CHALMERS
UNIVERSITY OF TECHNOLOGY

Batch SLAM with PMBM Data Association Sampling and Graph-Based Optimization

Citation for the original published paper (version of record):
Ge, Y., Kaltiokallio, O., Xia, Y. et al (2025). Batch SLAM with PMBM Data Association Sampling and Graph-Based Optimization. IEEE Transactions on Signal Processing, In Press: 1-15. http://dx.doi.org/10.1109/TSP.2025.3567916

(article starts on next page)

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3567916

Batch SLAM with PMBM Data Association Sampling and Graph-Based Optimization

Yu Ge, *Member, IEEE,* Ossi Kaltiokallio, *Member, IEEE,* Yuxuan Xia, *Member, IEEE,* Ángel F. García-Fernández, Hyowon Kim, *Member, IEEE,* Jukka Talvitie, *Member, IEEE,* Mikko Valkama, *Fellow, IEEE,* Henk Wymeersch, *Fellow, IEEE,* Lennart Svensson, *Senior Member, IEEE*

***Abstract*—Simultaneous localization and mapping (SLAM) methods need to both solve the data association (DA) problem and the joint estimation of the sensor trajectory and the map, conditioned on a DA. In this paper, we propose a novel integrated approach to solve both the DA problem and the batch SLAM problem simultaneously, combining random finite set (RFS) theory and the graph-based SLAM approach. A sampling method based on the Poisson multi-Bernoulli mixture (PMBM) density is designed for dealing with the DA uncertainty, and a graph-based SLAM solver is applied for the conditional SLAM problem. In the end, a post-processing approach is applied to merge SLAM results from different iterations. Using synthetic data, it is demonstrated that the proposed SLAM approach achieves performance close to the posterior Cramér-Rao bound, and outperforms state-of-the-art RFS-based SLAM filters in high clutter and high process noise scenarios.**

***Index Terms*—Batch processing, SLAM, DA, correlation, RFS, graph-based SLAM, sampling, PMBM.**

I. Introduction

The objective of the simultaneous localization and mapping (SLAM) problem [1], [2] is to deduce the dynamic pose of a mobile sensor over time, along with constructing a map of the surrounding environment, using measurements obtained from one or multiple sensors. Drawing inspiration from pioneering research in autonomous robotics [3], the SLAM problem has captured broad interest in recent decades: it holds significant importance with a multitude of applications spanning diverse fields, such as robotics [1], autonomous driving [4], virtual and augmented reality [5], indoor navigation [6], [7], integrated sensing and communication [8], [9] as exemplified in Fig. 1, and so on.

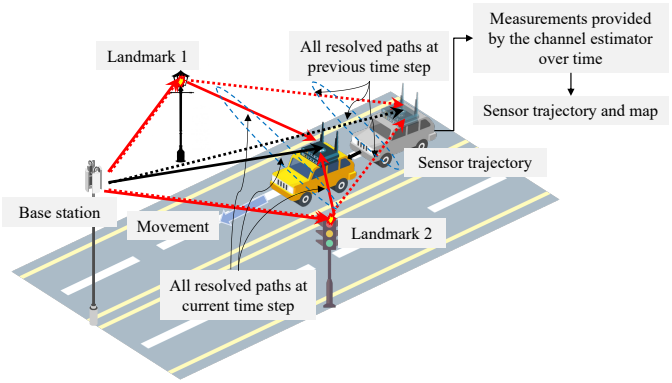


Fig. 1. Illustrative example of the SLAM application in integrated sensing and communication, where resolved measurements over time are leveraged to estimate both the sensor trajectory and the surrounding environment.

Traditional SLAM methods typically follow a two-step approach: i) solve the data association (DA) problem between the unknown number of landmarks which collectively define the map and imperfect measurements, which may include clutter and mis-detections, ii) estimate the joint posterior density of the sensor trajectory and the map, given measurements, control inputs, and the DA from step (i). Two important methodologies are the *filtering-based* and *graph-based* approaches. In filtering-based approaches [2], [3], [10], the map is modeled with an unknown number of physical landmarks with unknown spatial locations, and the map and the sensor state are then typically estimated sequentially. Prominent examples are extended Kalman filter (EKF)-SLAM [10] and FastSLAM [11], which has been demonstrated to work well in practice, but is sensitive to DA uncertainty [12].

On the other hand, in graph-based approaches [13]–[15], the sensor state at a specific time step or a physical landmark is represented as a node in a graph, and each edge represents probabilistic dependency between two sensor states, or between a landmark state and a sensor state. The sensor trajectory and the map can be simultaneously estimated by obtaining the maximum a posteriori estimation (MAP) estimate, optimizing over the whole graph. Unlike filtering-based approaches, graph-based SLAM typically takes all measurements and performs optimization techniques on the entire graph, maintaining cross-correlation information between the sensor trajectory and

Yu Ge, Henk Wymeersch, and Lennart Svensson are with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. Emails: {yuge,henkw,lennart.svensson}@chalmers.se.

Ossi Kaltiokallio, Jukka Talvitie, and Mikko Valkama are with the Unit of Electrical Engineering, Tampere University, Tampere, Finland. Emails: {ossi.kaltiokallio,jukka.talvitie,mikko.valkama}@tuni.fi.

Yuxuan Xia is with the Department of Automation, Shanghai Jiaotong University, Shanghai, China. Email: yuxuan.xia@sjtu.edu.cn.

Ángel F. García-Fernández is with the IPTC, ETSI de Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain. Email: angel.garcia.fernandez@upm.es.

Hyowon Kim is with the Department of Electronics Engineering, Chungnam National University, Daejeon, South Korea. Email: hyowon.kim@cnu.ac.kr.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation, the Vinnova Beyond5GPOS project, the Research Council of Finland (RCF) under the grants #345654, #352754, #357730, and #359095, Business Finland under the 6G-ISAC project, and the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (RS-2024-00464570).

the map. This results in more robust and accurate estimates, and makes graph-based SLAM perform batch processing and typically work offline. Among graph-based SLAM approaches, the GraphSLAM algorithm has become a prevalent offline SLAM solver for batch processing, due to its global consistency properties [16]. However, the performance of graph-based SLAM heavily relies on the quality of the DA. Statistical tests such as the χ^2 test, joint compatibility test, or other types of heuristics are often applied to solve the DA problem [17]. In GraphSLAM, the SLAM front-end is responsible for constructing constraints of the graph and a single error in the DA often leads to inconsistent maps [18]. Instead of assuming that the generated pose-graphs are free of outliers, typically the state-of-the-art SLAM back-ends are implemented such that they are able to naturally change the topological structure of the problem during the optimization itself [19]. Algorithmically this is done either by switching off potential outlier constraints [18], [19] or replacing the quadratic cost function in the nonlinear optimization problem to a robust cost function [20], [21], such as the Huber robust cost function used in ORB-SLAM [20].

One theoretically appealing approach to handling DAs is using random finite sets (RFSs) [22]. Modeling the map and measurements as RFSs enables a fully integrated Bayesian SLAM solution that treats the DA uncertainty as a part of the estimation process [23]. In RFS-based SLAM frameworks, different RFSs are used to model the map, resulting in probability hypothesis density (PHD)-SLAM filters in [23]–[25], the labeled multi-Bernoulli (LMB)-SLAM filters in [26], [27], the δ-generalized labeled multi-Bernoulli (δ-GLMB)-SLAM filters in [28], [29], the Poisson multi-Bernoulli mixture (PMBM)-SLAM filters in [30], [31], and the Poisson multi-Bernoulli (PMB)-SLAM filters in [32], [33]. Within these RFS-based SLAM solutions, the PMBM-SLAM filters can explicitly consider all possible DAs, resulting in better performance by sacrificing time efficiency. Many RFS-based SLAM solutions, such as [24], [30], [34], apply Rao-Blackwellized particle (RBP) filter, similar as the FastSLAM solution, sampling the sensor trajectory and taking RFS likelihoods into consideration in the particle weight computation. To reduce the computational complexity, the number of required particles can be reduced by using an approximation of the optimal importance density (OID) to draw samples efficiently [35], [36]. In addition, low complexity alternatives are introduced in [25], [31]–[33], [37], which rely on linearization and jointly updating the sensor state and the map, dropping the cross-correlation between the sensor and the map, while keeping the RFS format. These approaches can have relatively low computational complexities by sacrificing the SLAM performance and robustness.

Although batch solutions to the backend problem are known to yield superior performance, most of the existing RFS-based SLAM solutions focus on filtering. An RFS-based batch processing SLAM algorithm is vector-generalized labeled multi-Bernoulli (V-GLMB)-SLAM [38], which relies on the joint V-GLMB distribution. A drawback of this approach is that it uses a generalized labeled multi-Bernoulli (GLMB) density to represent the landmarks, which requires more hypotheses than a PMBM density [39]. A recent study [40] shows that the batch PMBM implementation outperforms the batch GLMB implementation, with both approaches using Gibbs sampling to address the multi-scan DA problem in the multi-target tracking (MTT) task. Considering that the DA problem in batch SLAM resembles the DA problem in the batch MTT problem, DA techniques designed for MTT can be leveraged to address the DA in graph-based SLAM. One possible solution to the DA association problem in MTT is to use sampling-based methods [41]–[43], which have been shown to exhibit excellent performance in challenging scenarios. Markov chain Monte Carlo (MCMC) sampling methods were proposed in [41], [42], [44] to handle the DA problem, using the Gibbs sampling [43] or/and the merge-split Metropolis-Hastings (MH) algorithms [45].

In this paper, we present a novel method that combines the advantages of batch processing with RFS for a theoretically optimal treatment of the DAs. The proposed approach can overcome the limitation of RFS-based SLAM methods, which are restricted to sequential processing, and graph-based SLAM methods, which rely on heuristics to handle the DA problem. Our approach is inspired by collapsed Gibbs sampling technique [46], [47], and iteratively applies two methods: (i) an MCMC sampling method based on the PMBM density to solve the DA uncertainty; (ii) a graph-based SLAM solver for a set of landmarks and the sensor trajectory conditioned on a specific DA and existences of landmarks. In the end, the final sensor trajectory and the map are acquired through a post-processing marginalization step, which involves merging the SLAM results from different iterations and considering the undetected landmarks. Our main contributions are summarized as follows:

- **The development of a novel SLAM algorithm**: We designed a new Graph PMBM-SLAM approach, which embodies a cyclic process of sampling, and graph-based SLAM. The framework bridges RFS theory and graph-based SLAM, where the RFS theory is leveraged to devise a sampling-based method for addressing the DA uncertainty, and graph-based SLAM serves as an optimal solution for tackling the SLAM problem given a determined DA and existences of landmarks. This integration provides a new effective and robust SLAM solution. Via simulation, this iterative refinement process achieves performance close to the posterior Cramér-Rao bound (PCRB), along with high accuracy and robustness in challenging scenarios.
- **The derivation of a new MCMC sampling method for batch SLAM**: Based on the RFS theory, a novel MCMC sampling method is formulated for addressing the DA problem. The proposed sampling method combines the Gibbs and the MH algorithms and exhibits superior performance compared to the Gibbs sampling and the MH algorithms on their own, providing reliable DA solutions for the batch SLAM problem.

- **The derivation of a novel marginalization algorithm for post-processing**: The GraphSLAM algorithm tackles the SLAM problem given a determined DA and existences of landmarks. By merging GraphSLAM results for different posterior samples of DA and existences of landmarks, and considering the undetected landmarks, the sensor trajectory and the PMB representation of the set of landmarks are estimated, including the existence probability of each detected landmark.

The subsequent sections of this article are structured as follows: Section II details the system models and introduces the fundamental concepts of the PMBM density. Section III focuses on the proposed Graph PMBM-SLAM approach designed for batch processing. Section IV elaborates on the representation of the DA problem and its solution using a sample-based approach. In Section V, the exploration is directed towards the GraphSLAM algorithm, conditioned on a specific DA sample and existences of landmarks. Section VI delineates the method for merging the map and sensor trajectory across iterations. The detailed demonstration of the simulated environment and the presentation of simulation results are provided in Section VII. Finally, Section VIII summarizes our concluding remarks.

Notations: Scalars (e.g., x) are denoted in italic, vectors (e.g., $\boldsymbol{x}$) in bold, matrices (e.g., $\boldsymbol{X}$) in bold capital letters, sets (e.g., $\mathcal{X}$) in calligraphic. The cardinality of a set or the number of elements in a sequence of sets is denoted by $|\cdot|$. The inner product of $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ is denoted by $\langle f; g\rangle = \int f(\boldsymbol{x})g(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$. The transpose is denoted by $(\cdot)^{\mathsf{T}}$, and the union of mutually disjoint sets is denoted by $\biguplus$. A multivariate Gaussian distribution with mean $\boldsymbol{u}$ and covariance $\boldsymbol{\Sigma}$ is denoted as $\mathcal{N}(\boldsymbol{u}, \boldsymbol{\Sigma})$, and $d_{\boldsymbol{x}} = \dim(\boldsymbol{x})$ is the dimension of $\boldsymbol{x}$. The i-th component of $\boldsymbol{x}$ is denoted by $[\boldsymbol{x}]_i$, and the component in the i-th row and j-th column of $\boldsymbol{X}$ is denoted by $[\boldsymbol{X}]_{i,j}$.

II. Models and PMBM Background

A. Sensor, Landmark, and Measurement Models

We assume static landmarks, and employ a point object model throughout this paper. Furthermore, we consider the sensor movement to follow a known transition function. The sensor state at time step k, denoted as $\boldsymbol{s}_k$, includes various components depending on the specific problem and scenario. The transition density of $\boldsymbol{s}_k$ can be expressed as [48]

$$f(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}) = \mathcal{N}(\boldsymbol{s}_k; \boldsymbol{v}(\boldsymbol{s}_{k-1}), \boldsymbol{Q}_{k-1}), \tag{1}$$

where $\boldsymbol{v}(\cdot)$ denotes a known transition function, and $\boldsymbol{Q}_{k-1}$ denotes a known covariance matrix. The map of the environment consists of various landmarks, which may correspond to small objects, such as street lamps, traffic signs, and similar features, reflecting surfaces, such as walls, building facades, and other structures. In this paper, all landmarks are assumed to remain static and persist throughout the entire time sequence. We model the map as an RFS, denoted as $\mathcal{X} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^I\}$, where $I = |\mathcal{X}|$ represents the total number of landmarks, and each element $\boldsymbol{x}^i \in \mathcal{X}$ denotes a specific landmark state. It is worth noting that both I and $\boldsymbol{x}^i \in \mathcal{X}$ are random, as $\mathcal{X}$ is modeled as an RFS [22, Section 2.3]. The prior distribution of the landmarks follows a Poisson point process (PPP).

We assume a point object model, where each landmark can generate at most one measurement per time instant. The detection probability $p_{\mathrm{D}}(\boldsymbol{x}^i, \boldsymbol{s}_k) \in [0, 1]$ is introduced to account for how likely there is a measurement from landmark $\boldsymbol{x}^i$, when the sensor has state $\boldsymbol{s}_k$. At time step k, a set of measurements $\mathcal{Z}_k = \{\boldsymbol{z}_k^1, \ldots, \boldsymbol{z}_k^{\hat{I}_k}\}$ is observed, where $\hat{I}_k$ is the number of measurements, and $\boldsymbol{z}_k^i \in \mathcal{Z}_k$ is a specific measurement. Assuming measurement $\boldsymbol{z}^i$ has originated from landmark $\boldsymbol{x}^i$, its likelihood is given by

$$f(\boldsymbol{z}_k^i|\boldsymbol{x}^i, \boldsymbol{s}_k) = \mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{h}(\boldsymbol{x}^i, \boldsymbol{s}_k), \boldsymbol{R}_k^i), \tag{2}$$

where $\boldsymbol{h}(\boldsymbol{s}_k, \boldsymbol{x}^i)$ is the known measurement function, which is a function of the sensor state and the landmark state, and $\boldsymbol{R}_k^i$ is the corresponding covariance matrix. It is important to note that usually $\hat{I}_k \neq I_k$, is due to clutter and missed detections. Apart from landmark-generated measurements, $\mathcal{Z}_k$ may contain clutter that is modeled as a PPP (see (3)), parameterized by the intensity function $c(\boldsymbol{z})$.

B. PMBM Density

An RFS is a random variable whose possible outcomes are sets with a finite number of unique elements. Specifically, in an RFS both the number of elements and the elements themselves are random [22, Section 2.3]. There are various types of RFS, with PMBM being one of the most significant [39]. In this subsection, a high-level introduction to the PMBM density is provided, without accounting for time dependency. The distribution of the landmarks in the map is a PMBM, given the sensor trajectory, measurements, and the considered models. In the PMBM representation of the map, the set of landmarks is separated into two disjoint sets: the set of undetected landmarks, which are the landmarks that have never been detected, and the set of detected landmarks, which are the landmarks that have been detected at least once. Therefore, $\mathcal{X}$ can be divided into two mutually disjoint sets.

The set of undetected landmarks $\mathcal{X}_{\mathrm{U}}$ is modeled as a PPP, and the set of detected landmarks is modeled as an multi-Bernoulli mixture (MBM) $\mathcal{X}_{\mathrm{D}}$, which results in $\mathcal{X} = \mathcal{X}_{\mathrm{U}} \biguplus \mathcal{X}_{\mathrm{D}}$ following a PMBM density [39], [42], [49]. The PPP density $f_{\mathrm{P}}(\mathcal{X}_{\mathrm{U}})$ is given by [22, Section 4.3.1]

$$f_{\mathrm{P}}(\mathcal{X}_{\mathrm{U}}) = e^{-\int \lambda_{\mathrm{U}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}} \prod_{\boldsymbol{x}\in\mathcal{X}_{\mathrm{U}}} \lambda_{\mathrm{U}}(\boldsymbol{x}), \tag{3}$$

where $\lambda_{\mathrm{U}}(\cdot)$ is the intensity function, and the density can be parameterized by $\lambda_{\mathrm{U}}(\boldsymbol{x})$. The MBM density $f_{\mathrm{MBM}}(\mathcal{X}_{\mathrm{D}})$ is

$$f_{\mathrm{MBM}}(\mathcal{X}_{\mathrm{D}}) = \sum_{j\in\mathbb{J}} w^j \sum_{\biguplus_{i\in\mathbb{I}^j} \mathcal{X}^i = \mathcal{X}_{\mathrm{D}}} \prod_{i=1}^{|\mathbb{I}^j|} f_{\mathrm{B}}^{j,i}(\mathcal{X}^i), \tag{4}$$

where $\mathbb{J}$ is the index set of all global hypotheses, which corresponds to DAs in the SLAM problem [49], $w^j \geq 0$ is the weight for j-th global hypothesis, satisfying $\sum_{j\in\mathbb{J}} w^j = 1$, and $\mathbb{I}^j$ is the index set of landmarks (i.e., the Bernoulli components)

under the j-th global hypothesis with density

$$f_{\mathrm{B}}^{j,i}(\mathcal{X}^i) = \begin{cases} 1 - r^{j,i} & \mathcal{X}^i = \varnothing, \\ r^{j,i} f^{j,i}(\boldsymbol{x}) & \mathcal{X}^i = \{\boldsymbol{x}\}, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where $r^{j,i} \in [0,1]$ is the existence probability, and $f^{j,i}(\cdot)$ is the state density [22, Section 4.3.3]. A larger $r^{j,i}$ means that the landmark is more likely to be present. If $r^{j,i} = 0$, the corresponding landmark does not exist, and if $r^{j,i} = 1$, the corresponding landmark surely exists. The MBM density can be parameterized as $\{w^j, \{r^{j,i}, f^{j,i}(\boldsymbol{x})\}_{i\in\mathbb{I}^j}\}_{j\in\mathbb{J}}$. Following the convolution formula [22, eq. (4.17)], the PMBM density $f(\mathcal{X})$ is given by [39]

$$f(\mathcal{X}) = \sum_{\mathcal{X}_{\mathrm{U}} \uplus \mathcal{X}_{\mathrm{D}} = \mathcal{X}} f_{\mathrm{P}}(\mathcal{X}_{\mathrm{U}}) f_{\mathrm{MBM}}(\mathcal{X}_{\mathrm{D}}), \tag{6}$$

which can be completely parameterized by $\lambda_{\mathrm{U}}(\boldsymbol{x})$ and $\{w^j, \{r^{j,i}, f^{j,i}(\boldsymbol{x})\}_{i\in\mathbb{I}^j}\}_{j\in\mathbb{J}}$. Please note that if there is only one mixture component in the MBM, i.e., there is only one global DA, then (4) reduces to an multi-Bernoulli (MB), and (6) reduces to a PMB. If there are no detected landmarks ($\mathcal{X}_{\mathrm{D}} = \varnothing$), (6) reduces to a PPP.

III. Graph PMBM-SLAM Algorithm

This section introduces the proposed Graph PMBM-SLAM algorithm, which combines RFS and GraphSLAM. The framework seeks to leverage the advantages of both methods to obtain a SLAM solution, where the RFS posterior serves for an elegant and theoretically sound treatment of the DA uncertainties, and the GraphSLAM serves as a computationally efficient and robust backend algorithm, conditioned on a DA and the existences of the landmarks.

A. Joint Posterior Expressions

1) Sensor Trajectory and Map Posterior: The posterior distribution is denoted by $f(\boldsymbol{s}_{0:K}, \mathcal{X} \mid \mathcal{Z}_{1:K})$, where $\boldsymbol{s}_{0:K}$ denotes the sensor trajectory, and $\mathcal{Z}_{1:K} = (\mathcal{Z}_1, \ldots, \mathcal{Z}_K)$ denotes the measurement batch (i.e., the sequence of measurements up to time step K). We can factorize $f(\boldsymbol{s}_{0:K}, \mathcal{X} \mid \mathcal{Z}_{1:K})$ as

$$f(\boldsymbol{s}_{0:K}, \mathcal{X} | \mathcal{Z}_{1:K}) = \frac{f(\boldsymbol{s}_0) f(\mathcal{X}) \prod_{k=1}^{K} f(\boldsymbol{s}_k | \boldsymbol{s}_{k-1}) g(\mathcal{Z}_{1:K} | \boldsymbol{s}_{1:K}, \mathcal{X})}{f(\mathcal{Z}_{1:K})}, \tag{7}$$

where $f(\boldsymbol{s}_0)$ denotes the sensor prior density, $f(\mathcal{X})$ denotes the prior set density of the landmark set, $f(\boldsymbol{s}_k | \boldsymbol{s}_{k-1})$ was introduced in (1), $g(\mathcal{Z}_{1:K} | \boldsymbol{s}_{1:K}, \mathcal{X})$ denotes the likelihood function of measurement batch $\mathcal{Z}_{1:K}$ given $\boldsymbol{s}_{1:k}$ and $\mathcal{X}$, and $f(\mathcal{Z}_{1:K})$ is the normalizing factor. By assuming that the prior is a PPP with intensity $\lambda(\boldsymbol{x})$, which does not depend on the measurements [42], and plugging all these expressions into (7), the joint posterior can be expressed in a more explicit form. Please note that the prior PPP intensity $\lambda(\boldsymbol{x})$ depends on $\boldsymbol{x}$, making the PPP inhomogeneous. However, the intensity can be constant in the area of interest, in which case the PPP is homogeneous.

We first proceed to define the required notation. As $j \in \mathbb{J}$ denotes the index set of DAs, we can partition $\mathcal{Z}_{1:K}$ into subsets $\mathcal{Z}_{1:K}^{j,1}, \ldots, \mathcal{Z}_{1:K}^{j,|\mathbb{I}^j|}$ based on the j-th DA. The i-th subset in this partition, $\mathcal{Z}_{1:K}^{j,i}$, contains all measurements associated with the same unique origin $\mathcal{Y}^{j,i}$. Here, $\mathcal{Y}^{j,i}$ is an RFS distributed according to the i-th Bernoulli component under the j-th global hypothesis, with its measurement sequence represented as $\mathcal{Z}_{1:K}^{j,i} = (\mathcal{Z}_1^{j,i}, \ldots, \mathcal{Z}_K^{j,i})$, where $\mathcal{Z}_k^{j,i}$ denotes the measurement set from the source $\mathcal{Y}^{j,i}$ at time step k. As the landmarks can only create one measurement per time step, we have that $|\mathcal{Z}_k^{j,i}| \leq 1$. When $\mathcal{Y}^{j,i} = \varnothing$ and $|\mathcal{Z}_{1:K}^{j,i}| = 1$, $\mathcal{Z}_{1:K}^{j,i}$ contains a single clutter measurement. Overall, it holds that $\mathcal{Z}_{1:K} = (\uplus_i \mathcal{Z}_1^{j,i}, \ldots, \uplus_i \mathcal{Z}_K^{j,i})$.

Theorem 1. *The joint posterior* (7) *can be expressed as*

$$f(\boldsymbol{s}_{0:K}, \mathcal{X} | \mathcal{Z}_{1:K}) = e^{-\int \lambda(\boldsymbol{x}) d\boldsymbol{x} - K \int c(\boldsymbol{z}) d\boldsymbol{z}} \sum_{j\in\mathbb{J}} \sum_{\mathcal{X}_{\mathrm{U}} \uplus \mathcal{Y}^{j,1} \uplus \cdots \uplus \mathcal{Y}^{j,|\mathbb{I}^j|} = \mathcal{X}} \prod_{\boldsymbol{x}\in\mathcal{X}_{\mathrm{U}}} \left(p_U(\boldsymbol{x}, \boldsymbol{s}_{1:K}) \lambda(\boldsymbol{x})\right) f(\boldsymbol{s}_0) \prod_{k=1}^{K} f(\boldsymbol{s}_k | \boldsymbol{s}_{k-1}) \prod_{i=1}^{|\mathbb{I}^j|} \left(t(\mathcal{Z}_{1:K}^{j,i} | \boldsymbol{s}_{1:K}, \mathcal{Y}^{j,i}) \lambda(\mathcal{Y}^{j,i})\right) / f(\mathcal{Z}_{1:K}). \tag{8}$$

In (8), $\mathcal{X}_{\mathrm{D}} = \mathcal{Y}^{j,1} \uplus \cdots \uplus \mathcal{Y}^{j,|\mathbb{I}^j|}$ *are all detected landmarks, and* $p_U(\boldsymbol{x}, \boldsymbol{s}_{1:K}) = \prod_{k=1}^{K} (1 - p_D(\boldsymbol{x}, \boldsymbol{s}_k))$ *denotes the misdetection probability for landmarks that have not been detected for the whole time period. Moreover,* $t(\mathcal{Z}_{1:K}^{j,i} | \boldsymbol{s}_{1:K}, \mathcal{Y}^{j,i})$ *denotes the likelihood of* $\mathcal{Z}_{1:K}^{j,i}$ *and is given by*

$$t(\mathcal{Z}_{1:K}^{j,i} | \boldsymbol{s}_{1:K}, \mathcal{Y}^{j,i}) = \begin{cases} c(\boldsymbol{z}) & |\mathcal{Z}_{1:K}^{j,i}| = 1, \mathcal{Y}^{j,i} = \varnothing, \\ \prod_{k=1}^{K} \ell(\mathcal{Z}_k^{j,i} | \boldsymbol{s}_k, \boldsymbol{x}^i) & |\mathcal{Z}_{1:K}^{j,i}| \geq 1, \mathcal{Y}^{j,i} = \{\boldsymbol{x}^i\}, \\ 0 & \text{otherwise}, \end{cases} \tag{9}$$

where

$$\ell(\mathcal{Z}_k^{j,i} | \boldsymbol{s}_k, \boldsymbol{x}^i) = \begin{cases} 1 - p_D(\boldsymbol{x}^i, \boldsymbol{s}_k) & \mathcal{Z}_k^{j,i} = \varnothing, \\ p_D(\boldsymbol{x}^i, \boldsymbol{s}_k) f(\boldsymbol{z} | \boldsymbol{x}^i, \boldsymbol{s}_k) & \mathcal{Z}_k^{j,i} = \{\boldsymbol{z}\}, \\ 0 & \text{otherwise}. \end{cases} \tag{10}$$

Finally, $\lambda(\mathcal{Y}^{j,i})$ *in* (8) *denotes the prior intensity defined on the set* $\mathcal{Y}^{j,i}$ *given by*

$$\lambda(\mathcal{Y}^{j,i}) = \begin{cases} 1 & \mathcal{Y}^{j,i} = \varnothing, \\ \lambda(\boldsymbol{x}^i) & \mathcal{Y}^{j,i} = \{\boldsymbol{x}^i\}, \\ 0 & \text{otherwise}. \end{cases} \tag{11}$$

Proof. See Appendix A. □

For using GraphSLAM, a weakness of the introduced notation is that the partition of $\mathcal{Z}_{1:K}$ into $\mathcal{Z}_{1:K}^{j,1}, \ldots, \mathcal{Z}_{1:K}^{j,|\mathbb{I}^j|}$ may contain subsets that only comprise a clutter measurement without any corresponding landmarks (see the first entry in (9)), which creates ambiguity in determining the actual number of landmarks. To address this, inspired by the MBM01 representation in [39, Section IV.A], we introduce an auxiliary variable $\psi^{j,i} \in \{0,1\}$, where[1] $\psi^{j,i} = 1$ indicates that $\mathcal{Y}^{j,i}$ is non-

[1]The usage of $\psi^{j,i}$ is similar to the expansion of a Bernoulli density into the sum of two Bernoulli densities with deterministic target existences in the MBM01 representation in [39, Section IV.A].

empty so that the corresponding landmark exists, while $\psi^{j,i}=0$ indicates that $\mathcal{Y}^{j,i}$ is empty so that the corresponding landmark does not exist and the measurement is clutter. Therefore, $\psi^{j,i}$ is entirely determined by the emptiness of $\mathcal{Y}^{j,i}$. With this auxiliary variable, we can write (9) and (11) as an MBM

$$t(\mathcal{Z}_{1:K}^{j,i}|\boldsymbol{s}_{1:K},\mathcal{Y}^{j,i})\lambda(\mathcal{Y}^{j,i}) = \tilde{f}(\mathcal{Z}_{1:K}^{j,i},\mathcal{Y}^{j,i}|\boldsymbol{s}_{1:K},\psi^{j,i}=0)+\tilde{f}(\mathcal{Z}_{1:K}^{j,i},\mathcal{Y}^{j,i}|\boldsymbol{s}_{1:K},\psi^{j,i}=1), \tag{12}$$

where

$$\tilde{f}(\mathcal{Z}_{1:K}^{j,i},\mathcal{Y}^{j,i}|\boldsymbol{s}_{1:K},\psi^{j,i}) = \begin{cases}(1-\psi^{j,i})c(\boldsymbol{z}) & \mathcal{Y}^{j,i}=\varnothing,\\ \psi^{j,i}\prod_{k=1}^{K}\ell(\mathcal{Z}_k^{j,i}|\boldsymbol{s}_k,\boldsymbol{x}^i)\lambda(\boldsymbol{x}^i) & \mathcal{Y}^{j,i}=\{\boldsymbol{x}^i\},\\ 0 & \text{otherwise.}\end{cases} \tag{13}$$

Then, we have

$$f(\boldsymbol{s}_{0:K},\mathcal{X}|\mathcal{Z}_{1:K}) = \frac{e^{-\int\lambda(\boldsymbol{x})\mathrm{d}\boldsymbol{x}-K\int c(\boldsymbol{z})\mathrm{d}\boldsymbol{z}}}{f(\mathcal{Z}_{1:K})} \sum_{j\in\mathbb{J}}\sum_{\mathcal{X}_{\mathrm{U}}\uplus\mathcal{Y}^{j,1}\uplus\cdots\uplus\mathcal{Y}^{j,|\mathbb{I}^j|}=\mathcal{X}}\prod_{\boldsymbol{x}\in\mathcal{X}_{\mathrm{U}}}(p_{\mathrm{U}}(\boldsymbol{x},\boldsymbol{s}_{1:K})\lambda(\boldsymbol{x}))\, f(\boldsymbol{s}_0) \prod_{k=1}^{K}f(\boldsymbol{s}_k|\boldsymbol{s}_{k-1})\prod_{i=1}^{|\mathbb{I}^j|}\sum_{\psi^{j,i}}\tilde{f}(\mathcal{Z}_{1:K}^{j,i},\mathcal{Y}^{j,i}|\boldsymbol{s}_{1:K},\psi^{j,i}). \tag{14}$$

2) Sensor Trajectory, Map, and DA Posterior: We now proceed to express the joint posterior of the map, the sensor trajectory, and the DA. We introduce $\mathcal{A}$ as the partition of $\mathcal{Z}_{1:K}$ into $\mathcal{Z}_{1:K}^{j,1},\ldots,\mathcal{Z}_{1:K}^{j,|\mathbb{I}^j|}$, which corresponds to a DA (see Section IV-A), and $\boldsymbol{\psi}^j=[\psi^{j,1},\ldots,\psi^{j,|\mathbb{I}^j|}]$ with $\psi^{j,i}\in\{0,1\},\forall i\in\{1,\ldots,|\mathbb{I}^j|\}$, which describes the existence of each corresponding landmark of $\mathcal{Z}_{1:K}^{j,1},\ldots,\mathcal{Z}_{1:K}^{j,|\mathbb{I}^j|}$. By further introducing $\tilde{\mathcal{A}}=(\mathcal{A},\boldsymbol{\psi}^j)$ as an auxiliary variable in (14), we have

$$f(\boldsymbol{s}_{0:K},\mathcal{X},\tilde{\mathcal{A}}|\mathcal{Z}_{1:K}) = \frac{e^{-\int\lambda(\boldsymbol{x})\mathrm{d}\boldsymbol{x}-K\int c(\boldsymbol{z})\mathrm{d}\boldsymbol{z}}}{f(\mathcal{Z}_{1:K})} \sum_{\mathcal{X}_{\mathrm{U}}\uplus\mathcal{Y}^{j,1}\uplus\cdots\uplus\mathcal{Y}^{j,|\mathbb{I}^j|}=\mathcal{X}}\prod_{\boldsymbol{x}\in\mathcal{X}_{\mathrm{U}}}(p_{\mathrm{U}}(\boldsymbol{x},\boldsymbol{s}_{1:K})\lambda(\boldsymbol{x}))\, f(\boldsymbol{s}_0) \prod_{k=1}^{K}f(\boldsymbol{s}_k|\boldsymbol{s}_{k-1})\prod_{i=1}^{|\mathbb{I}^j|}\tilde{f}(\mathcal{Z}_{1:K}^{j,i},\mathcal{Y}^{j,i}|\boldsymbol{s}_{1:K},\psi^{j,i}). \tag{15}$$

B. Overall Framework

To determine the posterior density $f(\boldsymbol{s}_{0:K},\mathcal{X}|\mathcal{Z}_{1:K})$, we take inspiration from the collapsed Gibbs sampling technique [46], [47]. The core idea of the proposed algorithm is to iteratively update 1) the DAs, and 2) the map, and the sensor trajectory. In principle, these two steps can be executed either through sampling, as in a Gibbs sampling algorithm, or through optimization, as in a coordinate descent algorithm. In our proposed approach, we sample the DAs and optimize the map and sensor trajectory using GraphSLAM. However, other combinations of these steps are also possible. We refer to the method as a modified collapsed Gibbs sampling algorithm. The term "collapsed" indicates that we condition the sampling step only on the sensor trajectory, with the map analytically marginalized, and "modified" denotes that we estimate the sensor trajectory instead of sampling it. The modified collapsed Gibbs sampling iterates the following two stages:

1) *Sampling DAs (See Section IV):* Sample a candidate $\tilde{\mathcal{A}}$ value from $f(\tilde{\mathcal{A}}|\boldsymbol{s}_{0:K},\mathcal{Z}_{1:K})$ based on the latest estimate of $\boldsymbol{s}_{0:K}$.
2) *GraphSLAM (See Section V):* Perform the GraphSLAM algorithm on $f(\boldsymbol{s}_{0:K},\mathcal{X}_{\mathrm{D}}|\mathcal{Z}_{1:K},\tilde{\mathcal{A}})$ to obtain conditional posteriors of the detected landmarks and a sensor trajectory, for the sampled $\tilde{\mathcal{A}}$.

Please note that this combination of sampling and GraphSLAM does not yield samples from (8). In a full Gibbs sampler, both the DAs and sensor trajectories would be sampled iteratively. In contrast, in our proposed approach, instead of drawing a sample from the corresponding conditional distribution of the sensor trajectory, we make a point estimate of the variable given by the maximum of the conditional distribution through GraphSLAM. In fact, if all sampling steps were replaced by point estimates, we would obtain the iterated conditional modes (ICM) algorithm [50, Section 8.3.3]. Therefore, our approach can be seen as a greedy approximation of the full Gibbs sampler [50, Section 11.3]. However, this approximation does not affect convergence, as both the Gibbs sampler and ICM converge.

The final sensor trajectory and the map are acquired through a post-processing step, which involves merging the SLAM results from different iterations and considering the undetected landmarks. The corresponding flowchart of the Graph PMBM-SLAM algorithm is summarized in Fig. 2.

IV. Data Association Sampling

In this section, the batch DA sampling problem is described. First, the DA representation is introduced, and then its weight is computed. To simplify the sampling process, instead of directly sampling $\tilde{\mathcal{A}}$, we firstly sample $\mathcal{A}$ from $f(\mathcal{A}|\mathcal{Z}_{1:K},\boldsymbol{s}_{0:K})$, and then sample $\boldsymbol{\psi}$ from $f(\boldsymbol{\psi}|\mathcal{Z}_{1:K},\boldsymbol{s}_{0:K},\mathcal{A})$.

A. Data Association Representation

Each DA is a valid assignment of the measurements to their sources (landmarks or clutter), which is equivalent to partition $\mathcal{Z}_{1:K}$ into valid non-empty subsets according to sources, i.e., $\mathcal{Z}_{1:K}^{j,1},\ldots,\mathcal{Z}_{1:K}^{j,|\mathbb{I}^j|}$ for the j-th DA in (8). In this section, we index the measurements in $\mathcal{Z}_{1:K}$ by $\boldsymbol{m}\in\mathbb{M}$, where $\boldsymbol{m}=(k,\alpha_k)$, with $k\in\{1,\ldots,K\}$ representing the time index and $\alpha_k\in\{1,\ldots,|\mathcal{Z}_k|\}$ representing the index of a measurement in scan $k\le K$ [49]. A DA can now be equivalently viewed as a valid partition of $\mathbb{M}$ into nonempty disjoint index subsets. Each subset (called a *cell* in this paper) contains indices of all measurements from the same source. Hence, consider $\mathcal{Z}_{1:K}^{j,i}$, then the i-th cell of global hypothesis j is $\mathcal{C}^{j,i}=\{\boldsymbol{m}|\boldsymbol{z}_{\boldsymbol{m}}\in\mathcal{Z}_{1:K}^{j,i}\}$.

A valid DA must satisfy several criteria: (i) each measurement can be associated with at most one landmark, so that two cells should be disjoint; (ii) due to the point object assumption, a landmark cannot generate more than one measurement at each time step. Therefore, any cell cannot contain more than

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2025.3567916

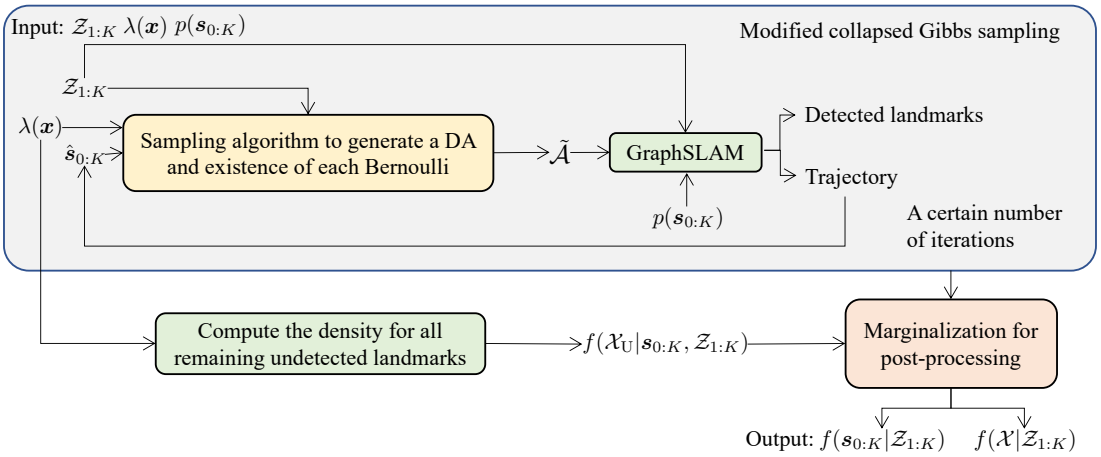


Fig. 2. The flowchart of the proposed Graph PMBM-SLAM algorithm. We generate a DA using a sampling algorithm, which is conditioned on the sensor trajectory from the last iteration (or the prior trajectory). Then we sample the existence probabilities of all the resulting Bernoulli components. Conditioned on the sampled DA and the existence of each Bernoulli, we apply the GraphSLAM algorithm to estimate the sensor trajectory and the detected landmarks. We iterate these two steps for a certain number of times. Finally, we merge GraphSLAM results from the last Γ iterations to output the estimates of the sensor trajectory and the map.

one measurement index with the same time index; (iii) the union of all cells is the index space $\mathbb{M}$. In summary, a valid partition (DA) $\mathcal{A}^j \in \mathbb{A}$ should satisfy

$$\begin{aligned}\mathcal{A}^j = \{\{\mathcal{C}^{j,1}, \ldots, \mathcal{C}^{j,|\mathbb{I}^j|}\}| \\ |\bigcup_{\boldsymbol{m}\in\mathcal{C}^{j,\beta}} \{\boldsymbol{m}|[\boldsymbol{m}]_1 = k\}| \leq 1, \forall\beta, \forall k; \\ \mathcal{C}^{j,\beta} \cap \mathcal{C}^{j,\gamma} = \varnothing, \forall\beta \neq \gamma; \bigcup_{\beta} \mathcal{C}^{j,\beta} = \mathbb{M}\},\end{aligned} \tag{16}$$

where $\beta, \gamma \in \{1, \ldots, |\mathbb{I}^j|\}$.

Example 1. *Let* $\mathbb{M} = \{(1,1),(1,2),(2,1),(2,2),(3,1),(3,2)\}$. *One possible DA could be* $\{\{(1,1),(2,1),(3,2)\},\{(2,2),(3,1)\},\{(1,2)\}\}$, *so that a landmark is detected at all time steps 1, 2 and 3 with measurements* $\boldsymbol{z}_{(1,1)}$, $\boldsymbol{z}_{(2,1)}$ *and* $\boldsymbol{z}_{(3,2)}$, *respectively; another landmark is misdetected at time step 1 and then detected with measurements* $\boldsymbol{z}_{(2,2)}$ *and* $\boldsymbol{z}_{(3,1)}$ *at time step 2 and 3, respectively; measurement* $\boldsymbol{z}_{(1,2)}$ *can either be a clutter or from a different landmarks. The partition is permutation invariant, so that different orders of cells or different orders of elements in cells do not create a new partition.*

B. Data Association Weight

Here, we compute the DA hypothesis weight. The measurement set sequence $\mathcal{Z}^{j,i}_{1:K} = (\mathcal{Z}^{j,i}_1, \ldots, \mathcal{Z}^{j,i}_K)$ of the cell $\mathcal{C}^{j,i}$ contains all measurements associated to the same source over time, with $\mathcal{Z}^{j,i}_k = \{\boldsymbol{z}_{\boldsymbol{m}}|[\boldsymbol{m}]_1 = k, \boldsymbol{m} \in \mathcal{C}^{j,i}\}$. Therefore, once $\{\mathcal{C}^{j,1}, \ldots, \mathcal{C}^{j,|\mathbb{I}^j|}\}$ is determined, the split of the measurement batch $\mathcal{Z}_{1:K} = (\mathcal{Z}^{j,1}_{1:K}, \ldots, \mathcal{Z}^{j,|\mathbb{I}^j|}_{1:K})$ is determined, and vice versa. The weight $f(\mathcal{A}^j|\mathcal{Z}_{1:K}, \boldsymbol{s}_{1:K})$, is equivalent to $f(\{\mathcal{C}^{j,1}, \ldots, \mathcal{C}^{j,|\mathbb{I}^j|}\}|\boldsymbol{s}_{1:K}, \mathcal{Z}_{1:K})$, which we will denote as w^j for notational convenience, and is given by

$$w^j = f(\mathcal{A}^j|\boldsymbol{s}_{1:K}, \mathcal{Z}_{1:K}) \propto \prod_{i=1}^{|\mathbb{I}^j|} l^{j,i}, \tag{17}$$

where the proportionality constant, given by the normalizing constant of the factor $f(\mathcal{Z}_{1:K})$ in (8), ensures that $\sum_{j\in\mathbb{J}} w^j = 1$, and $l^{j,i} = f(\mathcal{Z}^{j,i}_{1:K}|\boldsymbol{s}_{1:K})$, which can be obtained by applying set integral [49, eq. (4)] on (9) over $\mathcal{Y}^{j,i}$, as

$$\begin{aligned} f(\mathcal{Z}^{j,i}_{1:K}|\boldsymbol{s}_{1:K}) &= \int t(\mathcal{Z}^{j,i}_{1:K}|\boldsymbol{s}_{1:K}, \mathcal{Y}^{j,i}) \lambda(\mathcal{Y}^{j,i}) \delta\mathcal{Y}^{j,i} \\ &= \begin{cases} c(\boldsymbol{z}) + \langle \prod_{k=1}^K \ell^{j,i}_k; \lambda\rangle & |\mathcal{Z}^{j,i}_{1:K}| = 1, \\ \langle \prod_{k=1}^K \ell^{j,i}_k; \lambda\rangle & |\mathcal{Z}^{j,i}_{1:K}| > 1, \\ 0 & \text{otherwise}, \end{cases} \end{aligned} \tag{18}$$

where $\langle \prod_{k=1}^K \ell^{j,i}_k; \lambda\rangle = \langle \prod_{k=1}^K \ell(\mathcal{Z}^{j,i}_k|\boldsymbol{s}_k, \boldsymbol{x}^i); \lambda(\boldsymbol{x}^i)\rangle$. Please note that any invalid DA that does not satisfy the conditions in (16) will have a weight of 0.

C. Data Association Sampling

We will now discuss a method for obtaining a DA sample from (17). Due to the intractably large number of possible DAs, especially for the batch problem, direct sampling from (17) is unpractical. We will utilize a Gibbs sampling and a MH algorithm. The Gibbs sampling algorithm may perform poorly with an undesired initial DA, if several cells need to be merged to get the correct DA, and the MH algorithm needs to pass through intermediate DAs with comparatively lower likelihood before forming larger cells, both of which lead to slow mixing of the Markov chain [41], [45]. Therefore, we propose a new batch DA sampling algorithm for a point object model, which combines the Gibbs sampling and the MH algorithms, and is summarized in Algorithm 1. The proposed algorithm leverages the strengths of both the Gibbs sampling and the MH algorithms, allowing it to not only handle groups of components effectively but also form larger cells before passing through the MH step. The details of both algorithms are presented next. Note that the algorithm discards a certain number of iterations due to the burn-in period of MCMC sampling.

Algorithm 1 DA Sampling

Input: Measurement $\mathcal{Z}_{1:K}$, index set $\mathbb{M}$, sensor trajectory $\boldsymbol{s}_{1:K}$, initial DA $\mathcal{A}'$;
Output: DA $\mathcal{A}^*$;
1: **repeat**
2: Gibbs sampling (Algorithm 2);
3: MH sampling (Algorithm 3);
4: **until** A certain number of iterations;
5: Return the last sample as $\mathcal{A}^*$.

1) Gibbs Sampling: We denote the sample at the ι-th iteration of the Gibbs sampler as $\mathcal{A}^\iota$, which is a valid DA. The cells in $\mathcal{A}^\iota$ can be indexed by $\{1,\dots,|\mathcal{A}^\iota|\}$. To obtain the $(\iota+1)$-th sample using Gibbs sampler from $\mathcal{A}^\iota$, we firstly take a single measurement index $\boldsymbol{m} \in \mathcal{C}^{\iota,\beta} \in \mathcal{A}^\iota$, from cell $\mathcal{C}^{\iota,\beta}$. Then, we consider all possible moves/actions of the index (the case of no move is included in the two actions):

- Swap $\boldsymbol{m}$ from $\mathcal{C}^{\iota,\beta}$ with the index which has the same time index as $\boldsymbol{m}$ in the γ-th cell, $\gamma \in \{1,\dots,|\mathcal{A}^\iota|\}$, and if no such index exists, this action becomes a move of $\boldsymbol{m}$ from $\mathcal{C}^{\iota,\beta}$ to the γ-th cell. We denote the new resulting partition as $\mathcal{A}^\iota_{\beta\to\gamma}(\boldsymbol{m})$.[2]
- Move $\boldsymbol{m}$ from $\mathcal{C}^{\iota,\beta}$ to a new cell, which was an empty cell before the move, and we denote the new resulting partition as $\mathcal{A}^\iota_{\beta\to 0}(\boldsymbol{m})$.

In fact, both actions always yield a valid DA. No action duplicates $\boldsymbol{m}$, and $\boldsymbol{m}$ is only moved. Starting from a valid DA, if $\boldsymbol{m}$ is moved to another cell, it will be removed from its original cell, ensuring that two cells are always disjoint. Furthermore, we do not move $\boldsymbol{m}$ to a cell that already contains a component with the same time index; instead, we swap two measurement indices. This ensures that each cell contains at most one measurement index with the same time index. We do not add new measurement indices or remove existing ones, so the union of all cells remains equal to $\mathbb{M}$. In summary, the resulting DA always satisfies (16), and thus is always a valid DA. For notational brevity, we use the shorthand notation $w^\iota_{\beta\to\gamma}(\boldsymbol{m})$ to denote the transition probability of forming the new partition $\mathcal{A}^\iota_{\beta\to\gamma}(\boldsymbol{m})$. For each of all the possible options, $w^\iota_{\beta\to\gamma}(\boldsymbol{m})$ is computed, where a move is sampled from the resulting probability mass function (PMF) to form the new partition, denoted as

$$f(\mathcal{A}^{\iota+1} = \mathcal{A}^\iota_{\beta\to\gamma}(\boldsymbol{m}) | \mathcal{A}^\iota, \mathcal{Z}_{1:K}, \boldsymbol{s}_{1:K}) = w^\iota_{\beta\to\gamma}(\boldsymbol{m}), \quad (19)$$

for $\gamma = 0,1,\dots,|\mathcal{A}^\iota|$, and $\sum_{\gamma=0}^{|\mathcal{A}^\iota|} w^\iota_{\beta\to\gamma}(\boldsymbol{m}) = 1$.

As only two cells in $\mathcal{A}^\iota$ are changed at each sampling time, most of the factors in (17) are common, which reduces the computational cost significantly. In particular, $w^\iota_{\beta\to\gamma}(\boldsymbol{m})$ can be calculated more efficiently as [42]

$$w^\iota_{\beta\to\gamma}(\boldsymbol{m}) \propto \frac{\prod_{i=1}^{|\mathcal{A}^\iota|} l^{\iota,i}}{l^{\iota,\beta} l^{\iota,\gamma}} l^{\iota,\beta'} l^{\iota,\gamma'} \propto \frac{l^{\iota,\beta'} l^{\iota,\gamma'}}{l^{\iota,\beta} l^{\iota,\gamma}}, \quad (20)$$

where $l^{\iota,\beta}$ and $l^{\iota,\gamma}$ are the likelihood for $\mathcal{C}^{\iota,\beta}$ and $\mathcal{C}^{\iota,\gamma}$, respectively, and $l^{\iota,\beta'}$ and $l^{\iota,\gamma'}$ are the likelihood for $\mathcal{C}^{\iota,\beta'}$ and $\mathcal{C}^{\iota,\gamma'}$, respectively, which are the resulting cells after applying the action to $\mathcal{C}^{\iota,\beta}$ and $\mathcal{C}^{\iota,\gamma}$. Also, we have the special cases if $\mathcal{C}^{\iota,\gamma} = \varnothing$, $l^{\iota,\gamma} = 1$, and if $\mathcal{C}^{\iota,\beta'} = \varnothing$, $l^{\iota,\beta'} = 1$.[3] The resulting Gibbs sampling algorithm is summarized in Algorithm 2. This algorithm converges because the resulting Markov chain is ergodic. At each sampling iteration, the sampling space includes all possible DAs for an arbitrarily chosen measurement index at an arbitrary sampling step. Consequently, it is possible to transition between one valid DA and any other by modifying the differing measurement associations in a finite number of steps, ensuring the Markov chain is irreducible. Additionally, since there are many possible valid DAs at each sampling step, and it is also possible to remain in the current DA. Therefore, there is no guarantee that the chain will return to a DA after any number of transitions, which ensures that the Markov chain is aperiodic.

Example 2. *Let* $\mathcal{A}^\iota = \{\{(1,1),(2,1),(3,2)\},\{(1,2)\},\{(2,2),(3,1)\}\}$, *and* $\mathcal{A}^\iota_{1\to 3}((1,1))$ *denotes the resulting DA of moving the measurement index* $(1,1)$ *from its original cell to the third cell (the cell* $\{(2,2),(3,1)\}$*), which is* $\mathcal{A}^\iota_{1\to 3}((1,1)) = \{\{(2,1),(3,2)\},\{(1,2)\},\{(1,1),(2,2),(3,1)\}\}$, *and the transition probability of such move to forming* $\mathcal{A}^\iota_{1\to 3}((1,1))$ *is* $w^\iota_{1\to 3}((1,1))$.

Algorithm 2 Gibbs Sampling (one iteration)

Input: Batch measurements $\mathcal{Z}_{1:K}$, index set $\mathbb{M}$, sensor trajectory $\boldsymbol{s}_{1:K}$, DA $\mathcal{A}^{\text{in}}$;
Output: DA $\mathcal{A}^{\text{out}}$;
1: Set $\mathcal{A}^{\iota=0}$ as $\mathcal{A}^{\text{in}}$, and $\iota = 0$
2: **for** $n = 1 : |\mathbb{M}|$ **do**
3: Calculate transition prob. (19) for $\gamma \in \{0,1,\dots,|\mathcal{A}^\iota|\}$;
4: Draw sample $\mathcal{A}^{\iota+1}$;
5: $\iota \leftarrow \iota + 1$;
6: **end for**
7: Output the last sample as $\mathcal{A}^{\text{out}}$.

2) MH Algorithm: The Gibbs sampler can be slow as it only takes actions on one measurement index each time (or two for swapping), leading to slow mixing of the Markov chain [45]. To address these problems, the MH algorithm can be

[2]The process can be viewed as a form of Gibbs sampling. All resulting DAs are valid, even when swap actions are performed. Considering all possible resulting DAs, our MCMC kernel randomly selects one of these DAs by sampling from a distribution proportional to $f(\mathcal{A}^j|\mathcal{Z}_{1:K},\boldsymbol{s}_{0:K})$. While our notation for the conditioned event (or variables) may not be particularly elegant, our sampler still exhibits the Gibbs sampling property: it samples new variables from a distribution proportional to $f(\mathcal{A}^j|\mathcal{Z}_{1:K},\boldsymbol{s}_{0:K})$.

[3]Some implementation aspects: In (19), there could be some cases resulting $\mathcal{A}^{\iota+1} = \mathcal{A}^\iota$, i.e., if $\beta = \gamma$ in general, meaning the selected index stays in the same cell, or if $|\mathcal{C}^{\iota,\beta}| = 1$, the swapped cell only has one measurement index that has the same time index as $\boldsymbol{m}$ or $\gamma = 0$. To avoid considering the same move of the selected measurement index multiple times, we set $P(\mathcal{A}^{\iota+1} = \mathcal{A}^\iota_{\beta\to\gamma}(\boldsymbol{m})|\mathcal{A}^\iota,,\mathcal{Z}_{1:K},\boldsymbol{s}_{1:K}) = 0$ when $|\mathcal{C}^{\iota,\beta}| = 1$, for any γ satisfying $\gamma = \beta$ or $|\mathcal{C}^{\iota,\gamma}| = 1$ with its only measurement index having the same time index as $\boldsymbol{m}$. Avoiding the consideration of the same move can be viewed as modifying the sampling space of the proposal distribution used in MCMC. This adjustment does not impact the ergodicity of the Markov chain.

used. In the MH algorithm, instead of considering the action for a specific index at a sampling time, we consider the split of a cell or the merge of two cells. Since splits and merges change assignments for entire cells at each sampling time step, it leads to a faster algorithm and can avoid oscillations of a single index between these two cells. At each sampling time, if a single cell is chosen, the corresponding cell is considered for a cell split into two sub-cells; if two cells are chosen, the two corresponding cells are considered for a cell merge. There is only one way to merge two cells, while there are multiple ways to split a cell into two sub-cells. To avoid multiple possible split actions, we use the k-means++ algorithm [51] to split a cell, as suggested in [41].

By following [41, eq. (33)-(35)] and choosing the same proposal density, the acceptance probabilities of the split of cell $\mathcal{C}^{\iota,\beta}$ into two sub-cells $\mathcal{C}^{\iota,\beta,1}_{\text{split}}$ and $\mathcal{C}^{\iota,\beta,2}_{\text{split}}$, and the merge of two cells $\mathcal{C}^{\iota,\beta}$ and $\mathcal{C}^{\iota,\gamma}$ into a cell $\mathcal{C}^{\iota,\beta,\gamma}_{\text{merge}}$ are given by

$$P\{\text{split}\} = \min[1, \frac{l^{\iota,\beta,1}_{\text{split}} l^{\iota,\beta,2}_{\text{split}}}{l^{\iota,\beta}}], \tag{21}$$

$$P\{\text{merge}\} = \min[1, \frac{l^{\iota,\beta,\gamma}_{\text{merge}}}{l^{\iota,\beta} l^{\iota,\gamma}}], \tag{22}$$

respectively, with $l^{\iota,\beta,1}_{\text{split}}$, $l^{\iota,\beta,2}_{\text{split}}$, $l^{\iota,\beta,\gamma}_{\text{merge}}$, $l^{\iota,\beta}$, and $l^{\iota,\gamma}$ denoting the likelihood for the corresponding cells, obtained via (18). Please note that if $|\mathcal{C}^{\iota,\beta}| = 1$, the cell cannot be split, resulting in $P\{\text{split}\} = 0$ in (21); if $\mathcal{C}^{\iota,\beta}$ and $\mathcal{C}^{\iota,\gamma}$ contain indices with the same time index, two cells cannot be merged, resulting in $P\{\text{merge}\} = 0$ in (22) by setting $l^{\iota,\beta,\gamma}_{\text{merge}} = 0$ to effectively exclude this action from the sampling space.

The interpretation of (21) and (22) is: if the likelihood of the resulting DA is larger than the likelihood of the current DA, the action is for sure performed (with probability 1); if the likelihood of the resulting DA is smaller than the likelihood of the current DA, the action is performed with the probability of the value of the ratio of the likelihood of the resulting DA and the likelihood of the current DA, where we sample on this probability to decide if we perform the action or not. The resulting MH algorithm is summarized in Algorithm 3.

D. Sampling Existence Probabilities

Given the DA $\mathcal{A}^t = \{\mathcal{C}^{t,1}, \ldots, \mathcal{C}^{t,|\mathcal{A}^t|}\}$, there are $\mathcal{Y}^{t,1}, \ldots, \mathcal{Y}^{t,|\mathcal{A}^t|}$ Bernoullis in total, as each cell in $\mathcal{A}^t$ refers to a Bernoulli. Since $\boldsymbol{\psi}^{t,i}$ indicates whether the corresponding landmark exists or not (the Bernoulli $\mathcal{Y}^{t,i}$ is empty or contains the landmark), its corresponding probabilities are $p(\psi^{t,i} = 1 \mid \mathcal{A}^t, \mathcal{Z}_{1:K}, \boldsymbol{s}_{0:K}) = r^{t,i}_K$ and $p(\psi^{t,i} = 0 \mid \mathcal{A}^t, \mathcal{Z}_{1:K}, \boldsymbol{s}_{0:K}) = 1 - r^{t,i}_K$, respectively, with $r^{t,i}_K$ indicating the existence probability of $\mathcal{Y}^{t,i}$, given by [42, eq. (32)]

$$r^{t,i}_K = \begin{cases} \frac{\langle \prod_{k=1}^K \ell_k^{t,i}; \lambda \rangle}{c(\boldsymbol{z}) + \langle \prod_{k=1}^K \ell_k^{t,i}; \lambda \rangle} & |\mathcal{C}^{t,i}| = 1 \quad (|\mathcal{Z}^{t,i}_{1:K}| = 1), \\ 1 & |\mathcal{C}^{t,i}| > 1 \quad (|\mathcal{Z}^{t,i}_{1:K}| > 1), \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

Considering all components, the PMF of $\boldsymbol{\psi}$ is given by

$$f(\boldsymbol{\psi} \mid \mathcal{A}^t, \mathcal{Z}_{1:K}, \boldsymbol{s}_{0:K}) = \prod_{i=1}^{|\mathbb{I}^t|} p(\psi^{t,i} \mid \mathcal{A}^t, \mathcal{Z}_{1:K}, \boldsymbol{s}_{0:K}). \tag{24}$$

Algorithm 3 Metropolis-Hastings Algorithm (one iteration)

Input: Batch measurements $\mathcal{Z}_{1:K}$, index set $\mathbb{M}$, sensor trajectory $\boldsymbol{s}_{1:K}$, DA $\mathcal{A}^{\text{in}}$;
Output: DA $\mathcal{A}^{\text{out}}$;

Set $\mathcal{A}^{\iota=0}$ as $\mathcal{A}^{\text{in}}$, and $\iota = 0$
for $n = 1 : |\mathbb{M}|$ **do**
  for $n' = 1 : |\mathbb{M}|, n' \neq n$ **do**
    if the n-th and n'-th indices belong to the same cell in $\mathcal{A}^\iota$ **then**
      Split the cell into two sub-cells;
      Compute $P\{\text{split}\}$ and draw $\varsigma \sim U(0,1)$;
      if $\varsigma \leq P\{\text{split}\}$ **then** Split the cell;
      end if
    else
      Compute $P\{\text{merge}\}$ and draw $\varsigma \sim U(0,1)$;
      if $\varsigma \leq P\{\text{Merge}\}$ **then** Merge two cells;
      end if
    end if
    Set the resulting DA as $\mathcal{A}^{\iota+1}$;
    $\iota \leftarrow \iota + 1$;
  end for
end for
Output the last sample as $\mathcal{A}^{\text{out}}$.

Therefore, to draw a sample $\boldsymbol{\psi}^t$ from $p(\boldsymbol{\psi} \mid \mathcal{A}^t, \mathcal{Z}_{1:K}, \boldsymbol{s}_{0:K})$ is equivalently to sample $\psi^{t,i}$ from $p(\psi^{t,i} \mid \mathcal{A}^t, \mathcal{Z}_{1:K}, \boldsymbol{s}_{0:K})$ for all $i \in \{1, \ldots, |\mathcal{A}^t|\}$.[4]

V. GraphSLAM Given a Data Association

In Section IV, we generated a DA sample $\tilde{\mathcal{A}}$ based on the proposed sampling algorithm and performed sampling on the existence probabilities. In this section, we will focus on how to estimate $\boldsymbol{s}_{0:K}$ and $\mathcal{X}$ from $f(\boldsymbol{s}_{0:K}, \mathcal{X} | \mathcal{Z}_{1:K}, \tilde{\mathcal{A}})$ with the help of an MAP estimator.

A. Representation

By fixing the DA and the existence of each Bernoulli in (8), i.e, conditioning on $\tilde{\mathcal{A}}$, $f(\boldsymbol{s}_{0:K}, \mathcal{X} | \mathcal{Z}_{1:K}, \tilde{\mathcal{A}})$ follows

$$f(\boldsymbol{s}_{0:K}, \mathcal{X} | \mathcal{Z}_{1:K}, \tilde{\mathcal{A}}) = \frac{e^{-\int \lambda(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - K \int c(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}}}{f(\mathcal{Z}_{1:K} | \tilde{\mathcal{A}})} \sum_{\mathcal{X}_{\text{U}} \uplus \mathcal{Y}^{j,1} \uplus \cdots \uplus \mathcal{Y}^{j,|\mathbb{I}^j|} = \mathcal{X}} \prod_{\boldsymbol{x} \in \mathcal{X}_{\text{U}}} (p_{\text{U}}(\boldsymbol{x}, \boldsymbol{s}_{1:K}) \lambda(\boldsymbol{x})) \, f(\boldsymbol{s}_0) \prod_{k=1}^K f(\boldsymbol{s}_k | \boldsymbol{s}_{k-1}) \prod_{i \in \mathbb{I}^j} \tilde{f}(\mathcal{Z}^{j,i}_{1:K}, \mathcal{Y}^{j,i} | \boldsymbol{s}_{1:K}, \psi^{j,i}). \tag{25}$$

Once $\mathcal{A}$ is determined, $|\mathbb{I}^j|$ is fixed and the union $\mathcal{Y}^{j,1} \uplus \cdots \uplus \mathcal{Y}^{j,|\mathbb{I}^j|}$ indicates there are $|\mathbb{I}^j|$ landmark sets in

[4] Some implementation aspects: Although there are $|\mathcal{A}^t|$ components should be considered and $|\mathcal{A}^t|$ is not usually a small number, many of the components are 1, since these landmarks for sure exist and we can directly set the corresponding $\psi^{t,i}$ as 1, which corresponds to the second entry in (23). Then, only landmarks that correspond to the first entry in (23) (the landmark could either be a real landmark or a false alarm caused by clutter) need to be considered, and the number of which is usually not small. To further simplify the problem, we can also only directly set $\psi^{t,i}$ as 0 if the corresponding existence probability is lower than a threshold.

total, where the emptiness of each landmark set $\mathcal{Y}^{j,i}$ (i.e., the existence of each corresponding landmark) is determined by $\psi^{j,i}$. All the remaining landmarks $\mathcal{X}\backslash\mathcal{X}_{\mathrm{D}}$ are part of $\mathcal{X}_{\mathrm{U}}$. In (25), $\mathcal{X}_{\mathrm{U}}$ is dependent on $\boldsymbol{s}_{0:K}$ but independent to $(\mathcal{A},\boldsymbol{\psi})$. Therefore, we can obtain $f(\mathcal{X}_{\mathrm{U}}|\boldsymbol{s}_{0:K},\mathcal{Z}_{1:K})$, which is given by

$$f(\mathcal{X}_{\mathrm{U}}|\boldsymbol{s}_{0:K},\mathcal{Z}_{1:K}) = e^{-\int p_{\mathrm{U}}(\boldsymbol{x},\boldsymbol{s}_{1:K})\lambda(\boldsymbol{x})\mathrm{d}\boldsymbol{x}} \prod_{\boldsymbol{x}\in\mathcal{X}_{\mathrm{U}}} \left(p_{\mathrm{U}}(\boldsymbol{x},\boldsymbol{s}_{1:K})\lambda(\boldsymbol{x})\right), \quad (26)$$

which is a PPP density as shown in (3), with its intensity being $p_{\mathrm{U}}(\boldsymbol{x},\boldsymbol{s}_{1:K})\lambda(\boldsymbol{x})$. Please note that the resulting inhomogeneous PPP also depends on the sensor trajectory. In addition, by expanding $\tilde{f}(\mathcal{Z}_{1:K}^{j,i},\mathcal{Y}^{j,i}|\boldsymbol{s}_{1:K},\psi^{j,i})$ with (13), we have

$$f(\boldsymbol{s}_{0:K},\mathcal{Y}^{j,1},\ldots,\mathcal{Y}^{j,|\mathbb{I}^j|}|\mathcal{Z}_{1:K},\tilde{\mathcal{A}}) \propto f(\boldsymbol{s}_0)\prod_{k=1}^{K} f(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}) \times \prod_{i\in\mathbb{I}^j:\psi^{j,i}=1}\prod_{k=1}^{K} \ell(\mathcal{Z}_k^{j,i}|\boldsymbol{s}_k,\boldsymbol{x}^i)\lambda(\boldsymbol{x}^i), \quad (27)$$

where the proportionality also corresponds to $\tilde{f}(\mathcal{Z}_{1:K}^{j,i},\mathcal{Y}^{j,i}|\boldsymbol{s}_{1:K},\psi^{j,i}=0) = c(\boldsymbol{z})$. As $\boldsymbol{\psi}$ is fixed, $f(\boldsymbol{s}_{0:K},\mathcal{Y}^{j,1},\ldots,\mathcal{Y}^{j,|\mathbb{I}^j|}|\mathcal{Z}_{1:K},\tilde{\mathcal{A}})$ becomes an MB01 RFS, i.e., it is an MB RFS where the existence probabilities of all resulting Bernoullis are either 0 or 1 [39]. Our goal is to obtain estimates of $\boldsymbol{s}_{0:K}$ and $\mathcal{X}$ from (25).

B. GraphSLAM Approximations

For notational brevity, we drop the DA index j in the following two subsections. To enable the use of GraphSLAM [14], we apply several approximations. First, we note that the PPP part is independent of $(\mathcal{A},\boldsymbol{\psi})$ (26) and in most cases is not informative regarding the sensor state. Hence, we can first compute the MAP estimate of $\boldsymbol{s}_{0:K}$ and $\mathcal{Y}^1,\ldots,\mathcal{Y}^{|\mathcal{A}|}$ with GraphSLAM on (27), and then update the PPP intensity according to (26). Second, we drop non-existing landmarks based on $\boldsymbol{\psi}$. To this end, we introduce the number of existing landmarks as $\kappa = \sum_{i=1}^{|\mathcal{A}|}\psi^i$, and reorder $\mathcal{Y}^1,\ldots,\mathcal{Y}^{|\mathcal{A}|}$ to keep the first κ Bernoullis with $\psi^i=1$, and the rest $\kappa+1$ to $|\mathcal{A}|$ Bernoullis with $\psi^i=0$. We also introduce the random variable $\boldsymbol{q}_k^i = [\boldsymbol{s}_k^{\mathsf{T}},(\boldsymbol{x}^i)^{\mathsf{T}}]^{\mathsf{T}}$ comprising the sensor state at time k and the state of the corresponding landmark of $\mathcal{C}^i$, and the random variable $\boldsymbol{q} = [\boldsymbol{s}_0^{\mathsf{T}},\boldsymbol{s}_1^{\mathsf{T}},\ldots,\boldsymbol{s}_K^{\mathsf{T}},(\boldsymbol{x}^1)^{\mathsf{T}},\ldots,(\boldsymbol{x}^\kappa)^{\mathsf{T}}]^{\mathsf{T}}$. Then, $\boldsymbol{q}$ can be estimated by maximizing the posterior

$$\begin{aligned}&\arg\max_{\boldsymbol{q}}\ f(\boldsymbol{s}_{0:K},\mathcal{Y}^1,\ldots,\mathcal{Y}^{|\mathcal{A}|}|\mathcal{Z}_{1:K},\tilde{\mathcal{A}}) \quad (28)\\ &= \arg\max_{\boldsymbol{q}}\ f(\boldsymbol{s}_0)\prod_{k=1}^{K} f(\boldsymbol{s}_k|\boldsymbol{s}_{k-1})\prod_{i=1}^{\kappa}\prod_{k=1}^{K}\ell(\mathcal{Z}_k^i|\boldsymbol{s}_k,\boldsymbol{x}^i)\lambda(\boldsymbol{x}^i).\end{aligned}$$

Third, instead of using $\lambda(\boldsymbol{x}^i)$, we estimate a Gaussian distribution of $\boldsymbol{x}^i$, denoted as $f(\boldsymbol{x}^i)$, for those $i\le\kappa$, by following [31, Appendix A.C] with mean $\boldsymbol{u}^i$ determined by the first detected measurement (the measurement with the smallest time index in $\mathcal{Z}_{1:K}^i$, and we denote its corresponding index as $\boldsymbol{m}_{\mathrm{fir}}^i$) and the corresponding sensor state, and very large covariance $\boldsymbol{C}^i$. Fourth, we approximate $p_{\mathrm{D}}(\boldsymbol{x}^i,\boldsymbol{s}_k)$ in $\ell(\mathcal{Z}_k^i|\boldsymbol{s}_k,\boldsymbol{x}^i)$ from (10) to be a constant $p_{\mathrm{D}}>0$ in the field of view (FOV) of the sensor and 0 outside the FOV of the sensor[5]. Hence, (28) can be rewritten by

$$\arg\max_{\boldsymbol{q}}\ f(\boldsymbol{s}_0)\prod_{k=1}^{K} f(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}) \times\prod_{i=1}^{\kappa} f(\boldsymbol{x}^i)\prod_{k=1}^{K}\prod_{\boldsymbol{z}\in\mathcal{Z}_k^i:p_{\mathrm{D}}>0} f(\boldsymbol{z}|\boldsymbol{s}_k,\boldsymbol{x}^i). \quad (29)$$

C. GraphSLAM Optimization

By plugging (1) and (2) into (29), we can solve $\arg\min_{\boldsymbol{q}}\ \mathcal{E}(\boldsymbol{q})$ for the optimization problem in (29), where

$$\begin{aligned}\mathcal{E}(\boldsymbol{q}) = {}& (\boldsymbol{s}_0-\boldsymbol{\epsilon}_0)^{\mathsf{T}}\boldsymbol{P}_0^{-1}(\boldsymbol{s}_0-\boldsymbol{\epsilon}_0)+ \quad (30)\\ &\sum_{k=1}^{K}(\boldsymbol{s}_k-\boldsymbol{v}(\boldsymbol{\epsilon}_{k-1}))^{\mathsf{T}}\boldsymbol{Q}^{-1}(\boldsymbol{s}_k-\boldsymbol{v}(\boldsymbol{\epsilon}_{k-1}))+\\ &\sum_{i=1}^{\kappa}((\boldsymbol{x}^i-\boldsymbol{u}^i)^{\mathsf{T}}(\boldsymbol{C}^i)^{-1}(\boldsymbol{x}^i-\boldsymbol{u}^i)+\\ &\sum_{k=1}^{K}\sum_{\boldsymbol{z}\in\mathcal{Z}_k^i:p_{\mathrm{D}}>0}(\boldsymbol{z}-\boldsymbol{h}(\hat{\boldsymbol{q}}_k^i))^{\mathsf{T}}(\boldsymbol{R}_k^i)^{-1}(\boldsymbol{z}-\boldsymbol{h}(\hat{\boldsymbol{q}}_k^i))),\end{aligned}$$

with $\boldsymbol{\epsilon}_k$ and $\boldsymbol{P}_k$ denoting the mean and the covariance of $\boldsymbol{s}_k$ for $k\in\{0,\cdots,K\}$, respectively. To optimize (30), we start from an estimate $\hat{\boldsymbol{q}} = [\boldsymbol{\epsilon}_0^{\mathsf{T}},\boldsymbol{\epsilon}_1^{\mathsf{T}},\ldots,\boldsymbol{\epsilon}_K^{\mathsf{T}},(\boldsymbol{u}^1)^{\mathsf{T}},\ldots,(\boldsymbol{u}^\kappa)^{\mathsf{T}}]^{\mathsf{T}}$, and apply gradient descent [53, Chapter 9], as detailed in Appendix B. Here, $\boldsymbol{\epsilon}_k^{\mathsf{T}} = \boldsymbol{v}(\boldsymbol{\epsilon}_{k-1}^{\mathsf{T}})$, for $k\ge1$. After convergence, we obtain the final estimate $\hat{\boldsymbol{q}}$ and the associated information matrix $\boldsymbol{\Omega}$. The mean and covariance of $f(\boldsymbol{s}_{0:K}|\mathcal{Z}_{1:K},\tilde{\mathcal{A}})$ are then given by

$$\boldsymbol{\epsilon}_{0:K} = [\hat{\boldsymbol{q}}]_{1:\nu(K+1)}, \quad (31)$$

$$\boldsymbol{P}_{0:K} = [\boldsymbol{\Omega}^{-1}]_{1:\nu(K+1),1:\nu(K+1)}. \quad (32)$$

where $\nu=\dim(\boldsymbol{s}_k)$. Similarly, the mean and covariance of the map are given by

$$\boldsymbol{u}_{\mathrm{map}} = [\hat{\boldsymbol{q}}]_{\nu(K+1)+1:\mathrm{end}}, \quad (33)$$

$$\boldsymbol{C}_{\mathrm{map}} = [\boldsymbol{\Omega}^{-1}]_{\nu(K+1)+1:\mathrm{end},\nu(K+1)+1:\mathrm{end}}, \quad (34)$$

where $\boldsymbol{C}_{\mathrm{map}}$ is generally a full matrix, as landmarks are correlated to each other, when not conditioned on the sensor state trajectory. In addition, the updated mean and the covariance of each landmark can be directly obtained from $\boldsymbol{u}_{\mathrm{map}}$ and $\boldsymbol{C}_{\mathrm{map}}$ by taking the corresponding parts, denoted as

$$\boldsymbol{u}^i = [\boldsymbol{u}_{\mathrm{map}}]_{\mu(i-1)+1:\mu i}, \quad (35)$$

$$\boldsymbol{C}^i = [\boldsymbol{C}_{\mathrm{map}}]_{\mu(i-1)+1:\mu i,\mu(i-1)+1:\mu i}, \quad (36)$$

with $\mu=\dim(\boldsymbol{x}^i)$, and the existence probability $r^i=1$ since it exists for sure. For the remaining Bernoullis, the existence probability $r^i=0$, and the corresponding $\boldsymbol{u}^i$ and $\boldsymbol{C}^i$ do not exist. Therefore, we only need to output $\{r^i=1,\boldsymbol{u}^i,\boldsymbol{C}^i\}_{i\in\{1,\ldots,\kappa\}}$ for the map. It is important to note that $\boldsymbol{\Omega}$ usually has a high dimension, and taking the inverse is computationally costly.

[5]In principle, p_{D} is influenced by various factors, such as SNR and propagation environment [52]. However, the calculation of p_{D} is beyond the scope of this paper. The assumptions made here are intended to simplify the problem. Extending the approach to incorporate a more practical p_{D} is a straightforward task.

There are computationally efficiently methods to compute (31)–(36), e.g. [14, Section 5.5].

VI. Marginalization Over Samples

In this section, we describe how the final sensor trajectory, the MB of detected landmarks, and the PPP intensity of undetected landmarks are computed.

A. Marginalization

We keep the SLAM results from the last Γ iterations. Based on the collapsed Gibbs sampling theory, all Γ samples of $\tilde{\mathcal{A}}$, which we kept after the burn-in period, are equivalent to samples that are directly sampled from $f(\tilde{\mathcal{A}}|\mathcal{Z}_{1:K})$. For a specific sample $\tilde{\mathcal{A}}^t$, GraphSLAM provides $\boldsymbol{\epsilon}^t_{0:K}$ and $\boldsymbol{P}^t_{0:K}$, and $\{r^{t,i}, \boldsymbol{u}^{t,i}, \boldsymbol{C}^{t,i}\}_{i\in\{1,\ldots,\kappa^t\}}$ for $f(\boldsymbol{s}_{0:K}|\mathcal{Z}_{1:K}, \tilde{\mathcal{A}}^t)$ and $f(\mathcal{Y}^1, \ldots, \mathcal{Y}^{\kappa^t}|\mathcal{Z}_{1:K}, \tilde{\mathcal{A}}^t)$, respectively. Different samples of $\tilde{\mathcal{A}}^t$ lead to varying SLAM results. However, our objective is to derive a single, consolidated SLAM outcome as the final output, directly providing the sensor trajectory and the environmental map, which can be readily utilized for further analysis or applications. Therefore, the desired posterior approximations can be obtained by marginalizing across all $\tilde{\mathcal{A}}^t$ samples, with each sample having the same weight, given by

$$f(\boldsymbol{s}_{0:K}|\mathcal{Z}_{1:K}) \approx \frac{1}{\Gamma}\sum_{t=1}^{\Gamma} f(\boldsymbol{s}^t_{0:K}|\mathcal{Z}_{1:K}, \tilde{\mathcal{A}}^t), \tag{37}$$

$$f(\mathcal{Y}^1, \ldots, \mathcal{Y}^{|\mathbb{I}|}|\mathcal{Z}_{1:K}) \approx \frac{1}{\Gamma}\sum_{t=1}^{\Gamma} f(\mathcal{Y}^1, \ldots, \mathcal{Y}^{\kappa^t}|\mathcal{Z}_{1:K}, \tilde{\mathcal{A}}^t). \tag{38}$$

In terms of (37), the final updated trajectory has mean $\boldsymbol{\epsilon}_{0:K} \approx 1/\Gamma \sum_{t=1}^{\Gamma} \boldsymbol{\epsilon}^t_{0:K}$, and covariance $\boldsymbol{P}_{0:K} \approx \frac{1}{\Gamma}\sum_{t=1}^{\Gamma}(\boldsymbol{P}^t_{0:K} + (\boldsymbol{\epsilon}^t_{0:K} - \boldsymbol{\epsilon}_{0:K})(\boldsymbol{\epsilon}^t_{0:K} - \boldsymbol{\epsilon}_{0:K})^{\mathsf{T}})$. In terms of (38), each sample of the map follows the MB distribution, so that (38) is an MBM. To marginalize the MBM over all samples into a single MB, several practical aspects must be addressed: (i) the numbering of the landmarks across the samples $\tilde{\mathcal{A}}^t$; (ii) computation of the spatial density and existence probability of each MB: (iii) pruning and merging.

To address the first aspect, we introduce a vector to index all landmarks in the resulting MB of each sample, denoted as $\boldsymbol{\sigma}^t = [\sigma^t(1), \ldots, \sigma^t(\kappa^t)]$, defined as $\sigma^t(i) = \boldsymbol{m}^{t,i}_{\text{fir}}$, where we recall that $\boldsymbol{m}^{t,i}_{\text{fir}}$ is the index of the first (earliest) measurement in $\mathcal{C}^{t,i}$.[6] Different samples may have different $\boldsymbol{\sigma}^t$, since the source of each $\mathcal{C}^{t,i}$ may be different, which is $f(\boldsymbol{x}^i)$ generated using the corresponding measurement of $\boldsymbol{m}^{t,i}_{\text{fir}}$. To make $\boldsymbol{\sigma}^t$ consistent in all DAs, we pick up all unique $\boldsymbol{m}^{t,i}_{\text{fir}}$, and re-index them with $i \in \{1, \ldots, |\mathbb{I}|\}$, where $|\mathbb{I}|$ denotes the number of unique $\boldsymbol{m}^{t,i}_{\text{fir}}$ across all Γ samples, which represents all different landmarks over all DAs. Therefore, $\boldsymbol{\sigma}^t$ can be extended and rewritten as a vector $\tilde{\boldsymbol{\sigma}}^t$ with length $|\mathbb{I}|$ and components $\tilde{\sigma}^t(i) \in \{0, 1\}, \forall i \in \mathbb{I} = \{1, \ldots, |\mathbb{I}|\}$, where $\tilde{\sigma}^t(i) = 1$ means the corresponding landmark exists in the t-th sample, and $\tilde{\sigma}^t(i) = 0$ means the corresponding landmark non-exists in the t-th sample. We also extend and reorder $\{r^{t,i}, \boldsymbol{u}^{t,i}, \boldsymbol{C}^{t,i}\}_{i\in\{1,\ldots,\kappa^t\}}$ into $\{r^{t,i}, \boldsymbol{u}^{t,i}, \boldsymbol{C}^{t,i}\}_{i\in\mathbb{I}}$ by setting $r^{t,i} = 0$, if the corresponding $\tilde{\sigma}^t(i) = 0$. Then, the landmark MB for $i \in \mathbb{I}$ can be set to $r^i = \sum_{t=1}^{\Gamma} \tilde{\sigma}^t(i)/\Gamma$ and

$$\boldsymbol{u}^i = \frac{1}{\Gamma r^i}\sum_{t\in\{1,\ldots,\Gamma\}:\tilde{\sigma}^t(i)=1} \boldsymbol{u}^{t,i}, \tag{39}$$

$$\boldsymbol{C}^i = \frac{1}{\Gamma r^i}\sum_{t\in\{1,\ldots,\Gamma\}:\tilde{\sigma}^t(i)=1} (\boldsymbol{C}^{t,i} + (\boldsymbol{u}^{t,i} - \boldsymbol{u}^i)(\boldsymbol{u}^{t,i} - \boldsymbol{u}^i)^{\mathsf{T}}). \tag{40}$$

After marginalizing over $\tilde{\mathcal{A}}$, an updated MB to represent the map of all detected landmarks $\{r^i, \boldsymbol{u}^i, \boldsymbol{C}^i\}_{i\in\mathbb{I}}$ is acquired. Finally, we prune Bernoullis with low existence probabilities and merge Bernoullis which are very close to each other. The proposed method provides an efficient way to approximate the MBM into an MB. More accurate MB approximation methods exist, e.g., by finding the best-fitting MB that minimizes the Kullback–Leibler (KL) divergence [54].

B. PPP intensity for Undetected Landmarks

Apart from the detected landmarks, we have the updated PPP for all remaining undetected landmarks, $f(\mathcal{X}_{\text{U}}|\boldsymbol{s}_{0:K}, \mathcal{Z}_{1:K})$ in (26). We can also marginalize out the sensor trajectory to acquire $f(\mathcal{X}_{\text{U}}|\mathcal{Z}_{1:K})$, which results in the updated intensity as

$$\breve{\lambda}(\boldsymbol{x}) = \int f(\boldsymbol{s}_{0:K}|\mathcal{Z}_{1:K}) \prod_{k=1}^{K}(1 - p_{\text{D}}(\boldsymbol{s}_k, \boldsymbol{x}))\lambda(\boldsymbol{x})\mathrm{d}\boldsymbol{s}_{0:K}. \tag{41}$$

Together with the marginalized MB computed in Section VI-A, the final map is approximated as a PMB.

VII. Results

In this section, we assess the proposed algorithm in a simulated vehicular setting, conducting a comparison with a benchmark. We outline the simulation environment, detail the performance metrics, and describe the benchmark algorithm before analyzing SLAM performance.

A. Simulation Environment

We consider a propagation environment of bistatic radio SLAM, similar to [30], [31], featuring a single vehicle as the user equipment (UE), as shown in Fig 3. There is a single base station (BS) in the environment located at $[0\,\text{m}, 0\,\text{m}, 40\,\text{m}]^{\mathsf{T}}$, and 20 scattering points (SPs), corresponding to small objects in 8 distinct clusters, with each SP capable of generating at most one measurement per time step. Although real-world scenarios may involve various types of landmarks, we consider only SPs in the simulation to validate the proposed framework. It is also important to note that the ground truth locations of the SPs do not follow the PPP distribution assumed by the algorithm. The UE functions as the sensor, the BS is an a priori known

[6] This implies that we assume that if $\boldsymbol{m}^{t,i}_{\text{fir}}$ are the same, $\mathcal{C}^{t,i}$ are from the same source. It is possible that two cells in two different $\tilde{\mathcal{A}}$ with different $\boldsymbol{m}^{t,i}_{\text{fir}}$ could be still from the same source, where all the measurements assigned to a landmark are the same, expect the first one. Although these two cells are viewed as from different landmarks, they can be merged in the end, as they are close to each other (see later).

landmark for the UE and SPs are a priori unknown landmarks for the UE. The state of the single UE $\boldsymbol{s}_{k-1}$ comprises the 3D position $\boldsymbol{x}_{\mathrm{UE},k-1} = [x_{k-1}, y_{k-1}, z_{k-1}]^{\mathsf{T}}$, the heading ϖ_{k-1}, and clock bias B_{k-1}. The UE does a counterclockwise constant turn-rate movement around the BS on the ground, with $\boldsymbol{v}(\boldsymbol{s}_{k-1})$ in (1) defined as same as the transition function in [31, eq. (63)], i.e.,

$$\boldsymbol{v}(\boldsymbol{s}_{k-1}) = \boldsymbol{s}_{k-1} + \begin{bmatrix} \frac{\zeta}{\varrho}(\sin(\varpi_{k-1} + \varrho\Delta) - \sin\varpi_{k-1}) \\ \frac{\zeta}{\varrho}(-\cos(\varpi_{k-1} + \varrho\Delta) + \cos\varpi_{k-1}) \\ 0 \\ \varrho\Delta \\ 0 \end{bmatrix}, \tag{42}$$

where ζ is the translation speed, set as $22.22\,\mathrm{m/s}$, ϱ is the turn rate, set as $\pi/10\,\mathrm{rad/s}$, and Δ is the sampling time interval, set as $0.5\,\mathrm{s}$. As the UE moves on the ground, we assume that its position along the z-axis remains fixed and is known. The covariance of the process noise is assumed to be the same for all time steps, denoted as $\boldsymbol{Q}$. The UE has a concentrated prior regarding its initial position, but possesses no prior knowledge of the map, except for the BS location and the PPP intensity $\lambda(\boldsymbol{x}) = 1.5 \times 10^{-5} U_{\mathrm{ENV}}$ for the SPs, with U_{ENV} denoting a uniform distribution in the environment. It represents a homogeneous PPP intensity, as the UE has not yet explored the area and has no prior knowledge of the SPs locations at the beginning. We assume that $p_{\mathrm{D}} = 0.9$ (same as in [31]), where the FOV with respect to the BS is unlimited while for the SPs, it is limited to 50 m around the UE. All measurements from the BS and SPs are used, with measurements from the known BS and from SPs following different measurement functions $\boldsymbol{h}(\boldsymbol{x}^i, \boldsymbol{s}_k)$, which serves as a general representation, both detailed in [55, Section 2.2]. The measurement functions return an output of the form

$$\boldsymbol{h}(\boldsymbol{s}_k, \boldsymbol{x}^i) = [\tau_k^i, (\boldsymbol{\theta}_k^i)^{\mathsf{T}}, (\boldsymbol{\phi}_k^i)^{\mathsf{T}}]^{\mathsf{T}}, \tag{43}$$

where the geometric information is translated into a 5D measurements containing a time of arrival (TOA) τ_k^i, an angles of arrival (AOA) $\boldsymbol{\theta}_k^i$ and an angles of departure (AOD) $\boldsymbol{\phi}_k^i$. Both AOA and an AOD have azimuth and elevation components, for example, $\boldsymbol{\theta}_k^i = [\theta_{\mathrm{az},k}^i, \theta_{\mathrm{el},k}^i]^{\mathsf{T}}$. The measurement covariance matrix is fixed to $\boldsymbol{R} = \mathrm{diag}[0.1^2\,\mathrm{m}^2, 0.01^2\,\mathrm{rad}^2, 0.01^2\,\mathrm{rad}^2, 0.01^2\,\mathrm{rad}^2, 0.01^2\,\mathrm{rad}^2]$ (same as [25]). The clutter measurement intensity is given by $c(\boldsymbol{z}) = \Upsilon U_{\mathrm{FOV}}$, with U_{FOV} representing a uniform distribution inside the FOV and Υ representing the expected number of clutter measurements per time step. The simulation parameters are primarily selected based on existing literature to ensure consistency for comparison. However, in practical radio SLAM systems, these parameters are influenced by various factors, such as noise levels, transmitter power, and waveform design.

1) Scenarios: Four different scenarios are considered. *Scenario I:* low clutter and low process noise case; *Scenario II:* high clutter and low process noise case, *Scenario III:* low clutter and high process noise case; *Scenario IV:* high clutter and high process noise case. Here, the low clutter and the high clutter cases stand for cases with $\Upsilon = 1$ (same as in [31]) and $\Upsilon = 5$ (same as in [36]), respectively, and the low process noise and high process noise cases stand for cases with $\boldsymbol{Q} = \mathrm{diag}([0.2^2\,\mathrm{m}^2, 0.2^2\,\mathrm{m}^2, 0\,\mathrm{m}^2, 0.001^2\,\mathrm{rad}^2, 0.2^2\,\mathrm{m}^2])$ (same as in [31]) and $\boldsymbol{Q} = \mathrm{diag}([0.2^2\,\mathrm{m}^2, 0.2^2\,\mathrm{m}^2, 0\,\mathrm{m}^2, 0.001^2\,\mathrm{rad}^2, 0.2^2\,\mathrm{m}^2] \times 8)$, respectively.

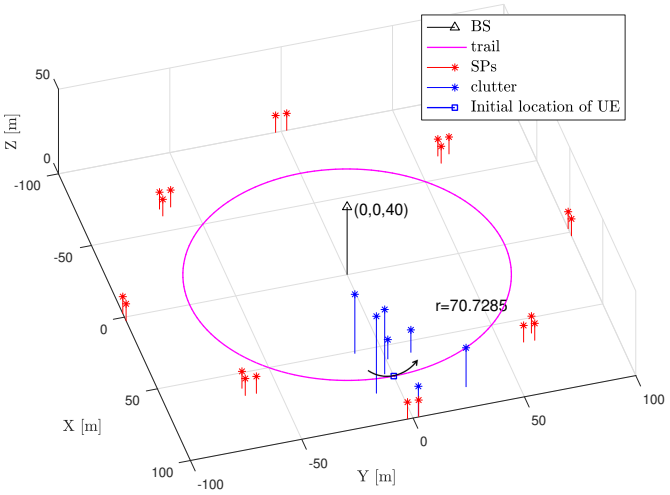


Fig. 3. Scenario with the environment of a BS, 20 SPs and 7 clutters. The UE moves counterclockwise along the trail centered at the BS.

2) Baselines: First, we assess the performance of the proposed sampling algorithm, comparing it to the Gibbs sampling algorithm and the MH algorithm, in Scenario IV with $\Gamma = 100$, which is the most challenging scenario among the four scenarios. Next, we assess the performance of the proposed Graph PMBM-SLAM algorithm with $\Gamma = 100$ by conducting a comparative analysis with respect to three baselines: the EK-PMB SLAM filter [31]; the RBP-PHD SLAM filter without optimal importance sampling [24] using 1000 samples; the RBP-PHD SLAM filter with optimal importance sampling [35] using 1000 samples.

3) Performance Metrics: The accuracy of DA accuracy is assessed using the average of normalized mutual informations (NMIs) [56] between each resulting DA and the ground-truth DA, where the NMI being 1 meaning the resulting DA and the ground-truth DA contain the same information, i.e, same to each other, and the closer NMI is to 1, the more accurate resulting DA is. The sensor state estimations are evaluated by the root mean squared error (RMSE) for the UE states over time. The mapping performance is quantified using the generalized optimal subpattern assignment (GOSPA) distance [57], where the cut-off distance is set to 5, and the exponent factor is set to 2. In total, we undertake 100 Monte Carlo (MC) simulations for all algorithms, and the final results are obtained by averaging over the independent MC simulations.

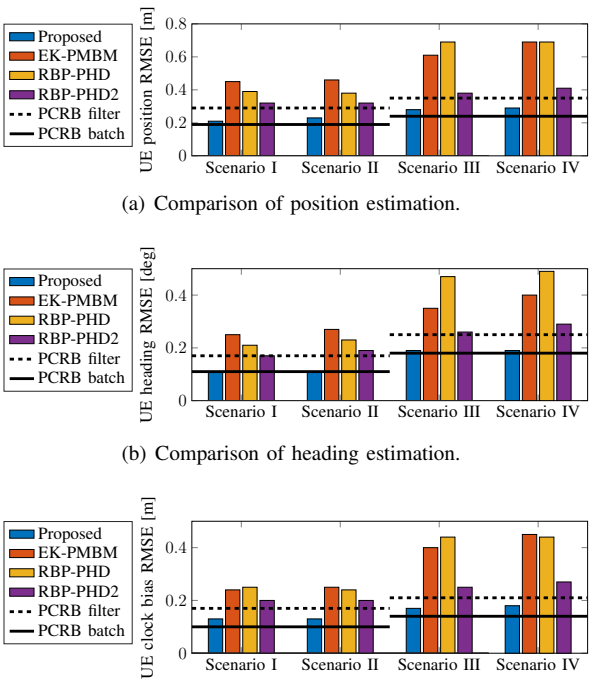
B. Results and Discussion

1) DA Accuracy: We initialize the sample with each measurement forming an individual cell. We measured the proposed sampling algorithm has better performance in accuracy than the

Gibbs sampling and the MH algorithms, which results in the NMI at 0.9971, compared to 0.9679 for Gibbs sampling and 0.9804 for the MH algorithm for the Scenario IV. The Gibbs sampling algorithm moves at most two indices at a time, which can cause measurements to oscillate between sub-cells and fail to transfer groups of measurements between cells, especially when each measurement starts as an individual cell, leading to poor DA results. Similarly, the MH algorithm performs poorly with this initialization, since it requires merging several cells to achieve correct DA, but may pass through intermediate DAs with lower likelihoods before forming larger cells. The proposed algorithm combines the Gibbs sampling and the MH algorithms, effectively handling groups of measurements and forming larger cells before using the MH algorithm. The inaccurate DAs result in poor SLAM results, for example, the resulting GOSPA distances are 10.37 m and 7.36 m if the proposed SLAM framework uses only the Gibbs sampling algorithm or the MH algorithm, respectively, compared to 1.55 m when the proposed sampling algorithm is used. While this proposed method outperforms the individual algorithms, it still does not perfectly solve the DA problem, as evidenced by its NMI being below 1. The primary reasons for this shortfall are the presence of cluttered scenarios and the misclassification of low-quality measurements as clutters.

2) Localization Performance: Next, the performance of the proposed framework in sensor state estimation is evaluated. Fig. 4 shows the RMSEs of the estimated sensor trajectories for four SLAM algorithms across four scenarios, compared to theoretical bounds. We observe that the proposed algorithm's bounds are approximately 30% lower than those of the filter-based algorithms. This difference arises because the proposed algorithm focus on the posterior $f(\boldsymbol{s}_{0:K}, \mathcal{X}|\mathcal{Z}_{1:K})$, which incorporates all measurements. In contrast, filter-based algorithms work on $f(\boldsymbol{s}_k, \mathcal{X}|\mathcal{Z}_{1:k})$ for $k \in \{1, \cdots, K\}$, conditioned only on measurements up to the current time step, resulting in higher bounds. For all algorithms, the bounds are higher in high-process noise scenarios (Scenarios III and IV) compared to low-process noise scenarios (Scenarios I and II). This is due to the PCRB considering the transition density; lower process noise, indicating a more accurate motion model, brings more posterior information and results in lower bounds. Therefore, all algorithms perform better in low process noise scenarios. Furthermore, all algorithms exhibit slightly worse performance in high clutter scenarios (Scenarios II and IV) compared to low clutter scenarios (Scenarios I and III), as the bars are higher. This decline is attributed to the presence of closely spaced clutter measurements in high clutter scenarios, which leads to false alarms and negatively impacts overall performance.

Among the four algorithms, the proposed algorithm demonstrates superior performance due to its batch processing approach, as evidenced by the blue bars being the lowest in Fig. 4. Additionally, the proposed algorithm is robust to both high clutter and high process noise, maintaining close-bound performance in all scenarios, which is indicated by the blue bars being very close to solid black lines in



(a) Comparison of position estimation.

(b) Comparison of heading estimation.

(c) Comparison of clock bias estimation.

Fig. 4. Comparison of sensor trajectory estimation for 4 algorithms under 4 scenarios.

Fig. 4. The proposed algorithm is also robust to low detection probabilities, as it maintains close-bound performance across all scenarios even with a reduced p_{D}, as summarized in Table. I. The robustness is due to the effective solution to the DA for the entire measurement batch and the joint optimization conditioned on the resulting DAs, allowing the algorithm to track all cross-correlations between the sensor trajectory and the map. Among filter-based algorithms, the EK-PMBM SLAM filter, which drops cross-correlations in computation, suffers from information loss, while the RBP-PHD and RBP-PHD2 SLAM filters retain cross-correlations through particles, requiring a sufficient number of particles for good performance. Consequently, the EK-PMBM SLAM filter performs the worst among the algorithms, when a sufficient number of particles are used for the two RBP-based algorithms, as the red bars are the highest in low process noise scenarios. However, 1000 particles are insufficient for the RBP-PHD SLAM filter in high process noise scenarios, leading to worse positioning performance for the RBP-PHD SLAM filter compared to the EK-PMBM SLAM filter, as reflected by the yellow bars being generally highest in high process noise scenarios. The RBP-PHD2 SLAM filter has close-bound performance as 1000 particles are sufficient, but it still underperforms to the proposed algorithm, due to its inherently higher bounds as a filter-based algorithm.

Fig. 5 demonstrates that the proposed algorithm consistently outperforms filter-based algorithms in Scenario IV, as the blue line consistently lies below the red, yellow, and purple lines, highlighting the efficacy of the proposed algorithm. Moreover, the proposed algorithm's bound remains stable throughout the trajectory, in contrast to the decreasing trend observed for filter-based bounds, as the solid black line remains stable,

TABLE I
THE PERFORMANCE OF THE PROPOSED ALGORITHM IN POSITION ESTIMATION ACROSS 4 SCENARIOS WITH $p_D = 0.7$.

	Scenario I	Scenario II	Scenario III	Scenario IV
Proposed	0.22 m	0.23 m	0.31 m	0.33 m
PCRB batch	0.21 m	0.21 m	0.29 m	0.29 m

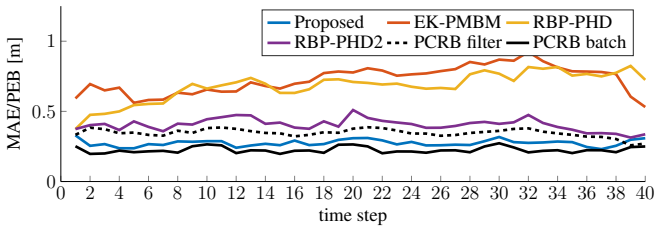


Fig. 5. Comparison of RMSE on sensor position estimates changing with time among four algorithms and two bounds for Scenario IV.

while the solid dashed line decreases in general. This stability arises from the batch-processing bounds conditioning on all measurements, unlike filter-based bound, which are conditioned only on measurements up to the current time step. As time progresses, more measurements can be incorporated, leading to improved performance.

3) Mapping Performance: Fig. 6 shows the RMSE of estimated landmark locations for four SLAM algorithms across different scenarios, compared to their respective bounds. We observe that the bounds for batch processing are lower than the bounds for filter-based algorithms, indicated by the solid black lines being lower than the dashed black lines in Fig. 6. This is because the batch processing incorporates the entire sensor trajectory into the posterior information matrix (PIM), compared to filter-based algorithms that only have snapshots of the sensor state in the bound computation. Consequently, batch processing yields lower bounds than filter-based methods even when all measurements are conditioned. In low process noise scenarios, all bounds are lower due to the more accurate transition model, which also benefits landmark state estimation.

The proposed method is robust to both high clutter and high process noise, and performs the best in landmark state estimation, as evidenced by the blue bars being close to solid black lines and lowest among four algorithms in all four scenarios. The superior performance and the robustness are attributed to the batch processing of the proposed method. In contrast, the EK-PMBM and RBP-PHD SLAM filters perform poorly. The EK-PMBM SLAM filter suffers from information loss, and the RBP-PHD SLAM does not utilize sufficient particles. Their performances degrade further in Scenario IV, due to the challenges posed by high clutter and high process noise for filter-based algorithms. The RBP-PHD2 filter, with sufficient particles, is also robust to high clutter and process noise but still underperforms compared to the proposed algorithm, due to its inherently filter-based processing.

Fig. 7 shows the GOSPA distance for the four algorithms across four scenarios. Consistent with previous results, the proposed algorithm exhibits the best performance, providing better landmark estimations with fewer false alarms and misdetections, as the blue bars are the lowest. This superior

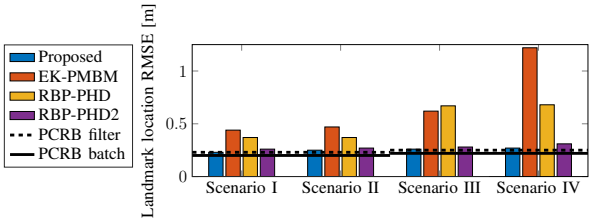


Fig. 6. Comparison of landmark estimations for 4 algorithms under 4 scenarios.

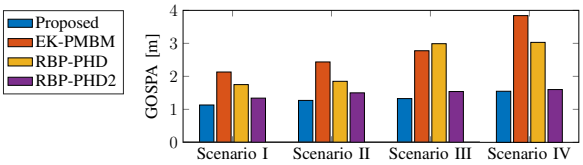


Fig. 7. Comparison of GOSPA distance for 4 algorithms under 4 scenarios.

performance is due to the effectiveness of the proposed DA solution for measurement batch and joint optimization in the SLAM results. Among the filter-based algorithms, the RBP-PHD2 filter performs the best due to its use of sufficient samples, where DA problem is solved effectively, resulting in fewer false alarms and misdetections. In contrast, the RBP-PHD and EK-PMBM SLAM filters perform poorly due to insufficient particles or information loss from marginalization, leading to more false alarms and misdetections compared to the RBP-PHD2 SLAM filter.

We also evaluate the proposed algorithm using a real-world radio SLAM dataset provided in [58]. Fig. 8 shows a comparison of our proposed algorithm with the RBF-PHD2 SLAM filter and the robust snapshot SLAM algorithm [59] using experimental data. We observed that all algorithms achieve good SLAM performance. However, the robust snapshot SLAM algorithm performs the worst, due to its snapshot-based nature, which ignores the temporal connections between time steps. Specifically, the RMSEs for position, heading, and clock bias were 0.28 m, 2.26°, and 0.23 m for the RBP-PHD2 SLAM filter, 0.40 m, 2.02°, 0.46 m for the robust snapshot SLAM algorithm, and 0.22 m, 2.19°, and 0.16 m for the proposed algorithm. These results demonstrate that our proposed algorithm performs better than the robust snapshot SLAM algorithm and the RBP-PHD2 SLAM filter on the real data, despite the multi-model implementation of the RBP-PHD2 SLAM filter. Furthermore, the performance of the proposed algorithm is expected to improve further if a multi-model implementation is incorporated.

VIII. CONCLUSIONS

This paper presents a novel Graph PMBM-SLAM algorithm, which bridges the RFS theory and graph-based SLAM together. By modeling the measurements and the landmarks as RFSs, a sampling-based algorithm, which combines the Gibbs sampling algorithm and the MH algorithm together, is proposed to solve the DA problem of all measurements given a sensor trajectory. The GraphSLAM algorithm is applied to estimate the best fit of the sensor trajectory and the map to the joint

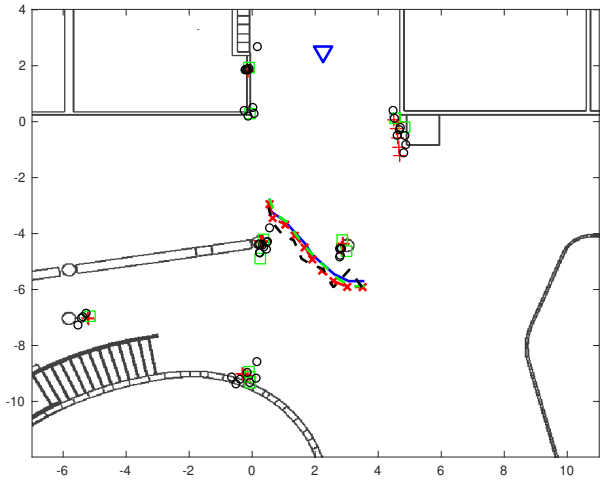


Fig. 8. Evaluation of the proposed algorithm using real-world radio SLAM data [58], collected from an indoor corridor. The blue solid line represents the ground truth of the UE trajectory, while the green dashed line and green squares depict the SLAM results of the proposed algorithm. The red dashed line and red plus signs represent the SLAM results from the RBP-PHD2 SLAM filter. The black dashed line and black circles represent the SLAM results from the robust snapshot SLAM algorithm. Please note that the RBP-PHD2 SLAM filter employs a multi-model implementation for different landmark types, whereas the other two algorithms do not. In the latter, the unknown environment is represented exclusively using multiple incidence points.

posterior of the sensor trajectory and the map conditioned on the resulting DA. The proposed framework iterates within these two steps until reaching a maximal number of iterations. The marginalization step to merge the SLAM resulting from iterations serves as the post-processing step to approximate the correct joint posterior, where the map is modeled as a RFS instead of a list of random vectors. Analysis was carried out in four simulated scenarios through MC simulations. Results demonstrated that the proposed framework can address the DA problem of all measurements accurately. Our results also demonstrated the close-to-bound performance of the proposed framework in mapping and positioning, as well as its accuracy and robustness in high clutter and high process noise scenarios.

Since the main goal of this paper is to demonstrate the feasibility of the proposed framework that integrates RFS theory with GraphSLAM, we focus on the most basic assumptions—point objects and static landmarks. While this paper primarily considers the point object assumption in a radio SLAM scenario, it is possible to extend the approach to include extended objects, incorporate a multi-model implementation, as well as apply the proposed algorithm to other datasets. Although the proposed algorithm currently operates offline and is limited to static landmarks, its practicality could be enhanced by incorporating techniques from conventional graph-based SLAM, enabling real-time operation [15] and adaptation to dynamic environments with moving objects [60]. Our future work will explore the extension to extended object models, evaluate the algorithm with additional experimental datasets, such as visual and LiDAR data, and compare its performance against other state-of-the-art SLAM algorithms using additional real-world datasets and widely adopted simulation platforms, such as the g2o software [61]. It will also be interesting to extend the proposed algorithm with a multi-model implementation that accounts for various types of landmarks, and investigate scenarios involving moving objects in the map, rather than only static landmarks.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[3] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*. Springer, 1990, pp. 167–193.

[4] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[5] X. Wang, S. K. Ong, and A. Y. Nee, "A comprehensive survey of augmented reality assembly research," *Advances in Manufacturing*, vol. 4, pp. 1–22, 2016.

[6] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.

[7] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, *et al.*, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2016.

[8] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, *et al.*, "Location-aware communications for 5G networks: How location information can improve scalability, latency, and robustness of 5G," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 102–112, 2014.

[9] Y. Ge, O. Kaltiokallio, H. Kim, J. Talvitie, *et al.*, "Mmwave mapping and SLAM for 5G and beyond," in *Integrated Sensing and Communications*. Springer, 2023, pp. 445–475.

[10] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, *et al.*, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Eighteenth National Conference on Artificial Intelligence*. USA: American Association for Artificial Intelligence, 2002, p. 593–598.

[12] J. Neira and J. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.

[13] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[14] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[15] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[16] A. Falchetti Pareja, *Random Finite Sets in Visual SLAM*. Universidad de Chile, 2017.

[17] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.

[18] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, *et al.*, "Robust map optimization using dynamic covariance scaling," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 62–69.

[19] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph SLAM," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 1254–1261.

[20] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[21] K. MacTavish and T. D. Barfoot, "At all costs: A comparison of robust cost functions for camera correspondence outliers," in *12th Conference on Computer and Robot Vision*, 2015, pp. 62–69.

[22] R. P. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House, 2014.

[23] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to Bayesian SLAM," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 268–282, 2011.

[24] H. Kim, K. Granström, L. Gao, G. Battistelli, *et al.*, "5G mmWave cooperative positioning and mapping using multi-model PHD filter and map fusion," *IEEE Transactions on Wireless Communications*, 2020.

[25] O. Kaltiokallio, Y. Ge, J. Talvitie, H. Wymeersch, *et al.*, "mmWave simultaneous localization and mapping using a computationally efficient EK-PHD filter," in *IEEE International Conference on Information Fusion (Fusion)*, 2021, pp. 1–6.

[26] H. Deusch, S. Reuter, and K. Dietmayer, "The labeled multi-Bernoulli SLAM filter," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1561–1565, 2015.

[27] H. Deusch, "Random finite set-based localization and SLAM for highly automated vehicles," Ph.D. dissertation, Universität Ulm, 2016.

[28] D. Moratuwage, M. Adams, and F. Inostroza, "δ-generalised labelled multi-Bernoulli simultaneous localisation and mapping," in *International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2018, pp. 175–182.

[29] D. Moratuwage, M. Adams, and F. Inostroza, "δ-generalized labeled multi-Bernoulli simultaneous localization and mapping with an optimal kernel-based particle filtering approach," *Sensors*, vol. 19, no. 10, p. 2290, 2019.

[30] Y. Ge, F. Wen, H. Kim, M. Zhu, *et al.*, "5G SLAM using the clustering and assignment approach with diffuse multipath," *Sensors (Basel, Switzerland)*, vol. 20, no. 16, August 2020.

[31] Y. Ge, O. Kaltiokallio, H. Kim, F. Jiang, *et al.*, "A computationally efficient EK-PMBM filter for bistatic mmwave radio SLAM," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 7, pp. 2179–2192, 2022.

[32] Y. Ge, Y. Wu, F. Jiang, O. Kaltiokallio, *et al.*, "Iterated posterior linearization PMB filter for 5G SLAM," in *IEEE International Conference on Communications*, 2022, pp. 877–882.

[33] H. Kim, A. F. García-Fernández, Y. Ge, Y. Xia, *et al.*, "Set-type belief propagation with applications to Poisson multi-Bernoulli SLAM," *IEEE Transactions on Signal Processing*, vol. 72, pp. 1989–2005, 2024.

[34] Y. Ge, H. Kim, F. Wen, L. Svensson, *et al.*, "Exploiting diffuse multipath in 5G SLAM," *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2020.

[35] O. Kaltiokallio, R. Hostettler, J. Talvitie, Y. Ge, *et al.*, "Towards real-time radio-SLAM via optimal importance sampling," in *IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022, pp. 1–5.

[36] O. Kaltiokallio, R. Hostettler, Y. Ge, H. Kim, *et al.*, "A multi-hypotheses importance density for SLAM in cluttered scenarios," *IEEE Transactions on Robotics*, vol. 40, pp. 1019–1035, 2023.

[37] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, *et al.*, "A belief propagation algorithm for multipath-based SLAM," *IEEE Transactions on Wireless Communications*, vol. 18, no. 12, pp. 5613–5629, 2019.

[38] F. Inostroza and M. Adams, "Combining the SLAM back and front ends with a joint vector-set distribution," *The International Journal of Robotics Research*, p. 02783649241303770, 2025.

[39] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: Direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1883–1901, 2018.

[40] Y. Xia, Á. F. García-Fernández, and L. Svensson, "Markov chain Monte Carlo multi-scan data association for sets of trajectories," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 6, pp. 7804–7819, 2024.

[41] K. Granström, L. Svensson, S. Reuter, Y. Xia, *et al.*, "Likelihood-based data association for extended object tracking using sampling methods," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 30–45, 2018.

[42] M. Fatemi, K. Granström, L. Svensson, F. J. Ruiz, *et al.*, "Poisson multi-Bernoulli mapping using Gibbs sampling," *IEEE Transactions on Signal Processing*, vol. 65, no. 11, pp. 2814–2827, 2017.

[43] G. Casella and E. I. George, "Explaining the Gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[44] Y. Xia, Á. F. García-Fernández, and L. Svensson, "An efficient implementation of the extended object trajectory PMB filter using blocked Gibbs sampling," in *International Conference on Information Fusion (FUSION)*. IEEE, 2023, pp. 1–8.

[45] S. Jain and R. M. Neal, "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model," *Journal of Computational and Graphical Statistics*, vol. 13, no. 1, pp. 158–182, 2004.

[46] C. P. Robert, G. Casella, and G. Casella, *Monte Carlo Statistical Methods*. Springer, 1999, vol. 2.

[47] J. S. Liu, "The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 958–966, 1994.

[48] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[49] J. L. Williams, "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA, and association-based MeMBer," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 1664–1687, 2015.

[50] C. M. Bishop, "Pattern recognition and machine learning," *Springer*, vol. 2, pp. 1122–1128, 2006.

[51] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[52] H. Wymeersch and G. Seco-Granados, "Adaptive detection probability for mmwave 5G SLAM," in *IEEE 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.

[53] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[54] J. L. Williams, "An efficient, variational approximation of the best fitting multi-Bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 258–273, 2014.

[55] Y. Ge, "Single base station mmwave radio positioning, mapping, and SLAM," Ph.D. dissertation, Chalmers Tekniska Hogskola (Sweden), 2024.

[56] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?" in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 1073–1080.

[57] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *20th IEEE International Conference on Information Fusion (Fusion)*, 2017.

[58] E. Rastorgueva-Foi, O. Kaltiokallio, Y. Ge, M. Turunen, *et al.*, "Millimeter-wave radio SLAM: End-to-end processing methods and experimental validation," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 9, pp. 2550–2567, 2024.

[59] O. Kaltiokallio, E. Rastorgueva-Foi, J. Talvitie, Y. Ge, *et al.*, "Robust snapshot radio SLAM," *IEEE Transactions on Vehicular Technology*, pp. 1–6, 2024.

[60] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. J. Leonard, "Dynamic pose graph SLAM: Long-term mapping in low dynamic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1871–1878.

[61] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, *et al.*, "G2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[62] H. L. Van Trees, *Detection, Estimation, and Dodulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory*. John Wiley & Sons, 2004.