPointT [69]	LiDAR	0.665	562	424
CBMOT [113]	LiDAR	0.675	494	-
Chiu et al. [78]	LiDAR+Camera	0.687	-	-
OGR3MOT [73]	LiDAR	0.693	262	332
SimpleTrack [75]	LiDAR	0.696	405	-
Immortal [76]	LiDAR	0.702	385	-
PMB [67]	LiDAR	0.707	650	345
BP	LiDAR	0.698	161	250
NEBP (proposed)	LiDAR	0.708	172	271

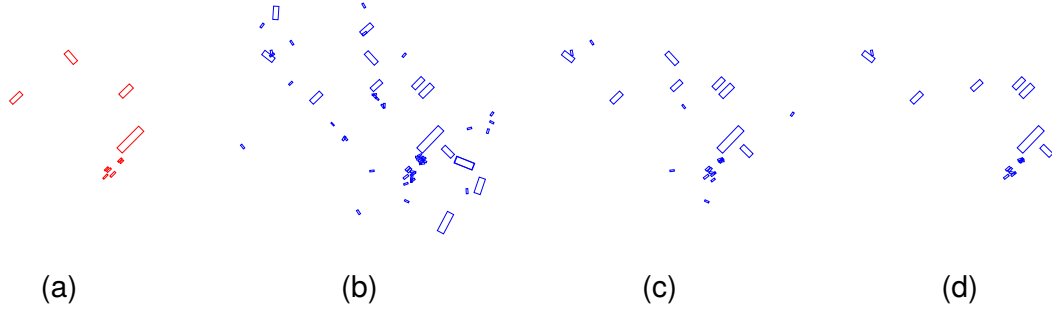


Figure 2.5. Top-down view and single time step for an example autonomous driving scene. Ground truth objects (a) as well as estimated object provided by CenterPointT (b), BP (c), and NEBP (d) are shown. Note that BP and NEBP do not provide object size and orientation estimates. Thus, for each estimated object, the size and orientation of the measurement with the largest association probability are shown.

Table 2.3. GOSPA results: nuScenes validation set

Method	GOSPA ↓	Localization ↓	False ↓	Missed ↓
BP	1.163	0.327	0.127	0.709
NEBP (proposed)	1.140	0.318	0.123	0.699

use of a statistical model to determine the initialization and termination of tracks [18], which is more robust compared to the heuristic track management performed by other reference methods. The improved AMOTA performance of NEBP over BP comes at the cost of a slightly increased IDS and Frag. Qualitative results for a single time step of an example autonomous driving scene are shown in Fig. 2.5. It can be seen that compared to CenterPointT, BP can reduce the number of false objects significantly, while NEBP can reduce the number of false alarms even further. We also report estimation performance results based on the generalized optimal sub-pattern assignment (GOSPA) metric [114] in Table 2.3. This metric can be split up into three components, namely, localization error, false estimated objects, and missed ground truth objects. NEBP outperforms BP in all three components.

We also compare the performance of the proposed NEBP method with BP based on measurements provided by different object detectors. In particular, in addition to measurements provided by the CenterPoint [69] detector, we also consider measurements provided by the

Table 2.4. Results for different detectors: nuScenes validation set

Detector	Method	AMOTA $\uparrow$	IDS $\downarrow$	Frag $\downarrow$
PointPillar [82]	BP	0.360	226	658
	NEBP (proposed)	0.398	270	418
Megvii [83]	BP	0.637	108	262
	NEBP (proposed)	0.651	136	252
CenterPoint [69]	BP	0.698	161	250
	NEBP (proposed)	0.708	172	271

PointPillar [82] and the Megvii [83] detectors. Results based on the nuScenes validation set are shown in Table 2.4. For all three detectors, NEBP can outperform BP in terms of AMOTA, and at the same time maintain a similar number of IDS and Frag. These results indicated that the proposed NEBP method is robust with respect to the chosen object detector.

All experiments were executed on a single Nvidia P100 GPU. For training, eight epochs of the nuScenes training set were performed. The total training time was measured as 30 hours. The inference times of BP and NEBP applied to the nuScenes validation set were measured as 658 seconds and 1137 seconds. These times do not include the execution of the object detector, which yields a runtime of 822 seconds. NEBP has a higher computational complexity compared to BP due to the additional operations discussed in Section 2.5.4.

2.7.5 Ablation Study

In this section, we analyze the contribution of different algorithmic components to the overall performance of our NEBP method. In particular, we analyze the degradations of NEBP performance that are the result of the ablation of specific algorithmic components. All ablation studies are based on the nuScenes validation set, and measurements provided by the CenterPoint detector [69].

We aim to quantify the performance benefits of NEBP resulting from using shape features as an additional input. We consider three NEBP variants: “NEBP-m”, “NEBP-r” and “NEBP-a”. NEBP-m does not make use of shape features. In particular, it initializes the GNN node embeddings only based on motion features, i.e., $\mathbf{h}_{a_i}^{(1)} = [\mathbf{h}_{a_i, \text{motion}}^T \mathbf{0}^T]^T$ and

Table 2.5. Ablation studies: nuScenes validation set

Method	AMOTA $\uparrow$	IDS $\downarrow$	Frag $\downarrow$
BP	0.698	161	250
NEBP-m	0.698	183	228
NEBP-a	0.703	195	259
NEBP-r	0.706	178	227
NEBP-nc	0.703	180	250
NEBP (proposed)	0.708	172	271

$\mathbf{h}_{b_j}^{(1)} = [\mathbf{h}_{b_j, \text{motion}}^T \quad \mathbf{0}^T]^T$. The two variants NEBP-r and NEBP-a are introduced to quantify the benefits of false alarm rejection and object shape association, respectively. More specifically, NEBP-r focuses on false alarm rejection only, i.e., in (2.24) object shape association is deactivated by setting $\mu_i(j) = 0, i \in \{1, \dots, I\}, j \in \{1, \dots, J\}$. Similarly, NEBP-a focuses on object shape association only, i.e., in (2.24) and (2.25) false alarm rejection is deactivated by setting $\omega_j = 1, j \in \{1, \dots, J\}$.

Table 2.5 shows the tracking performance of these NEBP variants, conventional BP, and the proposed NEBP method. It can be seen that NEBP-m cannot achieve any performance improvements compared to BP. This is not surprising since object motion is already modeled accurately by the statistical model and, compared to BP, NEBP-m does not make use of any additional information. On the other hand, both NEBP-a and NEBP-r can achieve an improved AMOTA performance compared to BP. This is because NEBP-a and NEBP-r incorporate additional information in the form of shape features and address the fact that the statistical model used by BP does not accurately model the true data-generating process. In particular, in the statistical model, false alarm measurements are uniformly distributed over the ROI. Furthermore, false alarm measurement and their number are also assumed to be independent and identically distributed across time. However, these assumptions do often not hold in real-world MOT applications such as the considered autonomous driving scenario. This is because physical structures and other reflecting features in the environment can generate so-called persistent false alarm measurements. These false alarm measurements are not uniformly distributed and are not

Table 2.6. AMOTA results for different sigmoid temperatures: nuScenes validation set

Temperature	bicycle	pedestrian	motorcycle	trailer	truck
$T = 0.5$	0.528	0.807	0.737	0.568	0.662
$T = 1$	0.534	0.809	0.727	0.565	0.663
$T = 2$	0.527	0.810	0.716	0.563	0.662
$T = 4$	0.550	0.815	0.727	0.566	0.664
$T = 8$	0.544	0.814	0.739	0.569	0.662
$T = 16$	0.540	0.813	0.730	0.562	0.662
$T = 32$	0.547	0.814	0.736	0.567	0.657
$T = +\infty$	0.550	0.813	0.739	0.559	0.656

independent across time. Thus, they are not accurately represented by the considered statistical model. This model mismatch degrades tracking performance and is addressed by false alarm rejection performed by NEBP and NEBP-r. Object shape association as performed by NEBP and NEBP-a improves data association by using object shape information provided by shape features.

Finally, Table 2.5 also reports performance improvements that result from the calibration process discussed in Section 2.7.3. In particular, “NEBP-nc” is the NEBP variant where no calibration has been performed, i.e., we have $T = 1$ and $\delta = 0$ for training and inference. It can be seen that calibration can significantly improve the performance of NEBP.

The effect of temperature parameters $T > 0$ for five representative object classes is shown in Table 2.6. For each class, the bias δ is fixed to the value provided above. Note that $T = +\infty$, is equivalent to discarding all measurements with $\omega_j < \sigma(\delta)$. It can be seen that NEBP does not always achieve the best AMOTA for $T = +\infty$. In cases where it is difficult to determine whether a measurement is a false alarm, using a temperature $T < +\infty$ can be more robust, as it does not directly discard the measurements with $\omega_j < \sigma(\delta)$, but instead reduces the estimated object score (2.36) of POs that are likely to generate these measurements.

2.8 Acknowledgement

Chapter 2, in part, is a reprint of the material as it appears in M. Liang and F. Meyer, “Neural enhanced belief propagation for multiobject tracking,” *IEEE Transactions on Signal Processing*, 2023 [1], and M. Liang and F. Meyer, “Neural enhanced belief propagation for data association in multiobject tracking,” in *International Conference on Information Fusion (FUSION)*, 2022 [2]. The dissertation author was the primary investigator and author of this material.

Chapter 3

Message Passing for Track-Before-Detect

Accurately tracking an unknown and time-varying number of objects in complex environments is a significant challenge but a fundamental capability in various applications, including applied ocean sciences, surveillance, autonomous driving, and wireless communications. Conventional Bayesian MOT methods typically employ a DTT approach, where a nonlinear frontend detector preprocesses raw sensor data to extract measurements for MOT. While simplifying subsequent processing, the irreversible nature of this preprocessing step can discard valuable object-related information, particularly impairing the ability to resolve weak or closely spaced objects. The TBD paradigm offers an alternative by operating directly on sensor data that has been preprocessed by a linear mapping, preserving information fidelity. However, existing TBD approaches introduce simplifications to facilitate the development of inference methods, such as assuming known signal amplitudes or conditional independence between sensor measurements given object states. These assumptions can lead to suboptimal performance and limit the applicability of the resulting TBD methods in realistic scenarios.

This chapter introduces a novel TBD method that is based on a comprehensive signal model for sensor data. The new measurement model accounts for sensor data correlations and amplitude fluctuation. It also enables the accurate representation of the physics of the data-generating process in the statistical model used for TBD. We further enhance the measurement model by introducing a general multi-snapshot, multi-dictionary formulation that enables multi-

sensor fusion or multi-frequency processing. The proposed model is suitable for a wide range of problems in active and passive radar, active and passive sonar, as well as integrated sensing and communication systems. Based on a factor graph representation of the new measurement model, a scalable BP method is developed to perform efficient Bayesian inference. Experimental results, performed with both synthetic and real data, demonstrate that the proposed method outperforms state-of-the-art conventional MOT methods.

3.1 Chapter Overview

The ability to perceive, interpret, and navigate surroundings has become a fundamental component in autonomous systems, applied ocean science, and wireless networks. MOT [1, 18, 30, 31, 32, 40, 42, 43, 44, 46, 47, 49, 50, 51, 52, 53, 59, 61, 65, 68, 89, 115] is an active area of research dedicated to addressing this challenge. In MOT, the time-varying states of multiple objects are estimated based on measurements from sensing technologies such as radar or sonar. However, accurately and reliably tracking objects in dynamic environments and in the presence of complex object interactions (e.g., occlusions and close proximity) remains challenging. In particular, it is typically unknown when and where objects appear and disappear. This necessitates robust mechanisms for object initialization and termination, also known as tracking management, which adds further complexity to the MOT task.

3.1.1 State-of-the-Art MOT

Conventional methods address the MOT task following a DTT framework that operates in two stages. In the first stage, a nonlinear detector preprocesses the raw sensor data in order to reduce data flow and computational complexity. Second, “point measurements” generated by the object detector are then used as input for tracking [18, 30, 32, 40, 59, 60, 61, 65, 68, 89]. The extracted “point measurements” are prone to measurement origin uncertainty. State-of-the-art DTT-based methods solve the MOT problem in a Bayesian estimation framework, utilizing statistical models to describe object birth, object motion, and measurement generation. Based

on the statistical models, data association can be performed to address measurement-origin uncertainty. The Bayesian inference engine sequentially estimates object states and dynamically introduces new object states in the state space. Prominent examples include joint probabilistic data association filters [59], multi-hypothesis trackers [89], RFS-based filters [32, 40, 65, 68], and BP-based MOT methods [18, 32, 60, 61]. Furthermore, neural networks have recently been integrated to refine the statistical models employed in MOT [1, 116]. However, a significant limitation of DTT-based MOT is its potential for suboptimal performance, stemming from the irreversible loss of valuable information during the initial detection stage.

To address the limitations of DTT, the TBD approach operates directly on received signals that have been preprocessed by a linear mapping. The linear mapping typically transforms the received signals into data cells, e.g., into angle and distance cells, which are then used as measurements for TBD. The potential advantage of the TBD strategy is that it can significantly reduce or completely avoid information loss related to irreversible object detection. Sequential Bayesian estimation techniques for TBD are of particular interest as they can be implemented in real time. Early TBD research focused on the simpler SOT problem, where the task is to determine the existence and state of at most one object [117, 118, 119, 120, 121, 122]. Building on this, subsequent work has extended the TBD paradigm to the more general and challenging MOT problem, which is the focus of this work.

For MOT scenarios, various TBD approaches have been developed. The resulting methods represent object states using either random vectors [42, 43, 44, 49, 53, 80, 115] or RFS [31, 46, 47, 50, 51, 52]. While adopting similar Bayesian frameworks, a fundamental distinction from DTT lies in the statistical measurement model: whereas DTT relates “point measurements” from detectors to object states, TBD establishes a statistical relationship between the sensor data and the object states. However, accurately modeling this relationship for sensor data is challenging, hence specific modeling constraints are often made. For instance, H-PMHT [42, 43] treats the sensor data as if they are point measurements and requires heuristics for track management. Others, like [46, 47], assume that a measurement cannot be affected by more than

one object.

A more physically motivated direction characterizes the raw sensor data using a superpositional signal model [49, 50, 51, 52, 53], which represents signals as a summation of contributions from different objects and noise. Crucially, the measurement likelihood depends not only on object states but also on their amplitudes, which describe the strength of signal contributions. These amplitudes vary between objects and fluctuate over time, significantly complicating the computation of the measurement likelihood. To reduce complexity, existing methods often rely on assumptions that simplify the signal model. Some assume amplitudes are known [49, 50], while others model them as random variables but assume measurements are conditionally independent given the object states and amplitudes [51, 52, 53]. While these assumptions facilitate the development of estimation methods, they represent idealized scenarios. Consequently, either a significant amount of model mismatch is introduced, leading to degraded tracking performance when applied to real data, or the resulting model is fundamentally too simplistic for effective application in real-world environments. Although a more general measurement model accounting for various amplitude fluctuations of complex valued sensor data was introduced in [45], a general MOT method based this model remains unavailable. Recently, [123] develops a variational message passing MOT method without detector for radar signals. It demonstrates superior tracking performance compared with the DTT-based counterparts, underscoring the substantial benefits achievable by leveraging more general signal models.

3.1.2 Contributions and Chapter Organization

In this chapter, we propose a novel Bayesian method for MOT that directly operates on sensor data, performing sequential Bayesian estimation without requiring a frontend object detector. In contrast to existing TBD methods that often rely on simplifying assumptions, our proposed method adopts a comprehensive superpositional signal model of the sensor data, which, for the first time, makes it possible to perform both state estimation and track management for TBD in a fully Bayesian manner, explicitly describing correlations among sensor measurements,

time-varying signal amplitudes, and unknown measurement noise levels. The ability to describe correlations among measurements allows us to make use of the information in the signal phase of measurements. For computationally efficient Bayesian inference, we employ BP based on a factor graph representation of the statistical model.

We further extend the signal model in [45] to accommodate multiple independent measurements, or "snapshots," as well as multiple concurrent signal models, or "dictionaries." This multi-dictionary formulation offers flexibility for multi-sensor fusion [124, 125], multi-frequency beamforming [126, 127], and multi-waveform localization [128]. Within this model, the signal model of a dictionary is a function of the object's kinematic state that contains its motion-related parameters. The amplitudes are modeled as zero-mean Gaussian variables, with variances governed by the object's signal power state. The signal power state consists of a binary existence variable and a continuous power parameter. The resulting Bernoulli-Gaussian model is inspired by the sparse signal reconstruction literature [129, 130, 131], where it is similarly used to describe the individual components of superpositional signals.

At each time step, the variance of the zero-mean additive noise, referred to as noise power state, is also considered a random variable that is inferred together with the kinematic and signal power states of the objects. In the Bayesian setting, the goal of our TBD method is to compute the marginal posterior PDFs of object states. Leveraging the conditional independence among random variables, we represent the new statistical model by a factor graph [12, 13]. The factor graph lays the foundation for developing a scalable BP method that can compute the desired marginal posteriors. BP operates by computing local "messages", which are real-value functions of random variables, along the edges of the factor graph. Messages that cannot be computed in closed form are represented by particles [21, 26] or by means and covariance matrices obtained via moment matching.

The key contributions of this chapter are as follows:

- We introduce a multi-snapshot and multi-dictionary measurement model that considers

correlation among measurements as well as fluctuations of amplitudes.

- We represent the proposed statistical models by a factor graph and develop a scalable BP method for estimating the existence and states of objects.
- We conduct comprehensive numerical experiments on both synthetic and real data and demonstrate state-of-the-art performance.

The organization of this chapter is as follows. Section 3.2 presents the superpositional signal model and the statistical model. Section 3.3 develops the proposed BP method for TBD. Section 3.4 discusses the results of the performed numerical study.

3.2 Signal Model and Problem Formulation

In this section, we introduce the signal model and statistical framework of the considered MOT problem. At each time step k , we assume that there are L_k objects in the environment. Measurements are generated simultaneously by I dictionaries, each producing J snapshots. The individual measurement vector $\mathbf{z}_{k,j}^{(i)} = [z_{k,j,1}^{(i)} \cdots z_{k,j,M}^{(i)}]^T \in \mathbb{C}^M, i \in \{1, \dots, I\}, j \in \{1, \dots, J\}$ is then modeled as

$$\mathbf{z}_{k,j}^{(i)} = \sum_{l=1}^{L_k} \varrho_{k,l,j}^{(i)} \mathbf{a}^{(i)}(\mathbf{x}_{k,l}) + \boldsymbol{\epsilon}_{k,j}^{(i)} \quad (3.1)$$

where $\mathbf{a}^{(i)}(\mathbf{x}_{k,l})$ is the known basis function for the i -th dictionary that encodes a forward model that maps the object's kinematic state $\mathbf{x}_{k,l}$, e.g., position and velocity, to the observed measurements. Multiple dictionaries may arise from distinct sensors [124, 125] or scenarios where objects transmit wideband (i.e. multi-frequency) signals [126, 127] or multiple waveforms [128]. The complex amplitude $\varrho_{k,l,j}^{(i)} \in \mathbb{C}$ quantifies the contribution of object l to snapshot j in dictionary i , while $\boldsymbol{\epsilon}_{k,j}^{(i)} \in \mathbb{C}^M$ is the noise vector. The number of objects L_k , complex amplitudes $\varrho_{k,l,j}^{(i)}$, and kinematics states $\mathbf{x}_{k,l}$ are unknown.

3.2.1 Measurement Model

The goal of MOT is to estimate the time-varying number of objects as well as their kinematic states. To facilitate this estimation, at each time step k , we employ a model that has N_k POs as proposed in [18]. Each PO $n \in \{1, \dots, N_k\}$ is characterized by a kinematic state $\mathbf{x}_{k,n} \in \{1, \dots, N_k\}$ and a binary existence random variable $r_{k,n} \in \{0, 1\}$, where $r_{k,n} = 1$ indicates that the PO corresponds to a physical object. The signal model in (3.1) can thus be reformulated to account for all POs, i.e.,

$$\mathbf{z}_{k,j}^{(i)} = \sum_{n=1}^{N_k} r_{k,n} \rho_{k,n,j}^{(i)} \mathbf{a}_{k,n}^{(i)} + \boldsymbol{\epsilon}_{k,j}^{(i)} \quad (3.2)$$

where $\mathbf{a}_{k,n}^{(i)} \triangleq \mathbf{a}^{(i)}(\mathbf{x}_{k,n})$ represents the contribution vector of PO $n \in \{1, \dots, N_k\}$ at dictionary $i \in \{1, \dots, I\}$. The complex amplitude $\rho_{k,n,j}^{(i)}$ is assumed zero-mean complex Gaussian with a common variance $\gamma_{k,n}^{(i)}$ across snapshots, i.e., $\rho_{k,n,j}^{(i)} \sim \mathcal{CN}(\rho_{k,n,j}^{(i)}; 0, \gamma_{k,n}^{(i)})$, $\forall j \in \{1, \dots, J\}$, corresponding to the *Swerling 1* fluctuation model [45, 132]. The amplitudes are also assumed mutually independent, and are independent of all $\mathbf{x}_{k,n}$, $r_{k,n}$, and $\boldsymbol{\epsilon}_{k,j}^{(i)}$. Note that $r_{k,n} \gamma_{k,n}^{(i)}$ is the “effective” signal power for each PO n on the i -th dictionary, where $r_{k,n} \in \{0, 1\}$ controls the PO’s existence and $\gamma_{k,n}^{(i)}$ scales the strength of its contribution. Hence, we define $\boldsymbol{\phi}_{k,n} = [\boldsymbol{\gamma}_{k,n}^T r_{k,n}]^T$ as the signal power state of PO n with $\boldsymbol{\gamma}_{k,n} = [\gamma_{k,n}^{(1)} \cdots \gamma_{k,n}^{(I)}]^T$. The noise vector $\boldsymbol{\epsilon}_{k,j}^{(i)}$ is modeled as zero-mean complex Gaussian with a common power for all snapshots, i.e., $\boldsymbol{\epsilon}_{k,j}^{(i)} \sim \mathcal{CN}(\boldsymbol{\epsilon}_{k,j}^{(i)}; \mathbf{0}, \eta_k^{(i)} \mathbf{I}_M)$, $\forall j \in \{1, \dots, J\}$. Conditioned on noise power $\eta_k^{(i)}$, the measurement noise are mutually independent across dictionaries and snapshots, and are independent of all $\mathbf{x}_{k,n}$, $r_{k,n}$, and $\rho_{k,n,j}^{(i)}$. Both the signal and noise power, $\gamma_{k,n}^{(i)}$ and $\eta_k^{(i)}$, are unknown and random. Based on the above assumptions, the likelihood function of individual measurement vector $\mathbf{z}_{k,j}^{(i)}$ is zero-mean complex Gaussian, i.e.,

$$f(\mathbf{z}_{k,j}^{(i)} | \mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)}) = \mathcal{CN}(\mathbf{z}_{k,j}^{(i)}; \mathbf{0}, \mathbf{C}_k^{(i)})$$

with covariance given by $\mathbf{C}_k^{(i)} = \sum_{n=1}^{N_k} r_{k,n} \gamma_{k,n}^{(i)} \mathbf{a}_{k,n}^{(i)} \mathbf{a}_{k,n}^{(i)H} + \eta_k^{(i)} \mathbf{I}_M$. We have $\mathbf{x}_k = [\mathbf{x}_{k,1}^T \cdots \mathbf{x}_{k,N_k}^T]^T$ and $\boldsymbol{\phi}_k = [\boldsymbol{\phi}_{k,1}^T \cdots \boldsymbol{\phi}_{k,N_k}^T]^T$. From the independence assumptions, the likelihood of the joint measurement vector can also be obtained as

$$f(\mathbf{z}_k | \mathbf{x}_k, \boldsymbol{\phi}_k, \boldsymbol{\eta}_k) = \prod_{i=1}^I \prod_{j=1}^J f(\mathbf{z}_{k,j}^{(i)} | \mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)}) \quad (3.3)$$

where we have $\mathbf{z}_k = [\mathbf{z}_k^{(1)T} \cdots \mathbf{z}_k^{(I)T}]^T$, $\mathbf{z}_k^{(i)} = [\mathbf{z}_{k,1}^{(i)T} \cdots \mathbf{z}_{k,J}^{(i)T}]^T$, and $\boldsymbol{\eta}_k = [\eta_k^{(1)} \cdots \eta_k^{(I)}]^T$.

Remark: When there is a single snapshot and single dictionary, i.e., $J = 1$ and $I = 1$, our signal model in (3.2) is similar to the one in [53]. However, a key difference is that the model in [53] relies on the assumption that the elements within the measurement vectors, i.e., $z_{k,j,m}^{(i)}, m \in \{1, \dots, M\}$ are independent conditioned on $\mathbf{x}_k, \boldsymbol{\phi}_k$, and $\eta_k^{(i)}$. This common assumption in TBD [51, 52, 115] ignores correlations across measurements and thus results in degraded tracking performance [45]. In practice, sensor data often exhibits correlation among individual measurements. To satisfy the independence assumption, preprocessing steps such as matched filtering or further decorrelation are typically required [51, Sec. 4.2]. In contrast, our model can take correlations across measurements into account, thereby reducing reliance on preprocessing steps.

The multi-dictionary model has been successfully applied in applications such as image reconstruction [133], multimodal classification [134], underwater source localization [135], and angle-of-arrival (AOA) estimation [127]. However, the aforementioned existing methods focus on static scenarios, neglecting state transitions inherent to dynamic tracking problems. Furthermore, existing methods typically model the complex amplitude $\varrho_{k,l,j}^{(i)}$ using Gaussian scale mixture priors, while here we adopted a Bernoulli-Gaussian model for signal amplitude. The Bernoulli-Gaussian model [129, 130, 131] can describe objects that dynamically appear and disappear over time together with time-varying signal strengths.

3.2.2 State-Transition Models

In addition to the measurement model, we also establish state-transition models to describe the dynamics of the kinematic states and the signal power states of POs as well as noise power states. A first-order Markov model is considered, i.e., we assume that conditioned on $\mathbf{x}_{k-1}, \boldsymbol{\phi}_{k-1}, \boldsymbol{\eta}_{k-1}$, the states $\underline{\mathbf{x}}_k, \underline{\boldsymbol{\phi}}_k, \boldsymbol{\eta}_k$ are independent of all previous states $\mathbf{x}_{k'}, \boldsymbol{\phi}_{k'}, \boldsymbol{\eta}_{k'}, k' < k-1$ with $\underline{\mathbf{x}}_k = [\mathbf{x}_{k,1}^T \cdots \mathbf{x}_{k,N_{k-1}}^T]^T$ and $\underline{\boldsymbol{\phi}}_k = [\boldsymbol{\phi}_{k,1}^T \cdots \boldsymbol{\phi}_{k,N_{k-1}}^T]^T$. The joint state-transition process is then governed by the conditional PDF $f(\underline{\mathbf{x}}_k, \underline{\boldsymbol{\phi}}_k, \boldsymbol{\eta}_k | \mathbf{x}_{k-1}, \boldsymbol{\phi}_{k-1}, \boldsymbol{\eta}_{k-1})$. Furthermore, we assume that the joint Markov process consists of multiple independent Markov processes. Specifically, $\mathbf{x}_{k,n}, \boldsymbol{\phi}_{k,n}, \eta_k^{(i)}, n \in \{1, \dots, N_{k-1}\}, i \in \{1, \dots, I\}$ are assumed to evolve independently, i.e., the joint state-transition PDF can be written as the product of individual state-transition PDFs as follows

$$f(\underline{\mathbf{x}}_k, \underline{\boldsymbol{\phi}}_k, \boldsymbol{\eta}_k | \mathbf{x}_{k-1}, \boldsymbol{\phi}_{k-1}, \boldsymbol{\eta}_{k-1}) = \prod_{n=1}^{N_{k-1}} f(\mathbf{x}_{k,n} | \mathbf{x}_{k-1,n}) f(\boldsymbol{\phi}_{k,n} | \boldsymbol{\phi}_{k-1,n}) \left(\prod_{i=1}^I f(\eta_k^{(i)} | \eta_{k-1}^{(i)}) \right).$$

Here, the kinematic state evolution of each PO $n \in \{1, \dots, N_{k-1}\}$ is described by the conditional PDF $f(\mathbf{x}_{k,n} | \mathbf{x}_{k-1,n})$. A common choice for this conditional PDF relies on a constant velocity model [108, Ch. 4]. The conditional PDF $f(\eta_k^{(i)} | \eta_{k-1}^{(i)})$ describes the state-transition of the noise power states. A suitable choice for this conditional PDF is the Gamma PDF [136]. The state-transition of PO existence and signal power are also assumed independent, i.e., $f(\boldsymbol{\phi}_{k,n} | \boldsymbol{\phi}_{k-1,n}) = p(r_{k,n} | r_{k-1,n}) f(\boldsymbol{\gamma}_{k,n} | \boldsymbol{\gamma}_{k-1,n})$. For the existence of PO $n \in \{1, \dots, N_{k-1}\}$, if it does not exist in the previous time step $k-1$, i.e., $r_{k-1,n} = 0$, then it also does not exist at time k , i.e., $p(r_{k,n} = 1 | r_{k-1,n} = 0) = 0$. If it exists at time $k-1$, i.e., $r_{k-1,n} = 1$, then it continues to exist at time k with survival probability p_s , i.e., $p(r_{k,n} = 1 | r_{k-1,n} = 1) = p_s$.

3.2.3 Object Birth Models

Newly appearing objects are modeled by a Poisson point process with mean μ_B and spatial PDF $f_B(\mathbf{x}, \boldsymbol{\gamma}) = f_B(\mathbf{x}) f_B(\boldsymbol{\gamma})$, where we assume the kinematic states and signal powers

of new POs are independent a priori. To systematically introduce new POs across the ROI $\mathcal{X}$ with $\mathbf{x} \in \mathcal{X}$, we partition the ROI into Q non-overlapping regions denoted by $\mathcal{X}_q$, $q \in \{1, \dots, Q\}$ with $\uplus_{q=1}^Q \mathcal{X}_q = \mathcal{X}$. Each subregion $\mathcal{X}_q$ spawns one new PO, resulting in $N_k = N_{k-1} + Q$ total POs at time k . The birth process within each $\mathcal{X}_q$ is also a Poisson point process with mean $\mu_{B,n} = \mu_B \int_{\mathcal{X}_q} f_B(\mathbf{x}) d\mathbf{x}$, $n = N_{k-1} + q$. Its spatial PDF is given by

$$f_{B,n}(\mathbf{x}) \propto \begin{cases} f_B(\mathbf{x}), & \mathbf{x} \in \mathcal{X}_q \\ 0, & \mathbf{x} \notin \mathcal{X}_q \end{cases}$$

where $\propto$ indicates equality up to a multiplicative constant.

When the region covered by each $\mathcal{X}_q$ is sufficiently small, we can assume that there is at most one new object in $\mathcal{X}_q$. Based on the Poisson probability mass function (PMF), the probabilities that $\mathcal{X}_q$ contains zero or one new objects are $e^{-\mu_{B,n}}$ and $\mu_{B,n}e^{-\mu_{B,n}}$, $n = N_{k-1} + q$. We thus define the existence probability of the corresponding new PO as

$$p_{B,n} = \frac{\mu_{B,n}e^{-\mu_{B,n}}}{\mu_{B,n}e^{-\mu_{B,n}} + e^{-\mu_{B,n}}} = \frac{\mu_{B,n}}{\mu_{B,n} + 1}.$$

As a result, the PDF of new PO $n \in \{N_{k-1} + 1, \dots, N_k\}$ is $f(\mathbf{x}_{k,n}, \boldsymbol{\phi}_{k,n}) = f(\mathbf{x}_{k,n})f(\boldsymbol{\phi}_{k,n})$ with

$$f(\mathbf{x}_{k,n}) = f_{B,n}(\mathbf{x}_{k,n})$$

$$f(\boldsymbol{\phi}_{k,n}) = p_B(r_{k,n})f_B(\boldsymbol{\gamma}_{k,n})$$

where $p_B(r_{k,n} = 1) = p_{B,n}$. The kinematic and signal power state $\mathbf{x}_{k,n}, \boldsymbol{\phi}_{k,n}$, $n \in \{N_{k-1} + 1, \dots, N_k\}$ of a new PO are independent of all current and previous states of other POs. The joint prior PDF of new POs can then be written as the product of prior PDFs of individual new POs, i.e., $f(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\phi}}_k) = \prod_{n=N_{k-1}+1}^{N_k} f(\mathbf{x}_{k,n}, \boldsymbol{\phi}_{k,n})$ with $\bar{\mathbf{x}}_k = [\mathbf{x}_{k,N_{k-1}+1}^T \cdots \mathbf{x}_{k,N_k}^T]^T$ and $\bar{\boldsymbol{\phi}}_k = [\boldsymbol{\phi}_{k,N_{k-1}+1}^T \cdots \boldsymbol{\phi}_{k,N_k}^T]^T$.

At time $k = 0$, the prior PDF $f(\mathbf{x}_{0,n})$, $f(\boldsymbol{\phi}_{0,n})$, $n \in \{1, \dots, N_0\}$, and $f(\eta_0^{(i)})$, $i \in \{1, \dots, I\}$ are assumed known. The random variables $\mathbf{x}_{0,n}$, $\boldsymbol{\phi}_{0,n}$, $n \in \{1, \dots, N_0\}$ and $\eta_0^{(i)}$, $i \in \{1, \dots, I\}$ are all independent of each other.

3.2.4 Joint Posterior PDF and Factor Graph

In the Bayesian setting, estimating the number of objects reduces to the computation of existence probabilities $p(r_{k,n}|\mathbf{z}_{1:k})$ that involve all the measurements $\mathbf{z}_{1:k} = [\mathbf{z}_1^T \cdots \mathbf{z}_k^T]^T$ up to time k . Object declarations are made by thresholding existence probabilities based on a threshold T_{dec} , i.e., a PO n is declared to exist if $p(r_{k,n}|\mathbf{z}_{1:k}) > T_{\text{dec}}$. The number of POs increases over time. To manage computational complexity, we remove a PO from the state space if its existence probability is lower than a threshold T_{pru} . To compute an estimate of the kinematic state estimation, we make use of the MMSE estimator given by

$$\hat{\mathbf{x}}_{k,n} = \int \mathbf{x}_{k,n} f(\mathbf{x}_{k,n}|\mathbf{z}_{1:k}) d\mathbf{x}_{k,n}.$$

Computing an MMSE estimate requires knowledge of the marginal posterior PDF $f(\mathbf{x}_{k,n}|\mathbf{z}_{1:k})$.

Provided the conditional independence assumptions in Section 3.2.1 to 3.2.3, the joint posterior PDF can be factorized as

$$\begin{aligned} f(\mathbf{x}_{0:k}, \boldsymbol{\phi}_{0:k}, \boldsymbol{\eta}_{0:k}|\mathbf{z}_{1:k}) &\propto \prod_{n=1}^{N_0} f(\mathbf{x}_{0,n}) f(\boldsymbol{\phi}_{0,n}) \left(\prod_{i=1}^I f(\eta_0^{(i)}) \right) \prod_{k'=1}^k \left(\prod_{i=1}^I f(\eta_{k'}^{(i)}|\eta_{k'-1}^{(i)}) \right) \\ &\times \prod_{n=1}^{N_{k'-1}} f(\mathbf{x}_{k',n}|\mathbf{x}_{k'-1,n}) f(\boldsymbol{\phi}_{k',n}|\boldsymbol{\phi}_{k'-1,n}) \prod_{n=N_{k'-1}+1}^{N_{k'}} f(\mathbf{x}_{k,n}) \\ &\times f(\boldsymbol{\phi}_{k,n}) \left(\prod_{i=1}^I \prod_{j=1}^J f(\mathbf{z}_{k,j}^{(i)}|\mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)}) \right) \end{aligned} \quad (3.4)$$

where we have $\mathbf{x}_{0:k} = [\mathbf{x}_0^T \cdots \mathbf{x}_k^T]^T$, $\boldsymbol{\phi}_{0:k} = [\boldsymbol{\phi}_0^T \cdots \boldsymbol{\phi}_k^T]^T$, and $\boldsymbol{\eta}_{0:k} = [\boldsymbol{\eta}_0^T \cdots \boldsymbol{\eta}_k^T]^T$. With the factorization (3.4), the joint posterior PDF can be represented by a factor graph [12, 13, 14]. This

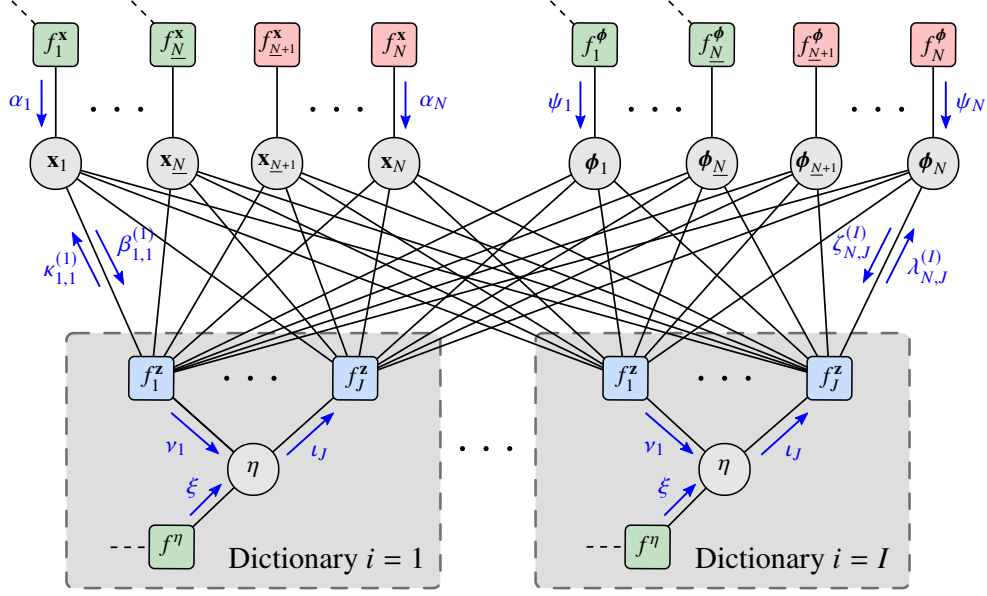


Figure 3.1. Factor graph representing the joint posterior PDF in (3.4) and the BP messages for a single time step k . The time index k is omitted. BP messages are shown as blue arrows. The following shorthand notations are used: $\underline{N} = N_{k-1}$, $N = N_k$, $\mathbf{x}_n = \mathbf{x}_{k,n}$, $\boldsymbol{\phi}_n = \boldsymbol{\phi}_{k,n}$, $\eta = \eta_k^{(i)}$, $f_j^z = f(\mathbf{z}_{k,j}^{(i)} | \mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)})$, $f_n^x = f(\mathbf{x}_{k,n} | \mathbf{x}_{k-1,n})$, $f_n^\phi = f(\boldsymbol{\phi}_{k,n} | \boldsymbol{\phi}_{k-1,n})$ for $n \in \{1, \dots, N_{k-1}\}$, $f_n^x = f(\mathbf{x}_{k,n})$, $f_n^\phi = f(\boldsymbol{\phi}_{k,n})$ for $n \in \{N_{k-1} + 1, \dots, N_k\}$, $\alpha_n = \alpha(\mathbf{x}_{k,n})$, $\psi_n = \psi(\boldsymbol{\phi}_{k,n})$, $\xi = \xi(\eta_k^{(i)})$, $\beta_{n,j}^{(i)} = \beta_j^{(i,t)}(\mathbf{x}_{k,n})$, $\zeta_{n,j}^{(i)} = \zeta_j^{(i,t)}(\boldsymbol{\phi}_{k,n})$, $\iota_j = \iota_j^{(t)}(\eta_k^{(i)})$, $\kappa_{n,j}^{(i)} = \kappa_j^{(i,t)}(\mathbf{x}_{k,n}; \mathbf{z}_{k,j}^{(i)})$, $\lambda_{n,j}^{(i)} = \lambda_j^{(i,t)}(\boldsymbol{\phi}_{k,n}; \mathbf{z}_{k,j}^{(i)})$, and $\nu_j = \nu_j^{(t)}(\eta_k^{(i)}; \mathbf{z}_{k,j}^{(i)})$.

factor graph comprises factor nodes encapsulating statistical models introduced in Sections 3.2.1 to 3.2.3, and variable nodes representing random variables, including the kinematic state, signal power state of POs, and the noise power states. Edges in the graph connect factor nodes and variable nodes, describing their statistical dependencies. The factor graph that represents (3.4), is shown in Fig. 3.1. Based on this factor graph, we develop the proposed Bayesian BP method [12, 13, 14], which enables the efficient computation of marginal posterior PDFs for object declaration and state estimation.

3.3 Proposed Method

In this section, we introduce the proposed BP method [12, 13, 14] for TBD. Based on the statistical model introduced in Section 3.2.4, we aim to compute the existence probabilities

$p(r_{k,n}|\mathbf{z}_{1:k})$ and the marginal posterior PDFs $f(\mathbf{x}_{k,n}|\mathbf{z}_{1:k})$ for each PO $n \in \{1, \dots, N_k\}$. However, directly marginalizing the joint posterior $f(\mathbf{x}_{0:k}, \boldsymbol{\phi}_{0:k}, \boldsymbol{\eta}_{0:k}|\mathbf{z}_{1:k})$ is computationally infeasible due to high-dimensional integrations over kinematic states, signal power states, and noise power states. Instead, we make use of BP. BP exploits the conditional independence structure represented by the factor graph for efficient and scalable computation of existence probabilities and marginal posterior PDFs. BP operates by computing so-called “messages” that are real-value functions of the random variables over the edges of the factor graph, propagating probabilistic information between variable and factor nodes. When the factor graph is a tree, existence probabilities and marginal posterior PDFs provided by BP, called “beliefs”, are the same as the true existence probabilities and marginal posterior PDFs. If the factor graph has loops, as in our TBD problem, beliefs serve as approximations of the true marginal posterior PDFs and theoretical performance guarantees are typically unavailable. Nonetheless, the approximation performed by BP is generally accurate. Loopy BP has been applied in a wide range of applications [16, 17, 18, 19].

The computation order of BP messages is flexible on loopy factor graphs. Depending on the chosen order, the resulting beliefs may vary. In this work, we adopt the following message passing order: (i) messages are only sent forward in time, (ii) within each time step, messages of the signal power states, noise power states, as well as kinematic states are computed sequentially and iteratively, (iii) messages in each dictionary are computed in parallel. BP messages involved in our considered MOT problem are shown in Fig. 3.1 and can be computed using standard sum-product rules [12, 13, 14]. We detail the computation of the messages in the rest of the section.

3.3.1 Prediction and Birth Messages

At each time step k , the prediction messages are computed based on the beliefs from the previous time step $k - 1$ and the state-transition model. For legacy PO $n \in \{1, \dots, N_{k-1}\}$, the

prediction messages of the kinematic states sent from “ $f(\mathbf{x}_{k,n}|\mathbf{x}_{k-1,n})$ ” to “ $\mathbf{x}_{k,n}$ ” are given by

$$\alpha(\mathbf{x}_{k,n}) = \int f(\mathbf{x}_{k,n}|\mathbf{x}_{k-1,n}) \tilde{f}(\mathbf{x}_{k-1,n}) d\mathbf{x}_{k-1,n}$$

and the prediction messages of the signal power states sent from “ $f(\boldsymbol{\phi}_{k,n}|\boldsymbol{\phi}_{k-1,n})$ ” to “ $\boldsymbol{\phi}_{k,n}$ ” can be expressed as

$$\psi(\boldsymbol{\phi}_{k,n}) = \sum_{r_{k-1,n} \in \{0,1\}} \int f(\boldsymbol{\gamma}_{k,n}, r_{k,n} | \boldsymbol{\gamma}_{k-1,n}, r_{k-1,n}) \tilde{f}(\boldsymbol{\gamma}_{k-1,n}, r_{k-1,n}) d\boldsymbol{\gamma}_{k-1,n} \quad (3.5)$$

where $\tilde{f}(\mathbf{x}_{k-1,n})$ and $\tilde{f}(\boldsymbol{\phi}_{k-1,n}) = \tilde{f}(\boldsymbol{\gamma}_{k-1,n}, r_{k-1,n})$ are the beliefs of the kinematic and signal power state at time $k-1$, respectively. Specifically, inserting the state-transition model defined in Section 3.2.2 into (3.5), we obtain

$$\psi(\boldsymbol{\gamma}_{k,n}, 1) = \int p_s f(\boldsymbol{\gamma}_{k,n} | \boldsymbol{\gamma}_{k-1,n}) \tilde{f}(\boldsymbol{\gamma}_{k-1,n}, 1) d\boldsymbol{\gamma}_{k-1,n}$$

and $\psi(\boldsymbol{\gamma}_{k,n}, 0) = \psi_{k,n} f_D(\boldsymbol{\gamma}_{k,n})$. Here, the constant $\psi_{k,n}$ is given by

$$\psi_{k,n} = \int (1 - p_s) \tilde{f}(\boldsymbol{\gamma}_{k-1,n}, 1) + \tilde{f}(\boldsymbol{\gamma}_{k-1,n}, 0) d\boldsymbol{\gamma}_{k-1,n}$$

and $f_D(\boldsymbol{\gamma}_{k,n})$ is a “dummy” PDF. Since it is “marginalized out” later, the functional form of the dummy PDF has no effect on the final solution of the BP method [18].

POs with indices $n \in \{N_{k-1} + 1, \dots, N_k\}$ are newly introduced at time k . The birth messages of their kinematic and signal power states are the ones sent from “ $f(\mathbf{x}_{k,n})$ ” to “ $\mathbf{x}_{k,n}$ ”, and from “ $f(\boldsymbol{\phi}_{k,n})$ ” to “ $\boldsymbol{\phi}_{k,n}$ ”, respectively. Since $f(\mathbf{x}_{k,n})$ and $f(\boldsymbol{\phi}_{k,n})$ are singleton factors, the corresponding messages are the factors itself, i.e., for $n \in \{N_{k-1} + 1, \dots, N_k\}$ we have

$$\alpha(\mathbf{x}_{k,n}) = f(\mathbf{x}_{k,n}) \quad \psi(\boldsymbol{\phi}_{k,n}) = f(\boldsymbol{\phi}_{k,n})$$

where $f(\mathbf{x}_{k,n})$ and $f(\boldsymbol{\phi}_{k,n})$ have been introduced in Section 3.2.3.

Similarly, for each dictionary $i \in \{1, \dots, I\}$, the prediction messages for noise power states from “ $f(\eta_k^{(i)}|\eta_{k-1}^{(i)})$ ” to “ $\eta_k^{(i)}$ ” are computed as

$$\xi(\eta_k^{(i)}) = \int f(\eta_k^{(i)}|\eta_{k-1}^{(i)}) \tilde{f}(\eta_{k-1}^{(i)}) d\eta_{k-1}^{(i)}$$

where $\tilde{f}(\eta_{k-1}^{(i)})$ is the belief of the noise power states at time $k-1$.

3.3.2 Iterative Measurement Update Messages

After the prediction messages are computed, iterative message passing is performed to incorporate the information from measurements $\mathbf{z}_k$. At iteration $t \in \{1, \dots, T\}$, the messages that pass from the variable nodes “ $\mathbf{x}_{k,n}, \phi_{k,n}, \eta_k^{(i)}$ ” to the factor nodes “ $f(\mathbf{z}_{k,j}^{(i)}|\mathbf{x}_k, \phi_k, \eta_k^{(i)})$ ” $n \in \{1, \dots, N_k\}, j \in \{1, \dots, J\}, i \in \{1, \dots, I\}$ are given as follows.

Measurement Update Signal Power State – Incoming

$$\zeta_j^{(i,t)}(\phi_{k,n}) = \psi(\phi_{k,n}) \prod_{i'=1}^I \prod_{\substack{j'=1 \\ (i',j') \neq (i,j)}}^J \lambda_{j'}^{(i',t)}(\phi_{k,n}; \mathbf{z}_{k,j'}^{(i')}). \quad (3.6)$$

Measurement Update Noise Power State – Incoming

$$\iota_j^{(t)}(\eta_k^{(i)}) = \xi(\eta_k^{(i)}) \prod_{\substack{j'=1 \\ j' \neq j}}^J \nu_{j'}^{(t)}(\eta_k^{(i)}; \mathbf{z}_{k,j'}^{(i)}). \quad (3.7)$$

Measurement Update Kinematic State – Incoming

$$\beta_j^{(i,t)}(\mathbf{x}_{k,n}) = \alpha(\mathbf{x}_{k,n}) \prod_{i'=1}^I \prod_{\substack{j'=1 \\ (i',j') \neq (i,j)}}^J \kappa_{j'}^{(i',t)}(\mathbf{x}_{k,n}; \mathbf{z}_{k,j'}^{(i')}). \quad (3.8)$$

The messages sent from factor node “ $f(\mathbf{z}_{k,j}^{(i)}|\mathbf{x}_k, \phi_k, \eta_k^{(i)})$ ” to the variable nodes are $\lambda_j^{(i,t)}(\phi_{k,n}; \mathbf{z}_{k,j}^{(i)})$, $\nu_j^{(t)}(\eta_k^{(i)}; \mathbf{z}_{k,j}^{(i)})$, and $\kappa_j^{(i,t)}(\mathbf{x}_{k,n}; \mathbf{z}_{k,j}^{(i)})$. These messages are presented below.

Measurement Update Signal Power State – Outgoing

$$\begin{aligned}
\lambda_j^{(i,t)}(\boldsymbol{\phi}_{k,n}; \mathbf{z}_{k,j}^{(i)}) &= \sum_{r_{k,1} \in \{0,1\}} \cdots \sum_{r_{k,n-1} \in \{0,1\}} \sum_{r_{k,n+1} \in \{0,1\}} \cdots \sum_{r_{k,N_k} \in \{0,1\}} \\
&\int \cdots \int f(\mathbf{z}_{k,j}^{(i)} | \mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)}) \iota_j^{(t-1)}(\eta_k^{(i)}) d\eta_k^{(i)} \\
&\times \prod_{\substack{n'=1 \\ n' \neq n}}^{N_k} \zeta_j^{(i,t-1)}(\boldsymbol{\phi}_{k,n'}) d\boldsymbol{\gamma}_{k,1} \cdots d\boldsymbol{\gamma}_{k,n-1} d\boldsymbol{\gamma}_{k,n+1} \cdots d\boldsymbol{\gamma}_{k,N_k} \\
&\times \prod_{n'=1}^{N_k} \beta_j^{(i,t-1)}(\mathbf{x}_{k,n'}) d\mathbf{x}_{k,1} \cdots d\mathbf{x}_{k,N_k}.
\end{aligned} \tag{3.9}$$

Measurement Update Noise Power State – Outgoing

$$\begin{aligned}
\nu_j^{(t)}(\eta_k^{(i)}; \mathbf{z}_{k,j}^{(i)}) &= \sum_{r_{k,1} \in \{0,1\}} \cdots \sum_{r_{k,N_k} \in \{0,1\}} \int \cdots \int f(\mathbf{z}_{k,j}^{(i)} | \mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)}) \\
&\times \prod_{n'=1}^{N_k} \beta_j^{(i,t-1)}(\mathbf{x}_{k,n'}) d\mathbf{x}_{k,1} \cdots d\mathbf{x}_{k,N_k} \\
&\times \prod_{n'=1}^{N_k} \zeta_j^{(i,t)}(\boldsymbol{\phi}_{k,n'}) d\boldsymbol{\gamma}_{k,1} \cdots d\boldsymbol{\gamma}_{k,N_k}.
\end{aligned} \tag{3.10}$$

Measurement Update Kinematic State – Outgoing

$$\begin{aligned}
\kappa_j^{(i,t)}(\mathbf{x}_{k,n}; \mathbf{z}_{k,j}^{(i)}) &= \sum_{r_{k,1} \in \{0,1\}} \cdots \sum_{r_{k,N_k} \in \{0,1\}} \int \cdots \int f(\mathbf{z}_{k,j}^{(i)} | \mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)}) \iota_j^{(t)}(\eta_k^{(i)}) d\eta_k^{(i)} \\
&\times \prod_{\substack{n'=1 \\ n' \neq n}}^{N_k} \beta_j^{(i,t-1)}(\mathbf{x}_{k,n'}) d\mathbf{x}_{k,1} \cdots d\mathbf{x}_{k,n-1} d\mathbf{x}_{k,n+1} \cdots d\mathbf{x}_{k,N_k} \\
&\times \prod_{n'=1}^{N_k} \zeta_j^{(i,t)}(\boldsymbol{\phi}_{k,n'}) d\boldsymbol{\gamma}_{k,1} \cdots d\boldsymbol{\gamma}_{k,N_k}.
\end{aligned} \tag{3.11}$$

Note that the messages in (3.9)-(3.11) depend on fixed sensor measurements $\mathbf{z}_{k,j}^{(i)}$, which is indicated by the semicolon notation “ $(\cdot; \mathbf{z}_{k,j}^{(i)})$ ”. The messages in (3.9)-(3.11) contain information related to measurements $\mathbf{z}_{k,j}^{(i)}$, while also taking most recent statistical information for the random signal power states, noise power states, and kinematic states into account. This most recent statistical information is given by (3.6)-(3.8). For example, the computation of $v_j^{(t)}(\eta_k^{(i)}; \mathbf{z}_{k,j}^{(i)})$ includes the likelihood function $f(\mathbf{z}_{k,j}^{(i)} | \mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)})$ of measurement $\mathbf{z}_{k,j}^{(i)}$, the product $\prod_{n'=1}^{N_k} \beta_j^{(i,t-1)}(\mathbf{x}_{k,n'})$ involving all kinematic states, and the product $\prod_{n'=1}^{N_k} \zeta_j^{(i,t)}(\boldsymbol{\phi}_{k,n'})$ involving on all signal power states.

The iteration is initialized by $\zeta_j^{(i,0)}(\boldsymbol{\phi}_{k,n}) = \psi(\boldsymbol{\phi}_{k,n})$, $\iota_j^{(0)}(\eta_k^{(i)}) = \xi(\eta_k^{(i)})$, and $\beta_j^{(i,0)}(\mathbf{x}_{k,n}) = \alpha(\mathbf{x}_{k,n})$. As defined in (3.9)-(3.11), within each iteration, messages are computed in the following order and sequentially, i.e., $\lambda_j^{(i,t)}(\boldsymbol{\phi}_{k,n}; \mathbf{z}_{k,j}^{(i)})$, $\zeta_j^{(i,t)}(\boldsymbol{\phi}_{k,n})$, $v_j^{(t)}(\eta_k^{(i)}; \mathbf{z}_{k,j}^{(i)})$, $\iota_j^{(t)}(\eta_k^{(i)})$, $\kappa_j^{(i,t)}(\mathbf{x}_{k,n}; \mathbf{z}_{k,j}^{(i)})$, and $\beta_j^{(i,t)}(\mathbf{x}_{k,n})$.

3.3.3 Message Approximation and Belief Calculation

Messages in (3.9)-(3.11) require the marginalization of all kinematic and signal power states. This marginalization cannot be performed in closed form. Numerical integration would lead to a computational complexity which grows exponentially with the number of POs N_k . Since $f(\mathbf{z}_{k,j}^{(i)} | \mathbf{x}_k, \boldsymbol{\phi}_k, \eta_k^{(i)})$ is zero-mean complex Gaussian with respect to $\mathbf{z}_{k,j}^{(i)}$, all three messages in (3.9)-(3.11) are a Gaussian mixture with an uncountably infinite number of components with respect to $\mathbf{z}_{k,j}^{(i)}$. All components have a zero mean. To ensure the scalability of the proposed method with respect to N_k , we employ moment matching to approximate these Gaussian mixtures by a single Gaussian component. The approximated messages read

$$\begin{aligned}\tilde{\lambda}_j^{(i,t)}(\boldsymbol{\phi}_{k,n}; \mathbf{z}_{k,j}^{(i)}) &= \mathcal{CN}(\mathbf{z}_{k,j}^{(i)}; \mathbf{0}, \mathbf{C}_{\lambda,k,n,j}^{(i,t)}) \\ \tilde{v}_j^{(t)}(\eta_k^{(i)}; \mathbf{z}_{k,j}^{(i)}) &= \mathcal{CN}(\mathbf{z}_{k,j}^{(i)}; \mathbf{0}, \mathbf{C}_{v,k,n,j}^{(i,t)}) \\ \tilde{\kappa}_j^{(i,t)}(\mathbf{x}_{k,n}; \mathbf{z}_{k,j}^{(i)}) &= \mathcal{CN}(\mathbf{z}_{k,j}^{(i)}; \mathbf{0}, \mathbf{C}_{\kappa,k,n,j}^{(i,t)})\end{aligned}$$

where the covariances $\mathbf{C}_{\lambda,k,j}^{(i,t)}$, $\mathbf{C}_{\nu,k,j}^{(i,t)}$, and $\mathbf{C}_{\kappa,k,j}^{(i,t)}$ preserve the second-order statistics of their respective original Gaussian mixture PDFs. Based on derivations previously presented in [5, 50, 53], covariance matrices can be computed as

$$\mathbf{C}_{\lambda,k,n,j}^{(i,t)} = r_{k,n} \gamma_{k,n}^{(i)} \boldsymbol{\Sigma}_{k,n,j}^{(i,t)} + \sum_{\substack{n'=1 \\ n' \neq n}}^{N_k} \hat{\gamma}_{k,n',j}^{(i,t)} \boldsymbol{\Sigma}_{k,n',j}^{(i,t)} + \hat{\eta}_{k,j}^{(i,t)} \mathbf{I}_M \quad (3.12)$$

$$\mathbf{C}_{\nu,k,n,j}^{(i,t)} = \sum_{n'=1}^{N_k} \hat{\gamma}_{k,n',j}^{(i,t)} \boldsymbol{\Sigma}_{k,n',j}^{(i,t)} + \eta_k^{(i)} \mathbf{I}_M \quad (3.13)$$

$$\mathbf{C}_{\kappa,k,n,j}^{(i,t)} = \hat{\gamma}_{k,n,j}^{(i,t)} \mathbf{a}_{k,n} \mathbf{a}_{k,n}^H + \sum_{\substack{n'=1 \\ n' \neq n}}^{N_k} \hat{\gamma}_{k,n',j}^{(i,t)} \boldsymbol{\Sigma}_{k,n',j}^{(i,t)} + \hat{\eta}_{k,j}^{(i,t)} \mathbf{I}_M \quad (3.14)$$

where

$$\hat{\gamma}_{k,n,j}^{(i,t)} = \sum_{r_{k,n} \in \{0,1\}} \int \gamma_{k,n}^{(i)} \zeta_j^{(i,t)}(\boldsymbol{\phi}_{k,n}) d\boldsymbol{\gamma}_{k,n} \quad (3.15)$$

$$\boldsymbol{\Sigma}_{k,n,j}^{(i,t)} = \int \mathbf{a}_{k,n} \mathbf{a}_{k,n}^H \beta_j^{(i,t)}(\mathbf{x}_{k,n}) d\mathbf{x}_{k,n} \quad (3.16)$$

$$\hat{\eta}_{k,j}^{(i,t)} = b_\eta \cdot \int \eta_k^{(i)} \iota_j^{(t)}(\eta_k^{(i)}) d\eta_k^{(i)}. \quad (3.17)$$

Here, b_η is the noise amplification factor. When $b_\eta = 1$, the matrices $\mathbf{C}_{\lambda,k,j}^{(i,t)}$, $\mathbf{C}_{\nu,k,j}^{(i,t)}$, and $\mathbf{C}_{\kappa,k,j}^{(i,t)}$ align with the covariances of the original messages. Setting $b_\eta > 1$ can improve robustness against model mismatch, e.g., when measurement noise $\boldsymbol{\epsilon}_{k,j}^{(i)}$ is colored.

After T iterations, the beliefs for each random variable can be obtained by multiplying

all the incoming messages at the corresponding variable node [12], i.e.,

$$\begin{aligned}\tilde{f}(\mathbf{x}_{k,n}) &\propto \alpha(\mathbf{x}_{k,n}) \prod_{i=1}^I \prod_{j=1}^J \tilde{\kappa}_j^{(i,T)}(\mathbf{x}_{k,n}; \mathbf{z}_{k,j}^{(i)}) \\ \tilde{f}(\boldsymbol{\phi}_{k,n}) &\propto \psi(\boldsymbol{\phi}_{k,n}) \prod_{i=1}^I \prod_{j=1}^J \tilde{\lambda}_j^{(i,T)}(\boldsymbol{\phi}_{k,n}; \mathbf{z}_{k,j}^{(i)}) \\ \tilde{f}(\eta_k^{(i)}) &\propto \xi(\eta_k^{(i)}) \prod_{j=1}^J \tilde{\nu}_j^{(T)}(\eta_k^{(i)}; \mathbf{z}_{k,j}^{(i)}).\end{aligned}$$

Beliefs are approximations of the corresponding marginal posterior PDFs, i.e., $\tilde{f}(\mathbf{x}_{k,n}) \approx f(\mathbf{x}_{k,n}|\mathbf{z}_{1:k})$ can then be used for state estimation as detailed in Section 3.2.4 and for inference in the subsequent time steps.

The integration in (3.15)-(3.16) is done individually for each random variable, i.e., a joint integration as in (3.9)-(3.11) can be avoided. The fact that integration is performed individually results in a computational complexity that scales linearly with the number of POs. Evaluating the Gaussian functions $\tilde{\lambda}_j^{(i,t)}(\boldsymbol{\phi}_{k,n}; \mathbf{z}_{k,j}^{(i)})$, $\tilde{\nu}_j^{(t)}(\eta_k^{(i)}; \mathbf{z}_{k,j}^{(i)})$, and $\tilde{\kappa}_j^{(i,t)}(\mathbf{x}_{k,n}; \mathbf{z}_{k,j}^{(i)})$ results in a computational complexity that scales as $O(M^3)$. Since there are $2IJN_k + IJ$ such messages, the overall complexity scales as $O(M^3IJN_k)$, i.e., linearly with respect to the number of POs, number of dictionaries, and number of snapshots. The expressions in (3.15)-(3.16) are computed by means of Monte Carlo integration, i.e., by means of random samples or particles.

3.4 Experimental Results

In this section, we evaluate the performance of our proposed method based on both synthetic and real data. In the considered scenario, the entries in the measurement vectors $\mathbf{z}_{k,j}^{(i)} = [z_{k,j,1}^{(i)} \cdots z_{k,j,M}^{(i)}]^T \in \mathbb{C}^M, i \in \{1, \dots, I\}, j \in \{1, \dots, J\}$ are strongly correlated across measurement index m , which makes conventional multiobject TBD methods such as [31, 42, 43, 44, 46, 47, 49, 50, 51, 52, 53, 115, 118] unsuitable.

3.4.1 Synthetic Radar Tracking Scenario

We consider a two-dimensional (2-D) multi-sensor radar system with I sensors equipped with a uniform linear array. Sensors are located at positions $\mathbf{p}^{(i)}, i \in \{1, \dots, I\}$. A linear chirp signal is transmitted and reflections from objects are received. This type of system is referred to as linear frequency modulated continuous wave (LFMCW) radar [137, 138]. At each time step k , the radar system demodulates, downconverts, and discretizes the analog received signal to obtain the baseband signal $\mathbf{z}_{k,j}^{(i)}$ in the time domain [137, Sec. 9.2.3]. The measurement vectors $\mathbf{z}_{k,j}^{(i)}$ adhere to the signal model in (3.1) through the position-dependent basis function, i.e.,

$$\mathbf{a}^{(i)}(\mathbf{p}) = \text{vec}(\mathbf{a}_d^{(i)}(\mathbf{p}) \mathbf{a}_\theta^{(i)\text{T}}(\mathbf{p})) \quad (3.18)$$

with delay and angular steering vectors defined as

$$\begin{aligned} \mathbf{a}_d^{(i)}(\mathbf{p}) &= [1 \quad e^{j4\pi\mu T_s d^{(i)}/c} \dots e^{j4\pi(M_d-1)\mu T_s d^{(i)}/c}]^T \\ \mathbf{a}_\theta^{(i)}(\mathbf{p}) &= [1 \quad e^{j\pi \sin(\theta^{(i)})} \dots e^{j\pi(M_\theta-1) \sin(\theta^{(i)})}]^T. \end{aligned}$$

Here $d^{(i)} = \|\mathbf{p} - \mathbf{p}^{(i)}\|$ denotes the distance between object at position $\mathbf{p}$ and sensor i , while $\theta^{(i)} = \angle(\mathbf{p} - \mathbf{p}^{(i)}) - \varphi^{(i)}$ is the AOA relative to the array orientation $\varphi^{(i)}$. The orientation of both arrays is $\varphi^{(i)} = \frac{\pi}{4}$. The parameters c , μ , and T_s represent the speed of light, chirp rate, and sampling interval, respectively. The total measurement size is $M = M_d M_\theta$, combining M_d delay samples and M_θ array elements, with $\text{vec}(\cdot)$ being the vectorization operation. Note that $\mathbf{a}_d^{(i)}(\mathbf{p})$ is a complex narrowband signal $e^{j\omega^{(i)}\tau}$ with distance-dependent frequency $\omega^{(i)} = 2\pi(2\mu d^{(i)}/c)$ sampled at times $\tau \in \{0, T_s, \dots, (M_d - 1)T_s\}$.

For data generation, the power of the complex amplitude $\varrho^{(i)}$ is characterized with the radar-range equation

$$|\varrho^{(i)}|^2 = \frac{c^2}{f_c^2 (4\pi)^3 (d^{(i)})^4} \quad (3.19)$$

where $f_c = 77$ GHz is the carrier frequency and its phase is uniformly distributed over $[0, 2\pi)$.

In this experiment, there are $I = 2$ radar sensors at positions $\mathbf{p}^{(1)} = [0 \ 0]^T$ and $\mathbf{p}^{(2)} = [50 \ 50]^T$. We set the chirp rate $\mu = 8\text{MHz}/\mu\text{s}$, the sampling interval $T_s = 0.2\mu\text{s}$, the number of samples in time domain $M_d = 20$, and the number of array elements $M_\theta = 10$. The kinematic state $\mathbf{x}_{k,n} = [\mathbf{p}_{k,n}^T \ \mathbf{v}_{k,n}^T]^T$ of PO $n \in \{1, \dots, N_k\}$ contains its 2-D position $\mathbf{p}_{k,n}$ and 2-D velocity $\mathbf{v}_{k,n}$. A constant velocity model $\mathbf{x}_{k,n} = \mathbf{F}\mathbf{x}_{k-1,n} + \mathbf{W}\mathbf{q}_k$ [108, Ch. 4] is employed to describe the dynamics of the kinematic states, where $\mathbf{q}_k \in \mathbb{R}^2$ is the zero-mean Gaussian driving noise with covariance $10^{-4}\mathbf{I}_2$. The state-transition of the signal power for each dictionary is assumed to be independent, i.e., $f(\gamma_{k,n}|\gamma_{k-1,n}) = \prod_{i=1}^I f(\gamma_{k,n}^{(i)}|\gamma_{k-1,n}^{(i)})$, and the individual state-transition PDFs are given by $f(\gamma_{k,n}^{(i)}|\gamma_{k-1,n}^{(i)}) = \mathcal{G}(\gamma_{k,n}^{(i)}; \gamma_{k-1,n}^{(i)}/c_\gamma, c_\gamma)$ with $c_\gamma = 10^3$. Here, $\mathcal{G}(\cdot; a, b)$ is the Gamma PDF with the shape parameter a and the scale parameter b . The surveillance region spans $[0, 50] \times [0, 50]$, defining a ROI defined as $\mathcal{X} = [0, 50] \times [0, 50] \times \mathbb{R}^2$. The spatial PDF of the object birth model is given by $f_B(\mathbf{x}) = f_B(\mathbf{p})f_B(\mathbf{v})$ with $f_B(\mathbf{p})$ uniformly distributed over the surveillance region and $f_B(\mathbf{v})$ being zero-mean Gaussian with covariance $0.25\mathbf{I}_2$. The expected number of new objects is set to $\mu_B = 10^{-6}$. As discussed in Section 3.2.3, we discretize the surveillance region using a step size of 1 m in both dimensions, leading to $Q = 2500$ non-overlapping regions $\mathcal{X}_q, q \in \{1, \dots, Q\}$ for the initialization of new POs. The signal power of new POs is uniformly distributed from 0 to 1 and independent across different sensors. Note that our normalization of the data implies that the signal power of POs cannot exceed one. The dynamics of the noise power states are modeled by a Gamma PDF $f(\eta_k^{(i)}|\eta_{k-1}^{(i)}) = \mathcal{G}(\eta_k^{(i)}; \eta_{k-1}^{(i)}/c_\eta, c_\eta)$ with $c_\eta = 100$. The noise amplification factor in (3.17) is set to $b_\eta = 1$ as there is no model mismatch. At the initial time step $k = 0$, the prior of noise powers $\eta_0^{(i)}, i \in \{1, 2\}$ are uniformly distributed over $[0, 10^{-3}]$ and there is no prior information about the objects, i.e., $N_0 = 0$. We set the survival probability to $p_s = 0.9$, the declaration threshold to $T_{\text{dec}} = 0.5$, and the pruning threshold to $T_{\text{pru}} = 10^{-2}$.

We simulate with 80 time steps, i.e., $k \in \{1, \dots, 80\}$ and five objects. Two of the objects appear at time $k = 10$ and disappear at time $k = 70$. The two objects are initialized at positions

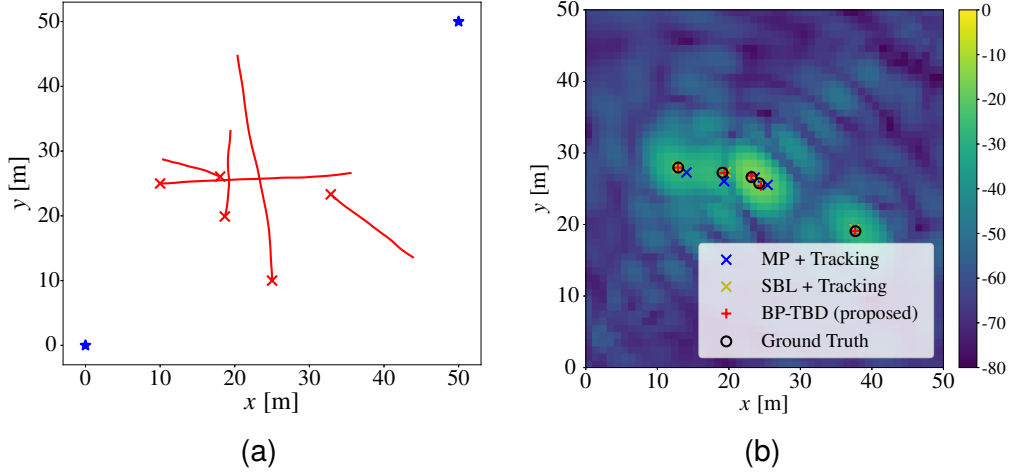
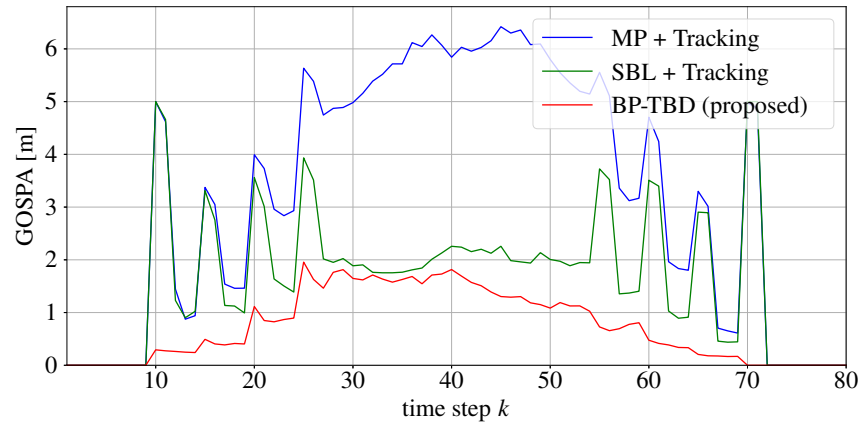


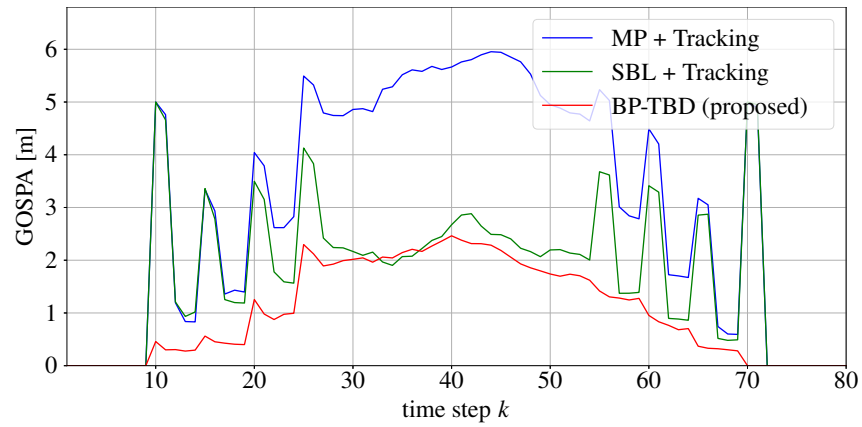
Figure 3.2. Visualization of the synthetic radar tracking scenario in an example simulation run: (a) Trajectories of the objects with their initial positions marked as red crosses and sensor positions marked as blue stars. (b) Tracking results of different methods at time step $k = 41$. An SNR of -3 dB was considered. The Bartlett spectrum obtained by matched filtering is shown as a background.

$[10 \ 25]^T$ and $[25 \ 10]^T$ with velocities $[0.5 \ 0]^T$ and $[0 \ 0.5]^T$, respectively, creating intersecting trajectories at the surveillance region center. The other three objects appear at $k = 15, 20$, and 25 , and disappear at $k = 55, 60$, and 65 , respectively. Their initial positions are uniformly distributed over $[15, 35] \times [15, 35]$ and velocities are drawn from a zero-mean Gaussian PDF with covariance $0.04\mathbf{I}_2$. Objects also disappear if they leave the surveillance region. We conduct 100 simulation runs. Fig. 3.2a shows representative trajectories. The measurements $\mathbf{z}_{k,j}^{(i)}$ are generated based on (3.1), (3.18), and (3.19) for $J = 3$ snapshots at each time step. For data generation, noise power states are set to $\eta_k^{(i)} = 0.5 \times 10^{-14}$, 1×10^{-14} , and 2×10^{-14} , corresponding to the “input” signal-to-noise ratios (SNRs) $|\varrho_{k,l}^{(i)}|^2 / \eta_k^{(i)} = -6, -3$, and 0 dB for objects at the surveillance region center $[25 \ 25]^T$.

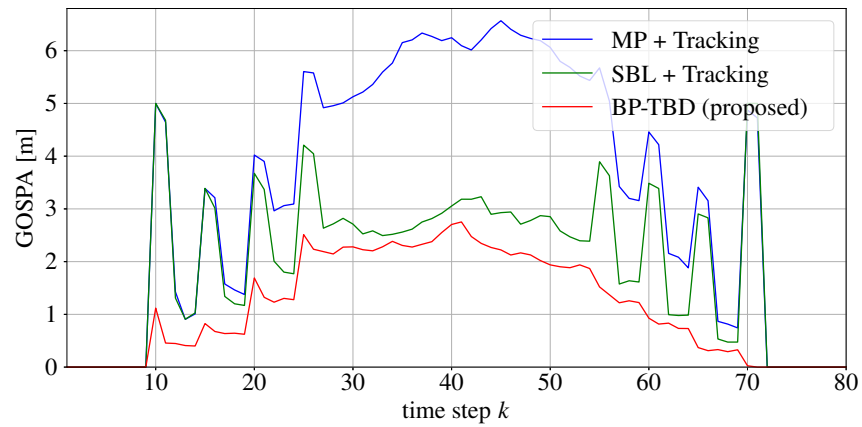
We employ methods based on the DTT strategy, comprising frontend detectors paired with a conventional tracking method as a reference. For object detection, we utilize two distinct detectors: matching pursuit (MP) [139] and sparse Bayesian learning (SBL) [127, 135]. These detectors provide 2-D position measurements to the BP-based MOT method [18]. We refer to the two variants as “MP + Tracking” and “SBL + Tracking”, respectively. The pairing of



(a)



(b)



(c)

Figure 3.3. GOSPA error of different tracking methods for synthetic radar tracking averaged over 100 simulation runs at (a) 0 dB, (b) -3 dB, and (c) -6 dB.

SBL with detect-then-track MOT in [18], was first introduced in [80]. Our proposed “BP-TBD” method is implemented using particle-based BP [21], with 30,000 particles for kinematic states, 1000 particles for signal power states, and 300 particles for measurement noise variances. Conventional BP-based tracking [18] is also implemented using particle-based BP [21] with 30,000 particles. To further reduce the complexity, instead of initializing new POs for all grid cells, we only initialize new POs on cells with a MP detection. As a result, BP-TBD has an equal number of POs at each time step as MP + Tracking. The tracking performance is evaluated using the GOSPA [114] metric with cutoff parameter $c = 5$ and order $p = 1$. The GOSPA, averaged over the 100 simulation runs, is shown in Fig. 3.3. Among all methods, BP-TBD achieves the lowest GOSPA for different noise levels. The advantage stems from the ability of the proposed statistical model to accurately characterize the data generation process. The qualitative tracking result for -3 dB SNR at a single time step is shown in Fig. 3.2b. Here, the background is the “Bartlett” spectrum, i.e., a average of the output of matched filtering defined as $\prod_{i=1}^I \sum_{j=1}^J \|(\mathbf{a}^{(i)}(\mathbf{p}))^H \mathbf{z}_{k,j}^{(i)}\|_2^2$.

While DTT-based methods fail to estimate closely spaced objects due to information loss in the detection stage, BP-TBD can estimate them accurately by directly processing raw measurements. We further show the noise variance $\hat{\eta}_k^{(i)} = \int \eta_k^{(i)} \tilde{f}(\eta_k^{(i)}) d\eta_k^{(i)}$ estimated by the proposed method in Fig. 3.4. After a few initial time steps, our method accurately estimates the noise level of sensor measurements.

3.4.2 High-SNR Passive Acoustic Scenario

In this section, we evaluate the proposed method using real data in a shallow-water underwater scenario. A 20-minute segment from a passive acoustic dataset [126] is considered. A mobile source transmits seven tonals at frequencies $f_i \in \{49, 79, 112, 148, 201, 283, 388\}$ Hz simultaneously. Signals are recorded by a vertical linear array (VLA) with $M = 21$ elements at a known position. The seven tones define $I = 7$ distinct dictionaries, each corresponding to one tonal transmission. This set of tones is referred to as the “high tonal set”. The measurements

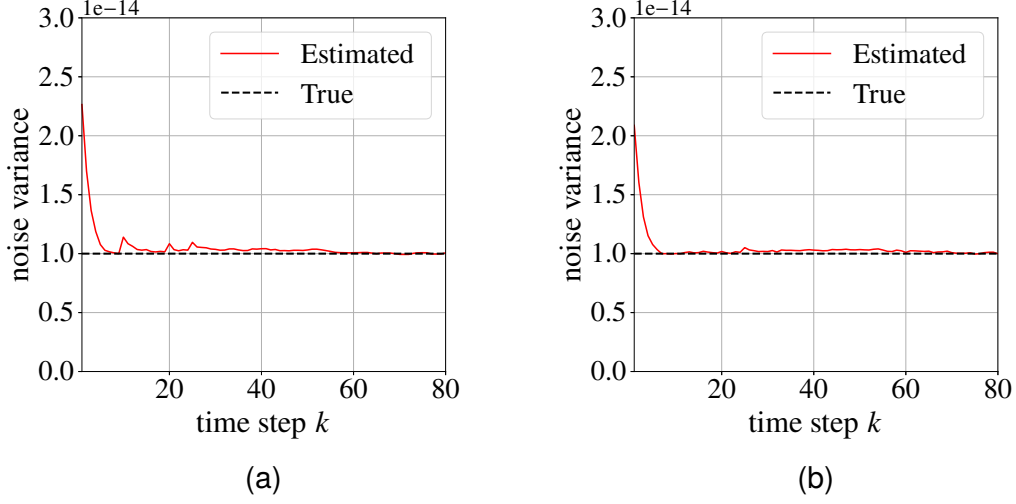


Figure 3.4. Estimated and true noise variance $\eta_k^{(i)}$ averaged over 100 simulations runs for (a) the sensor at $[0\ 0]^T$ and (b) the sensor at $[50\ 50]^T$. An SNR of -3 dB was considered.

$\mathbf{z}_{k,j}^{(i)}$ are extracted from an 8192-point short-time Fourier transform (STFT) with 75% overlap applied to the received acoustic signals. In particular, frequency bins for $i \in \{1, \dots, I\}$ are extracted by taking the Doppler shift into account. Doppler compensation is performed by selecting the maximum-power bin among the ± 5 discrete Fourier transform (DFT) bins adjacent to the nominal DFT bin. We include $J = 3$ snapshots per time step, resulting in a total of 292 time steps. The duration of each time step is 4.09 seconds. We make use of a range-independent geoacoustic model [140] of the considered shallow water waveguide [126]. Here, $\mathbf{a}^{(i)}(x_r, x_d) \in \mathbb{C}^M$ corresponds to the complex wavefield of a source at frequency f_i , located at a range of x_r , and a depth of x_d , received by the M -element VLA. This wavefield of the underwater environment was computed based on the range-independent geoacoustic model and by using the KRAKEN normal mode program [141]. In particular, we use KRAKEN to precompute the propagating modes of the range-independent acoustic waveguide and then, for each relevant (x_r, x_d) value, synthesize the wavefield from modes during runtime. The mirage effect in shallow water [142] as well as an array tilt of 1° has been taken into account.

To illustrate that our proposed method is capable of tracking multiple objects, we create a two-source scenario artificially. In particular, a copy of the measurement at the last time step,

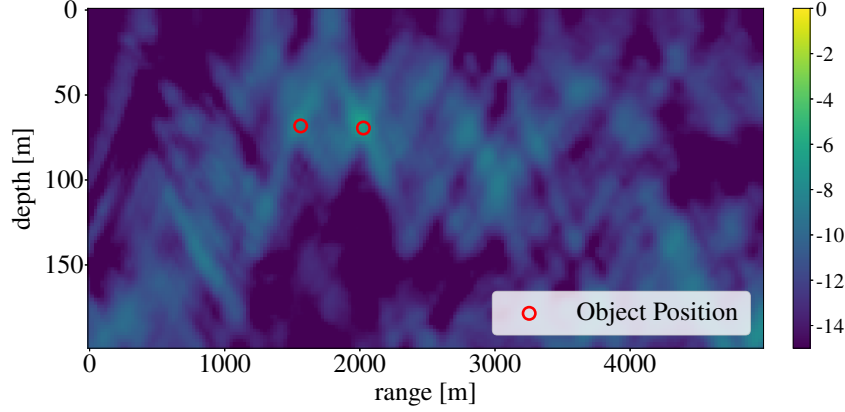


Figure 3.5. Bartlett spectrum of high tonal set in dB at time $k = 246$. The red circles are the true positions of the two sources.

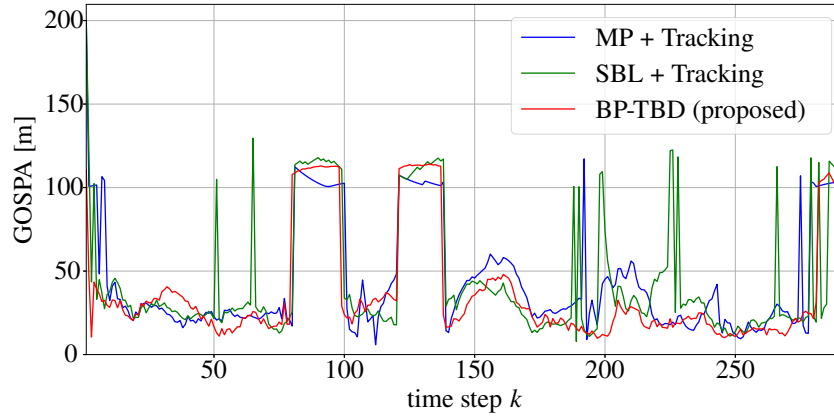


Figure 3.6. GOSPA error in the high-SNR passive acoustic scenario. Different tracking methods based on the high tonal set are considered.

multiplied by a random phase and a scaling factor of 2, is added to the original measurements [135] in frequency domain. This creates a scenario with one static and one moving source transmitting at the same frequencies, where the power of the moving source is 3 dB weaker than the static source. At times step k , the goal is to estimate the number of sources (two in this case) as well as their ranges from the VLA and their depths using measurements $\mathbf{z}_{1:k}$. The Bartlett spectrum of the measurement, i.e., the output of matched filtering $\frac{1}{IJ} \prod_{i=1}^I \sum_{j=1}^J \|(\mathbf{a}^{(i)}(x_r, x_d))^H \mathbf{z}_{k,j}\|_2^2$, at time $k = 246$, is shown in Fig. 3.5.

The kinematic state $\mathbf{x}_{k,n} \in \mathbb{R}^4$ of PO $n \in \{1, \dots, N_k\}$ consists of its range $x_{r,k,n}$, depth $x_{d,k,n}$, range rate $v_{r,k,n}$, and depth rate $v_{d,k,n}$. A constant rate model $\mathbf{x}_{k,n} = \mathbf{F}\mathbf{x}_{k-1,n} +$

$\mathbf{W}\mathbf{q}_k$ [108, Ch. 4] is employed to describe the dynamics of the kinematic states, where $\mathbf{q}_k \in \mathbb{R}^2$ is the zero-mean Gaussian driving noise with covariance $\text{diag}(10^{-2}, 10^{-5})$. The state-transition of the signal power for each dictionary is assumed to be independent, i.e., $f(\gamma_{k,n}|\gamma_{k-1,n}) = \prod_{i=1}^I f(\gamma_{k,n}^{(i)}|\gamma_{k-1,n}^{(i)})$, and the individual state-transition PDFs are given by a Gamma PDF $f(\gamma_{k,n}^{(i)}|\gamma_{k-1,n}^{(i)}) = \mathcal{G}(\gamma_{k,n}^{(i)}; \gamma_{k-1,n}^{(i)}/c_\gamma, c_\gamma)$ with $c_\gamma = 10^4$. The ROI is defined as $\mathcal{X} = [0, 5000] \times [0, 200] \times \mathbb{R}^2$, corresponding to a maximum range of 5000 m and depth of 200 m. The expected number of new objects is set to $\mu_B = 10^{-4}$, with a spatial PDF for range and depth that is uniform on $[0, 5000] \times [0, 200]$. The range rate and depth rate are modeled as independent zero-mean Gaussian PDFs, with standard deviations of 4m/s and 1m/s, respectively. The disjoint sets $\mathcal{X}_q, q \in \{1, \dots, Q\}$ are constructed by a range and depth discretization with intervals 25 m and 2 m, resulting in a 200-by-100 grid over the ROI. Our normalization of the data implies that the signal power of POs cannot exceed one. Consequently, the signal power of new POs is assumed to be uniformly distributed from 0 to 1 and is independent across tones. The dynamics of the measurement noise variance is modeled by a Gamma PDF $f(\eta_k^{(i)}|\eta_{k-1}^{(i)}) = \mathcal{G}(\eta_k^{(i)}; \eta_{k-1}^{(i)}/c_\eta, c_\eta)$ with $c_\eta = 10^2$. The noise amplification factor in (3.17) is set to $b_\eta = 10$. The initial prior of noise variances $\eta_0^{(i)}, i \in \{1, \dots, I\}$ is uniform on $[0, 2 \times 10^{-4}]$, and we set $N_0 = 0$. Particle-based BP [21] is used for the implementation of the proposed method, using 10,000 particles for kinematic states, 1000 particles each for signal power states and measurement noise variances. In addition, we set the survival probability to $p_s = 0.95$, the declaration threshold to $T_{\text{dec}} = 0.5$, and the pruning threshold to $T_{\text{pru}} = 10^{-2}$.

Consistent with our synthetic experiment, we adopt DTT as our baseline, with MP [139] and SBL [127, 135] as object detectors combined with BP-based MOT [18], termed MP + Tracking and SBL + Tracking, respectively. Also here, due to strongly correlated measurements, conventional TBD methods such as [31, 42, 43, 44, 46, 47, 49, 50, 51, 52, 53, 115, 117, 118, 119, 120, 121, 122] are unsuitable for the considered scenario.

The object detectors provide range and depth measurements. BP-based MOT [18] is implemented using 30,000 particles. For the proposed BP-TBD method, we only initialize new

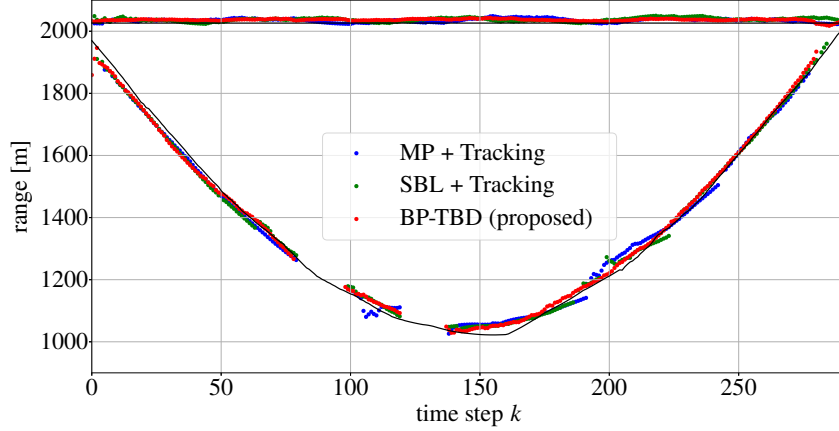


Figure 3.7. Range estimates of different methods in the high-SNR passive acoustic scenario. The true range of the two sources is indicated by black lines.

POs for grid cells with an MP detection to reduce computational complexity. The tracking performance is evaluated using the GOSPA [114] metric with cutoff parameter $c = 200$ and order $p = 1$. Our results presented in Fig. 3.6 demonstrate that the proposed method achieves the lowest GOSPA. A visualization of tracking results is shown in Fig. 3.7. The proposed method shows superior performance compared to the two-step reference methods, especially around $k = 200$. The primary reason for the superior performance is that the conventional two-step approach can suffer from poor detector performance due to side lobes and model mismatch. The proposed method can improve performance by avoiding a potentially error-prone detector. Note that the GOSPA error peaks around time steps 90 and 140 are related to the time steps where sources stop transmitting the aforementioned tones.

3.4.3 Low-SNR Passive Acoustic Scenario

We further extend our experiment to a more challenging low-SNR scenario. A 10-minute segment is extracted from the same dataset as in Section 3.4.2. The considered tonal signals are at frequencies $f_i \in \{52, 82, 115, 151, 204, 286, 391\}$ Hz. Compared to the “high tonal set” used in Section 3.4.2, the power of the “low tonal set” is reduced by 26 dB. The low tonal set is too quiet for tracking based on the model used for the high tonal set in Section 3.4.2. To increase SNR, we adopt the range-coherent model in [143] that relies on a virtual-array formulation.

In particular, N_{syn} measurement vectors are stacked to form a super measurement vector, i.e., $\mathbf{z}_{\text{syn},k,j}^{(i)} = [\mathbf{z}_{k_1,j}^{(i)\text{T}} \cdots \mathbf{z}_{k_{N_{\text{syn}}},j}^{(i)\text{T}}]^{\text{T}}$. Let there be $k_1, \dots, k_{N_{\text{syn}}}$ time steps with a duration of τ_{syn} . For a source with a range of x_r at the initial time step, a constant range rate v_r , and a constant depth x_d , the synthetic array response becomes

$$\mathbf{a}_{\text{syn}}^{(i)}(x_r, x_d, v_r) = \begin{bmatrix} \mathbf{a}^{(i)}(x_r, x_d) \\ \mathbf{a}^{(i)}(x_r + v_r \tau_{\text{syn}}, x_d) \\ \vdots \\ \mathbf{a}^{(i)}(x_r + (N_{\text{syn}} - 1)v_r \tau_{\text{syn}}, x_d) \end{bmatrix}$$

which generates a planar synthetic array with the source at the endfire. Note that $\mathbf{a}^{(i)}(x_r, x_d) \in \mathbb{C}^M$ is the wavefield model discussed in Sec. 3.4.2. This modeling approach effectively generates a virtual planar array that increases SNR via coherent integration. The stacked measurement and contribution vector are used in the measurement model (3.2) throughout the rest of the section. The segment that we consider for the experiment contains 45 time steps with a time interval of 10.92 seconds. There are $J = 4$ snapshots at each time step, and the number of subarrays in the planar synthetic array is $N_{\text{syn}} = 5$.

The statistical model and parameter setup are similar to those used for the high tonal set discussed in Section 3.4.2. We set the power and noise state-transition parameters to $c_\gamma = 1000$ and $c_\eta = 10$. The noise amplification factor is set to $b_\eta = 5$. The expected number of new objects is set to $\mu_B = 10^{-4}$. The initial prior of noise variances $\eta_0^{(i)}, i \in \{1, \dots, I\}$ are uniform over $[0, 6 \times 10^{-3}]$ and $N_0 = 0$. There are 3000 particles for kinematic states, 1000 particles for signal power states and 300 particles for measurement noise variances. Other parameters remain unchanged compared to Sec. 3.4.2.

To extract measurements from the received acoustic signals, we first compute the 16384-point STFT spectrum with 50% overlap. Since the SNR in the low tonal set is too low, we are not able to obtain $\mathbf{z}_{k,j}^{(i)}$ by choosing DFT bins based on power as in the high SNR case

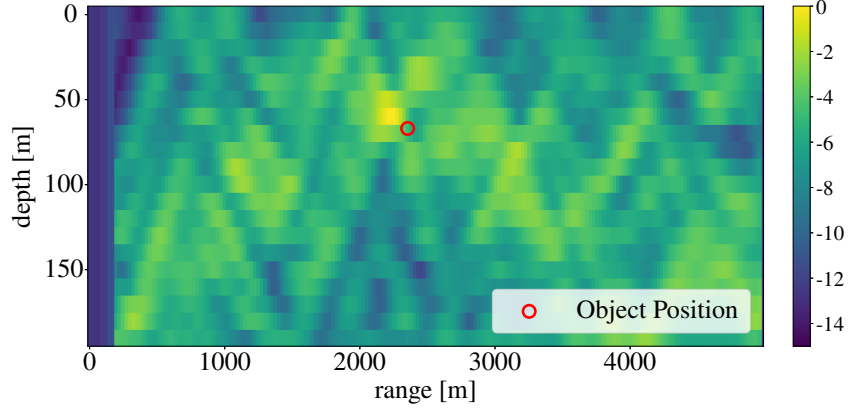


Figure 3.8. Bartlett spectrum in dB at time $k = 1$ in the low-SNR passive acoustic scenario with virtual array processing. The red circle is the true position of the source.

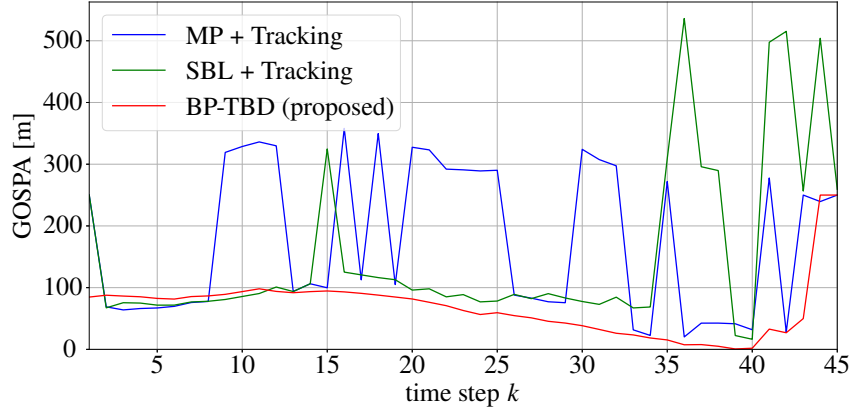


Figure 3.9. GOSPA error of different tracking methods in the low-SNR passive acoustic scenario.

(cf. Section 3.4.2). Instead, we propose selecting bins for each PO separately based on their estimated range rate. For PO $n \in \{1, \dots, N_k\}$, we first compute its estimated range rate $\hat{v}_{r,k-1,n}$ at the previous time step from its belief $\tilde{f}(\mathbf{x}_{k-1,n})$ of the kinematic state. Considering the Doppler effect, the observed frequency of the tones are now $(1 + \hat{v}_{r,k-1,n}/c)f_i$ where c is the sound speed, which can then be used for selecting the corresponding DFT bins. This frequency-adjusted bin selection is applied independently to each PO, meaning that POs may use different measurement vectors for evaluating BP messages. When computing the covariance matrices in (3.12) and (3.14) for PO n , we set $\Sigma_{k,n',j}^{(i,t)}$ to zero if PO $n' \neq n$ selects a different DFT bin on tone $i \in \{1, \dots, I\}$.

The tracking performance is again evaluated using the GOSPA [114] metric with cutoff parameter $c = 500$ and order $p = 1$. The two reference methods, MP + Tracking and SBL + Tracking, are identical to those used in the high tonal set. To reduce computational complexity, the proposed BP-TBD method only initializes new POs for grid cells with a MP detection. The ROI is defined as $\mathcal{X} = [0, 5000] \times [0, 200] \times [-3, -1] \times \mathbb{R}$. The spatial PDF is uniform on the ROI for range, depth, and range rate, respectively, while the depth rate is standard normal distributed. The disjoint sets $\mathcal{X}_q, q \in \{1, \dots, Q\}$ are constructed by discretizing range, depth, and range rate with intervals 25 m, 10 m, and 0.1 m/s, resulting in a $200 \times 20 \times 20$ voxel grid covering the ROI. Results are shown in Fig. 3.9. Our proposed method achieves the lowest GOSPA. It can accurately identify that there is exactly one object, while the reference methods often introduce false objects. Note that all methods fail at the end of the data segment, where the object does not have a constant range rate. The variable range rate introduces a significant mismatch with the range-coherent model.

3.5 Acknowledgment

Chapter 3, in part, is a reprint of the material as it appears in M. Liang and F. Meyer, “Message passing for track-before-detect,” which was submitted to *IEEE Transactions on Signal Processing* in 2025, the preprint of which is available on arXiv [3], and M. Liang and F. Meyer, “An approach of directly tracking of multiple objects,” in *Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2024 [4]. The dissertation author was the primary investigator and author of this material.

Chapter 4

Direct Multipath-Based SLAM

Radio-based localization approaches that make use of reflections in the propagation environment to improve the accuracy and robustness of location estimates have a variety of potential applications in future wireless communication networks. Multipath-based SLAM aims at estimating the position of agents and reflecting features in the environment by exploiting the relationship between the local geometry and MPCs in received radio signals. Existing multipath-based SLAM methods preprocess received radio signals using a channel estimator. The channel estimator lowers the data rate by extracting a set of dispersion parameters for each MPC. These parameters are then used as measurements for SLAM. Bayesian estimation for multipath-based SLAM is facilitated by the lower data rate. However, due to finite resolution capabilities limited by signal bandwidth, channel estimation is prone to errors and MPC parameters may be extracted incorrectly and lead to a reduced SLAM performance. We propose a multipath-based SLAM approach that directly uses received radio signals as inputs. A new statistical model that can effectively be represented by a factor graph is introduced. The factor graph is the starting point for the development of an efficient BP method for multipath-based SLAM that avoids data preprocessing by a channel estimator. Numerical results based on synthetic and real data in challenging single-input, single-output (SISO) scenarios demonstrate that the proposed method outperforms conventional methods in terms of localization and mapping accuracy.

4.1 Chapter Overview

SLAM [19, 54, 55, 57, 144, 145, 146, 147, 148, 149] aims at estimating the position together with a unknown map of the environment based on noisy measurements provided by sensors hosted on mobile platforms. SLAM is a key signal processing task in a variety of applications, including land robotics, autonomous driving, underwater navigation, and wireless communication. In particular, in wireless communication, SLAM can be performed by exploiting multipath propagation of radio signals in urban or indoor scenarios. The resulting knowledge of the agent position and the propagation environment can then be used for, e.g., precise beam tracing or effective resource allocation [150, 151].

4.1.1 Multipath-Based SLAM and State of the Art

In wireless networks, localization in scenarios with multiple propagation paths is typically performed using ultra-wideband or mmWave radios. Their relatively high bandwidth makes it possible to resolve MPCs.

Early work on radio-based indoor localization aims at mitigating multipath effects [152]. On the contrary, multipath-based SLAM exploits multiple propagation by inferring the relation between the geometry of the propagation environment and the MPCs in received radio signals. Existing multipath-based SLAM methods can be categorized as feature-based SLAM [153, 154, 155, 156]. Here, the map of the environment is represented by an unknown number of individual *map features* with unknown positions. In received radio signals, MPCs generated by LOS path and specular reflections at flat surfaces often dominate. The map features are referred to as PAs and VAs, where PAs are base stations and VAs are the mirror images of PAs at the reflecting surfaces. The number and position of VAs are unknown. The signals resulting from a transmission from mobile agent to PA (or vice versa) provide information on the agent's position, the PA, and the VAs associated with it. Consequently, the goal of many multipath-based SLAM approaches is to estimate the positions of the mobile agents together with the number of

aforementioned features and their positions. In addition to features related to specular reflections, some multipath-based SLAM methods also aim to estimate the position of point scatterer [57] or the parameters of a probability distribution characterizing diffuse multipath [149].

Existing multipath-based SLAM methods include methods based on extended Kalman filtering [54], Rao-Blackwellized particle filtering [55, 57, 145], sequential estimation of random finite sets [146, 147], and BP [157, 158, 159]. All existing approaches perform data processing in two stages. In stage one, a channel estimator is applied to the received radio signal. The channel estimator detects MPCs and extracts their dispersion parameters, e.g., delay, angle of arrival, angle of departure, or Doppler frequency [130, 160, 161, 162, 163, 164, 165, 166]. In stage two, a sequential Bayesian SLAM engine estimates the mobile agent position and the number and position of map features. The extracted dispersion parameters of MPCs are prone to measurement origin uncertainty. In particular, (i) it is not known which propagation path generated which extracted MPC (ii) certain propagation paths are missed in the sense that no MPC is extracted, and (iii) some MPCs that do not correspond to any propagation path may be erroneously extracted.

The SLAM engine sequentially estimates agent positions and map features representing propagation paths, addresses measurement-origin uncertainty by performing data association [35, 41, 59], and dynamically introduces new map features in the state space. The use of a channel estimator as a preprocessing stage is widely used as it reduces the data flow and thus leads to SLAM engines that have reduced computational complexity due to simplified models. For example, a key assumption in data association is that each extracted MPC can be associated with at most one map feature. On the other hand, the use of a channel estimator can lead to a significant loss of information and thus impede the robustness and accuracy of SLAM. For example, in single-input, single-output (SISO) scenarios, where only delays, i.e., range measurements, can be extracted from the received radio signal, it is often common that multiple VAs have similar ranges to the agent. In such challenging geometries, the channel estimator may not be able to resolve the MPC delays and thus extract a single range “merged” measurement for

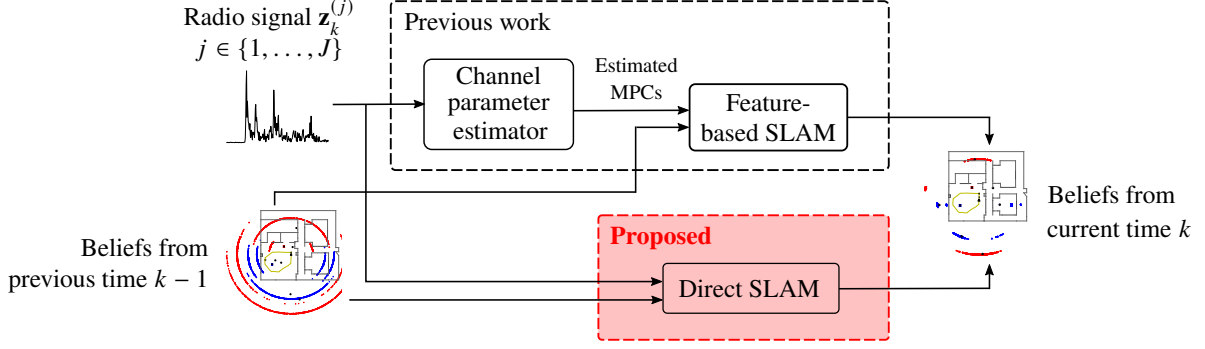


Figure 4.1. Flow diagram of the proposed Direct-SLAM compared to conventional multipath-based SLAM. Discrete time is indexed by k . Conventional methods process received radio signals in two stages. In stage one, a channel estimator extracts the parameters of MPCs, e.g., the delays, from J samples of the received radio signal, $\mathbf{z}_k^{(j)}$, $j \in \{1, \dots, J\}$. These parameters are then used as the measurements for SLAM in stage two. On the contrary, Direct-SLAM uses the received radio signal $\mathbf{z}_k^{(j)}$, $j \in \{1, \dots, J\}$ as measurements. Both approaches perform sequential Bayesian estimation, i.e., in addition to the received radio signal, approximate marginal posterior distributions (“beliefs”) from the previous time step $k-1$ are also used as the input at time k .

multiple VAs. The SLAM engine is then poised to make suboptimum data associations, which can lead to incorrect multipath SLAM estimation results. Since the ability to resolve MPC is directly proportional to signal bandwidth, performance degradation related to challenging geometries is exacerbated in low-bandwidth systems.

4.1.2 Contributions and Chapter Organization

In this chapter, we propose *Direct-SLAM* for multipath-aided localization and mapping. Raw radio signals are directly used as measurements for sequential Bayesian estimation without being preprocessed by a channel estimator to reduce information loss. The proposed method is inspired by TBD [45, 46, 53, 167, 168] approaches for MOT. In particular, it relies on a superpositional measurement model for received radio signals that describes LOS propagation and specular reflections. Other propagation phenomena that can potentially also generate MPCs, such as point scattering and diffusion, are modeled as interference.

We introduce a new measurement model for multipath-based SLAM that (i) accurately describes the superposition of specular MPCs in received radio signals and a birth model that (ii)

introduces new feature hypotheses referred to as potential features (PFs). At each time step, in addition to the state of the mobile agent and the features, the noise variance is also modeled as a random variable. The new statistical model is designed such that it can be efficiently represented by a factor graph [12, 13]. It can be viewed as a combination of the Swerling 1 model for correlated measurements proposed in [45] and the Bernoulli existence model introduced in [53], while also resembling the statistical model used in variational line spectral estimation [130, 169, 170]. The factor graph provides the blueprint for a BP method [12, 14, 90] that can sequentially estimate the number of features and their positions as well as the state of the agent in a scalable and efficient way. BP, also known as the sum-product-algorithm [12, 14, 90], computes real-value functions called “messages” along the edges of the factor graph by exploiting conditional statistical independencies. Most BP messages cannot be calculated in closed form. For an accurate and feasible numerical evaluation, following our previous work in [53, 60, 171], we represent some of the BP messages by random samples “particles” and others by means and covariance matrices obtained via moment matching. Contrary to existing SLAM methods, the processing of received radio signals by a channel estimator and explicit data association can be avoided, alleviating performance degradation in challenging SLAM geometries. Basic block diagrams of conventional and proposed methods are shown in Fig. 4.1.

The key contributions of this chapter can be summarized as follows.

- We introduce a statistical model for multipath-based SLAM that accurately models the data-generating process of received radio signals.
- We develop a scalable BP method for the sequential estimation of agent position and map features based on the factor graph representing the new model.
- We conduct comprehensive numerical experiments on synthetic and real data and demonstrate state-of-the-art performance.

The chapter is organized as follows. Section 4.2 presents the radio signal model and

the new statistical model. Section 4.3 develops the proposed BP method for multipath-based SLAM. Section 4.4 introduces a feasible implementation of the BP method. Section 4.5 reports the results of the performed numerical study.

4.2 System Model and Problem Formulation

In what follows, we introduce the proposed system model and discuss the problem formulation of the considered multipath-based SLAM problem.

4.2.1 Radio Signal and Specular MPCs

Multipath effects in radio channels are directly related to the geometry of the propagation environment [56, 57, 157]. In particular, the parameters of specular MPCs, e.g., delays, can be described by the position of the mobile agent and the positions of PAs and VAs. In what follows, we assume the mobile agent acts as a transmitter, and the PAs act as receivers. Note, however, that the proposed model can be easily reformulated for the case where the roles of PAs and mobile agent are swapped.

VA positions are the mirror images of PA positions at flat reflecting surfaces and encode the geometry of the environment. In particular, the length of the path from the mobile agent to the VA is equal to the length of the path from the mobile agent to the reflecting surface to the PA. This principle is illustrated in Fig. 4.2.

For the considered multipath-based SLAM problem, we consider a single mobile agent with an unknown position. At discrete time k , the position of the agent is denoted as $\mathbf{p}_k$. In addition, there are J PAs with known positions $\mathbf{p}_1^{(j)}, j \in \{1, \dots, J\}$. At time step k , each PA j has $L_k^{(j)} - 1$ associated VAs, with unknown positions $\mathbf{p}_l^{(j)}, l \in \{2, \dots, L_k^{(j)}\}$. PAs and VAs, here referred to as features, provide a partial characterization of the propagation environment. While PA positions are known, due to the unavailability of a floor plan, VA positions are unknown and have to be estimated jointly with the agent positions. The fact that the absolute position of certain features in the environment, i.e., the PAs, is known is a major difference of

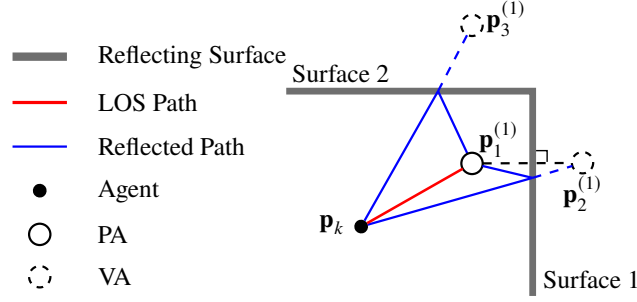


Figure 4.2. Scenario with one mobile agent, one physical anchor (PA), and two virtual anchors (VAs). The radio signal transmitted by the agent at position $\mathbf{p}_k$ propagates to the receiver at position $\mathbf{p}_1^{(1)}$ on the LOS path and on the reflected paths. The path length of the reflected paths are equal to the length of the paths from the agent at position $\mathbf{p}_k$ to the VAs at position $\mathbf{p}_2^{(1)}$ and $\mathbf{p}_3^{(1)}$, respectively.

multipath-aided SLAM compared to traditional robotic SLAM.

The mobile agent transmits a signal at center frequency f_c and bandwidth B . In the frequency domain, the transmitted signal in baseband is $S(f) \in \mathbb{C}$. At each discrete time step k , M samples of the received signal waveform are recorded by the PAs. In the frequency domain, the resulting $M = B/\Delta + 1$ samples have a frequency spacing of Δ^1 . Let $\mathbf{h}(t)$ denote the sampled transmitted signal, delayed by time t , in baseband and frequency domain, i.e.,

$$\mathbf{h}(t) = [S(-(M-1)/2\Delta)e^{j2\pi(\frac{M-1}{2}\Delta)t} \quad \dots \quad S((M-1)/2\Delta)e^{-j2\pi(\frac{M-1}{2}\Delta)t}]^T. \quad (4.1)$$

The measurement model for the sampled received signal at PA j , i.e., $\mathbf{z}_k^{(j)} = [z_k^{(j)}(-(M-1)/2\Delta) \quad \dots \quad z_k^{(j)}((M-1)/2\Delta)]^T = [z_{k,1}^{(j)} \quad \dots \quad z_{k,M}^{(j)}]^T$, can now be expressed as [172, 173]

$$\mathbf{z}_k^{(j)} = \sum_{l=1}^{L_k^{(j)}} \varrho_{k,l}^{(j)} \mathbf{h}(t_{k,l}^{(j)}) + \boldsymbol{\epsilon}_k^{(j)} \quad (4.2)$$

where $t_{k,l}^{(j)} = \|\mathbf{p}_k - \mathbf{p}_l^{(j)}\|/c$ is the delay related to the l -th MPC, $\varrho_{k,l}^{(j)} \in \mathbb{C}$ is the complex amplitude of propagation path j , and c is the speed of light.² For example, a simple physical model for the

¹The corresponding unambiguous maximum observation distance is given by $d_{\max} = c/\Delta$.

²The signal model in (4.2) can be straightforwardly extended to include additional MPC parameters, such as angle-of-departure, angle-of-arrival and Doppler frequencies [166, 174].

complex amplitude, $\varrho_{k,l}^{(j)} \in \mathbb{C}$, is given by

$$\varrho_{k,l}^{(j)} = a_{k,l}^{(j)} e^{-j2\pi f_c t_{k,l}^{(j)}} \frac{c}{4\pi f_c \|\mathbf{p}_k - \mathbf{p}_l^{(j)}\|} \quad (4.3)$$

where $a_{k,l}^{(j)} \in \mathbb{C}$ is an unknown cumulative reflection coefficient resulting from all interactions of the radio signal with flat surfaces³. Furthermore, $\boldsymbol{\epsilon}_k^{(j)} = [\epsilon_k^{(j)}(-\frac{M-1}{2}\Delta) \cdots \epsilon_k^{(j)}(\frac{M-1}{2}\Delta)]^T \in \mathbb{C}^M$ is the noise vector.

4.2.2 State Vectors and Measurement Model

The state of the mobile agent is denoted as $\mathbf{x}_k$. It includes the agent's position $\mathbf{p}_k$ and possibly further motion-related parameters. The number of features $L_k^{(j)}$, $j \in \{1, \dots, J\}$ associated with each PA depends on the position of the agent and is generally unknown and time-varying. For the estimate of the unknown $L_k^{(j)}$ at each time k , so called PFs indexed by $n \in \{1, \dots, N_k^{(j)}\}$ are introduced [19, 60]. Here $N_k^{(j)}$ is the maximum possible number of features associated with the j -th PA at time step k . The existence of each PF $j \in \{1, \dots, J\}$ is modeled by a binary random variable $r_{k,n}^{(j)} \in \{0, 1\}$, $n \in \{1, \dots, N_k^{(j)}\}$. A PF exists, i.e., corresponds to a physical feature in the environment contributing to the signal received at PA j , if and only if $r_{k,n}^{(j)} = 1$.

The state of PFs is defined as $\mathbf{y}_{k,n}^{(j)} = [\boldsymbol{\phi}_{k,n}^{(j)T} r_{k,n}^{(j)}]^T$ where $\boldsymbol{\phi}_{k,n}^{(j)} = [\mathbf{p}_{k,n}^{(j)T} \gamma_{k,n}^{(j)}]^T$ includes the position $\mathbf{p}_{k,n}^{(j)}$ and the intensity $\gamma_{k,n}^{(j)}$. The intensity represents the signal strength of the propagation path represented by the PF. Since each PA is also modeled as a PF, we have $N_k^{(j)} \geq 1$. In particular, we always use the first index, $n = 1$, for PFs representing PAs. For future reference, we establish the notation $\mathbf{y}_k^{(j)} = [\mathbf{y}_{k,1}^{(j)T} \cdots \mathbf{y}_{k,N_k^{(j)}}^{(j)T}]^T$.

Based on the PF representation, the measurement model in (4.2), can also be expressed

³The signal model in (4.2) implies uniform radiation patterns for transmit and receive antennas.

as

$$\mathbf{z}_k^{(j)} = \sum_{n=1}^{N_k^{(j)}} r_{k,n}^{(j)} \rho_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)} + \boldsymbol{\epsilon}_k^{(j)} \quad (4.4)$$

where $\mathbf{h}_{k,n}^{(j)} \triangleq \mathbf{h}(\tau_{k,n}^{(j)}) \in \mathbb{C}^M$ denotes the known contribution vector related to PF with indexes $j \in \{1, \dots, J\}$ and $n \in \{1, \dots, N_k^{(j)}\}$, i.e., $\tau_{k,n}^{(j)} = \|\mathbf{p}_k - \mathbf{p}_{k,n}^{(j)}\|/c$. Following [130, 160, 161, 162, 163], we here assume that the complex amplitudes, $\rho_{k,n}^{(j)}$, are random and unknown. In particular, random amplitudes are modeled by zero-mean complex Gaussian random variables with variance $\gamma_{k,n}^{(j)}$, i.e., $\rho_{k,n}^{(j)} \sim \mathcal{CN}(\rho_{k,n}^{(j)}; 0, \gamma_{k,n}^{(j)})$. This amplitude model is referred to as *Swerling 1* [45, 132]. The complex amplitudes $\rho_{k,n}^{(j)}$ are independent across k, n , and j . The noise $\boldsymbol{\epsilon}_k^{(j)}$ represents different noise sources, including unmodeled non-specular propagation effects and white receiver noise. For the sake of simplicity $\boldsymbol{\epsilon}_k^{(j)}$ is assumed distributed according to $\mathcal{CN}(\boldsymbol{\epsilon}_k^{(j)}; \mathbf{0}, \eta_k^{(j)} \mathbf{I}_M)$ and independent across k and all j , as well as independent of all complex amplitudes.

As a result of this model, conditioned on the agent state $\mathbf{x}_k$, the joint PF state $\mathbf{y}_k^{(j)}$, and the noise variance $\eta_k^{(j)}$, the measurement $\mathbf{z}_k^{(j)}$ is also zero-mean complex Gaussian, i.e.,

$$f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)}) = \mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \mathbf{C}_k^{(j)}) \quad (4.5)$$

with covariance $\mathbf{C}_k^{(j)} = \eta_k^{(j)} \mathbf{I}_M + \sum_{n=1}^{N_k^{(j)}} r_{k,n}^{(j)} \gamma_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)H}$. It follows from (4.4) that, conditioned on the $\mathbf{x}_k, \mathbf{y}_k^{(j)}$ and $\eta_k^{(j)}$, the measurement $\mathbf{z}_k^{(j)}$ is independent of all the other measurements and independent of $\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1}^{(j)}$ and $\eta_{1:k-1}^{(j)}$. As a result, the joint likelihood function factorizes as

$$f(\mathbf{z}_{1:k} | \mathbf{x}_{1:k}, \mathbf{y}_{1:k}, \boldsymbol{\eta}_{1:k}) = \prod_{k'=1}^k \prod_{j=1}^J f(\mathbf{z}_{k'}^{(j)} | \mathbf{x}_{k'}, \mathbf{y}_{k'}^{(j)}, \eta_{k'}^{(j)}) \quad (4.6)$$

where we introduce $\mathbf{z}_k = [\mathbf{z}_k^{(1)T} \dots \mathbf{z}_k^{(J)T}]^T$, $\mathbf{z}_{1:k} = [\mathbf{z}_1^T \dots \mathbf{z}_k^T]^T$, $\mathbf{x}_{1:k} = [\mathbf{x}_1^T \dots \mathbf{x}_k^T]^T$, $\mathbf{y}_k = [\mathbf{y}_k^{(1)T} \dots \mathbf{y}_k^{(J)T}]^T$, $\mathbf{y}_{1:k} = [\mathbf{y}_1^T \dots \mathbf{y}_k^T]^T$, $\boldsymbol{\eta}_k = [\eta_k^{(1)} \dots \eta_k^{(J)}]^T$, and $\boldsymbol{\eta}_{1:k} = [\boldsymbol{\eta}_1^T \dots \boldsymbol{\eta}_k^T]^T$.

4.2.3 State-Transition Models and Prior PDFs

The dynamics of the agent state $\mathbf{x}_k$ are described statistically by a first-order Markov model. Its state transition PDF is denoted as $f(\mathbf{x}_k|\mathbf{x}_{k-1})$, which can, e.g., be the result of a constant velocity model [108, Ch. 4]. The dynamics of the noise variance is described by a PDF $f(\eta_k^{(j)}|\eta_{k-1}^{(j)})$, which can, e.g., be a Gamma distribution. Each PF state $\mathbf{y}_{k,n}^{(j)}$ with $n \in \{1, \dots, N_{k-1}^{(j)}\}$ and $j \in \{1, \dots, J\}$ is also assumed to evolve independently following a first-order Markov model. The individual state-transition PDF of the PF with indexes $n \in \{1, \dots, N_{k-1}^{(j)}\}$ and $j \in \{1, \dots, J\}$, is denoted by $f(\mathbf{y}_{k,n}^{(j)}|\mathbf{y}_{k-1,n}^{(j)}) = f(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)}|\boldsymbol{\phi}_{k-1,n}^{(j)}, r_{k-1,n}^{(j)})$. If PF (j, n) does not exist at time step $k-1$, i.e., $r_{k-1,n}^{(j)} = 0$, then it does not exist at time step k either. The state-transition PDF for $r_{k-1,n}^{(j)} = 0$ is thus given by

$$f(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)}|\boldsymbol{\phi}_{k-1,n}^{(j)}, 0) = \begin{cases} f_D(\boldsymbol{\phi}_{k,n}^{(j)}), & r_{k,n}^{(j)} = 0 \\ 0, & r_{k,n}^{(j)} = 1 \end{cases} \quad (4.7)$$

where $f_D(\boldsymbol{\phi}_{k,n}^{(j)})$ is an arbitrary “dummy” PDF. However, if the PF exists at time step $k-1$, i.e., $r_{k-1,n}^{(j)} = 1$, then it continues to exist at time step k with survival probability $0 < p_s \leq 1$. If it still exists at time k , then $\boldsymbol{\phi}_{k,n}^{(j)}$ is distributed according to $f(\boldsymbol{\phi}_{k,n}^{(j)}|\boldsymbol{\phi}_{k-1,n}^{(j)})$, which can be, e.g., a random walk model. For $r_{k-1,n}^{(j)} = 0$, the state-transition PDF thus reads

$$f(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)}|\boldsymbol{\phi}_{k-1,n}^{(j)}, 1) = \begin{cases} (1 - p_s) f_D(\boldsymbol{\phi}_{k,n}^{(j)}), & r_{k,n}^{(j)} = 0 \\ p_s f(\boldsymbol{\phi}_{k,n}^{(j)}|\boldsymbol{\phi}_{k-1,n}^{(j)}), & r_{k,n}^{(j)} = 1. \end{cases} \quad (4.8)$$

The birth of newly appearing features associated with PA j and time k is modeled by a point Poisson process with mean $\mu_B^{(j)}$ and spatial PDF $f_B^{(j)}(\boldsymbol{\phi}|\mathbf{x}_k)$. Assuming that (i) the number of newly appearing features is significantly smaller than the number of measurements and (ii) newly appearing features are well separated, we introduce M new PFs at each time step, i.e., $N_k^{(j)} = N_{k-1}^{(j)} + M$ [53]. The initial PDF of each new PF, conditioned on $\mathbf{x}_k$, can thus be written

as

$$f_{B,n}^{(j)}(\boldsymbol{\phi}|\mathbf{x}_k) \propto \begin{cases} f_B^{(j)}(\boldsymbol{\phi}|\mathbf{x}_k), & \mathbf{p} \in \mathcal{P}_m(\mathbf{x}_k) \\ 0, & \mathbf{p} \notin \mathcal{P}_m(\mathbf{x}_k) \end{cases}$$

where $\propto$ indicates equality up to a multiplicative constant and the region $\mathcal{P}_m(\mathbf{x}_k)$ that is occupied by index $m = n - N_{k-1}^{(j)} \in \{1, \dots, M\}$, is given by

$$\mathcal{P}_m(\mathbf{x}_k) = \{\mathbf{p} \mid (m-1) \cdot T_s \leq \|\mathbf{p} - \mathbf{p}_k\|/c \leq m \cdot T_s\}. \quad (4.9)$$

To define the existence probability of new PFs, we first note that the number of new features in cell $\mathcal{P}_m(\mathbf{x}_k)$ is also Poisson distributed with mean $\mu_{B,n}^{(j)} = \mu_B^{(j)} \iint_{\mathcal{P}_m(\mathbf{x}_k)} f_B^{(j)}(\boldsymbol{\phi}|\mathbf{x}_k) d\mathbf{p} d\gamma$, and the probabilities that each cell has zero and one feature are $e^{-\mu_{B,n}^{(j)}}$ and $\mu_{B,n}^{(j)} e^{-\mu_{B,n}^{(j)}}$, respectively. Based on the assumption that there is at most one new feature in $\mathcal{P}_m(\mathbf{x}_k)$, we define the existence probability of the corresponding PF as

$$p_{B,n}^{(j)} = \frac{\mu_{B,n}^{(j)} e^{-\mu_{B,n}^{(j)}}}{\mu_{B,n}^{(j)} e^{-\mu_{B,n}^{(j)}} + e^{-\mu_{B,n}^{(j)}}} = \frac{\mu_{B,n}^{(j)}}{\mu_{B,n}^{(j)} + 1}.$$

As a result, the PDF for individual new PFs reads

$$f(\mathbf{y}_{k,n}^{(j)}|\mathbf{x}_k) = \begin{cases} (1 - p_{B,n}^{(j)}) f_D(\boldsymbol{\phi}_{k,n}^{(j)}), & r_{k,n}^{(j)} = 0 \\ p_{B,n}^{(j)} f_{B,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}|\mathbf{x}_k), & r_{k,n}^{(j)} = 1. \end{cases} \quad (4.10)$$

Note that based on the binary random variables $r_{k,n}^{(j)} \in \{0, 1\}$ (cf. Section 4.2.2) that indicate feature existence and the Poisson point process used to describe the appearance of new PFs discussed above, we can represent dynamic SLAM environments with features that appear/disappear due to limited visibility, e.g., blockage of propagation paths and changes in the environment.

By using the fact that agent states, PFs, and noise variance follow first-order Markov

dynamics, we can write

$$f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) = f(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\eta}_0) \prod_{k'=1}^k f(\mathbf{x}_{k'}, \mathbf{y}_{k'}, \boldsymbol{\eta}_{k'} | \mathbf{x}_{k'-1}, \mathbf{y}_{k'-1}, \boldsymbol{\eta}_{k'-1}). \quad (4.11)$$

At time $k = 0$, the random variables $\mathbf{x}_0$, $\eta_0^{(j)}$, $j \in \{1, \dots, J\}$, and $\mathbf{y}_{0,n}^{(j)}$, $n \in \{1, \dots, N_0^{(j)}\}$, $j \in \{1, \dots, J\}$ are all independent of each other, i.e.,

$$f(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\eta}_0) = f(\mathbf{x}_0) \prod_{j=1}^J f(\eta_0^{(j)}) \left(\prod_{n=1}^{N_0^{(j)}} f(\mathbf{y}_{0,n}^{(j)}) \right). \quad (4.12)$$

The prior distributions $f(\mathbf{x}_0)$, $f(\eta_0^{(j)})$, $j \in \{1, \dots, J\}$, and $f(\mathbf{y}_{0,n}^{(j)})$, $n \in \{1, \dots, N_0^{(j)}\}$, $j \in \{1, \dots, J\}$ are known.

It is furthermore assumed that agent states and noise variance evolve independently in time and that the evolution of time of PFs also depends on the current agent state, i.e.,

$$f(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\eta}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, \boldsymbol{\eta}_{k-1}) = f(\mathbf{x}_k | \mathbf{x}_{k-1}) f(\boldsymbol{\eta}_k | \boldsymbol{\eta}_{k-1}) f(\mathbf{y}_k | \mathbf{x}_k, \mathbf{y}_{k-1}). \quad (4.13)$$

Conditioned on the corresponding previous PF state $\mathbf{y}_{k-1,n}^{(j)}$, the legacy PF state $\mathbf{y}_{k,n}^{(j)}$, $j \in \{1, \dots, J\}$, $n \in \{1, \dots, N_{k-1}^{(j)}\}$, is assumed independent of the current and previous states of the other PFs. For the new PF state $\mathbf{y}_{k,n}^{(j)}$, $j \in \{1, \dots, J\}$, $n \in \{N_{k-1}^{(j)} + 1, \dots, N_k^{(j)}\}$, it is assumed that conditioned on $\mathbf{x}_k$, the new PF state $\mathbf{y}_{k,n}^{(j)}$ is independent of all current and previous states of all other PFs, i.e.,

$$f(\mathbf{y}_k | \mathbf{x}_k, \mathbf{y}_{k-1}) = \prod_{j=1}^J \prod_{n=1}^{N_{k-1}^{(j)}} f(\mathbf{y}_{k,n}^{(j)} | \mathbf{y}_{k-1,n}^{(j)}) \prod_{n=N_{k-1}^{(j)}+1}^{N_k^{(j)}} f(\mathbf{y}_{k,n}^{(j)} | \mathbf{x}_k). \quad (4.14)$$

It is also assumed that, conditioned on $\eta_{k-1}^{(j)}$, the noise variance $\eta_k^{(j)}$ is independent of all the other noise variances, i.e.,

$$f(\boldsymbol{\eta}_k | \boldsymbol{\eta}_{k-1}) = \prod_{j=1}^J f(\eta_k^{(j)} | \eta_{k-1}^{(j)}). \quad (4.15)$$

Based on these assumptions, we can develop the joint prior PDF $f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k})$. In particular, by first using (4.12) and (4.13) for $f(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\eta}_0)$ and $f(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\eta}_k | \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, \boldsymbol{\eta}_{k-1})$ in (4.11) and subsequently replacing $f(\mathbf{y}_k | \mathbf{x}_k, \mathbf{y}_{k-1})$ and $f(\boldsymbol{\eta}_k | \boldsymbol{\eta}_{k-1})$, by the expressions in (4.14) and (4.15), respectively, we obtain

$$\begin{aligned} f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) &= f(\mathbf{x}_0) \prod_{j=1}^J f(\eta_0^{(j)}) \left(\prod_{n=1}^{N_0^{(j)}} f(\mathbf{y}_{0,n}^{(j)}) \right) \prod_{k'=1}^k f(\mathbf{x}_{k'} | \mathbf{x}_{k'-1}) \\ &\quad \times \prod_{j=1}^J f(\eta_{k'}^{(j)} | \eta_{k'-1}^{(j)}) \left(\prod_{n=1}^{N_{k'-1}^{(j)}} f(\mathbf{y}_{k',n}^{(j)} | \mathbf{y}_{k'-1,n}^{(j)}) \right) \left(\prod_{n=N_{k'-1}^{(j)}+1}^{N_{k'}^{(j)}} f(\mathbf{y}_{k',n}^{(j)} | \mathbf{x}_{k'}) \right). \end{aligned} \quad (4.16)$$

4.2.4 Declaration and Estimation

At each time step k , given all the measurements $\mathbf{z}_{1:k} = [\mathbf{z}_1^T \cdots \mathbf{z}_k^T]^T$, the goal of multipath-based SLAM is to declare the existence of PFs as well as estimate the states $\boldsymbol{\phi}_{k,n}^{(j)}$ of PF that have been declared to exist, the agent state $\mathbf{x}_k$, and the noise variance $\eta_k^{(j)}$. In our Bayesian setting, the problem comprises the computation of the marginal PDFs $f(r_{k,n}^{(j)} | \mathbf{z}_{1:k})$, $f(\boldsymbol{\phi}_{k,n}^{(j)} | r_{k,n}^{(j)} = 1, \mathbf{z}_{1:k})$, $f(\mathbf{x}_k | \mathbf{z}_{1:k})$, and $f(\eta_k^{(j)} | \mathbf{z}_{1:k})$. We refer to $f(r_{k,n}^{(j)} = 1 | \mathbf{z}_{1:k})$, $n \in \{1, \dots, N_k^{(j)}\}$ as the existence probability of PFs. A PF is declared to exist, if its existence probability is larger than a certain fixed threshold T_{dec} .

Based on these marginal PDFs, the MMSE estimate of the agent states, PFs declared to exist, and noise variance at time k , can be obtained as

$$\hat{\mathbf{x}}_{k,\text{MMSE}} = \int \mathbf{x}_k f(\mathbf{x}_k | \mathbf{z}_{1:k}) d\mathbf{x}_k \quad (4.17)$$

$$\hat{\boldsymbol{\phi}}_{k,n,\text{MMSE}}^{(j)} = \int \boldsymbol{\phi}_{k,n}^{(j)} f(\boldsymbol{\phi}_{k,n}^{(j)} | r_{k,n}^{(j)} = 1, \mathbf{z}_{1:k}) d\boldsymbol{\phi}_{k,n}^{(j)} \quad (4.18)$$

$$\hat{\eta}_{k,\text{MMSE}}^{(j)} = \int \eta_k^{(j)} f(\eta_k^{(j)} | \mathbf{z}_{1:k}) d\eta_k^{(j)}. \quad (4.19)$$

Note that $f(r_{k,n}^{(j)} = 1 | \mathbf{z}_{1:k})$ and $f(\boldsymbol{\phi}_{k,n}^{(j)} | r_{k,n}^{(j)} = 1, \mathbf{z}_{1:k})$ can be computed from the marginal posterior PDF $f(\mathbf{y}_{k,n}^{(j)} | \mathbf{z}_{1:k}) = f(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)} | \mathbf{z}_{1:k})$.

Following Bayes rule and inserting (4.6) and (4.16), the joint posterior PDF $f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k} | \mathbf{z}_{1:k})$ can be factorized as

$$\begin{aligned} f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k} | \mathbf{z}_{1:k}) &\propto f(\mathbf{z}_{1:k} | \mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) \\ &= f(\mathbf{z}_{1:k} | \mathbf{x}_{1:k}, \mathbf{y}_{1:k}, \boldsymbol{\eta}_{1:k}) f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) \\ &= f(\mathbf{x}_0) \left(\prod_{j=1}^J f(\eta_0^{(j)}) \prod_{n=1}^{N_0^{(j)}} f(\mathbf{y}_{0,n}^{(j)}) \right) \prod_{k'=1}^k f(\mathbf{x}_{k'} | \mathbf{x}_{k'-1}) \\ &\quad \times \prod_{j=1}^J \left(\prod_{n=1}^{N_{k'-1}^{(j)}} f(\mathbf{y}_{k',n}^{(j)} | \mathbf{y}_{k'-1,n}^{(j)}) \right) \left(\prod_{n=N_{k'-1}^{(j)}+1}^{N_{k'}^{(j)}} f(\mathbf{y}_{k',n}^{(j)} | \mathbf{x}_{k'}) \right) \\ &\quad \times f(\eta_{k'}^{(j)} | \eta_{k'-1}^{(j)}) f(\mathbf{z}_{k'}^{(j)} | \mathbf{x}_{k'}, \mathbf{y}_{k'}^{(j)}, \eta_{k'}^{(j)}). \end{aligned} \quad (4.20)$$

A single time step of the factor graph corresponding to the factorization in (4.20), is shown in Fig. 4.3. This factorization facilitates the development of an efficient BP method for the computation of approximate marginal posterior PDFs $\tilde{f}(\mathbf{y}_{k,n}^{(j)}) \approx f(\mathbf{y}_{k,n}^{(j)} | \mathbf{z}_{1:k})$, $\tilde{f}(\mathbf{x}_k) \approx f(\mathbf{x}_k | \mathbf{z}_{1:k})$, and $\tilde{f}(\eta_k^{(j)}) \approx f(\eta_k^{(j)} | \mathbf{z}_{1:k})$, typically referred to as beliefs.

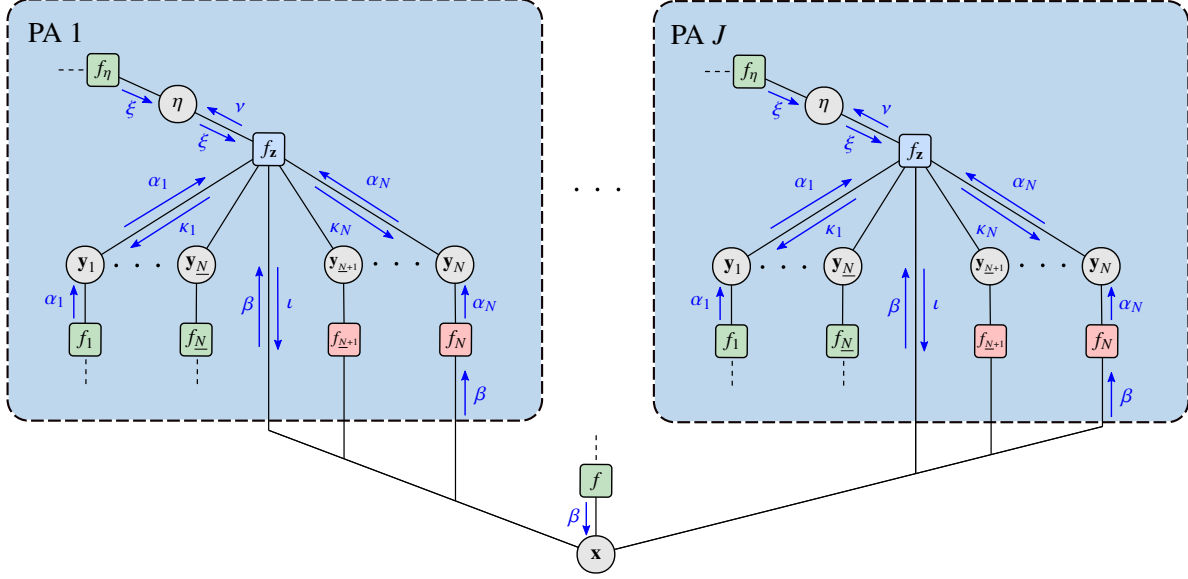


Figure 4.3. Factor graph representing the joint posterior distribution in (4.20) and the BP messages for a single time step k . Each box with light blue background indicates a different PAs. BP messages are shown as blue arrows. The time index k is omitted. The following shorthand notation is used: $\underline{N} = N_{k-1}^{(j)}$, $N = N_k^{(j)}$, $\mathbf{x} = \mathbf{x}_k$, $\mathbf{y}_n = \mathbf{y}_{k,n}^{(j)}$, $\eta = \eta_k^{(j)}$, $f = f(\mathbf{x}_k | \mathbf{x}_{k-1})$, $f_\eta = f(\eta_k^{(j)} | \eta_{k-1}^{(j)})$, $f_z = f(\mathbf{z}_k | \mathbf{x}_k, \mathbf{y}_k, \eta_k^{(j)})$, $f_n = f(\mathbf{y}_{k,n}^{(j)} | \mathbf{y}_{k-1,n}^{(j)})$ for $n \in \{1, \dots, N_{k-1}^{(j)}\}$, and $f_n = f(\mathbf{y}_{k,n}^{(j)} | \mathbf{x}_k)$ for $n \in \{N_{k-1}^{(j)} + 1, \dots, N_k^{(j)}\}$. Furthermore, $\beta = \beta(\mathbf{x}_k)$, $\alpha_n = \alpha(\mathbf{y}_{k,n}^{(j)})$, $\xi = \xi(\eta_k^{(j)})$, $\iota = \iota(\mathbf{x}_k; \mathbf{z}_k^{(j)})$, $\kappa_n = \kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)})$, and $v = v(\eta_k^{(j)}; \mathbf{z}_k^{(j)})$.

4.3 The Proposed BP Algorithm

In this section, we derive the proposed BP method for multipath-based SLAM. BP is an efficient approach for solving high-dimension Bayesian inference problems. A BP method is derived by performing local operations called “messages” on the edges of the factor graph [12, 14, 90] that represents the statistical model of the Bayesian estimation problem. When the factor graph is a tree, the solutions provided the BP, called “beliefs”, are equal to the true marginal posterior PDFs required for the computation of optimum estimates. However, the factor graph representing the statistical model for direct SLAM shown in Fig. 4.3 has loops. Thus, the beliefs computed by BP are only approximations of the true marginal posteriors [12, 14, 90]. Despite the fact that theoretical performance guarantees are typically unavailable, performing BP on graphs with loops is a widely used strategy to compute accurate approximations of marginal

posterior PDFs with application in, e.g., modern channel coding [16], cooperative localization [17], multitarget tracking [18], and SLAM [19]. In graphs with loops, there is some flexibility regarding the order in which messages are computed. Different message-passing orders may lead to different beliefs. In this work, we make use of the following message passing computation order: (i) messages are passed only forward in time, (ii) along the edges connecting an agent state variable node “ $\mathbf{x}_k$ ” and a new PF state variable node “ $\mathbf{y}_{k,n}^{(j)}$ ”, messages are only sent from the former to the latter [19], (iii) all messages are only computed once, i.e., messages are not updated iteratively. BP messages are also shown in Fig. 4.3. They are computed in parallel for all PAs $j \in \{1, \dots, J\}$. All messages developed in what follows are obtained from conventional BP message rules for messages passed from factor to variable nodes, and messages passed from variable nodes to factor nodes [12, 14, 90].

4.3.1 Prediction and Birth Messages

First, we present the prediction message that is passed from factor node “ $f(\mathbf{x}_k|\mathbf{x}_{k-1})$ ” to variable node “ $\mathbf{x}_k$ ”. This message is obtained as

$$\beta(\mathbf{x}_k) = \int f(\mathbf{x}_k|\mathbf{x}_{k-1}) \tilde{f}(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1} \quad (4.21)$$

where $\tilde{f}(\mathbf{x}_{k-1})$ is the belief of the agent state at the previous time step $k-1$.

Next, we develop the messages that are sent from the factor nodes “ $f(\mathbf{y}_{k,n}^{(j)}|\mathbf{y}_{k-1,n}^{(j)})$ ”, $n \in \{1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ to the corresponding PF variable nodes “ $\mathbf{y}_{k,n}^{(j)}$ ”. For each legacy PF state $\mathbf{y}_{k,n}^{(j)}$ with $n \in \{1, \dots, N_{k-1}^{(j)}\}$, $j \in \{1, \dots, J\}$, this message is given by

$$\alpha(\mathbf{y}_{k,n}^{(j)}) = \sum_{r_{k-1,n}^{(j)} \in \{0,1\}} \int f(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)} | \boldsymbol{\phi}_{k-1,n}^{(j)}, r_{k-1,n}^{(j)}) \tilde{f}(\boldsymbol{\phi}_{k-1,n}^{(j)}, r_{k-1,n}^{(j)}) d\boldsymbol{\phi}_{k-1,n}^{(j)} \quad (4.22)$$

where $\tilde{f}(\mathbf{y}_{k-1,n}^{(j)}) = \tilde{f}(\boldsymbol{\phi}_{k-1,n}^{(j)}, r_{k-1,n}^{(j)})$ is the corresponding belief at time step $k-1$. Specifically,

by inserting (4.7) and (4.8) into (4.22), for $r_{k-1,n}^{(j)} = 1$, we obtain

$$\alpha(\boldsymbol{\phi}_{k,n}^{(j)}, 1) = \int p_s f(\boldsymbol{\phi}_{k,n}^{(j)} | \boldsymbol{\phi}_{k-1,n}^{(j)}) \tilde{f}(\boldsymbol{\phi}_{k-1,n}^{(j)}, 1) d\boldsymbol{\phi}_{k-1,n}^{(j)}.$$

Similarly, for $r_{k-1,n}^{(j)} = 0$ we have $\alpha(\boldsymbol{\phi}_{k,n}^{(j)}, 0) = \alpha_{k,n}^{(j)} f_D(\boldsymbol{\phi}_{k,n}^{(j)})$ with

$$\alpha_{k,n}^{(j)} \triangleq \int (1 - p_s) \tilde{f}(\boldsymbol{\phi}_{k-1,n}^{(j)}, 1) + \tilde{f}(\boldsymbol{\phi}_{k-1,n}^{(j)}, 0) d\boldsymbol{\phi}_{k-1,n}^{(j)}.$$

PF states $\mathbf{y}_{k,n}^{(j)}$, $n \in \{N_{k-1}^{(j)} + 1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ are newly introduced at time k .

The message sent from factor node “ $f(\mathbf{y}_{k,n}^{(j)} | \mathbf{x}_k)$ ” to variable node “ $\mathbf{y}_{k,n}^{(j)}$ ” is given by

$$\alpha(\mathbf{y}_{k,n}^{(j)}) = \int f(\mathbf{y}_{k,n}^{(j)} | \mathbf{x}_k) \beta(\mathbf{x}_k) d\mathbf{x}_k. \quad (4.23)$$

This message provides feature birth information. By inserting (4.10) into (4.23) we obtain

$$\alpha(\boldsymbol{\phi}_{k,n}^{(j)}, 1) = p_{B,n}^{(j)} \int f_{B,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)} | \mathbf{x}_k) \beta(\mathbf{x}_k) d\mathbf{x}_k$$

and $\alpha(\boldsymbol{\phi}_{k,n}^{(j)}, 0) = (1 - p_{B,n}^{(j)}) f_D(\boldsymbol{\phi}_{k,n}^{(j)})$.

Finally, for each $j \in \{1, \dots, J\}$, the prediction message that is sent from the factor node “ $f(\eta_k^{(j)} | \eta_{k-1}^{(j)})$ ” to the noise variance node “ $\eta_k^{(j)}$ ”, is given by

$$\xi(\eta_k^{(j)}) = \int f(\eta_k^{(j)} | \eta_{k-1}^{(j)}) \tilde{f}(\eta_{k-1}^{(j)}) d\eta_{k-1}^{(j)}. \quad (4.24)$$

where $\tilde{f}(\eta_{k-1}^{(j)})$ is the belief of the noise variance computed at the previous time step $k - 1$.

Following BP message passing rules, messages sent from the PF state nodes “ $\mathbf{y}_{k,n}^{(j)}$ ”, $n \in \{1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ to the factor nodes “ $f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)})$ ” are equal to $\alpha(\mathbf{y}_{k,n}^{(j)})$, while messages sent from the noise variance nodes “ $\eta_k^{(j)}$ ”, $j \in \{1, \dots, J\}$ to “ $f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)})$ ” are equal to $\xi(\eta_k^{(j)})$. In addition, due to our message schedule stated above, no message is passed from “ $f(\mathbf{y}_{k,n}^{(j)} | \mathbf{x}_k)$ ”, $n \in \{N_{k-1}^{(j)} + 1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ to “ $\mathbf{x}_k$.” Thus, the

messages sent from “ $\mathbf{x}_k$ ” to factor nodes “ $f(\mathbf{y}_{k,n}^{(j)}|\mathbf{x}_k)$ ”, $n \in \{N_{k-1}^{(j)} + 1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ and “ $f(\mathbf{z}_k^{(j)}|\mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)})$ ”, $j \in \{1, \dots, J\}$ are equal to $\beta(\mathbf{x}_k)$. It is worth mentioning that the messages resulting from the equations (4.21)-(4.24) are all PDFs, i.e., they all normalized to one. This is because the messages needed as an input in (4.21)-(4.24), namely $\tilde{f}(\mathbf{x}_{k-1})$, $\tilde{f}(\eta_k^{(j)})$, $\beta(\mathbf{x}_k)$, and $\tilde{f}(\eta_{k-1}^{(j)})$ are also PDFs. Message computations in (4.21)-(4.24) can then be seen as marginalization from joint distributions that consist of two variables.

4.3.2 Measurement Update Messages

After prediction and birth messages are obtained, a measurement update is performed where messages that provide the information of measurements $\mathbf{z}_k$ are obtained. Specifically, messages sent from factor nodes “ $f(\mathbf{z}_k^{(j)}|\mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)})$ ” to agent state “ $\mathbf{x}_k$ ”, PF states “ $\mathbf{y}_{k,n}^{(j)}$ ”, $n \in \{1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ and noise variance “ $\eta_k^{(j)}$ ”, $j \in \{1, \dots, J\}$ are computed. In particular, the messages $\iota(\mathbf{x}_k; \mathbf{z}_k^{(j)})$, $j \in \{1, \dots, J\}$ sent from “ $f(\mathbf{z}_k^{(j)}|\mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)})$ ” to “ $\mathbf{x}_k$ ” are obtained as

$$\begin{aligned} \iota(\mathbf{x}_k; \mathbf{z}_k^{(j)}) = & \sum_{r_{k,1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,N_k^{(j)}}^{(j)} \in \{0,1\}} \int \cdots \int f(\mathbf{z}_k^{(j)}|\mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)}) \xi(\eta_k^{(j)}) \\ & \times \prod_{n=1}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n}^{(j)}) d\phi_{k,1}^{(j)} \cdots d\phi_{k,N_k^{(j)}}^{(j)} d\eta_k^{(j)}. \end{aligned} \quad (4.25)$$

Note that the measurements $\mathbf{z}_k^{(j)}$, $j \in \{1, \dots, J\}$ are observed and thus fixed. This is indicated by the “;” notation in $\iota(\mathbf{x}_k; \mathbf{z}_k^{(j)})$. These messages provide new agent location information related to measurement $\mathbf{z}_k^{(j)}$ by leveraging knowledge on PFs states and the noise variance.

Next, the messages $\kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)})$, $n \in \{1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ passed from “ $f(\mathbf{z}_k^{(j)}|\mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)})$ ” to PF “ $\mathbf{y}_{k,n}^{(j)}$ ” are given by

$$\begin{aligned}
\kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)}) &= \sum_{r_{k,1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,n-1}^{(j)} \in \{0,1\}} \sum_{r_{k,n+1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,N_k}^{(j)} \in \{0,1\}} \\
&\int \cdots \int f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)}) \xi(\eta_k^{(j)}) \beta(\mathbf{x}_k) \prod_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n'}^{(j)}) \\
&\mathrm{d}\boldsymbol{\phi}_{k,1}^{(j)} \cdots \mathrm{d}\boldsymbol{\phi}_{k,n-1}^{(j)} \mathrm{d}\boldsymbol{\phi}_{k,n+1}^{(j)} \cdots \mathrm{d}\boldsymbol{\phi}_{k,N_k}^{(j)} \mathrm{d}\eta_k^{(j)} \mathrm{d}\mathbf{x}_k.
\end{aligned} \tag{4.26}$$

These messages provide new location information related to measurement $\mathbf{z}_k^{(j)}$ for each PF state by taking the knowledge on all the other PFs states, on the agent state, and on the noise variance into account.

Finally, the messages $\nu(\eta_k^{(j)}; \mathbf{z}_k^{(j)})$, $j \in \{1, \dots, J\}$ sent from “ $f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)})$ ” to the noise variance nodes “ $\eta_k^{(j)}$ ” are calculated as

$$\begin{aligned}
\nu(\eta_k^{(j)}; \mathbf{z}_k^{(j)}) &= \sum_{r_{k,1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,N_k}^{(j)} \in \{0,1\}} \int \cdots \int f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)}) \prod_{n=1}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n}^{(j)}) \\
&\times \beta(\mathbf{x}_k) \mathrm{d}\boldsymbol{\phi}_{k,1}^{(j)} \cdots \mathrm{d}\boldsymbol{\phi}_{k,N_k}^{(j)} \mathrm{d}\mathbf{x}_k.
\end{aligned} \tag{4.27}$$

These messages provide new noise variance information related to measurement $\mathbf{z}_k^{(j)}$ by taking the knowledge on all the PFs states and on the agent state into account.

4.3.3 Approximate Update and Belief Calculation

The computation of measurement update messages in (4.25)–(4.27) involves marginalization of $N_k^{(j)}$ or $N_k^{(j)} - 1$ PF states. The complexity of this marginalization scales exponentially with the number of PFs $N_k^{(j)}$. To improve the scalability of the proposed algorithm, we perform the following approximations. Since the likelihood function $f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)})$ is zero-mean complex Gaussian with respect to $\mathbf{z}_k^{(j)}$, the messages computed in (4.25)–(4.27), i.e., $\iota(\mathbf{x}_k; \mathbf{z}_k^{(j)})$,

$\kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)})$, and $\nu(\eta_k^{(j)}; \mathbf{z}_k^{(j)})$, can be interpreted as zero-mean complex Gaussian mixtures PDFs of $\mathbf{z}_k^{(j)}$. We here approximate them with zero-mean Gaussian PDFs via moment matching [53, 168], i.e., we compute the following approximated messages

$$\tilde{\iota}(\mathbf{x}_k; \mathbf{z}_k^{(j)}) = \mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \mathbf{C}_{\iota,k}^{(j)})$$

$$\tilde{\kappa}(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)}) = \mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \mathbf{C}_{\kappa,k,n}^{(j)})$$

$$\tilde{\nu}(\mathbf{C}_{\epsilon,k}^{(j)}; \mathbf{z}_k^{(j)}) = \mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \mathbf{C}_{\nu,k}^{(j)})$$

such that $\mathbf{C}_{\iota,k}^{(j)}(\mathbf{x}_k)$, $\mathbf{C}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j)})$, and $\mathbf{C}_{\nu,k}^{(j)}$ are the covariance of the original Gaussian mixtures $\iota(\mathbf{x}_k; \mathbf{z}_k^{(j)})$, $\kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)})$, and $\nu(\eta_k^{(j)}; \mathbf{z}_k^{(j)})$, respectively. In particular, the matrices $\mathbf{C}_{\iota,k}^{(j)}(\mathbf{x}_k)$, $\mathbf{C}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j)})$, and $\mathbf{C}_{\nu,k}^{(j)}$ are obtained as [175, Sec. 1 of the supporting document]

$$\mathbf{C}_{\iota,k}^{(j)}(\mathbf{x}_k) = \sum_{n=1}^{N_k^{(j)}} \mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_k) + \eta_{\xi,k}^{(j)} \mathbf{I}_M \quad (4.28)$$

$$\mathbf{C}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j)}) = r_{k,n}^{(j)} \mathbf{C}_{2,k,n}^{(j)}(\phi_{k,n}^{(j)}) + \sum_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \mathbf{C}_{3,k,n'}^{(j)} + \eta_{\xi,k}^{(j)} \mathbf{I}_M \quad (4.29)$$

$$\mathbf{C}_{\nu,k}^{(j)} = \eta_k^{(j)} \mathbf{I}_M + \sum_{n=1}^{N_k^{(j)}} \mathbf{C}_{3,k,n}^{(j)} \quad (4.30)$$

where we introduced the following matrices that have to be computed in a preparatory step, i.e.,

$$\mathbf{H}_{k,n}^{(j)}(\mathbf{x}_k, \phi_{k,n}^{(j)}) = \gamma_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)H} \quad (4.31)$$

$$\mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_k) = \int \mathbf{H}_{k,n}^{(j)}(\mathbf{x}_k, \phi_{k,n}^{(j)}) \alpha(\phi_{k,n}^{(j)}, 1) d\phi_{k,n}^{(j)} \quad (4.31)$$

$$\mathbf{C}_{2,k,n}^{(j)}(\phi_{k,n}^{(j)}) = \int \mathbf{H}_{k,n}^{(j)}(\mathbf{x}_k, \phi_{k,n}^{(j)}) \beta(\mathbf{x}_k) d\mathbf{x}_k \quad (4.32)$$

$$\mathbf{C}_{3,k,n}^{(j)} = \int \mathbf{H}_{k,n}^{(j)}(\mathbf{x}_k, \phi_{k,n}^{(j)}) \alpha(\phi_{k,n}^{(j)}, 1) \beta(\mathbf{x}_k) d\phi_{k,n}^{(j)} d\mathbf{x}_k \quad (4.33)$$

$$\eta_{\xi,k}^{(j)} = \int \eta_k^{(j)} \xi(\eta_k^{(j)}) d\eta_k^{(j)}. \quad (4.34)$$

Recall that $\mathbf{h}_{k,n}^{(j)}$ is the contribution of the PF with indexes (j, n) as introduced in (4.4). This contribution is a function of $\mathbf{y}_{k,n}^{(j)}$ and $\mathbf{x}_k$. Note that for $r_{k,n}^{(j)} = 0$, $\mathbf{C}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j)}) = \mathbf{C}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)})$ in (4.29) is independent of $\boldsymbol{\phi}_{k,n}^{(j)}$.

With the approximated BP messages, the beliefs of agent state $\mathbf{x}_k$, PF states $\mathbf{y}_{k,n}^{(j)}$, $n \in \{1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ and noise variance $\eta_k^{(j)}$, $j \in \{1, \dots, J\}$ can be obtained as the product of all messages sent to the corresponding variable node, i.e.,

$$\tilde{f}(\mathbf{x}_k) \propto \beta(\mathbf{x}_k) \prod_{j=1}^J \tilde{t}(\mathbf{x}_k; \mathbf{z}_k^{(j)}) \quad (4.35)$$

$$\tilde{f}(\mathbf{y}_{k,n}^{(j)}) \propto \alpha(\mathbf{y}_{k,n}^{(j)}) \tilde{\kappa}(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)}) \quad (4.36)$$

$$\tilde{f}(\eta_k^{(j)}) \propto \xi(\eta_k^{(j)}) \tilde{v}(\eta_k^{(j)}; \mathbf{z}_k^{(j)}). \quad (4.37)$$

Computation of these beliefs involves proper normalization such that they integrate and sum to one. The calculated beliefs can then be used for declaration and state estimation as discussed in Section 4.2.4.

4.4 Particle-Based Implementation

In general, it is not possible to calculate the beliefs and BP messages discussed in Section 4.3 in closed form. This is because the integrations in BP message passing equations (4.21)–(4.24) and (4.31)–(4.34) as well as message multiplication in belief calculations in (4.35)–(4.37) cannot be performed analytically.

Hence, in this section, we present a computationally feasible particle-based implementation [21, 26, 60] of the proposed BP method. Here, the beliefs $\tilde{f}(\mathbf{x}_k)$, $\tilde{f}(\mathbf{y}_{k,n}^{(j)}) = \tilde{f}(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)})$, and $\tilde{f}(\eta_k^{(j)})$ are represented by sets of weighted particles $\{(\mathbf{x}_k^{(p)}, w_{\mathbf{x},k}^{(p)})\}_{p=1}^P$, $\{(\boldsymbol{\phi}_{k,n}^{(j,p)}, w_{\mathbf{y},k,n}^{(j,p)})\}_{p=1}^P$, $n \in \{1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ and $\{(\eta_k^{(j,p')}, w_{\eta,k}^{(j,p')})\}_{p'=1}^{P'}$, $j \in \{1, \dots, J\}$. To keep the computational complexity linear in the number of particles, we adopted an approach introduced in [171] where the particles representing incoming BP messages for each agent-PF pair are stacked

into a joint incoming message. This stacking approach requires that the agent state and the PF states are represented by the same number of particles, P . The measurement noise variance can be represented by a different number of particles P' . Note that this stacking approach is asymptotically optimal [171]. The weights representing PF states, $w_{\mathbf{y},k,n}^{(j,p)}$, do not sum to one, instead we have $p_{k,n}^{(j)} = \sum_{p=1}^P w_{\mathbf{y},k,n}^{(j,p)} \approx \int \tilde{f}(\phi_{k,n}^{(j)}, 1) d\phi_{k,n}^{(j)}$, which approximately equals to the existence probability $f(r_{k,n}^{(j)} = 1 | \mathbf{z}_{1:k})$.

4.4.1 Prediction and Birth

At each time step k , we have sets of weighted particles $\{(\mathbf{x}_{k-1}^{(p)}, w_{\mathbf{x},k-1}^{(p)})\}_{p=1}^P$, $\{(\phi_{k-1,n}^{(j,p)}, w_{\mathbf{y},k-1,n}^{(j,p)})\}_{p=1}^P$, $n \in \{1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$ and $\{(\eta_{k-1}^{(j,p')}, w_{\eta,k-1}^{(j,p')})\}_{p'=1}^{P'}$, $j \in \{1, \dots, J\}$ representing beliefs at the previous step $k-1$. Since the messages $\beta(\mathbf{x}_k)$, $\alpha(\mathbf{y}_{k,n}^{(j)})$, and $\xi(\eta_k^{(j)})$ are PDFs as discussed in Section 4.3.1, they can be represented with weighted particles $\{(\mathbf{x}_k^{(p)}, w_{\beta,k}^{(p)})\}_{p=1}^P$, $\{(\phi_{k,n}^{(j,p)}, w_{\alpha,k,n}^{(j,p)})\}_{p=1}^P$, and $\{(\eta_k^{(j,p')}, w_{\xi,k}^{(j,p')})\}_{p'=1}^{P'}$. These new sets of particles can be obtained as follows [21, 26, 60].

For the agent state, one particle $\mathbf{x}_k^{(p)}$, $p \in \{1, \dots, P\}$ is drawn from $f(\mathbf{x}_k | \mathbf{x}_{k-1}^{(p)})$ for each $\mathbf{x}_{k-1}^{(p)}$. Similarly, for the noise covariance matrix, one particle $\eta_k^{(j,p')}$, $p' \in \{1, \dots, P'\}$ is drawn from $f(\eta_k^{(j)} | \eta_{k-1}^{(j,p')})$ for each $\eta_{k-1}^{(j,p')}$. Their weights remain unchanged, i.e., $w_{\beta,k}^{(p)} = w_{\mathbf{x},k-1}^{(p)}$ and $w_{\xi,k}^{(j,p')} = w_{\eta,k-1}^{(j,p')}$. For legacy PF states, one particle $\phi_{k,n}^{(j,p)}$, $p \in \{1, \dots, P\}$, $n \in \{1, \dots, N_{k-1}^{(j)}\}$ is drawn from $f(\phi_{k,n}^{(j)} | \phi_{k-1,n}^{(j,p)})$ for each $\phi_{k-1,n}^{(j,p)}$, and the particle weights is multiplied by the survival probability, i.e., $w_{\alpha,k,n}^{(j,p)} = p_s w_{\mathbf{y},k-1}^{(j,p)}$. The predicted existence probability of a PF state is thus $\sum_{p=1}^P w_{\alpha,k,n}^{(j,p)} = p_s p_{k-1,n}^{(j)}$. Finally, for new PF states $n \in \{N_{k-1}^{(j)} + 1, \dots, N_k^{(j)}\}$, one particle $\phi_{k,n}^{(j,p)}$, $p \in \{1, \dots, P\}$ is drawn from $f_{B,n}^{(j)}(\phi_{k,n}^{(j)} | \mathbf{x}_k^{(p)})$ for each $\mathbf{x}_k^{(p)}$, and the weights are set to $w_{\alpha,k,n}^{(j,p)} = p_{B,n}^{(j)} / P$.

4.4.2 Measurement Update and Belief Calculation

As discussed in Section 4.6.1, we approximate the measurement update messages in (4.25)–(4.27) with zero-mean complex Gaussian PDFs. Thus, for each particle, we only need to

compute approximations of the covariance matrices in (4.28)–(4.30) as follows

$$\begin{aligned}\tilde{\mathbf{C}}_{\iota,k}^{(j)}(\mathbf{x}_k^{(p)}) &= \sum_{n=1}^{N_k^{(j)}} \tilde{\mathbf{C}}_{1,k,n}^{(j)}(\mathbf{x}_k^{(p)}) + \tilde{\eta}_{\xi,k}^{(j)} \mathbf{I}_M \\ \tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j,p)}) &= r_{k,n}^{(j)} \tilde{\mathbf{C}}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j,p)}) + \sum_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \tilde{\mathbf{C}}_{3,k,n'}^{(j)} + \tilde{\eta}_{\xi,k}^{(j)} \mathbf{I}_M \\ \tilde{\mathbf{C}}_{\nu,k}^{(j,p')} &= \eta_k^{(j,p')} \mathbf{I}_M + \sum_{n=1}^{N_k^{(j)}} \tilde{\mathbf{C}}_{3,k,n}^{(j)}.\end{aligned}$$

Recall that for $r_{k,n}^{(j)} = 0$, $\tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j)}) = \tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)})$ is independent of $\boldsymbol{\phi}_{k,n}^{(j)}$. The resulting constant is denoted by $\tilde{\mathbf{C}}_{\kappa,k,n}^{(j)} = \tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j,p)}, r_{k,n}^{(j)} = 0)$.

Here, the covariance in (4.31)–(4.34) are performed by means on Monte Carlo integration [176]

$$\tilde{\mathbf{H}}_{k,n}^{(j,p)} = \gamma_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)H} \quad (4.38)$$

$$\tilde{\mathbf{C}}_{1,k,n}^{(j)}(\mathbf{x}_n^{(p)}) = \left(\sum_{p'=1}^P w_{\alpha,k,n}^{(j,p')} \right) \tilde{\mathbf{H}}_{k,n}^{(j,p)} \quad (4.39)$$

$$\tilde{\mathbf{C}}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j,p)}) = \tilde{\mathbf{H}}_{k,n}^{(j,p)} \quad (4.40)$$

$$\tilde{\mathbf{C}}_{3,k,n}^{(j)} = \sum_{p=1}^P w_{\beta,k}^{(p)} \left(\sum_{p'=1}^P w_{\alpha,k,n}^{(j,p')} \right) \tilde{\mathbf{H}}_{k,n}^{(j,p)} \quad (4.41)$$

$$\tilde{\eta}_{\xi,k}^{(j)} = \sum_{p'=1}^{P'} w_{\xi,k}^{(j,p')} \eta_k^{(j,p')} \quad (4.42)$$

where $\mathbf{h}_{k,n}^{(j,p)} = \mathbf{h}(\tau_{k,n}^{(j,p)})$ is the contribution vector with delay $\tau_{k,n}^{(j,p)} = \|\mathbf{p}_k^{(p)} - \mathbf{p}_{k,n}^{(j,p)}\|/c$ (cf. Section 4.2.2) evaluated at the “stacked” particle $(\mathbf{x}_k^{(p)}, \boldsymbol{\phi}_{k,n}^{(j,p)})$. Details on how (4.38)–(4.42) are obtained based on the stacking approach [171] are provided in [175, Sec. 2 of the supporting document]. We now compute a particle-based representation of (4.35)–(4.37) by means of importance sampling. In particular, for $p \in \{1, \dots, P\}$, we first calculate unnormalized weights

as

$$\tilde{w}_{\mathbf{x},k}^{(p)} = w_{\beta,k}^{(p)} \prod_{j=1}^J \mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \tilde{\mathbf{C}}_{\iota,k}^{(j)}(\mathbf{x}_k^{(p)})) \quad (4.43)$$

$$\tilde{w}_{\mathbf{y},k,n}^{(j,p)} = w_{\alpha,k,n}^{(j,p)} \mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j,p)}, r_{k,n}^{(j)} = 1)). \quad (4.44)$$

We also introduce the constant $\tilde{w}_{\mathbf{y},k,n}^{(j)} = \mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \tilde{\mathbf{C}}_{\kappa,k,n}^{(j)})$. Similarly, for $p' \in \{1, \dots, P'\}$, we obtain unnormalized weights as

$$\tilde{w}_{\eta,k}^{(j,p')} = w_{\xi,k}^{(j,p')} \mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \tilde{\mathbf{C}}_{\nu,k}^{(j,p')}). \quad (4.45)$$

Here, we have replaced $\tilde{l}(\mathbf{x}_k; \mathbf{z}_k^{(j)})$, $\tilde{\kappa}(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)})$, and $\tilde{\nu}(\mathbf{C}_{\epsilon,k}^{(j)}; \mathbf{z}_k^{(j)})$ in (4.35)–(4.37), by $\mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \tilde{\mathbf{C}}_{\iota,k}^{(j)}(\mathbf{x}_k^{(p)}))$, $\mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \tilde{\mathbf{C}}_{\kappa,k,n}^{(j,p)}(\mathbf{y}_{k,n}^{(j,p)}))$, and $\mathcal{CN}(\mathbf{z}_k^{(j)}; \mathbf{0}, \tilde{\mathbf{C}}_{\nu,k}^{(j,p')})$, respectively.

Next, the weighted particles $\{(\mathbf{x}_k^{(p)}, w_{\mathbf{x},k}^{(p)})\}_{p=1}^P$ and $\{(\eta_k^{(j,p')}, w_{\eta,k}^{(j,p')})\}_{p'=1}^{P'}$ representing the beliefs $\tilde{f}(\mathbf{x}_k)$ and $\tilde{f}(\eta_k^{(j)})$ are obtained by normalizing their corresponding weights, i.e., $w_{\mathbf{x},k}^{(p)} = \tilde{w}_{\mathbf{x},k}^{(p)} / \sum_{p=1}^P \tilde{w}_{\mathbf{x},k}^{(p)}$ and $w_{\eta,k}^{(j,p')} = \tilde{w}_{\eta,k}^{(j,p')} / \sum_{p'=1}^{P'} \tilde{w}_{\eta,k}^{(j,p')}$. For PF states, the particle representations $\{(\boldsymbol{\phi}_{k,n}^{(j,p)}, w_{\mathbf{y},k,n}^{(j,p)})\}_{p=1}^P$ of the beliefs $\tilde{f}(\mathbf{y}_{k,n}^{(j)}) = \tilde{f}(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)})$ are obtained in a similar way. Here, the sum of weights represents the existence probability of the corresponding PF, i.e., $p_{k,n}^{(j)} = \sum_{p=1}^P w_{\mathbf{y},k,n}^{(j,p)} \approx \int \tilde{f}(\boldsymbol{\phi}_{k,n}^{(j)}, 1) d\boldsymbol{\phi}_{k,n}^{(j)}$. Properly normalized weights are obtained as (cf. [60, Sec. VI-C])

$$w_{\mathbf{y},k,n}^{(j,p)} = \frac{\tilde{w}_{\mathbf{y},k,n}^{(j,p)}}{\sum_{p=1}^P \tilde{w}_{\mathbf{y},k,n}^{(j,p)} + \tilde{w}_{\mathbf{y},k,n}^{(j)} (1 - \sum_{p=1}^P w_{\alpha,k,n}^{(j,p)})}.$$

Here, the denominator is a particle-based computation of the normalization constant $\sum_{r_{k,n}^{(j)} \in \{0,1\}} \int \alpha(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)}) \tilde{\kappa}(\boldsymbol{\phi}_{k,n}^{(j)}, r_{k,n}^{(j)}; \mathbf{z}_k^{(j)}) d\boldsymbol{\phi}_{k,n}^{(j)}$. To avoid particle degeneracy, a resampling step [21, 26] is performed for the agent and each PF at each time step.

Based on the probabilities $p_{k,n}^{(j)}$, $n \in \{1, \dots, N_k^{(j)}\}$, $j \in \{1, \dots, J\}$, the existence of PFs can now be determined. In particular, if $p_{k,n}^{(j)}$ is larger than the threshold T_{dec} , PF with indexes (j, n) is declared to exist (cf. Section 4.2.4). An approximation of the MMSE estimates in

(4.17)-(4.18) at time k for agent states and PFs declared to exist can now be computed as

$$\hat{\mathbf{x}}_k = \sum_{p=1}^P w_{\mathbf{x},k}^{(p)} \mathbf{x}_k^{(p)}$$

$$\hat{\phi}_{k,n}^{(j)} = \frac{1}{P_{k,n}^{(j)}} \sum_{p=1}^P w_{\mathbf{y},k,n}^{(j,p)} \phi_{k,n}^{(j,p)}$$

and the MMSE estimate of measurement noise variance in (4.19) can be approximated as

$$\hat{\eta}_k^{(j)} = \sum_{p'=1}^{P'} w_{\eta,k}^{(j,p')} \eta_k^{(j,p')}.$$

The number of PFs increases with each time step. To limit computational complexity, we prune PF states that do not represent the PA and have a low existence probability. In particular, if $p_{k,n}^{(j)}$ for $n \in \{2, \dots, N_k^{(j)}\}$ and $j \in \{1, \dots, J\}$ is lower than a certain threshold T_{pru} , the corresponding PF state is removed from the state space.

4.4.3 Complexity Analysis

The computational complexity of our particle-based implementation is analyzed in what follows. To simplify the notation, it is assumed that $P = P'$. The computation of prediction and birth messages only involves drawing particles and multiplying constants to weights for the agent, each PF, and each measurement noise variance individually. It can easily be verified that the computational complexity of these operations scales as $O(P(N_k + J))$, where $N_k = \sum_{j=1}^J N_k^{(j)}$ is the total number of PFs. The computation of the measurement update messages and beliefs includes the covariance matrix calculations and the Gaussian PDF evaluations. The complexity of the operations in (4.38)–(4.42) needed for the computation of covariance matrices is $O(PN_k M^2)$. For complex Gaussian PDF evaluations in (4.43)–(4.45), the inverses of $\tilde{\mathbf{C}}_{i,k}^{(j)}(\mathbf{x}_k^{(p)})$, $\tilde{\mathbf{C}}_{\kappa,k,n}^{(j,p)}(\mathbf{y}_{k,n}^{(p)})$, and $\tilde{\mathbf{C}}_{v,k}^{(j,p)}$ are needed for each particle and each PF. The complexity of these operations scales as $O(PN_k M^3)$. Thus, the overall complexity of the proposed particle-based BP algorithm scales as $O(PN_k M^3)$, i.e., linearly in the number of map features and cubically in the number of measurements. To further reduce computational complexity, we have developed an alternative approximate computation of BP messages. The complexity of the

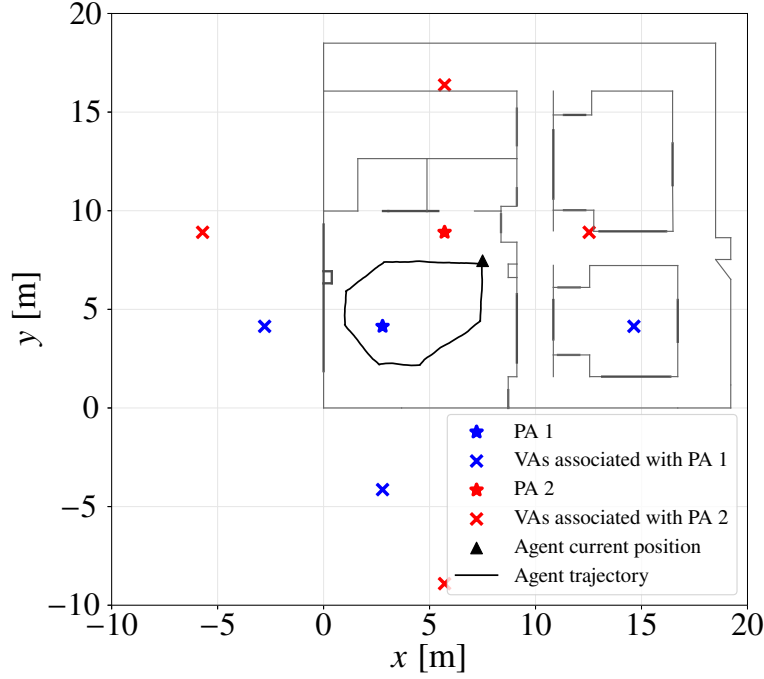


Figure 4.4. The considered floor plan for performance evaluation. Stars indicate the PAs and the black line indicates the agent trajectory. The black triangle is the position of the agent at the current time step $k = 1$. The crosses are the first-order VAs and their colors indicate the PAs they associate with.

resulting BP message passing method scales quadratically in the number of measurements M and is thus particularly useful if the signal bandwidth is high. The alternative approximation and resulting BP messages are presented in [175, Sec. 3 of the supporting document]. Experimental results show that direct SLAM with the alternative approximation performs almost identical to the original SLAM method but yields significantly lower runtime when the signal bandwidth is high [175, Sec. 3 of the supporting document].

4.5 Numerical Results

In this section, to analyze the performance of the proposed direct SLAM method, we present numerical results using synthetic and real data in 2-D scenarios.

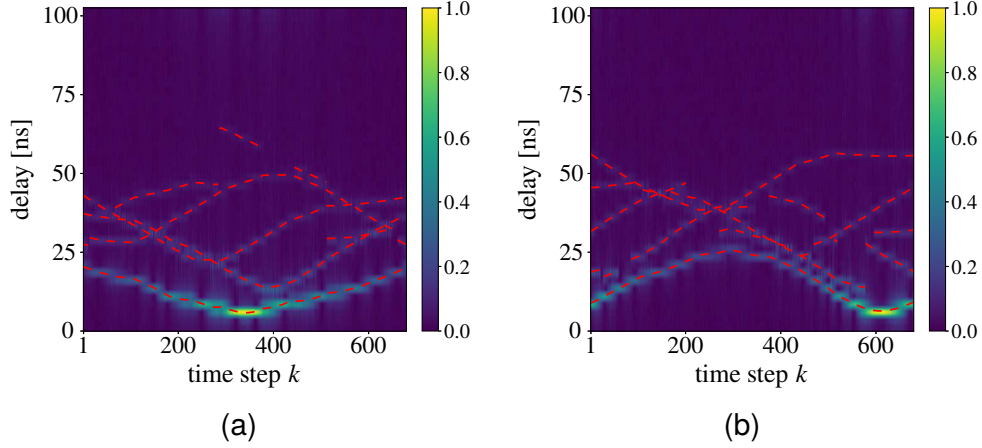


Figure 4.5. The magnitude $|\tilde{z}_{k,m}^{(j)}|$ of synthetic data with 400MHz bandwidth normalized to $[0, 1]$ from PA (a) $j = 1$ and (b) $j = 2$. The red dashed lines are the delay between the agent and ground truth features (PAs plus VAs). VA positions are provided by ray tracing.

4.5.1 Experimental Setup

System Model

The agent state $\mathbf{x}_k = [\mathbf{p}_k^T \mathbf{v}_k^T]^T \in \mathbb{R}^4$ consists of the agent's 2-D position $\mathbf{p}_k$ and 2-D velocity $\mathbf{v}_k$. Its state-transition function $f(\mathbf{x}_k|\mathbf{x}_{k-1})$ is obtained from a constant-velocity motion model [108, Sec. 4]. A Gamma distribution is used for the dynamics of the measurement noise variance, i.e., $f(\eta_k^{(j)}|\eta_{k-1}^{(j)}) = \mathcal{G}(\eta_k^{(j)}; c_\eta, \eta_{k-1}^{(j)}/c_\eta)$, which is of mean $\eta_{k-1}^{(j)}$ and variance $(\eta_{k-1}^{(j)})^2/c_\eta$.

The evolution of the PF position $\mathbf{p}_{k,n}^{(j)}$ and intensity $\gamma_{k,n}^{(j)}$ are assumed to be independent, i.e., $f(\boldsymbol{\phi}_{k,n}^{(j)}|\boldsymbol{\phi}_{k-1,n}^{(j)}) = f(\mathbf{p}_{k,n}^{(j)}|\mathbf{p}_{k-1,n}^{(j)})f(\gamma_{k,n}^{(j)}|\gamma_{k-1,n}^{(j)})$. PFs are assumed static, i.e., the state-transition of PF position would be a Dirac delta function. However, for the sake of robustness of our particle-based implementation, we use the random walk model with a small driving noise, i.e., $f(\mathbf{p}_{k,n}^{(j)}|\mathbf{p}_{k-1,n}^{(j)}) = \mathcal{N}(\mathbf{p}_{k,n}^{(j)}; \mathbf{p}_{k-1,n}^{(j)}, \sigma_{\mathbf{p},n}^2 \mathbf{I}_2)$. The state transition of the intensity is modeled by a random walk model $f(\gamma_{k,n}^{(j)}|\gamma_{k-1,n}^{(j)}) = \mathcal{N}(\gamma_{k,n}^{(j)}; \gamma_{k-1,n}^{(j)}, \sigma_\gamma^2)$. For new PFs, the birth of position and intensity are also assumed to be independent, i.e., $f_{B,n}^{(j)}(\boldsymbol{\phi}|\mathbf{x}_k) = f_{B,n}^{(j)}(\mathbf{p}|\mathbf{x}_k)f_{B,n}^{(j)}(\gamma|\mathbf{x}_k)$, where $f_{B,n}^{(j)}(\mathbf{p}|\mathbf{x}_k)$ is the uniform PDF over $\mathcal{P}_m(\mathbf{x}_k)$ (cf. (4.9)) and $f_{B,n}^{(j)}(\gamma|\mathbf{x}_k) = f_{B,n}^{(j)}(\gamma)$ is the uniform PDF over $[0, 2]$.

Parameters and Implementation

We consider an indoor environment with two PAs, and the floor plan is shown in Fig. 4.4. There are 679 time steps in total and the agent track is depicted by the black line. The driving noise variance of the agent state is set to $\sigma_{\mathbf{x}}^2 = 10^{-4} \text{ m}^2/\text{s}^2$. The driving noise variance for legacy PF positions $\sigma_{\mathbf{p},n}^2$ is set to 10^{-8} m^2 for PAs, i.e., $n = 1$, and is set to $9 \times 10^{-6} \text{ m}^2$ for other VAs, i.e., $n \geq 2$. PAs have lower driving noise variance as their positions are known. The driving noise variance for legacy PFs' intensity is $\sigma_{\gamma}^2 = 10^{-4}$ and the survival probability in (4.8) is $p_s = 0.999$. The birth probability in (4.10) is set to $p_{\text{B},n}^{(j)} = 10^{-4}$ for all new PFs. The state-transition parameter for noise variance is $c_{\eta} = 10$. In addition, we set the declaration threshold to $T_{\text{dec}} = 0.5$ and the pruning threshold to $T_{\text{pru}} = 10^{-2}$.

For the particle-based implementation, we choose $P = 10^4$ particles for the agent state and PF states, and $P' = 1000$ particles for measurement noise variance. The particles for the initial agent position, $\{\mathbf{p}_0^{(p)}\}_{p=1}^P$, are drawn uniformly from a disk with center $\mathbf{p}_1$ and radius 0.5 m, where $\mathbf{p}_1$ is the true agent position at $k = 1$. The initial agent velocity particles $\{\mathbf{v}_0^{(p)}\}_{p=1}^P$ are uniformly drawn from $[-0.01, 0.01] \text{ m/s}$ in both dimensions. The initial agent state particles are uniformly weighted, i.e., $w_{\mathbf{x},0}^{(p)} = 1/P$. The particles for initial PA positions $\{\mathbf{p}_{0,1}^{(j,p)}\}_{p=1}^P$ are sampled from a Gaussian PDF $\mathcal{N}(\mathbf{p}_{0,1}^{(j,p)}; \mathbf{p}_1^{(j)}, \sigma_{\mathbf{p},1}^2 \mathbf{I}_2)$ where $\mathbf{p}_1^{(j)}$ is the position of PA $j \in \{1, 2\}$. The particles for initial PA intensity $\{\gamma_{0,1}^{(j,p)}\}_{p=1}^P$ are drawn uniformly from $[0, 2]$. There is no prior information for VAs. Since the two PAs exist at the beginning for the considered agent track, we set the PA state weights as $w_{\mathbf{y},1}^{(j,p)} = 1/P$. For the measurement noise variance, its initial particles $\{\eta_0^{(j,p')}\}_{p'=1}^{P'}$ are sampled uniformly from 0 to 0.1.

Synthetic Data Generation

For synthetic data, only first-order VAs are considered, i.e., every considered propagation path is reflected at most once from a feature in the environment. The ground truth parameters of propagation paths are computed using ray-tracing [177, 178]. The VAs computed at $k = 1$

are shown in Fig. 4.4, where we have $L_1^{(1)} = 4$ and $L_1^{(2)} = 5$ features corresponds to PA 1 and 2, respectively. The ground truth VAs are used exclusively for data generation and performance evaluation, and are not by the SLAM methods. The transmitted signal spectrum $S(f)$ has a root-raised-cosine shape with a roll-off factor of 0 (flat spectrum) and a bandwidth B (specified in Sections 4.5.2 and 4.5.3). We define the maximum observation distance as $d_{\max} = 30$ m, resulting in a frequency spacing of $\Delta = 10$ MHz. The complex amplitudes $\varrho_{k,l}^{(j)}$ are generated according to (4.3), where the magnitude $a_{l,k}^{(j)} = 1$ for PAs ($l = 1$) and $a_l^{(j)} = 0.7$ for VAs ($l \geq 2$). The phase of $a_{l,k}^{(j)}$ is drawn uniformly from $[0, 2\pi)$. The SNR output at 1 m distance to the mobile agent is set to be 42 dB resulting in $10^{-4.2}$ measurement noise variance.

To reduce computational complexity and increase robustness, instead of introducing all the M new PFs at each time step, we here only establish a new PF over a subset $\mathcal{N}_k^{(j)} \subseteq \{N_{k-1}^{(j)} + 1, \dots, N_{k-1}^{(j)} + M\}$. To do that, we first compute $\tilde{z}_{k,m}^{(j)} = \mathbf{h}^H(\tau_m) \mathbf{z}_k^{(j)}$ with $\tau_m = (m-1)T_s$ and $\mathbf{h}(t)$ introduced in (4.1). Note that the positive scalar values $|\tilde{z}_{k,m}^{(j)}|$, $m \in \{1, \dots, M\}$ represent the 1-D magnitude spectrum over the delay grid $[\tau_1 \cdots \tau_M]^T$. A new PF $n \in \mathcal{N}_k^{(j)}$, corresponding to $\tilde{z}_{k,m}^{(j)}$, $m = n - N_{k-1}^{(j)}$ and occupying the region $\mathcal{P}_m(\mathbf{x}_k)$ defined in (4.9), must satisfy the following two conditions: (i) the absolute value $|\tilde{z}_{k,m}^{(j)}|$ is larger than a certain threshold, $\gamma_{\text{init}}^{(j)}$, and a local maximum with respect to m ; (ii) the delay satisfies $(m-1)T_s \cdot c < \min_p \|\mathbf{p}_k^{(p)} - \mathbf{p}_{k,1}^{(j,p)}\|$ or $(m-1)T_s \cdot c > \max_p \|\mathbf{p}_k^{(p)} - \mathbf{p}_{k,1}^{(j,p)}\|$. Here, $\|\mathbf{p}_k^{(p)} - \mathbf{p}_{k,1}^{(j,p)}\|$ is the Euclidean distance between the agent and PA particles. The first condition ensures that only the peaks of the spectrum with values exceeding the threshold are selected. The second condition ensures that no PF with a delay similar to the PA is introduced. The threshold is set to $\gamma_{\text{init},k}^{(j)} = \sqrt{\hat{\eta}_{k-1}^{(j)}} \cdot 10$.

4.5.2 Results of Synthetic Data

We perform 100 simulation runs for synthetic data, i.e., we generate 100 different measurements $\mathbf{z}_k^{(j)}$, $j \in \{1, 2\}$ as discussed in Section 4.5.1 and apply different methods independently. A bandwidth of $B = 300, 400$, and 600 MHz is considered. This results in a signal vector $\mathbf{z}_k^{(j)}$, $j \in \{1, 2\}$ with $M = B/\Delta + 1 = 31, 41$, and 61 samples, respectively. A visualization

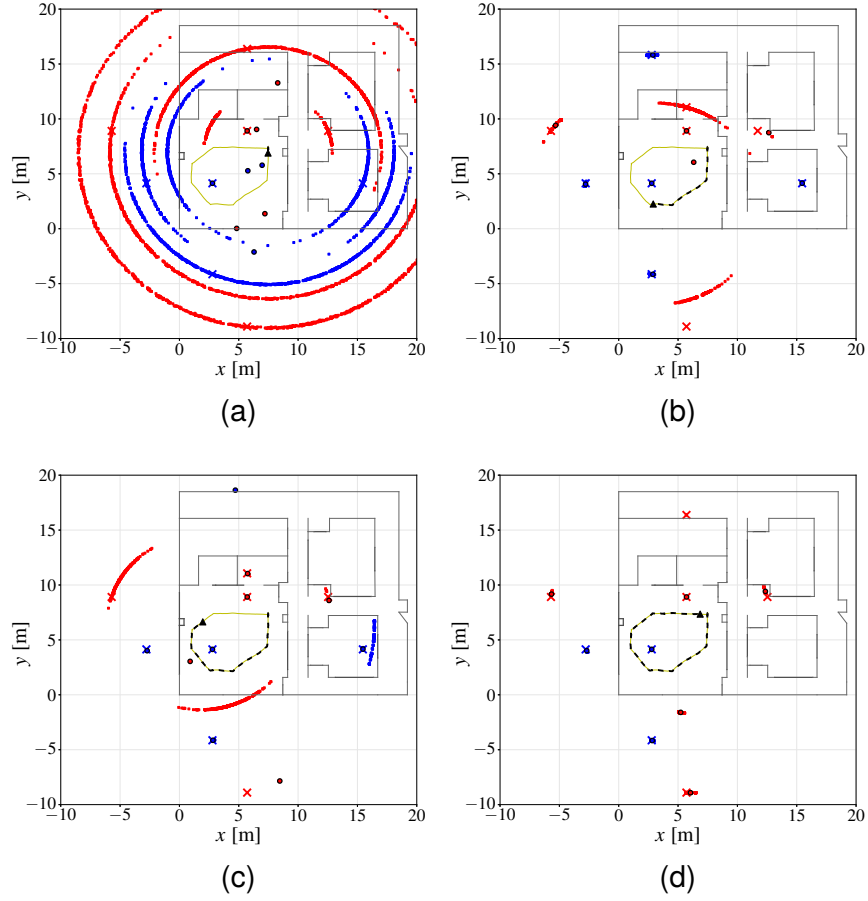


Figure 4.6. Example of a single simulation run using synthetic data with 400MHz bandwidth at times (a) $k = 18$, (b) $k = 282$, (c) $k = 476$, and (d) $k = 655$. Blue and red indicate features associated with PA 1 and PA 2, respectively. Crosses indicate the true positions of features. Bullets indicate the position particles of declared PFs. The yellow line is the true agent trajectory, the black dashed line is the estimated agent trajectory, and the black triangle is the agent position at the current time step.

of $|\tilde{z}_{k,m}^{(j)}|$ with 400MHz bandwidth is shown in Fig. 4.5. The results of a single simulation run are depicted in Fig. 4.6.

For performance evaluation, the following multipath-based SLAM methods are used as benchmarks: “Channel-SLAM” [57], “BP-SLAM” [157] and “BP-SLAM-AI” [179]. All the reference methods use a sparse Bayesian learning channel estimator [161, 162, 166] to estimate MPCs from radio signals. The estimated MPCs are then fed to the downstream algorithm for

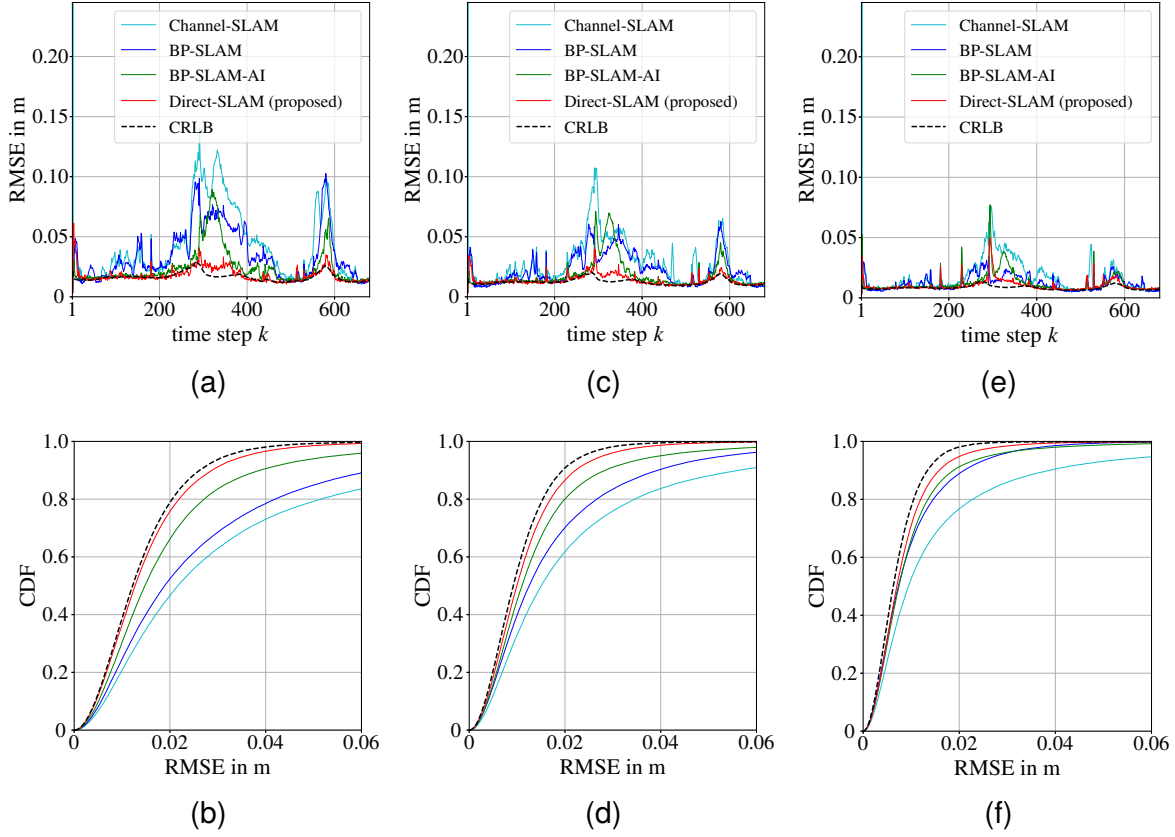


Figure 4.7. Performance of agent localization on synthetic data: (a) Agent position RMSEs averaged over 100 simulation runs and (b) empirical CDFs of the RMSEs with 300 MHz bandwidth. (c)\(d\) and (e)\(f\) are the counterparts of (a)\(b\) for bandwidth 400 MHz and 600 MHz, respectively.

SLAM. Compared with BP-SLAM, in addition to using the delay information, BP-SLAM-AI further exploits the amplitude information in estimated MPCs. For the three considered signal different bandwidths, Fig. 4.7 shows the root mean square errors (RMSEs) of estimated agent positions vs. time step k and the empirical cumulative distribution function (CDF) of RMSEs. The RMSEs of estimated agent positions are averaged over all 100 simulation runs, while the empirical CDFs is also averaged over all time steps. Note that, in the visual SLAM literature [180], the RMSE is also referred to as absolute trajectory error (ATE).

Fig. 4.7 also shows the Cramér-Rao lower bound (CRLB) of the agent localization error. A detailed derivation of the CRLB for the considered multipath-based SLAM problem is

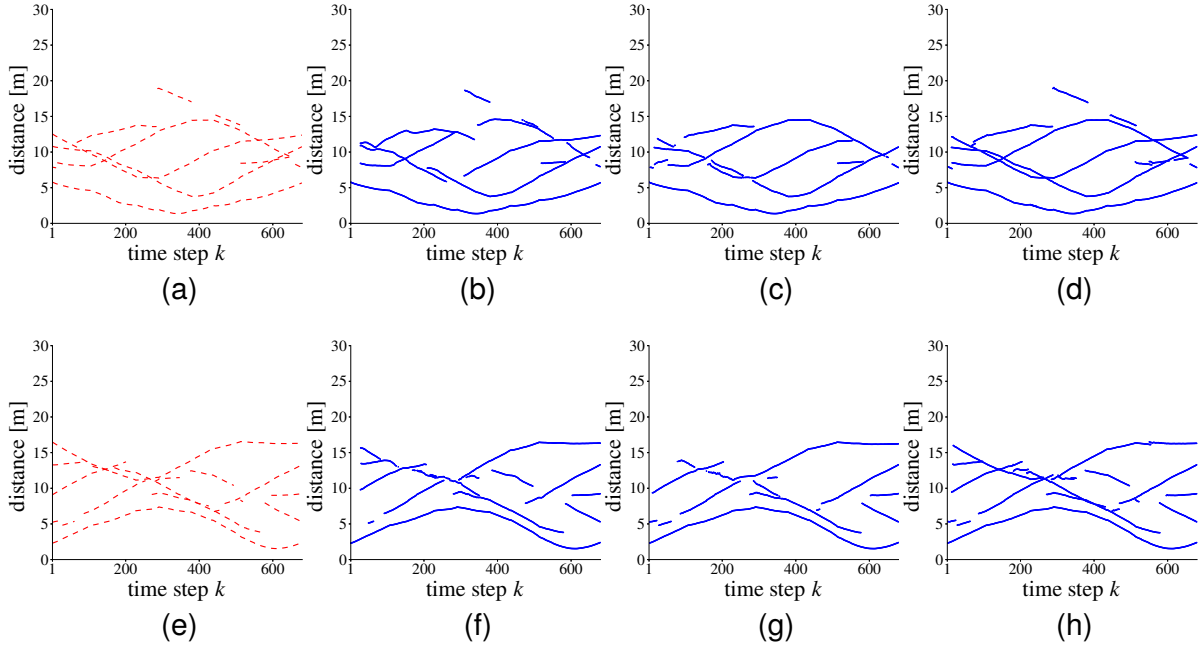


Figure 4.8. Visualization of the distance between the agent position and the true and estimated feature position. The first column with (a)/(e) shows the true distances for PA 1 and PA 2, respectively. The following columns show the averaged distance between the true agent position and the particles of PF positions obtained using (b)/(f) BP-SLAM, (c)/(g) BP-SLAM-AI, and (d)/(h) Direct SLAM with 400MHz bandwidth and for PA 1 and PA 2, respectively.

provided in [181, 182, 183]. Note that, for the different bandwidths, in 3-5 runs of the total 100 runs, Channel-SLAM leads to an “agent track loss”. An agent track loss is a run where the RMSE rises above 1 m and continues to increase. Runs where the application of Channel-SLAM results in an agent track loss are not taken into account in the performance plots provided in Fig. 4.7 (a), (c), and (e). In all 100 runs, the application of BP-SLAM, BP-SLAM-AI, and Direct-SLAM does not lead to an agent track loss. It can be seen that the proposed Direct-SLAM method closely follows the CRLB and outperforms the reference methods in terms of localization error. Especially, the performance gain is significant during time steps $k \in [100, 400]$ and around $k = 600$, where the geometry of the propagation environment is complicated, and many features have similar distances to the agent. We plot the average distance between the particles of the PF positions and the true agent positions for one exemplary simulation run with 400 MHz in Figure 4.8b-4.8d and 4.8f-4.8h. Since BP-SLAM and BP-SLAM-AI rely on a channel estimator

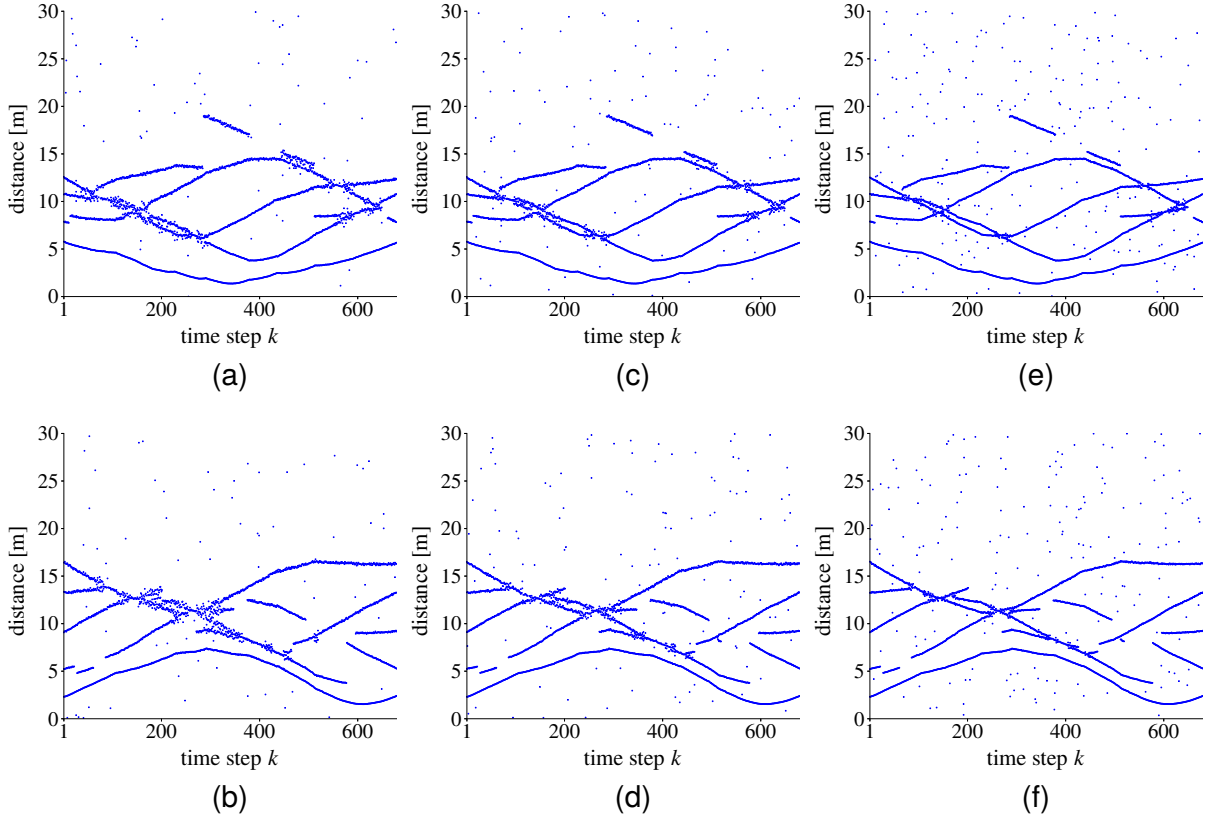


Figure 4.9. Visualization of the range measurements (blue dots) provided by the sparse Bayesian learning estimator for (a) PA 1 and (b) PA 2 with bandwidth 300 MHz. (c)\(d\) and (e)\(f\) are the counterparts of (a)\(b\) for bandwidth 400 MHz and 600 MHz, respectively.

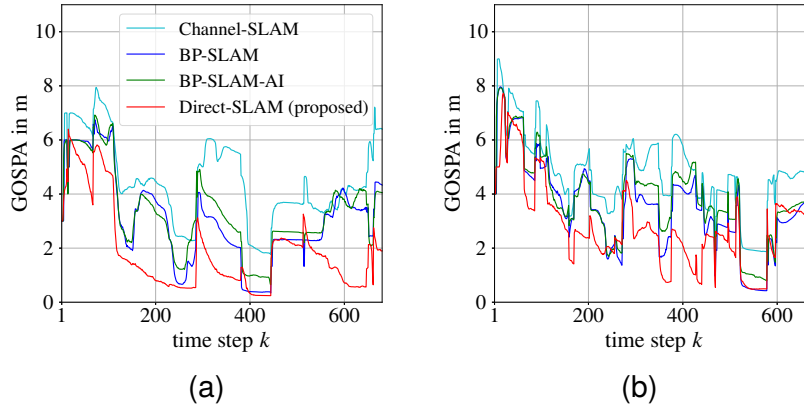


Figure 4.10. Performance of mapping on synthetic data with 400MHz bandwidth: GOSPA error of estimated VAs associated with PA 1 (a) and PA 2 (b), averaged over 100 simulation runs.

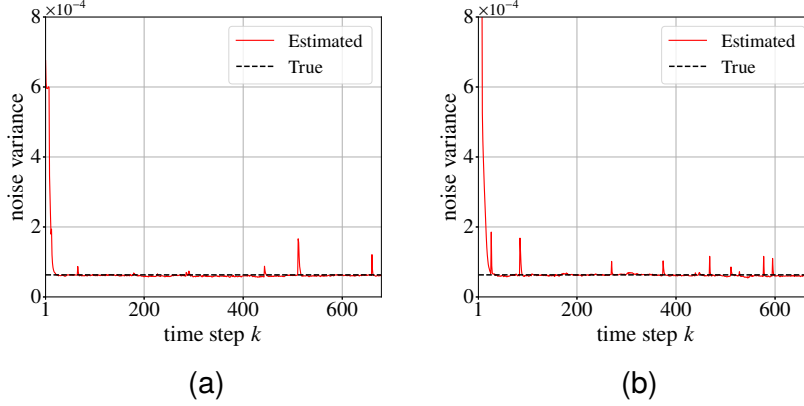


Figure 4.11. True and estimated measurement noise variance $\eta_k^{(j)}$ averaged over 100 simulation runs for PA 1 (a) and PA 2 (b) at 400MHz bandwidth.

that cannot resolve MPCs having similar path lengths, both BP-SLAM and BP-SLAM-AI have problems when features “cross” or have similar distances with respect to the agent position. The range measurements provided by the state-of-the-art sparse Bayesian learning channel parameter estimator are shown in Fig. 4.9. As can be seen, these range measurements are of low quality during time steps $k \in [100, 400]$ and around $k = 600$ when different MPCs result in similar delays. In addition, as can also be seen in Fig. 4.9, this inaccuracy exacerbates as the signal bandwidth decreases, resulting in a reduced localization performance. Conversely, Direct-SLAM directly operates on the radio signal and does consider the superposition of MPCs in the statistical model. For this reason, Direct-SLAM alleviates performance issues in situations with complicated propagation environments.

To quantify the mapping performance, we use the Euclidean distance-based GOSPA metric [114] with cutoff parameter $c = 2$ m and order $p = 1$. Results are shown in Fig. 4.10. The GOSPA error of Direct-SLAM is generally smaller than the reference methods, and its performance gain is large in challenging situations, e.g., during $k \in [100, 400]$. Videos of this experiment that indicated a faster mapping performance of Direct-SLAM compared to BP-SLAM can be found in [184]. Fig. 4.11 compares the true and estimated measurement noise variance at all time steps. The estimated variance is large at the beginning as the particles are initialized over a large interval $[0, 0.1]$, then it quickly converges to the true value.

Table 4.1. Average runtime of different methods per time step

Method	Time (s)		
	300MHz	400MHz	600MHz
BP-SLAM [19]	1.03	1.04	1.04
BP-SLAM-AI [179]	1.06	1.06	1.07
Direct-SLAM (proposed)	1.19	1.53	2.58

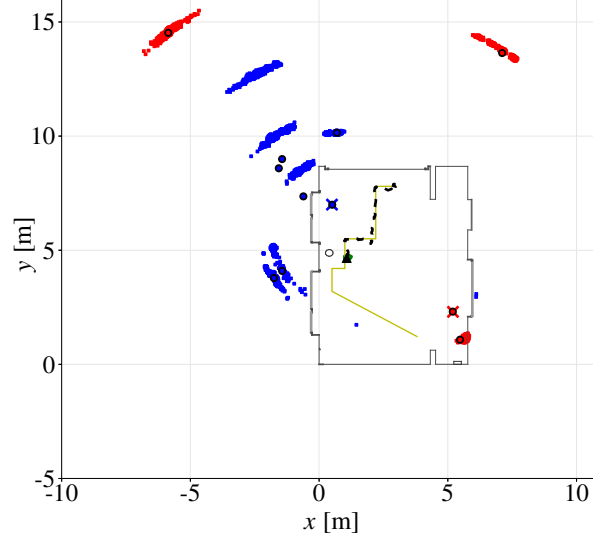


Figure 4.12. Results using real data for $k = 102$. Blue and red indicate features associated with the two PAs. Crosses depict the positions of PAs. Bullets are particles of declared PF positions, and circles with black edges are estimated positions of declared PFs. The yellow line is the expected agent trajectory, the black dashed line is the estimated agent trajectory, and the black triangle is the estimated agent position at the current time step.

All experiments are conducted on a single core of Intel Xeon Gold 5222 CPU. The average computation time of different methods for each time step is reported in Table 4.1. Note the runtime of BP-SLAM and BP-SLAM-AI includes the 0.98, 0.99, and 0.99 s processing time of the fast variational sparse Bayesian learning channel estimator [161] for 300, 400, and 600 MHz bandwidth, respectively. The runtime of Direct-SLAM is somewhat higher than the ones of the two reference methods. For 300 MHz, where Direct-SLAM yields the most significant performance gain. Here, the runtime is only increased by 15.5% compared to reference methods.

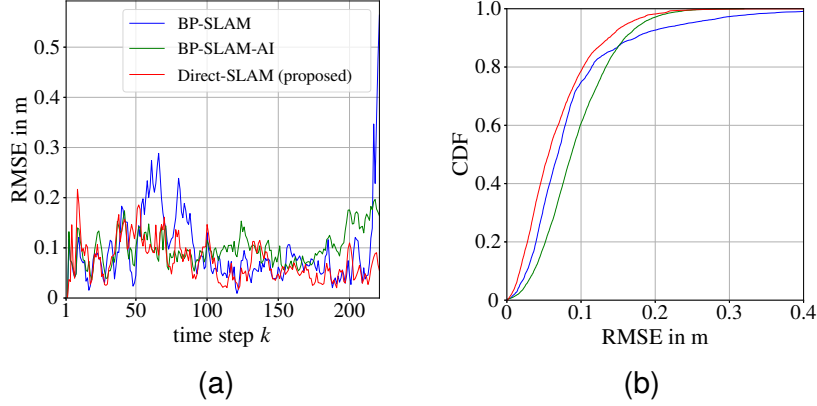


Figure 4.13. Performance of agent localization on real data: (a) Agent position RMSEs averaged over 30 simulation runs and (b) empirical CDFs of the RMSEs.

4.5.3 Results of Real Data

The performance of the proposed method has been evaluated on real data collected in a classroom previously used in [185]. The floor plan is shown in Fig. 4.12. The two crosses are the position of the two PAs and the yellow line is the agent trajectory. The trajectory consist of 220 agent positions with 0.05 m spacing along the trajectory. This signal was measured using an M-sequence correlative channel sounder with frequency range 3 – 10 GHz and antennas with an approximately uniform radiation pattern in the azimuth plane and zeros in the floor and ceiling directions. Within the measured band, the actual signal band was selected by a filter with rooted-raised-cosine impulse response with a roll-off factor of 0, a bandwidth of $B = 400$ MHz, and a center frequency of $f_c = 6$ GHz, leading to measurements $\mathbf{z}_k^{(j)}$, $j \in \{1, 2\}$ with $M = 41$.

For the experiments running on real data, we set the driving noise variance for legacy PF positions $\sigma_{\mathbf{p},n}^2$ to $1 \times 10^{-6} \text{ m}^2$ for PAs, and to $6.4 \times 10^{-5} \text{ m}^2$ for other VAs. The driving noise variance for legacy PFs' intensity is $\sigma_\gamma^2 = 10^{-2}$. In the likelihood function (4.5), we artificially increase the measurement noise variance by a factor of 2. Note that, compared to the simulated scenario in Section 4.5.2, we increase the driving noise variance and the measurement noise variance to capture model mismatch, e.g., due to not explicitly modeling non-specular propagation effects. In this way, we represent the mismatch between our statistical model and

the true data-generating process as an additional noise source. The number of particles of the agent and PF states is $P = 30000$ and the number of particles of noise variances is $P' = 3000$. The survival probability is $p_s = 0.98$. The driving noise variance of the agent state is set to $\sigma_x^2 = 4 \times 10^{-4} \text{ m}^2/\text{s}^2$. All other parameters remain the same.

For the reference method BP-SLAM and BP-SLAM-AI, 150000 particles are used. In total, the data is processed in 30 runs, where only particle realizations are different across runs. The qualitative result of Direct-SLAM at a single time step in a single simulation run is shown in Fig. 4.12. The RMSE of the estimated agent positions at different time steps averaged over all 30 runs, as well as the empirical CDF of individual RMSEs evaluated at all time steps and all simulation runs are shown in Fig. 4.13, and Direct-SLAM outperforms the reference methods with a clear margin. As the reference methods, the proposed method also does not consider non-specular propagation effects such as diffuse scattering and diffraction. Nonetheless, it can achieve robust performance in real scenarios.

4.6 Appendices

4.6.1 Computation of Matrices $\mathbf{C}_{\iota,k}^{(j)}(\mathbf{x}_k)$, $\mathbf{C}_{\kappa,k,n}^{(j)}(\phi_{k,n}^{(j)})$, and $\mathbf{C}_{\nu,k}^{(j)}$

In this section, we present the derivation of the covariance matrices in (4.28)-(4.30). By definition, $\mathbf{C}_{\kappa,k,n}^{(j)}$ is computed as

$$\begin{aligned}
& \mathbf{C}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}) \\
&= \int \mathbf{z}_k^{(j)} \mathbf{z}_k^{(j)\text{H}} \kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)}) \, d\mathbf{z}_k^{(j)} \\
&= \sum_{r_{k,1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,n-1}^{(j)} \in \{0,1\}} \sum_{r_{k,n+1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,N_k}^{(j)} \in \{0,1\}} \\
&\quad \times \int \cdots \int \left(\int \mathbf{z}_k^{(j)} \mathbf{z}_k^{(j)\text{H}} f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)}) \, d\mathbf{z}_k^{(j)} \right) \\
&\quad \times \xi(\eta_k^{(j)}) \beta(\mathbf{x}_k) \prod_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n'}^{(j)}) d\boldsymbol{\phi}_{k,1}^{(j)} \cdots d\boldsymbol{\phi}_{k,n-1}^{(j)} d\boldsymbol{\phi}_{k,n+1}^{(j)} \cdots d\boldsymbol{\phi}_{k,N_k}^{(j)} d\eta_k^{(j)} d\mathbf{x}_k \quad (4.46)
\end{aligned}$$

where $\int \mathbf{z}_k^{(j)} \mathbf{z}_k^{(j)\text{H}} f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)}) \, d\mathbf{z}_k^{(j)} = \mathbf{C}_{k,n}^{(j)} = \eta_k^{(j)} \mathbf{I}_M + \sum_{n=1}^{N_k^{(j)}} r_{k,n}^{(j)} \gamma_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)\text{H}}$ as discussed in Section 4.2.2. Plugging in the result we obtain

$$\begin{aligned}
& \mathbf{C}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}) \\
&= \sum_{r_{k,1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,n-1}^{(j)} \in \{0,1\}} \sum_{r_{k,n+1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,N_k^{(j)}}^{(j)} \in \{0,1\}} \int \cdots \int \left(\eta_k^{(j)} \mathbf{I}_M + \sum_{n=1}^{N_k^{(j)}} r_{k,n}^{(j)} \gamma_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)H} \right) \\
&\quad \times \xi(\eta_k^{(j)}) \beta(\mathbf{x}_k) \prod_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n'}^{(j)}) d\boldsymbol{\phi}_{k,1}^{(j)} \cdots d\boldsymbol{\phi}_{k,n-1}^{(j)} d\boldsymbol{\phi}_{k,n+1}^{(j)} \cdots d\boldsymbol{\phi}_{k,N_k^{(j)}}^{(j)} d\eta_k^{(j)} d\mathbf{x}_k \\
&= \sum_{r_{k,1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,n-1}^{(j)} \in \{0,1\}} \sum_{r_{k,n+1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,N_k^{(j)}}^{(j)} \in \{0,1\}} \int \cdots \int \left(\sum_{n=1}^{N_k^{(j)}} r_{k,n}^{(j)} \gamma_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)H} \right) \\
&\quad \times \beta(\mathbf{x}_k) \prod_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n'}^{(j)}) d\boldsymbol{\phi}_{k,1}^{(j)} \cdots d\boldsymbol{\phi}_{k,n-1}^{(j)} d\boldsymbol{\phi}_{k,n+1}^{(j)} \cdots d\boldsymbol{\phi}_{k,N_k^{(j)}}^{(j)} + \int \eta_k^{(j)} \mathbf{I}_M \xi(\eta_k^{(j)}) d\eta_k^{(j)} \\
&= \sum_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \sum_{r_{k,n'}^{(j)} \in \{0,1\}} \int r_{k,n'}^{(j)} \gamma_{k,n'}^{(j)} \mathbf{h}_{k,n'}^{(j)} \mathbf{h}_{k,n'}^{(j)H} \alpha(\mathbf{y}_{k,n'}^{(j)}) \beta(\mathbf{x}_k) d\boldsymbol{\phi}_{k,n'}^{(j)} d\mathbf{x}_k \\
&\quad + r_{k,n}^{(j)} \gamma_{k,n}^{(j)} \int \mathbf{h}_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)H} \beta(\mathbf{x}_k) d\mathbf{x}_k + \int \eta_k^{(j)} \mathbf{I}_M \xi(\eta_k^{(j)}) d\eta_k^{(j)} \\
&= \sum_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \int \gamma_{k,n'}^{(j)} \mathbf{h}_{k,n'}^{(j)} \mathbf{h}_{k,n'}^{(j)H} \alpha(\boldsymbol{\phi}_{k,n'}, 1) \beta(\mathbf{x}_k) d\boldsymbol{\phi}_{k,n'}^{(j)} d\mathbf{x}_k \\
&\quad + r_{k,n}^{(j)} \gamma_{k,n}^{(j)} \int \mathbf{h}_{k,n}^{(j)} \mathbf{h}_{k,n}^{(j)H} \beta(\mathbf{x}_k) d\mathbf{x}_k + \int \eta_k^{(j)} \mathbf{I}_M \xi(\eta_k^{(j)}) d\eta_k^{(j)} \\
&= r_{k,n}^{(j)} \mathbf{C}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}) + \sum_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \mathbf{C}_{3,k,n'}^{(j)} + \eta_{\xi,k}^{(j)} \mathbf{I}_M
\end{aligned} \tag{4.47}$$

which is the same as (4.28)-(4.30). The matrices $\mathbf{C}_{\iota,k}^{(j)}(\mathbf{x}_k)$ and $\mathbf{C}_{\nu,k}^{(j)}$ can be derived in a similar way.

4.6.2 Monte Carlo Integration of $\mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_k)$, $\mathbf{C}_{2,k,n}^{(j)}(\phi_{k,n}^{(j)})$, and $\mathbf{C}_{3,k,n}^{(j)}$

In this section, we discuss how the “stacking technique” from (4.39)-(4.41) is used within our direct SLAM framework. We start from the “non-stacked” Monte Carlo integration of $\mathbf{C}_{3,k,n}^{(j)}$

$$\mathbf{C}_{3,k,n}^{(j)} \approx \sum_{p=1}^P \sum_{p'=1}^P w_{\beta,k}^{(p)} w_{\alpha,k,n}^{(j,p')} \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \phi_{k,n}^{(j,p')}). \quad (4.48)$$

This “non-stacked” version has complexity $O(P^2)$. To avoid the square complexity, we stack the particles and only evaluate pairs (p, p') , $p = p'$. As a result,

$$\begin{aligned} \mathbf{C}_{3,k,n}^{(j)} &\approx \sum_{p=1}^P w_{\beta,k}^{(p)} \left(\sum_{p'=1}^P w_{\alpha,k,n}^{(j,p')} \right) \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \phi_{k,n}^{(j,p)}) \\ &= \sum_{p=1}^P w_{\beta,k}^{(p)} \left(\sum_{p'=1}^P w_{\alpha,k,n}^{(j,p')} \right) \tilde{\mathbf{H}}_{k,n}^{(j,p)} = \tilde{\mathbf{C}}_{3,k,n}^{(j)}. \end{aligned} \quad (4.49)$$

Similarly, $\mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_k)$ and $\mathbf{C}_{2,k,n}^{(j)}(\phi_{k,n}^{(j)})$ are approximated as

$$\begin{aligned} \mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_n^{(p)}) &\approx \sum_{p'=1}^P w_{\alpha,k,n}^{(j,p')} \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \phi_{k,n}^{(j,p')}) \\ &\approx \left(\sum_{p'=1}^P w_{\alpha,k,n}^{(j,p')} \right) \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \phi_{k,n}^{(j,p)}) \\ &= \left(\sum_{p'=1}^P w_{\alpha,k,n}^{(j,p')} \right) \tilde{\mathbf{H}}_{k,n}^{(j,p)} = \tilde{\mathbf{C}}_{1,k,n}^{(j)}(\mathbf{x}_n^{(p)}) \end{aligned} \quad (4.50)$$

$$\begin{aligned}
\mathbf{C}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j,p)}) &\approx \sum_{p'=1}^P w_{\beta,k}^{(j,p')} \mathbf{H}_{k,n}(\mathbf{x}_k^{(p')}, \boldsymbol{\phi}_{k,n}^{(j,p)}) \\
&\approx \left(\sum_{p'=1}^P w_{\beta,k}^{(j,p')} \right) \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \boldsymbol{\phi}_{k,n}^{(j,p)}) \\
&= \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \boldsymbol{\phi}_{k,n}^{(j,p)}) = \tilde{\mathbf{H}}_{k,n}^{(j,p)} = \tilde{\mathbf{C}}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j,p)}). \tag{4.51}
\end{aligned}$$

4.6.3 Alternative Approximation of Measurement Update Messages

As discussed in Section 4.6.1, the measurement update messages $\iota(\mathbf{x}_k; \mathbf{z}_k^{(j)})$, $\kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)})$, and $\nu(\eta_k^{(j)}; \mathbf{z}_k^{(j)})$ are approximated by Gaussian density functions with respect to $\mathbf{z}_k^{(j)}$, and the covariance matrices of the Gaussian PDFs are computed using moment matching. Since evaluating Gaussian PDFs requires the inverse of the covariance matrices, the computational complexity scales cubically with the number of measurements M . In this section, we provide an alternative approximation that reduces the complexity.

The main idea used for the derivation of alternative measurement update messages is to use the outer product of the mean vector to approximate the correlation (covariance) matrix. Specifically, we now compute an alternative approximation of (4.33) based on the outer product $\mathbf{C}_{3,k,n}^{(j)} = \boldsymbol{\mu}_{3,k,n}^{(j)} \boldsymbol{\mu}_{3,k,n}^{(j)\text{H}}$, where

$$\boldsymbol{\mu}_{3,k,n}^{(j)} = \int \sqrt{\gamma_{k,n}^{(j)}} \mathbf{h}_{k,n}^{(j)} \alpha(\boldsymbol{\phi}_{k,n}^{(j)}, 1) \beta(\mathbf{x}_k) d\boldsymbol{\phi}_{k,n}^{(j)} d\mathbf{x}_k.$$

A particle-based computation of this outer product (4.41) is then computed as $\tilde{\mathbf{C}}_{3,k,n}^{(j)} = \tilde{\boldsymbol{\mu}}_{3,k,n}^{(j)} \tilde{\boldsymbol{\mu}}_{3,k,n}^{(j)\text{H}}$ with

$$\tilde{\boldsymbol{\mu}}_{3,k,n}^{(j)} = \sum_{p=1}^P w_{\beta,k}^{(p)} \left(\sum_{p'=1}^P w_{\alpha,k,n}^{(j,p')} \right) \sqrt{\gamma_{k,n}^{(j,p)}} \mathbf{h}_{k,n}^{(j,p)}.$$

Based on this approximation, the matrix $\tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j,p)})$ now reads

$$\begin{aligned}
\tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j,p)}) &= r_{k,n}^{(j)} \gamma_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)\text{H}} + \sum_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \tilde{\boldsymbol{\mu}}_{3,k,n'}^{(j)} \tilde{\boldsymbol{\mu}}_{3,k,n'}^{(j)\text{H}} + \tilde{\eta}_{\xi,k}^{(j)} \mathbf{I}_M \\
&= r_{k,n}^{(j)} \gamma_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)\text{H}} + \mathbf{B}_{k,n}^{(j)} \mathbf{B}_{k,n}^{(j)\text{H}} + \tilde{\eta}_{\xi,k}^{(j)} \mathbf{I}_M \\
&= r_{k,n}^{(j)} \gamma_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)\text{H}} + \mathbf{A}_{k,n}^{(j)}
\end{aligned} \tag{4.52}$$

with $\mathbf{B}_{k,n}^{(j)} = [\tilde{\boldsymbol{\mu}}_{3,k,1}^{(j)} \cdots \tilde{\boldsymbol{\mu}}_{3,k,n-1}^{(j)} \tilde{\boldsymbol{\mu}}_{3,k,n+1}^{(j)} \cdots \tilde{\boldsymbol{\mu}}_{3,k,N_k^{(j)}}^{(j)}] \in \mathbb{C}^{M \times (N_k^{(j)}-1)}$ and $\mathbf{A}_{k,n}^{(j)} = \mathbf{B}_{k,n}^{(j)} \mathbf{B}_{k,n}^{(j)\text{H}} + \tilde{\eta}_{\xi,k}^{(j)} \mathbf{I}_M$. To evaluate $C\mathcal{N}(\mathbf{z}_k^{(j)}; \mathbf{0}, \tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j,p)}))$ for $\tilde{\mathbf{w}}_{\mathbf{y},k,n}^{(j,p)}$ in (4.44), one needs to compute the quantities $\det(\tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j,p)}))$ and $\mathbf{z}_k^{(j)\text{H}}(\tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j,p)}))^{-1} \mathbf{z}_k^{(j)}$, which can be computed efficiently using the matrix determinant lemma and matrix inversion lemma, respectively, as

$$\begin{aligned}
\det(\tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j,p)})) &= \det(r_{k,n}^{(j)} \gamma_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)\text{H}} + \mathbf{A}_{k,n}^{(j)}) \\
&= \det\left(1 + r_{k,n}^{(j)} \gamma_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)\text{H}} (\mathbf{A}_{k,n}^{(j)})^{-1} \mathbf{h}_{k,n}^{(j,p)}\right) \det(\mathbf{A}_{k,n}^{(j)})
\end{aligned} \tag{4.53}$$

$$\begin{aligned}
\mathbf{z}_k^{(j)\text{H}}(\tilde{\mathbf{C}}_{\kappa,k,n}^{(j)}(\mathbf{y}_{k,n}^{(j,p)}))^{-1} \mathbf{z}_k^{(j)} &= \mathbf{z}_k^{(j)\text{H}}(r_{k,n}^{(j)} \gamma_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)\text{H}} + \mathbf{A}_{k,n}^{(j)})^{-1} \mathbf{z}_k^{(j)} \\
&= \mathbf{z}_k^{(j)\text{H}} (\mathbf{A}_{k,n}^{(j)})^{-1} \mathbf{z}_k^{(j)} - \frac{r_{k,n}^{(j)} \gamma_{k,n}^{(j,p)} |\mathbf{h}_{k,n}^{(j,p)\text{H}} (\mathbf{A}_{k,n}^{(j)})^{-1} \mathbf{z}_k^{(j)}|^2}{1 + r_{k,n}^{(j)} \gamma_{k,n}^{(j,p)} \mathbf{h}_{k,n}^{(j,p)\text{H}} (\mathbf{A}_{k,n}^{(j)})^{-1} \mathbf{h}_{k,n}^{(j,p)}}.
\end{aligned} \tag{4.54}$$

Next, $(\mathbf{A}_{k,n}^{(j)})^{-1}$ can be further factorized again using the matrix inversion lemma

$$\begin{aligned}
(\mathbf{A}_{k,n}^{(j)})^{-1} &= (\mathbf{B}_{k,n}^{(j)} \mathbf{B}_{k,n}^{(j)\text{H}} + \tilde{\eta}_{\xi,k}^{(j)} \mathbf{I}_M)^{-1} \\
&= (\tilde{\eta}_{\xi,k}^{(j)})^{-1} \mathbf{I}_M - (\tilde{\eta}_{\xi,k}^{(j)})^{-1} \mathbf{B}_{k,n}^{(j)} (\mathbf{B}_{k,n}^{(j)\text{H}} \mathbf{B}_{k,n}^{(j)} + \tilde{\eta}_{\xi,k}^{(j)} \mathbf{I}_{N_k^{(j)}-1})^{-1} \mathbf{B}_{k,n}^{(j)\text{H}}
\end{aligned} \tag{4.55}$$

with which the quantities $\mathbf{z}_k^{(j)\text{H}} (\mathbf{A}_{k,n}^{(j)})^{-1} \mathbf{z}_k^{(j)}$, $\mathbf{h}_{k,n}^{(j,p)\text{H}} (\mathbf{A}_{k,n}^{(j)})^{-1} \mathbf{z}_k^{(j)}$, and $\mathbf{h}_{k,n}^{(j,p)\text{H}} (\mathbf{A}_{k,n}^{(j)})^{-1} \mathbf{h}_{k,n}^{(j,p)}$ can

Table 4.2. Average runtime of fast implementation per time step

Method	Time (s)		
	300MHz	400MHz	600MHz
Direct-SLAM	1.19	1.53	2.58
Direct-SLAM-fast	0.74	0.88	1.25

be computed in $\mathcal{O}(N_k^{(j)}M + (N_k^{(j)})^3)$ instead of in $\mathcal{O}(M^3)$. Note that since $\det(\mathbf{A}_{k,n}^{(j)})$ appears in both the numerator and denominator of $\tilde{w}_{\mathbf{y},k,n}^{(j,p)}$ in (4.44) and does thus not need to be computed. The same method can be applied to the computation of $\tilde{w}_{\mathbf{x},k}^{(p)}$ and $\tilde{w}_{\eta,k}^{(j,p')}$ in (4.43) and (4.45). Including the operations of all particles and PFs, as well as the initialization of new PFs, the total complexity is $\mathcal{O}\left(M^2 + P\left(\sum_{j=1}^J (N_k^{(j)})^2 M + \sum_{j=1}^J (N_k^{(j)})^4\right)\right)$ which scales quadratically with the number of measurements M . The number of PFs $N_k^{(j)}$ is typically smaller than ten and having five features is already sufficient to accurately localize the agent. Hence, this alternative approximation has time complexity advantages over the original one in high-bandwidth cases.

Next we repeat the experiments in Section 4.5.2 to show the efficacy of the alternative approximation that we refer to as “Direct-SLAM-fast”. All the parameter settings remain the same. The results are shown in Fig. 4.14, from which we can see that Direct-SLAM-fast has almost identical performance compared with the original Direct-SLAM. The averaged runtime shown in Table 4.2 verifies that Direct-SLAM-fast achieves lower computational complexity in the considered bandwidths.

4.6.4 Qualitative Real Data Results

The real radio measurements have been collected in a room with the floorplan as shown in Fig. 4.4, and the two PAs are in the same positions as well. The mobile agent moves following a track composed of 679 consecutive positions shown as yellow line in Fig. 4.4, yet the accurate position of the agent at each time step is not available. The radio signals are measured with a vector network analyzer (VNA) over a bandwidth of $B = 2$ GHz at a center frequency of $f_c = 3$ GHz. The dipole-like ultra-wideband transmit antenna mounted at the

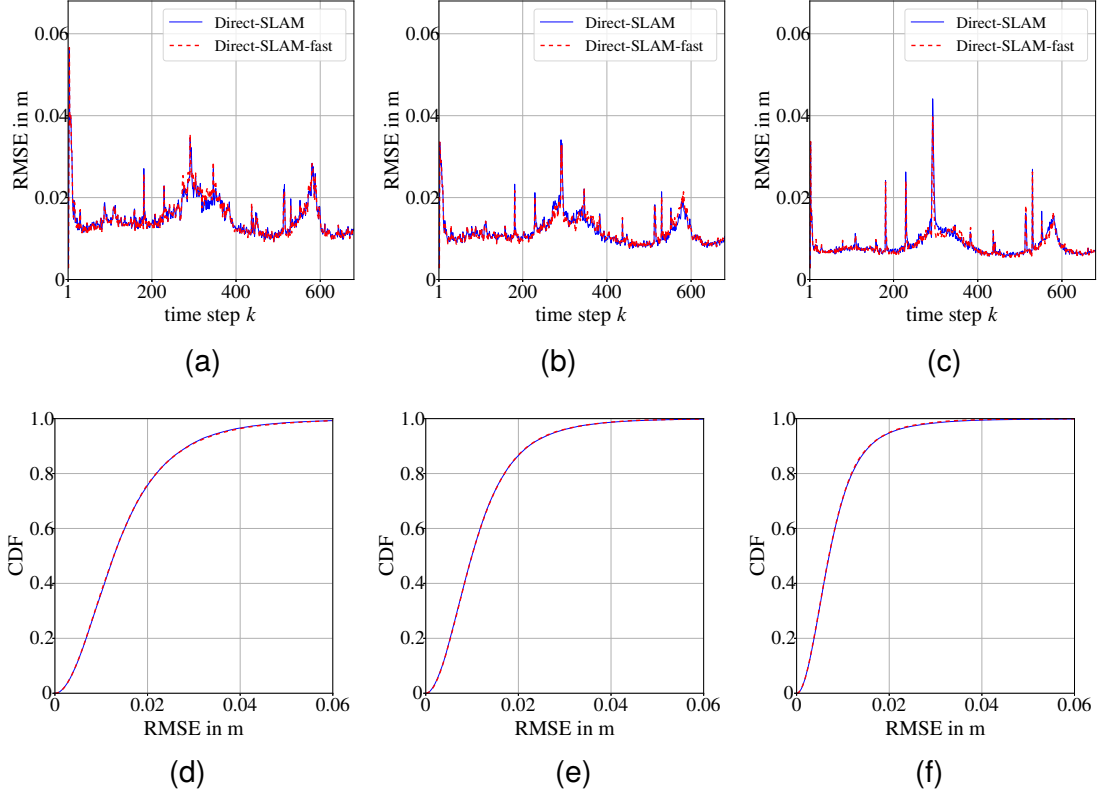


Figure 4.14. Performance of agent localization on synthetic data: The first row shows the agent position RMSEs averaged over 100 simulation runs with (a) 300 MHz, (b) 400 MHz, and (c) 600 MHz bandwidth, respectively. The second row shows the empirical CDFs of the RMSEs with (d) 300 MHz, (e) 400 MHz, and (f) 600 MHz bandwidth, respectively.

mobile agent and the dipole-like ultra-wideband receiver antennas mounted at two static PAs have a uniform radiation pattern in the azimuth plane and nulls in the ground and ceiling directions. Using $S(f)$, the measured radio signals are filtered to a bandwidth of $B = 400$ MHz and demodulated to received baseband signals $\mathbf{z}_k^{(j)}$, $j \in \{1, 2\}$ with $M = 41$. The magnitude spectrum $|\tilde{z}_{k,m}^{(j)}|$, $j \in \{1, 2\}$ of the real measurements used by the Direct-SLAM algorithm is shown in Fig. 4.15a-4.15b. Compared to simulated radio signals shown in Fig. 4.5, it can be observed that the real measurements contain more measurement noise and non-specular multipath effects, including dense multipath component (DMC).

For the experiments running on real data, we set the driving noise variance for legacy PF positions $\sigma_{\mathbf{p},n}^2$ to $2.5 \times 10^{-5} \text{ m}^2$ for PAs, and to $6.4 \times 10^{-5} \text{ m}^2$ for other VAs. Increasing driving

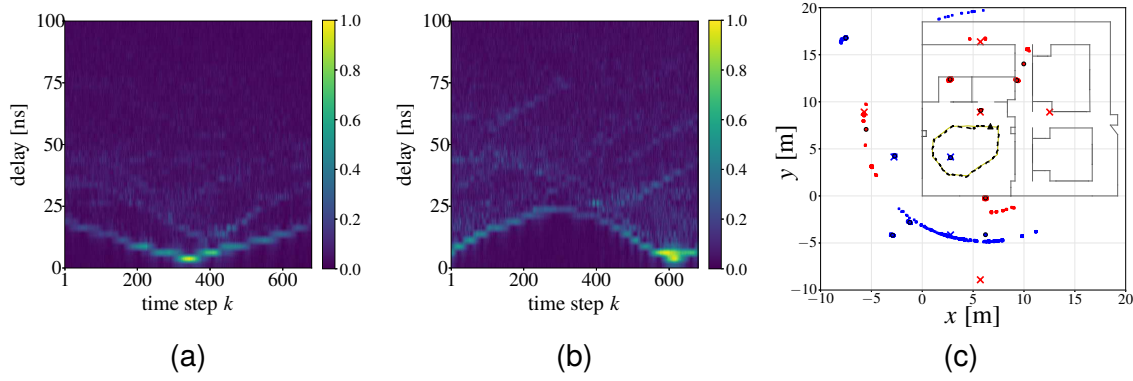


Figure 4.15. The magnitude spectrum $|z_{k,m}^{(j)}|$ of real data with 400MHz bandwidth and normalized to $[0, 1]$ of PA (a) $j = 1$ and (b) $j = 2$.

noise variance improves robustness in this scenario. All other parameters remain the same as in Section 4.5.1. Since there are no ground truth agent positions, we present a qualitative result in Fig. 4.15c. It can be seen that some estimated PFs do not relate to any specular path. These result from the fact that real measurements include non-specular MPCs and there is a model mismatch. Nonetheless, the proposed Direct-SLAM algorithm is still able to provide reasonable estimates for the agent position and the features in the environment.

4.7 Acknowledgement

Chapter 4, in part, is a reprint of the material as it appears in M. Liang, E. Leitingner, and F. Meyer, “Direct multipath-based SLAM,” *IEEE Transactions on Signal Processing*, 2025 [5], and M. Liang, E. Leitingner, and F. Meyer, “A belief propagation approach for direct multipath-based SLAM,” in *Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2023 [6]. The dissertation author was the primary investigator and author of this material.

Chapter 5

Conclusions

This dissertation has advanced the state of the art in machine perception by developing a unified, graph-based Bayesian framework for solving complex estimation problems. Centered on the use of factor graphs to represent statistical models and belief propagation for scalable inference, this work has introduced novel algorithms for both MOT and multipath-based SLAM. This conclusion provides a summary of the principal contributions of this thesis and reflects on their impact.

First, we addressed the persistent challenge of model mismatch in DTT-based MOT. We introduced NEBP for MOT, a hybrid method that bridges that gap between model-based BP and the adaptive power of data-driven neural networks. By designing a GNN to mirror the topology of the underlying factor graph, our method learns to compute data-informed correction terms for the BP messages. Our method also retains the principled structure and scalability of the probabilistic graphical model while harnessing the expressive power of neural networks to adapt to raw sensor data where hand-crafted models are often ineffective. Specifically, the GNN component learns to improve the data association and to effectively reject false alarms, enhancing both object declaration and state estimation accuracy without sacrificing the computational scalability of BP. The evaluation of NEBP on the challenging nuScenes autonomous driving dataset validated this hybrid approach. It not only achieved state-of-the-art accuracy, but also demonstrated superior robustness compared with non-BP-based methods. This enhanced consistency is essential for the

safety of downstream tasks like trajectory prediction and path planning in autonomous systems.

Second, to overcome the fundamental information loss inherent in the DTT paradigm, we developed a message passing algorithm for TBD-based MOT. The cornerstone of this work is a physically-grounded superpositional measurement model that operates directly on raw sensor data and is applicable to a wide range of sensing systems. It simultaneously accounts for challenging real-world effects like sensor data correlation and unknown amplitude fluctuations. The multi-dictionary and multi-snapshot structure allows it to be adapted to a wide range of applications, such as multi-sensor fusion, multi-frequency beamforming, and multi-waveform localization. The new measurement model is combined with statistical models for object dynamics, object birth, and time-varying sensor noise levels. We demonstrated that even with this sophisticated model, scalable inference remains feasible by representing the statistical model as a factor graph and employing Gaussian and particle-based approximations within a tailored BP algorithm. The superior performance over DTT in synthetic radar scenarios validated that preserving information from raw sensor data through a direct signal model yields significant gains. Furthermore, its successful application to a real-world passive acoustic scenario underscored the model’s generality and its robustness to real-world imperfections like colored noise.

Finally, the idea of “direct processing” is extended from tracking to SLAM. We introduced a novel direct multipath-based SLAM algorithm, which is the first method capable of performing SLAM directly on raw radio signals without relying on a lossy frontend channel estimator. This is achieved by a statistical model that directly links the raw received signal to the agent’s state and the locations of map features. We also establish a birth model that makes it possible to infer the unknown and time-varying number of map features. A factor graph representation is then established with an efficient BP algorithm derived from it. To reduce computational complexity, certain BP messages are approximated by complex Gaussian PDFs via moment matching. The resulting complexity only scales linearly with the number of map features. Our numerical studies demonstrated significantly improved localization and mapping accuracy, particularly in challenging environments where conventional two-stage methods fail

due to unresolvable MPCs. The experiments on real data verified its robustness to non-ideal effects like non-specular reflections, showing the power of this direct, graph-based approach for robust perception in complex indoor environments.

The framework established in this dissertation opens several promising directions for future research. One direction involves enriching the physical fidelity of the models, for instance by incorporating diffuse multipath components and non-ideal reflecting surfaces in multipath-based SLAM, or by considering non-constant bathymetry in underwater acoustic sensing. Another promising direction lies in further integrating machine learning, such as using neural networks to learn the sophisticated, non-Gaussian statistics of environmental noise and interference, thereby further bridging the gap between analytical models and real-world complexity.

Ultimately, the contributions presented in this dissertation offer a unified and extensible methodology for machine perception. The synthesis of detailed statistical models and scalable inference provides a robust foundation for the future of perception systems. By leveraging the factor graph as a common language to integrate structured, physics-based knowledge and insights learned from sensor data, this work establishes a powerful paradigm for creating intelligent systems that can perceive their environments with greater accuracy, reliability, and insight.

Bibliography

- [1] M. Liang and F. Meyer, “Neural enhanced belief propagation for multiobject tracking,” *IEEE Trans. Signal Process.*, vol. 72, pp. 15–30, Sept. 2023.
- [2] M. Liang and F. Meyer, “Neural enhanced belief propagation for data association in multiobject tracking,” in *Proc. FUSION-22*, pp. 1–7, July 2022.
- [3] M. Liang and F. Meyer, “Message passing for track-before-detect,” *arXiv preprint arXiv:2506.18177*, 2025.
- [4] M. Liang and F. Meyer, “An approach of directly tracking multiple objects,” in *Proc. IEEE Asilomar-24*, pp. 995–999, Oct. 2024.
- [5] M. Liang, E. Leitingner, and F. Meyer, “Direct multipath-based SLAM,” *IEEE Trans. Signal Process.*, vol. 73, pp. 2336–2352, Mar. 2025.
- [6] M. Liang, E. Leitingner, and F. Meyer, “A belief propagation approach for direct multipath-based SLAM,” in *Proc. IEEE Asilomar-23*, pp. 588–593, Nov. 2023.
- [7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ: Prentice-Hall, 1993.
- [8] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [9] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Machine learning*, vol. 37, pp. 183–233, Feb. 1999.
- [10] T. P. Minka, “Expectation propagation for approximate Bayesian inference,” in *Proc. UAI-01*, pp. 362–369, Aug. 2001.
- [11] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo: Morgan Kaufmann, 1988.
- [12] 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.
- [13] H.-A. Loeliger, “An introduction to factor graphs,” *IEEE Signal Process. Mag.*, vol. 21, pp. 28–41, Jan. 2004.
 - [14] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press, 2009.
 - [15] K. P. Murphy, Y. Weiss, and M. I. Jordan, “Loopy belief propagation for approximate inference: an empirical study,” in *Proc. UAI*, vol. 15, pp. 467–475, July 1999.
 - [16] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.
 - [17] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative localization in wireless networks,” *Proc. IEEE*, vol. 97, pp. 427–450, Feb. 2009.
 - [18] F. Meyer, T. Kropfreiter, J. L. Williams, R. 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.
 - [19] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Z. Win, “A belief propagation algorithm for multipath-based SLAM,” *IEEE Trans. Wireless Commun.*, vol. 18, pp. 5613–5629, Dec. 2019.
 - [20] D. Bickson, *Gaussian belief propagation: Theory and application*. PhD thesis, The Hebrew University of Jerusalem, 2008.
 - [21] A. Ihler and D. McAllester, “Particle belief propagation,” in *Proc. AISTATS-09*, vol. 5, pp. 256–263, Apr. 2009.
 - [22] S. Kramer and H. Sorenson, “Bayesian parameter estimation,” *IEEE Trans. Autom. Control*, vol. 33, pp. 217–222, Feb. 1988.
 - [23] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Eng.*, vol. 82, pp. 35–45, Mar. 1960.
 - [24] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Corporation, 2013.
 - [25] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proc. IEEE*, vol. 92, pp. 401–422, Mar. 2004.
 - [26] 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.

- [27] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: principles and techniques*, vol. 19. YBS publishing Storrs, CT, 1995.
- [28] D. Musicki, R. Evans, and S. Stankovic, “Integrated probabilistic data association,” *IEEE Trans. Autom. Control*, vol. 39, no. 6, pp. 1237–1241, 1994.
- [29] D. Musicki and R. Evans, “Joint integrated probabilistic data association: JIPDA,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, pp. 1093–1099, July 2004.
- [30] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2007.
- [31] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina, “A tutorial on Bernoulli filters: Theory, implementation and applications,” *IEEE Trans. Signal Process.*, vol. 61, pp. 3406–3430, Apr. 2013.
- [32] 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, July 2015.
- [33] D. Salmond and N. Gordon, “Group and extended object tracking,” in *Proc. IEE Colloq. Target Tracking: Algorithms Appl.*, pp. 16/1–16/4, Nov. 1999.
- [34] W. Koch and G. Van Keuk, “Multiple hypothesis track maintenance with possibly unresolved measurements,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, pp. 883–892, July 1997.
- [35] H. W. Kuhn, “The Hungarian method for the assignment problem,” *NRL quarterly*, vol. 2, pp. 83–97, Mar. 1955.
- [36] D. B. Reid, “An algorithm for tracking multiple targets,” *IEEE Trans. Autom. Control*, vol. 24, pp. 843–854, Dec. 1979.
- [37] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” *Automatica*, vol. 11, pp. 451–460, Sept. 1975.
- [38] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *IEEE Control Syst. Mag.*, vol. 29, pp. 82–100, Nov. 2009.
- [39] K. G. Murty, “An algorithm for ranking all the assignments in order of increasing cost,” *Oper. Res.*, vol. 16, pp. 682–687, May 1968.

- [40] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, “Poisson multi-Bernoulli mixture filter: Direct derivation and implementation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, pp. 1883–1901, Feb. 2018.
- [41] 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.
- [42] R. L. Streit, M. L. Graham, and M. J. Walsh, “Multitarget tracking of distributed targets using histogram-PMHT,” *Digital Signal Processing*, vol. 12, pp. 394–404, Nov. 2002.
- [43] S. J. Davey and H. X. Gaetjens, *Track-Before-Detect Using Expectation Maximisation*. Singapore: Springer, 2018.
- [44] M. Orton and W. Fitzgerald, “A Bayesian approach to tracking multiple targets using sensor arrays and particle filters,” vol. 50, pp. 216–223, Feb. 2002.
- [45] A. Lepoutre, O. Rabaste, and F. Le Gland, “Multitarget likelihood computation for track-before-detect applications with amplitude fluctuations of type Swerling 0, 1, and 3,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, pp. 1089–1107, June 2016.
- [46] B.-N. Vo, B.-T. Vo, N.-T. Pham, and D. Suter, “Joint detection and estimation of multiple objects from image observations,” *IEEE Trans. Signal Process.*, vol. 58, pp. 5129–5141, Oct. 2010.
- [47] T. Kropfreiter, J. L. Williams, and F. Meyer, “A scalable track-before-detect method with Poisson/multi-Bernoulli model,” in *Proc. FUSION-21*, pp. 1–8, Nov. 2021.
- [48] T. Kropfreiter, J. L. Williams, and F. Meyer, “Association-based track-before-detect with object contribution probabilities,” *arXiv preprint arXiv:2504.16814*, 2025.
- [49] N. Ito and S. Godsill, “A multi-target track-before-detect particle filter using superpositional data in non-Gaussian noise,” vol. 27, pp. 1075–1079, June 2020.
- [50] E. S. Davies and Á. F. García-Fernández, “Information exchange track-before-detect multi-Bernoulli filter for superpositional sensors,” *IEEE Trans. Signal Process.*, vol. 72, pp. 607–621, Jan. 2024.
- [51] B. Ristic, L. Rosenberg, D. Y. Kim, X. Wang, and J. Williams, “Bernoulli track-before-detect filter for maritime radar,” *IET Radar, Sonar Navig.*, vol. 14, pp. 356–363, Mar. 2020.
- [52] D. Y. Kim, B. Ristic, R. Guan, and L. Rosenberg, “A Bernoulli track-before-detect filter for interacting targets in maritime radar,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57,

pp. 1981–1991, June 2021.

- [53] M. Liang, T. Kropfreiter, and F. Meyer, “A BP method for track-before-detect,” *IEEE Signal Process. Lett.*, vol. 30, pp. 1137–1141, July 2023.
- [54] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *IEEE Trans. Robot. Autom.*, vol. 17, pp. 229–241, June 2001.
- [55] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *Proc. AAAI-02*, pp. 593–598, July 2002.
- [56] P. Meissner, *Multipath-Assisted Indoor Positioning*. PhD thesis, Graz University of Technology, 2014.
- [57] C. Gentner, T. Jost, W. Wang, S. Zhang, A. Dammann, and U.-C. Fiebig, “Multipath assisted positioning with simultaneous localization and mapping,” *IEEE Trans. Wireless Commun.*, vol. 15, pp. 6104–6117, June 2016.
- [58] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, and M. Z. Win, “High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend,” *IEEE Signal Process. Mag.*, vol. 33, pp. 59–70, Mar. 2016.
- [59] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT: Yaakov Bar-Shalom, 2011.
- [60] 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.
- [61] 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.
- [62] G. Soldi, F. Meyer, P. Braca, and F. Hlawatsch, “Self-tuning algorithms for multisensor-multitarget tracking using belief propagation,” *IEEE Trans. Signal Process.*, vol. 67, pp. 3922–3937, Aug. 2019.
- [63] W. Zhang and F. Meyer, “Graph-based multiobject tracking with embedded particle flow,” in *Proc. IEEE RadarConf-21*, pp. 1–6, May 2021.
- [64] P. Sharma, A. A. Saucan, D. J. Bucci, and P. K. Varshney, “Decentralized Gaussian filters for cooperative self-localization and multi-target tracking,” *IEEE Trans. Signal Process.*,

vol. 67, pp. 5896–5911, Nov. 2019.

- [65] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, “Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering,” in *Proc. IEEE IV-18*, pp. 433–440, June 2018.
- [66] S. Pang, D. Morris, and H. Radha, “3D multi-object tracking using random finite set-based multiple measurement models filtering (RFS-M³) for autonomous vehicles,” in *Proc. ICRA-21*, pp. 13701–13707, June 2021.
- [67] J. Liu, L. Bai, Y. Xia, T. Huang, and B. Zhu, “GNN-PMB: A simple but effective online 3D multi-object tracker without bells and whistles,” *IEEE Trans. Intell. Veh.*, vol. 8, pp. 1176–1189, Oct. 2023.
- [68] A. A. Saucan, M. J. Coates, and M. Rabbat, “A multisensor multi-Bernoulli filter,” *IEEE Trans. Signal Process.*, vol. 65, pp. 5495–5509, Oct. 2017.
- [69] T. Yin, X. Zhou, and P. Krahenbuhl, “Center-based 3D object detection and tracking,” in *Proc. CVPR-21*, pp. 11784–11793, June 2021.
- [70] F. Meyer and M. Z. Win, “Scalable data association for extended object tracking,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 491–507, May 2020.
- [71] H.-k. Chiu, A. Prioletti, J. Li, and J. Bohg, “Probabilistic 3D multi-object tracking for autonomous driving,” *arXiv preprint arXiv:2001.05673*, 2020.
- [72] X. Weng, J. Wang, D. Held, and K. Kitani, “3D multi-object tracking: A baseline and new evaluation metrics,” in *Proc. IROS-20*, pp. 10359–10366, Oct. 2020.
- [73] J.-N. Zaech, D. Dai, A. Liniger, M. Danelljan, and L. Van Gool, “Learnable online graph representations for 3D multi-object tracking,” *IEEE Robot. Autom. Lett.*, vol. 7, pp. 5103–5110, Jan. 2022.
- [74] A. Ranges, P. Maheshwari, M. Gebre, S. Mhatre, V. Ramezani, and M. M. Trivedi, “TrackMPNN: A message passing graph neural architecture for multi-object tracking,” *arXiv preprint arXiv:2101.04206*, 2021.
- [75] Z. Pang, Z. Li, and N. Wang, “SimpleTrack: Understanding and rethinking 3D multi-object tracking,” *arXiv preprint arXiv:2111.09621*, 2021.
- [76] Q. Wang, Y. Chen, Z. Pang, N. Wang, and Z. Zhang, “Immortal Tracker: Tracklet never dies,” *arXiv preprint arXiv:2111.13672*, 2021.
- [77] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, “GNN3DMOT: Graph neural network

- for 3D multi-object tracking with 2D-3D multi-feature learning,” in *Proc. CVPR-20*, pp. 6499–6508, June 2020.
- [78] H.-k. Chiu, J. Li, R. Ambrus, and J. Bohg, “Probabilistic 3D multi-modal, multi-object tracking for autonomous driving,” in *Proc. ICRA-21*, pp. 14227–14233, June 2021.
 - [79] M. Sun, M. E. Davies, I. K. Proudler, and J. R. Hopgood, “Adaptive kernel Kalman filter based belief propagation algorithm for maneuvering multi-target tracking,” *IEEE Signal Process. Lett.*, vol. 29, pp. 1452–1456, 2022.
 - [80] X. Li, E. Leitinger, A. Venus, and F. Tufvesson, “Sequential detection and estimation of multipath channel parameters using belief propagation,” *IEEE Trans. Wireless Commun.*, vol. 21, pp. 8385–8402, Apr. 2022.
 - [81] Y. Zhou and O. Tuzel, “VoxelNet: End-to-end learning for point cloud based 3D object detection,” in *Proc. CVPR-18*, pp. 4490–4499, June 2018.
 - [82] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast encoders for object detection from point clouds,” in *Proc. CVPR-19*, pp. 12697–12705, June 2019.
 - [83] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, “Class-balanced grouping and sampling for point cloud 3D object detection,” *arXiv:1908.09492*, 2019.
 - [84] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. NeurIPS-15*, vol. 28, pp. 91–99, Dec. 2015.
 - [85] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” in *Proc. CVPR-19*, pp. 770–779, June 2019.
 - [86] A. Simonelli, S. R. Buló, L. Porzi, M. López-Antequera, and P. Kotschieder, “Disentangling monocular 3D object detection,” in *Proc. CVPR-19*, pp. 1991–1999, June 2019.
 - [87] K. Okuma, A. Taleghani, N. d. Freitas, J. J. Little, and D. G. Lowe, “A boosted particle filter: Multitarget detection and tracking,” in *Proc. ECCV-04*, pp. 28–39, May 2004.
 - [88] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, “Online multiperson tracking-by-detection from a single, uncalibrated camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 1820–1833, Dec. 2010.
 - [89] S. S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 19, pp. 5–18, Jan. 2004.

- [90] J. Yedidia, W. Freemand, and Y. Weiss, “Constructing free-energy approximations and generalized belief propagation algorithms,” *IEEE Trans. Inf. Theory*, vol. 51, pp. 2282–2312, July 2005.
- [91] M. Gori, G. Monfardini, and F. Scarselli, “A new model for learning in graph domains,” in *Proc. INNS/IEEE IJCNN-05*, vol. 2, pp. 729–734, Aug. 2005.
- [92] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proc. ICML-17*, pp. 1263–1272, Aug. 2017.
- [93] K. Yoon, R. Liao, Y. Xiong, L. Zhang, E. Fetaya, R. Urtasun, R. Zemel, and X. Pitkow, “Inference in probabilistic graphical models by graph neural networks,” in *Proc. IEEE Asilomar-19*, pp. 868–875, Nov. 2019.
- [94] V. G. Satorras and M. Welling, “Neural enhanced belief propagation on factor graphs,” in *Proc. AISTATS-21*, pp. 685–693, Apr. 2021.
- [95] M. Liang and F. Meyer, “Neural enhanced belief propagation for cooperative localization,” in *Proc. IEEE SSP-21*, pp. 326–330, July 2021.
- [96] K. Granström, M. Baum, and S. Reuter, “Extended object tracking: Introduction, overview and applications,” *J. Adv. Inf. Fusion*, vol. 12, pp. 139–174, Dec. 2017.
- [97] K. Granström, M. Fatemi, and L. Svensson, “Poisson multi-Bernoulli mixture conjugate prior for multiple extended target filtering,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, pp. 208–225, Feb. 2020.
- [98] H. Wymeersch, *Iterative Receiver Design*. Cambridge University Press, 2007.
- [99] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” Tech. Rep. TR-649, University of California, Berkeley, Sept. 2003.
- [100] M. Bayati, D. Shah, and M. Sharma, “Max-product for maximum weight matching: Convergence, correctness, and LP duality,” *IEEE Trans. Inf. Theory*, vol. 54, pp. 1241–1251, Feb. 2008.
- [101] J. L. Williams and R. A. Lau, “Data association by loopy belief propagation,” in *Proc. FUSION-10*, pp. 1–8, July 2010.
- [102] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, second ed., 1994.
- [103] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. ICML-15*, pp. 448–456, July 2015.

- [104] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proc. ICML-17*, pp. 1321–1330, Aug. 2017.
- [105] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, “Imbalance problems in object detection: A review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, pp. 3388–3415, Mar. 2020.
- [106] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuScenes: A multimodal dataset for autonomous driving,” in *Proc. CVPR-20*, pp. 11621–11631, June 2020.
- [107] A. Neubeck and L. Van Gool, “Efficient non-maximum suppression,” in *Proc. ICPR-06*, vol. 3, pp. 850–855, Aug. 2006.
- [108] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY: Wiley, 2002.
- [109] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: the CLEAR MOT metrics,” *EURASIP J. Image Video Proc.*, vol. 2008, pp. 1–10, May 2008.
- [110] B. Wu and R. Nevatia, “Tracking of multiple, partially occluded humans based on static body part detection,” in *Proc. of CVPR-06*, vol. 1, pp. 951–958, June 2006.
- [111] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Proc. NeurIPS-89*, vol. 2, pp. 396–404, Nov. 1989.
- [112] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [113] N. Benbarka, J. Schröder, and A. Zell, “Score refinement for confidence-based 3D multi-object tracking,” in *Proc. IROS-21*, pp. 8083–8090, Sept. 2021.
- [114] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, “Generalized optimal sub-pattern assignment metric,” in *Proc. FUSION-17*, pp. 1–8, July 2017.
- [115] Y. Boers and J. Driessen, “Multitarget particle filter track before detect application,” *IEE Proc. Radar Sonar Navig.*, vol. 151, pp. 351–357, Dec. 2004.
- [116] S. Wei, M. Liang, and F. Meyer, “A new architecture for neural enhanced multiobject tracking,” in *Proc. FUSION-24*, pp. 1–8, July 2024.
- [117] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina, “A tutorial on Bernoulli filters: Theory, implementation and applications,” *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3406–3430, 2013.

- [118] D. Salmond and H. Birch, “A particle filter for track-before-detect,” in *Proc. ACC-01*, pp. 3755–3760, June 2001.
- [119] O. Rabaste, C. Riché, and A. Lepoutre, “Long-time coherent integration for low SNR target via particle filter in track-before-detect,” in *Proc. FUSION-12*, pp. 127–134, July 2012.
- [120] S. J. Davey, M. G. Rutten, and B. Cheung, “Using phase to improve track-before-detect,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, pp. 832–849, Jan. 2012.
- [121] K. Kim, M. Üney, and B. Mulgrew, “Coherent track-before-detect with micro-Doppler signature estimation in array radars,” *IET Radar, Sonar Navig.*, vol. 14, pp. 572–585, Mar. 2020.
- [122] M. Üney, P. Horridge, B. Mulgrew, and S. Maskell, “Coherent long-time integration and Bayesian detection with Bernoulli track-before-detect,” *IEEE Signal Process. Lett.*, vol. 30, pp. 239–243, Mar. 2023.
- [123] A. M. Westerkam, J. Möderl, E. Leitingner, and T. Pedersen, “Variational message passing-based multiobject tracking for MIMO-radars using raw sensor signals,” *arXiv preprint arXiv:2503.15246*, 2025.
- [124] S. Carson, D. Kilfoyle, M. Potter, and J. Vance, “A passive, multi-static radar system,” in *Proc. IET RADAR-07*, pp. 1–4, Oct. 2007.
- [125] J. Möderl, A. M. Westerkam, A. Venus, and E. Leitingner, “A block-sparse Bayesian learning algorithm with dictionary parameter estimation for multi-sensor data fusion,” *arXiv preprint arXiv:2503.12913*, 2025.
- [126] N. Booth, A. Abawi, P. Schey, and W. Hodgkiss, “Detectability of low-level broad-band signals using adaptive matched-field processing with vertical aperture arrays,” *IEEE J. Ocean. Eng.*, vol. 25, pp. 296–313, July 2000.
- [127] S. Nannuru, K. L. Gemba, P. Gerstoft, W. S. Hodgkiss, and C. F. Mecklenbräuker, “Sparse Bayesian learning with multiple dictionaries,” *Signal Processing*, vol. 159, pp. 159–170, June 2019.
- [128] M. Temiz, H. Griffiths, and M. A. Ritchie, “Improved target localization in multiwaveform multiband hybrid multistatic radar networks,” *IEEE Sensors J.*, vol. 22, pp. 20785–20796, Nov. 2022.
- [129] J. Kormylo and J. Mendel, “Maximum likelihood detection and estimation of Bernoulli - Gaussian processes,” *IEEE Trans. Inf. Theory*, vol. 28, pp. 482–488, May 1982.

- [130] M.-A. Badiu, T. L. Hansen, and B. H. Fleury, “Variational Bayesian inference of line spectra,” *IEEE Trans. Signal Process.*, vol. 65, pp. 2247–2261, Jan. 2017.
- [131] T. L. Hansen, B. H. Fleury, and B. D. Rao, “Superfast line spectral estimation,” *IEEE Trans. Signal Process.*, vol. 66, pp. 2511–2526, Feb. 2018.
- [132] M. I. Skolnik, *Introduction to Radar Systems*. New York: Mcgraw-Hill, 3rd ed., 2002.
- [133] S. Ji, D. Dunson, and L. Carin, “Multitask compressive sensing,” *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 92–106, 2009.
- [134] I. Fedorov, B. D. Rao, and T. Q. Nguyen, “Multimodal sparse Bayesian dictionary learning applied to multimodal data classification,” in *Proc. ICASSP-17*, pp. 2237–2241, Mar. 2017.
- [135] K. L. Gemba, S. Nannuru, P. Gerstoft, and W. S. Hodgkiss, “Multi-frequency sparse Bayesian learning for robust matched field processing,” *J. Acoust. Soc. Am.*, vol. 141, pp. 3411–3420, May 2017.
- [136] K. Granström and U. Orguner, “New prediction for extended targets with random matrices,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, pp. 1577–1589, July 2014.
- [137] N. Levanon and E. Mozeson, *Radar signals*. John Wiley & Sons, 2004.
- [138] J. Zhu, M. Xu, R. Guo, F. Wang, G. Zheng, and F. Qu, “Joint multitarget detection and tracking with mmWave radar,” *arXiv preprint arXiv:2412.17211*, 2024.
- [139] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, vol. 41, pp. 3397–3415, Dec. 1993.
- [140] F. B. Jensen, W. A. Kuperman, M. B. Porter, H. Schmidt, and A. Tolstoy, *Computational ocean acoustics*, vol. 2011. Springer, 2011.
- [141] M. B. Porter, “The KRAKEN normal mode program,” *SACLANT Undersea Research Centre Memorandum (SM-245) and Naval Research Laboratory Memorandum Report No. 6920*, 1991.
- [142] G. D’Spain, J. Murray, W. Hodgkiss, N. Booth, and P. Schey, “Mirages in shallow water matched field processing,” *J. Acoust. Soc. Am.*, vol. 105, pp. 3245–3265, June 1999.
- [143] F. Hunter Akins and W. Kuperman, “Range-coherent matched field processing for low signal-to-noise ratio localization,” *J. Acoust. Soc. Am.*, vol. 150, pp. 270–280, July 2021.
- [144] 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, pp. 1309–1332, Dec. 2016.
- [145] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I,” *IEEE Robot. Autom. Mag.*, vol. 13, pp. 99–110, June 2006.
- [146] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, “A random-finite-set approach to Bayesian SLAM,” vol. 27, pp. 268–282, Feb. 2011.
- [147] H. Deusch, S. Reuter, and K. Dietmayer, “The labeled multi-Bernoulli SLAM filter,” *IEEE Signal Process. Lett.*, vol. 22, pp. 1561–1565, Mar. 2015.
- [148] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” *IEEE Trans. Intell. Veh.*, vol. 2, pp. 194–220, Sept. 2017.
- [149] F. Wen, J. Kulmer, K. Witrisal, and H. Wymeersch, “5G positioning and mapping with diffuse multipath,” *IEEE Trans. Wireless Commun.*, vol. 20, pp. 1164–1174, Oct. 2021.
- [150] V. Raghavan, J. Cezanne, S. Subramanian, A. Sampath, and O. Koymen, “Beamforming tradeoffs for initial UE discovery in millimeter-wave MIMO systems,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, pp. 543–559, Jan. 2016.
- [151] C. De Lima, D. Belot, R. Berkvens, A. Bourdoux, D. Dardari, M. Guillaud, M. Isomursu, E.-S. Lohan, Y. Miao, A. N. Barreto, *et al.*, “Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges,” *IEEE Access*, vol. 9, pp. 26902–26925, Jan. 2021.
- [152] H. Wymeersch, S. Marandò, W. M. Gifford, and M. Z. Win, “A machine learning approach to ranging error mitigation for UWB localization,” *IEEE Trans. Wireless Commun.*, vol. 60, pp. 1719–1728, Apr. 2012.
- [153] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [154] 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, pp. 1309–1332, Dec 2016.
- [155] Y. Wang, K. C. Ho, and L. Huang, “Room geometry estimation using the multipath delays,” *IEEE Signal Process. Lett.*, vol. 28, pp. 1380–1384, June 2021.
- [156] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak, L. Nogueira, M. Palieri, P. Petráček, M. Petr-

- lík, A. Reinke, V. Krátký, S. Zhao, A.-A. Agha-Mohammadi, K. Alexis, C. Heckman, K. Khosoussi, N. Kottege, B. Morrell, M. Hutter, F. Pauling, F. Pomerleau, M. Saska, S. Scherer, R. Siegwart, J. L. Williams, and L. Carlone, “Present and future of SLAM in extreme environments: The DARPA SubT Challenge,” *IEEE Trans. Robot.*, vol. 40, pp. 936–959, Oct. 2023.
- [157] E. Leitinger, S. Grebien, and K. Witrisal, “Multipath-based SLAM exploiting AoA and amplitude information,” in *Proc. IEEE ICCW-19*, pp. 1–7, May 2019.
- [158] E. Leitinger, A. Venus, B. Teague, and F. Meyer, “Data fusion for multipath-based SLAM: Combining information from multiple propagation paths,” *IEEE Trans. Signal Process.*, vol. 71, pp. 4011–4028, Sept. 2023.
- [159] J. Yang, C.-K. Wen, and S. Jin, “Hybrid active and passive sensing for SLAM in wireless communication systems,” *IEEE J. Sel. Areas Commun.*, vol. 40, pp. 2146–2163, Mar. 2022.
- [160] D. P. Wipf and B. D. Rao, “An empirical Bayesian strategy for solving the simultaneous sparse approximation problem,” *IEEE Trans. Signal Process.*, vol. 55, pp. 3704–3716, July 2007.
- [161] D. Shutin, W. Wang, and T. Jost, “Incremental sparse Bayesian learning for parameter estimation of superimposed signals,” in *Proc. SAMPTA-13*, no. 1, pp. 6–9, Sept. 2013.
- [162] P. Gerstoft, C. F. Mecklenbräuker, A. Xenaki, and S. Nannuru, “Multisnapshot sparse bayesian learning for DOA,” *IEEE Signal Process. Lett.*, vol. 23, pp. 1469–1473, Aug. 2016.
- [163] A. Sant, M. Leinonen, and B. D. Rao, “Block-sparse signal recovery via general total variation regularized sparse bayesian learning,” *IEEE Trans. Signal Process.*, vol. 70, pp. 1056–1071, 2022.
- [164] M. Bayraktar, J. Palacios, N. González-Prelcic, and C. J. Zhang, “Multidimensional orthogonal matching pursuit-based RIS-aided joint localization and channel estimation at mmWave,” in *Proc. SPAWC-22*, pp. 1–5, July 2022.
- [165] J. Palacios and N. González-Prelcic, “Separable multidimensional orthogonal matching pursuit and its application to joint localization and communication at mmWave,” in *Proc. GC Wkshps-23*, pp. 1421–1426, Dec. 2023.
- [166] S. Grebien, E. Leitinger, K. Witrisal, and B. H. Fleury, “Super-resolution estimation of UWB channels including the dense component – An SBL-inspired approach,” *IEEE Trans. Wireless Commun.*, vol. 23, pp. 10301–10318, Feb. 2024.

- [167] B. Ristic, L. Rosenberg, D. Y. Kim, X. Wang, and J. Williams, “Bernoulli filter for track-before-detect: Swerling-1 target in K-distributed clutter,” in *Proc. RadarConf-19*, pp. 1–5, Apr. 2019.
- [168] E. S. Davies and Á. F. García-Fernández, “A multi-Bernoulli Gaussian filter for track-before-detect with superpositional sensors,” in *Proc. FUSION-22*, pp. 1–8, July 2022.
- [169] T. L. Hansen, B. H. Fleury, and B. D. Rao, “Superfast line spectral estimation,” *IEEE Trans. Signal Process.*, vol. 66, pp. 2511–2526, Feb. 2018.
- [170] J. Möderl, F. Pernkopf, K. Witrisal, and E. Leitinger, “Variational inference of structured line spectra exploiting group-sparsity,” *IEEE Trans. Signal Process.*, Nov. 2024.
- [171] 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.
- [172] E. Leitinger, S. Grebien, B. H. Fleury, and K. Witrisal, “Detection and estimation of a spectral line in MIMO systems,” in *Proc. Asilomar-20*, pp. 1090–1095, Oct. 2020.
- [173] A. Richter, *Estimation of Radio Channel Parameters: Models and Algorithms*. PhD thesis, Ilmenau University of Technology, 2005.
- [174] L. Wielandner, A. Venus, T. Wilding, K. Witrisal, and E. Leitinger, “MIMO multipath-based SLAM for non-ideal reflective surfaces,” in *Proc. FUSION-24*, 2024. accepted.
- [175] M. Liang, E. Leitinger, and F. Meyer, “Direct multipath-based SLAM: Supporting documents,” 2024. <https://fmeyer.ucsd.edu/LiaLeiMey-2024-SD.pdf>.
- [176] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [177] J. Borish, “Extension of the image model to arbitrary polyhedra,” *J. Am. Statist. Assoc.*, vol. 75, pp. 1827–1836, Mar. 1984.
- [178] F. Antonio, “Iv.6 - faster line segment intersection,” in *Graphics Gems III*, pp. 199–202, San Francisco: Morgan Kaufmann, 1992.
- [179] E. Leitinger, S. Grebien, X. Li, F. Tufvesson, and K. Witrisal, “On the use of MPC amplitude information in radio signal based SLAM,” in *Proc. IEEE SSP-18*, pp. 633–637, June 2018.
- [180] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *Proc. IROS-12*, pp. 573–580, Oct. 2012.

- [181] P. Tichavsky, C. Muravchik, and A. Nehorai, “Posterior Cramér-Rao bounds for discrete-time nonlinear filtering,” *IEEE Trans. Signal Process.*, vol. 46, pp. 1386–1396, May 1998.
- [182] O. Kaltiokallio, Y. Ge, J. Talvitie, H. Wymeersch, and M. Valkama, “mmWave simultaneous localization and mapping using a computationally efficient EK-PHD filter,” in *Proc. FUSION-21*, pp. 1–8, Nov. 2021.
- [183] E. Leitinger, P. Meissner, C. Rudisser, G. Dumphart, and K. Witrisal, “Evaluation of position-related information in multipath components for indoor positioning,” *IEEE J. Sel. Areas Commun.*, vol. 33, pp. 2313–2328, Nov. 2015.
- [184] M. Liang, E. Leitinger, and F. Meyer, “Direct multipath-based SLAM: Supporting results,” 2024. <https://fmeyer.ucsd.edu/LiaLeiMey-2024-ConventionalVsDirect.mp4>.
- [185] P. Meissner, E. Leitinger, and K. Witrisal, “UWB for robust indoor tracking: Weighting of multipath components for efficient estimation,” *IEEE Wireless Commun. Lett.*, vol. 3, pp. 501–504, July 2014.