



## **A Multi-Hypotheses Importance Density for SLAM in Cluttered Scenarios**

Downloaded from: <https://research.chalmers.se>, 2025-12-04 23:06 UTC








Citation for the original published paper (version of record):

Kaltiokallio, O., Hostettler, R., Ge, Y. et al (2024). A Multi-Hypotheses Importance Density for SLAM in Cluttered Scenarios. IEEE Transactions on Robotics, 40: 1019-1035.  
<http://dx.doi.org/10.1109/TRO.2023.3338975>

N.B. When citing this work, cite the original published paper.

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

# A Multihypotheses Importance Density for SLAM in Cluttered Scenarios

Ossi Kaltiokallio , *Member, IEEE*, Roland Hostettler , *Member, IEEE*, Yu Ge , *Graduate Student Member, IEEE*, Hyowon Kim , *Member, IEEE*, Jukka Talvitie , *Member, IEEE*, Henk Wymeersch , *Fellow, IEEE*, and Mikko Valkama , *Fellow, IEEE*

**Abstract**—One of the most fundamental problems in simultaneous localization and mapping (SLAM) is the ability to take into account data association (DA) uncertainties. In this article, this problem is addressed by proposing a multihypotheses sampling distribution for particle filtering-based SLAM algorithms. By modeling the measurements and landmarks as random finite sets, an importance density approximation that incorporates DA uncertainties is derived. Then, a tractable Gaussian mixture model approximation of the multihypotheses importance density is proposed, in which each mixture component represents a different DA. Finally, an iterative method for approximating the mixture components of the sampling distribution is utilized and a partitioned update strategy is developed. Using synthetic and experimental data, it is demonstrated that the proposed importance density improves the accuracy and robustness of landmark-based SLAM in cluttered scenarios over state-of-the-art methods. At the same time, the partitioned update strategy makes it possible to include multiple DA hypotheses in the importance density approximation, leading to a favorable linear complexity scaling, in terms of the number of landmarks in the field-of-view.

**Index Terms**—Importance density, particle filter (PF), probability hypotheses density, random finite set (RFS), simultaneous localization and mapping (SLAM).

## NOMENCLATURE

### Major Notations of This Article

$k$	Current time index.
$\mathbf{x}_k$	Pose of the robot at time $k$ .

Manuscript received 9 March 2023; revised 8 August 2023; accepted 31 October 2023. Date of publication 4 December 2023; date of current version 8 January 2024. This paper was recommended for publication by Associate Editor M. Walter and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by the Academy of Finland under Grant #328214, Grant #323244, Grant #319994, Grant #338224, Grant #345654, and Grant #352754, in part by the Finnish Funding Agency for Innovation under the 6G-ISAC project, and in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP) through Knut and Alice Wallenberg Foundation, and the Chalmers Area of Advance Transport 6G-Cities project. (Corresponding author: Ossi Kaltiokallio.)

Ossi Kaltiokallio, Jukka Talvitie, and Mikko Valkama are with the Unit of Electrical Engineering, Tampere University, 33100 Tampere, Finland (e-mail: ossi.kaltiokallio@tuni.fi; jukka.talvitie@tuni.fi; mikko.valkama@tuni.fi).

Roland Hostettler is with the Department of Electrical Engineering, Uppsala University, 752 36 Uppsala, Sweden (e-mail: roland.hostettler@angstrom.uu.se).

Yu Ge and Henk Wymeersch are with the Department of Electrical Engineering, Chalmers University of Technology, 412 58 Gothenburg, Sweden (e-mail: yuge@chalmers.se; henkw@chalmers.se).

Hyowon Kim is with the Department of Electronics Engineering, Chungnam National University, Daejeon 34134, South Korea (e-mail: hyowon.kim@cnu.ac.kr).

Digital Object Identifier 10.1109/TRO.2023.3338975

$\mathbf{u}_k$	Control input of the robot at time $k$ .
$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$	Transition function of the robot.
$\mathbf{Q}_{k-1}$	Process noise covariance at time $k - 1$ .
$\mathcal{M}_k$	Map at time $k$ .
$M_k$	Number of landmarks in $\mathcal{M}_k$ .
$\mathbf{m}^i$	$i$ th landmark in $\mathcal{M}_k$ .
$\mathcal{Z}_k$	Set of measurements at time $k$ .
$J_k$	Number of measurements in $\mathcal{Z}_k$ .
$\mathbf{z}^j$	$j$ th measurement in $\mathcal{Z}_k$ .
$g(\mathcal{Z}_k   \mathcal{M}_k, \mathbf{x}_k)$	random finite set (RFS) likelihood.
$g(\mathbf{z}_k   \mathbf{m}, \mathbf{x}_k)$	Measurement likelihood.
$\mathbf{R}_k$	Measurement noise covariance.
$\lambda_c$	Poisson rate.
$c(\mathbf{z})$	Clutter intensity.
$p_D(\mathbf{m}^i   \mathbf{x}_k)$	Detection probability.
$\phi_k^{t,i}$	Association variable.
$\mathbf{x}_k^n$	$n$ th particle of the probability hypothesis density (PHD)-simultaneous localization and mapping (SLAM) density.
$w_k^n$	Weight of $n$ th particle of the PHD-SLAM density.
$N$	Number of particles.
$v_k(\mathbf{m}   \mathbf{x}_{0:k}^n)$	PHD conditioned on the $n$ th trajectory.
$\eta_k^{n,i}$	Weight of the $i$ th component of the Gaussian mixture (GM)-PHD.
$\hat{\mathbf{m}}_k^{n,i}$	Mean of the $i$ th component of the GM-PHD.
$\mathbf{P}_k^{n,i}$	Covariance of the $i$ th component of the GM-PHD.
$\Gamma$	Number of Gaussian mixture importance density (GM-ID) components.
$\tilde{\gamma}^t$	Weight of the $t$ th GM-ID component.
$\boldsymbol{\mu}^t$	Mean of the $t$ th GM-ID component.
$\boldsymbol{\Sigma}^t$	Covariance of the $t$ th GM-ID component.

## I. INTRODUCTION

**F**OLLOWING pioneering research in autonomous robotics [1], the simultaneous localization and mapping (SLAM) problem has gained widespread interest over the past decades, with numerous applications ranging from mobile robotics [2] to visual odometry [3]. In recursive probabilistic form, the SLAM problem requires a robot to incrementally build a map of the unknown environment, simultaneously localize itself within the map, and estimate the related uncertainties [4]. There are

numerous representations for the robot pose and map, including landmark-based [1], [4], grid-based [5], [6], and graph-based [7], [8] approaches. Landmark-based approaches decompose the physical environmental landmarks, such as trees and other obstacles, into parametric representations, such as a point, which then form a map with an unknown number of landmarks at unknown locations.

Solving the probabilistic SLAM problem requires propagating the joint posterior density of the robot trajectory and map over time [4]. Furthermore, building the map requires the joint estimation of both the number and location of landmarks that have been covered by the sensor's field-of-view (FOV). Conventional SLAM solutions comprise of a three-step approach in which the measurement to landmark association is solved first. Then, given the found data association (DA), the joint posterior density is estimated using Bayesian filtering. Lastly, a separate map management routine is required to create landmarks that enter the sensor's FOV and delete landmarks that originate from false detections. Numerous works have demonstrated that such an approach works well in practice [9], [10], [11], but is sensitive to DA uncertainty [12]. To account for data ambiguities in a theoretically sound manner requires that DA uncertainty is treated as a part of the estimation process. This can be done by modeling the map as a finite set and using RFS theory for propagating the joint posterior distribution in time [13]. RFS-based methods are particularly attractive since they enable a fully integrated Bayesian framework for SLAM since DA uncertainty is factored in to the estimation process, uncertainties on both the number of landmarks in the map and their state are taken into account, there is no ordering of the landmarks, and the map management routine is integrated in to the filtering recursion. The RFS formulation for SLAM was first proposed in [14] and a tractable first-order approximation coined as PHD-SLAM filter soon followed [13]. Over the years, various approximations have been used to model the landmark map resulting in various RFS-SLAM filters including the labeled multi-Bernoulli (LMB)-SLAM filter [15],  $\delta$ -generalized LMB-SLAM filter [16], and the Poisson multi-Bernoulli mixture (PMBM)-SLAM filter [17], [18].

In terms of implementation and representation, many SLAM algorithms take advantage of an important characteristic of the SLAM problem: by conditioning the map to the robot's trajectory, the landmarks are conditionally independent [19], making it natural to apply Rao-Blackwellized particle filter (RBPF) solutions. Such a factored solution was adopted in FastSLAM, which uses a particle filter (PF) to sample over robot paths and each particle possesses numerous low-dimensional extended Kalman filters (EKF), one for each of the landmarks [11]. The filtering recursion of PHD-SLAM is similar to that in FastSLAM, but the use of RFS likelihoods affects how the particle weights are computed and in addition, PHD filters are used for mapping [13]. The standard PHD-SLAM algorithm [13], similar to that of FastSLAM 1.0 [11], is a bootstrap filter in which the dynamic model is used as the importance density. If the robot motion is affected by large disturbances, a large number of particles are required to adequately explore the sample space, which can be computationally expensive.

In this work, we propose an improved importance density in which poses are sampled under consideration of both the motion of the robot and the measurements, as well as DA uncertainty. To account for data ambiguities, a multihypotheses importance density (MH-ID) approximation is derived and a Gaussian mixture model (GMM) representation of the multihypotheses importance density (MH-ID) is proposed, in which each component of the GM represents a single DA. The proposed GM-ID enables us to incorporate DA uncertainty within the importance density approximation, rather than assume that the most likely DA is correct as in [20]. Then, for each GM component, we exploit an iterative method for approximating parameters of the importance density, which is based on using generalized statistical linear regression (SLR), combined with iterated posterior linearization (IPL) [21], [22]. The proposed solution exploits the measurement model structure and uses partitioned updates, that is, the importance density is updated one measurement at a time. The partitioned algorithm scales linearly with the number of landmarks  $M$  within the FOV, instead of  $\mathcal{O}(M^3)$  when using a joint approximation approach as proposed in [20]. Using synthetic and experimental data, it is demonstrated that the proposed GM-ID improves the accuracy and robustness of PHD-SLAM, while the partitioned update strategy makes it possible to include multiple DA hypotheses in the importance density approximation with low computational complexity.

The contributions of this article are summarized below.

- 1) *A novel GM-ID for probabilistic SLAM:* A GM proposal distribution is developed for PHD-SLAM. In the proposed GM-ID, individual mixture components represent different DAs to account for data ambiguities in high clutter scenarios. The developed GM-ID provides a new method for taking into account DA uncertainty, which allows developing more robust and efficient versions of existing PHD-SLAM filters.
- 2) *Low-complexity computation of a GM-ID approximation:* An iterative method for approximating the mixture components of the GM-ID is utilized and a partitioned update strategy is developed. The partitioned importance density algorithm is computationally feasible since it scales according to  $\mathcal{O}(M)$ .
- 3) *Comparison against state of the art:* We validate the development efforts using synthetic and experimental data and compare the proposed algorithm to three other PHD-SLAM filters. The results indicate that the proposed GM-ID improves PHD-SLAM performance allowing accurate, efficient, and robust SLAM even in high clutter scenarios. We provide MATLAB code that runs PHD-SLAM using the presented GM-ID approximation. In addition, we provide C/C++ source codes that can be compiled to MATLAB MEX-files to enable a highly efficient implementation of the developed algorithm.<sup>1</sup>

The rest of this article is organized as follows. In Section II, the related work is presented. In Section III, the probabilistic

<sup>1</sup>[Online]. Available: <https://github.com/okaltio/PHD-SLAM-3.0>

SLAM problem is formulated, the underlying models are introduced, and the Bayesian filtering recursion of RFS-SLAM is presented. Section IV summarizes a first-order solution to RFS-SLAM, which utilizes an Rao-Blackwellized particle filter (RBPF) for propagating the robot posterior and a GM-PHD filter for estimating the map. The proposed GM-ID is introduced in Section V, with Section VI presenting and discussing its performance. Finally, Section VII concludes this article. Major notations of this article are given in the Nomenclature.

## II. RELATED WORK

In SLAM, non-Gaussian noise and unknown DA have been the primary drivers for developing multi-hypotheses methods. Particle filtering approaches, such as FastSLAM [11] and RFS-SLAM [13], are able to approximate arbitrary probability distributions through a finite number of samples. In FastSLAM, DA can be determined on a per-particle basis, and hence, different particles can be associated with different landmarks, which gives the filter the possibility of dealing with the multihypotheses DA problem [23]. In contrast, PHD-SLAM avoids explicit DAs, and DA uncertainty is factored in to the estimation process. More recent works have considered multihypotheses methods in the context of GraphSLAM [24], [25], [26]. In [24], a max-mixture Gaussian distribution is proposed that can be used for example in implementing robust cost functions to handle uncertain loop closures. Respectively, the works in [25] and [26] introduce methods for probabilistic DA. The method in [25] considered probabilistic DA making use of expectation–minimization whereas the work in [26] uses a max-mixture Gaussian distribution similar to the work in [24]. In this article, a sum-mixture of Gaussians is used to model the sampling distribution and each component of the GM represents a single DA.

The aforementioned filtering approaches are typically referred to as online SLAM methods since they recursively estimate the joint posterior density as new measurements become available. On the contrary, smoothing approaches utilize the entire data to estimate the full trajectory and a common approach to solve the problem is using GraphSLAM [7], [8], which consists of two tasks. The front-end task constructs the graph using odometry and sensor measurements, and the back-end task solves the optimization problem to determine the best trajectory that satisfies constraints of the graph. The GraphSLAM algorithms typically exploit the sparse structure of the problem to efficiently solve the optimization problem [27], [28], [29]. Throughout this article, we focus on online SLAM methods and therefore will only consider such approaches in the following. The interested reader is referred to [30] for a comparison between PHD-SLAM and GraphSLAM in which it is demonstrated that the RFS formulation can be beneficial in scenarios in which DA is very challenging.

One major weakness of particle filtering-based SLAM solutions is the large number of particles required to propagate the posterior accurately over time. The culprit is utilizing the motion model as the proposal distribution, which results in an inefficient use of particles as most of them will have an insignificant weight after the posterior update. The FastSLAM

2.0 algorithm addressed this issue by using an improved importance density in which poses are sampled under consideration of both the motion of the robot and the measurements [31]. With known DA, the proposal distribution used in FastSLAM 2.0 is an approximation of the optimal importance density (OID), which is optimal in terms of minimizing the variance of the incremental particle weights [32]. Similarly, most PHD-SLAM algorithms suffer from inefficient use of particles since they use the transition density as the proposal distribution [13], [33], [34]. Inspired by the FastSLAM 2.0 algorithm, two recent works have proposed improved importance densities for PHD-SLAM [20], [35]. In [35], the proposal distribution is computed by conditioning the predictive distribution of the robot state to the measurements independently and thereafter, the independent conditional distributions are merged using weighted averaging. On the other hand, the method in [20] approximates the importance density by finding the joint approximation of the robot and map first, and then by conditioning on the measurements of the most likely DA. While both methods improve the performance with respect to the original PHD-SLAM algorithm, they still require quite many particles to function properly. The proposal of Gao et al. [35] was very conservative, and numerous particles are needed to propagate the posterior accurately, whereas the work in [20] is sensitive to incorrect DA and the filter requires many particles to handle data ambiguities. In this work, we propose an improved importance density in which poses are sampled under consideration of both the motion of the robot and the measurements, as well as DA uncertainty.

## III. PROBLEM FORMULATION

Consider a mobile robot exploring an unknown environment. Its location and orientation at time  $k$  are described by state vector,  $\mathbf{x}_k$ , and the movement is governed by motion command  $\mathbf{u}_k$ . The robot is equipped with a sensor and it is taking relative observations of a number of unknown landmarks while moving in the environment. The  $i$ th landmark is described by state vector,  $\mathbf{m}^i$ , and an observation of a landmark is denoted as  $\mathbf{z}_k^j$ . Since the robot is moving in an uncharted environment, the number of landmarks within the sensor's FOV, denoted as  $\text{FOV}(\mathbf{x}_k)$ , is unknown and time-varying. Moreover, DA between the observations and landmarks is unknown, clutter measurements can cause false alarms and landmarks can be misdetected. Formulating the problem using RFS theory enables a fully integrated Bayesian framework for SLAM under DA uncertainty and unknown landmark number [13]. The underlying models and the RFS-SLAM algorithm are summarized in the following.

### A. Models

A Gaussian density with zero-mean additive noise is a common representation of the motion model in SLAM [4]. In addition, the state transition is assumed independent of the landmarks and observations, and it is modeled as a Markov process in which  $\mathbf{x}_k$  only depends on  $\mathbf{x}_{k-1}$  and  $\mathbf{u}_k$ . Mathematically, the transition model of the mobile robot can be expressed as

$$f(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k), \mathbf{Q}_{k-1}) \quad (1)$$



where  $\mathbf{f}(\cdot)$  denotes a known transition function and  $\mathbf{Q}_{k-1}$  the covariance.

Let  $\mathcal{M}_{k-1}$  denote the explored map up to time  $k-1$ , then the RFS transition density of the map is given by

$$f(\mathcal{M}_k | \mathcal{M}_{k-1}, \mathbf{x}_k) \quad (2)$$

and the explored map evolves in time according to [36, eq. (7)]

$$\mathcal{M}_k = \mathcal{M}_{k-1} \cup \mathcal{B}_k(\mathbf{x}_k) \quad (3)$$

in which  $\mathcal{B}_k(\mathbf{x}_k)$  is the birth RFS describing the set of landmarks observed for the first time. Let  $\mathcal{M}_{k-1}^c$  denote the complement of  $\mathcal{M}_{k-1}$  so that the newly detected landmarks can be defined as  $\mathcal{B}_k(\mathbf{x}_k) = \text{FOV}(\mathbf{x}_k) \cap \mathcal{M}_{k-1}^c$ . Essentially, the RFS transition density of the map describes how the explored map grows monotonically as the FOV of the robot's sensor covers more of the unexplored environment [36].

In this article, we consider a point target measurement model, where each landmark can create at most one observation per time instant. Then, the RFS likelihood function is obtained by summing over all hypotheses [37, eq. (7.2)]

$$g(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_k) = \sum_{t=1}^{\Gamma_{\text{all}}} \exp(-\lambda_c) \prod_{j=1}^{|\mathcal{Z}_k|} c(\mathbf{z}_k^j) \times \prod_{i: \phi_k^{t,i}=0} [1 - p_D(\mathbf{m}^i | \mathbf{x}_k)] \prod_{i: \phi_k^{t,i}>0} \frac{p_D(\mathbf{m}^i | \mathbf{x}_k) g(\mathbf{z}_k^{\phi_k^{t,i}} | \mathbf{m}^i, \mathbf{x}_k)}{c(\mathbf{z}_k^{\phi_k^{t,i}})}. \quad (4)$$

In (4),  $\Gamma_{\text{all}}$  is the total number of hypotheses,  $\phi_k^{t,i}$  is an association variable,  $\lambda_c = \int c(\mathbf{z}) d\mathbf{z}$  is the Poisson rate,  $c(\mathbf{z})$  is the clutter intensity,  $p_D(\mathbf{m}^i | \mathbf{x}_k) \in [0, 1]$  is the detection probability, and  $g(\cdot)$  is the likelihood function. In SLAM literature, the likelihood function is typically modeled as a zero-mean Gaussian, given by [4]

$$g(\mathbf{z}_k | \mathbf{m}, \mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; \mathbf{g}(\mathbf{m}, \mathbf{x}_k), \mathbf{R}_k) \quad (5)$$

where  $\mathbf{g}(\mathbf{m}, \mathbf{x}_k)$  is a known measurement model and  $\mathbf{R}_k$  denotes the covariance. The association variable is defined as

$$\phi_k^{t,i} = \begin{cases} j & \text{if } \mathbf{m}^i \text{ is associated with } \mathbf{z}_k^j \\ 0 & \text{if } \mathbf{m}^i \text{ is undetected.} \end{cases} \quad (6)$$

## B. RFS-SLAM

The objective of probabilistic SLAM is to estimate the joint posterior density of the map and robot trajectory [4]

$$p(\mathcal{M}_k, \mathbf{x}_{1:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) \quad (7)$$

given the initial pose of the robot,  $\mathbf{x}_0$ , measurements, and controls up to the current time instant  $k$ . The RFS-SLAM filter follows the prediction and update steps of the Bayesian filtering recursion applied with RFSs [13]. The prediction step of the filter is given by [13, eq. (7)]

$$p(\mathcal{M}_k, \mathbf{x}_{1:k} | \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}, \mathbf{x}_0) = f(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \times \int f(\mathcal{M}_k | \mathcal{M}_{k-1}, \mathbf{x}_k) p(\mathcal{M}_{k-1}, \mathbf{x}_{1:k-1}) \delta \mathcal{M}_{k-1} \quad (8)$$

where  $\delta \mathcal{M}_{k-1}$  denotes a set integral [38] and  $p(\mathcal{M}_{k-1}, \mathbf{x}_{1:k-1})$  is the shorthand notation for the posterior at the previous time step. Once observing  $\mathcal{Z}_k$ , the predicted density can be updated using the Bayes' rule to obtain the posterior at time  $k$  [13, eq. (8)]

$$p(\mathcal{M}_k, \mathbf{x}_{1:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) = \frac{g(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_k) p(\mathcal{M}_k, \mathbf{x}_{1:k} | \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}, \mathbf{x}_0)}{g(\mathcal{Z}_k | \mathcal{Z}_{0:k-1}, \mathbf{x}_0)} \quad (9)$$

where the RFS likelihood is given in (4), and the term in the denominator is a normalization constant.

## IV. RAO-BLACKWELLIZED PHD-SLAM FILTER

An RBPF implementation of the PHD-SLAM filter is summarized in this section. Fundamentally, the robot trajectory is estimated with a PF, and a PHD filter is used for estimating each trajectory conditioned map.

### A. Factorized RFS-SLAM Filter

Analogous to FastSLAM [11], the joint posterior RFS-SLAM density in (7) can be factorized as [13]

$$p(\mathcal{M}_k, \mathbf{x}_{1:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) = p(\mathbf{x}_{1:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) p(\mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}). \quad (10)$$

The recursion for the joint RFS-SLAM density presented in Section III-B is equivalent to jointly propagating the posterior density of the robot trajectory,  $p(\mathbf{x}_{1:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$ , and the posterior density of the map that is conditioned on the trajectory,  $p(\mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k})$ . The posterior of the robot trajectory is computed as [13, eq. (16)]

$$p(\mathbf{x}_{1:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) = p(\mathbf{x}_{1:k-1} | \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k-1}, \mathbf{x}_0) \times \frac{g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) f(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)}{g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1})}. \quad (11)$$

The recursion to propagate the map posterior in time follows the generalization of the recursive Bayesian filter applied to sets [38] for which the prediction step is given by the Chapman-Kolmogorov equation

$$p(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) = \int f(\mathcal{M}_k | \mathcal{M}_{k-1}, \mathbf{x}_k) \times p(\mathcal{M}_{k-1} | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k-1}) \delta \mathcal{M}_{k-1} \quad (12)$$

and the update by applying the Bayes' rule

$$p(\mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}) = \frac{g(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_k) p(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k})}{g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k})}. \quad (13)$$

As detailed in [13], the factorized solution defined by (11)–(13) is similar to that of FastSLAM [11], [31] and the effect of conditioning on  $\mathbf{x}_{0:k}$  is to render each landmark estimate conditionally independent. However, adopting RFS likelihoods affects how (11) is evaluated. Furthermore, FastSLAM is conditioned on DA assignments, which are essentially unknown, whereas RFS-SLAM is not and the recursion defined by (12) and (13) is that of an RFS-based mapping with known poses.

### B. Representation of the RFS-SLAM Density

We follow an RBPF approach as in [13], [36], and [33] and approximate the posterior density of the robot trajectory using a weighted set of  $N$  particles

$$p(\mathbf{x}_{1:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) \approx \sum_{n=1}^N w_k^n \delta(\mathbf{x}_{1:k} - \mathbf{x}_{1:k}^n) \quad (14)$$

where  $\delta(\cdot)$  is the Dirac delta function,  $\mathbf{x}_{1:k}^n$  is the  $n$ th particle, and  $w_k^n$  is the associated weight. The RFS-SLAM density is parameterized using

$$\{w_k^n, \mathbf{x}_{0:k}^n, p(\mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}^n)\}_{n=1}^N \quad (15)$$

and each particle represents a single trajectory, and a trajectory conditioned map is associated with each of the particles.

We utilize a map representation based on the PHD by approximating the posterior RFS map using a Poisson point process (PPP) following assumed-density filtering [13], [39]

$$p(\mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}^n) \approx \frac{\prod_{\mathbf{m} \in \mathcal{M}} v_k(\mathbf{m} | \mathbf{x}_{0:k}^n)}{\exp\left(\int v_k(\mathbf{m} | \mathbf{x}_{0:k}^n) d\mathbf{m}\right)}. \quad (16)$$

In (16),  $v_k(\mathbf{m} | \mathbf{x}_{0:k}^n)$  is the shorthand notation for  $v_k(\mathbf{m} | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}^n)$  and it denotes the PHD conditioned on the  $n$ th trajectory. Using a GM parametrization, the PHD is given by [40]

$$v_k^n(\mathbf{m} | \mathbf{x}_{0:k}^n) = \sum_{i=1}^{M_k^n} \eta_k^{n,i} \mathcal{N}(\hat{\mathbf{m}}_k^{n,i}, \mathbf{P}_k^{n,i}) \quad (17)$$

where  $M_k^n$  is the number of GM components at time  $k$  and,  $\eta_k^{n,i}$ ,  $\hat{\mathbf{m}}_k^{n,i}$ , and  $\mathbf{P}_k^{n,i}$  are the weight, mean, and covariance of landmark  $i$  for particle  $n$ , respectively. It is to be noted that the weights  $\eta_k^{n,i}$  represent the number of landmarks at  $\hat{\mathbf{m}}_k^{n,i}$ , and  $\sum_{i=1}^{M_k^n} \eta_k^{n,i}$  gives the expected number of landmarks in the estimated map. In PHD-SLAM, the trajectory-conditioned map is estimated using a PHD filter and the overall PHD-SLAM density at time  $k$  is represented by

$$\left\{w_k^n, \mathbf{x}_{0:k}^n, \left\{\eta_k^{n,i}, \hat{\mathbf{m}}_k^{n,i}, \mathbf{P}_k^{n,i}\right\}_{i=1}^{M_k^n}\right\}_{n=1}^N. \quad (18)$$

### C. PHD Filter for Mapping

The utilized mapping algorithm follows the GM implementation of the PHD filter [13], [40], which is summarized in the following for the sake of completeness. If the PHD at the previous time instant is a GM, then it follows that the predicted PHD is also a GM, given by [13, eq. (22)]

$$v_{k|k-1}^n(\mathbf{m} | \mathbf{x}_{0:k}^n) = v_{k-1}^n(\mathbf{m} | \mathbf{x}_{0:k-1}^n) + v_k^B(\mathbf{m} | \mathcal{Z}_{k-1}^B, \mathbf{x}_{k-1}^n) \quad (19)$$

and it is the union of the prior map,  $v_{k-1}^n(\cdot)$ , and the PHD of the birth RFS,  $v_k^B(\cdot)$ , which is used to model landmarks that enter FOV( $\mathbf{x}_k$ ) for the first time, as discussed in Section III-A. The birth PHD at time  $k$  is modeled as a GM with  $M_k^{B,n}$  components, representing a subset of measurements at the previous time step,  $\mathcal{Z}_{k-1}^B \subseteq \mathcal{Z}_{k-1}$  [see definition of  $\mathcal{Z}_{k-1}^B$  after (22)].<sup>2</sup> Thus,

<sup>2</sup>A similar approach has been used in [13] and [36] but using all the measurements from time step  $k-1$ , that is,  $\mathcal{Z}_{k-1}^B = \mathcal{Z}_{k-1}$ .

the predicted PHD has  $M_{k|k-1}^n = M_{k-1}^n + M_k^{B,n}$  components, and parameters of the prior map are unchanged since the landmarks are static, that is,  $\eta_{k|k-1}^{n,i} = \eta_{k-1}^{n,i}$ ,  $\hat{\mathbf{m}}_{k|k-1}^{n,i} = \hat{\mathbf{m}}_{k-1}^{n,i}$ , and  $\mathbf{P}_{k|k-1}^{n,i} = \mathbf{P}_{k-1}^{n,i}$ .

Since the likelihood in (5) has a Gaussian form, it follows that the posterior PHD is also a GM, given by [13, eq. (23)]

$$v_k^n(\mathbf{m} | \mathbf{x}_{0:k}^n) = v_{k|k-1}^n(\mathbf{m} | \mathbf{x}_{0:k-1}^n) \left[ 1 - p_D(\mathbf{m} | \mathbf{x}_k^n) + \sum_{\mathbf{z} \in \mathcal{Z}_k} \frac{p_D(\mathbf{m} | \mathbf{x}_k^n) g(\mathbf{z} | \mathbf{m}, \mathbf{x}_k^n)}{c(\mathbf{z}) + \int \Lambda(\mathbf{m}', \mathbf{z} | \mathbf{x}_{0:k}^n) d\mathbf{m}'} \right] \quad (20)$$

where

$$\Lambda(\mathbf{m}', \mathbf{z} | \mathbf{x}_{0:k}^n) = p_D(\mathbf{m}' | \mathbf{x}_k^n) g(\mathbf{z} | \mathbf{m}', \mathbf{x}_k^n) v_{k|k-1}^n(\mathbf{m}' | \mathbf{x}_{0:k-1}^n). \quad (21)$$

In practice, parameters of the updated PHD can be estimated using any standard Gaussian filtering technique, such as the extended, unscented, or cubature Kalman filter (see, e.g., [41]). In this article, a first-order Taylor series-based Gaussian approximation is used allowing EKF style updates, and the reader is referred to [13] and [36] for further details. To reduce computational complexity of the update step, ellipsoidal gating [42] is utilized and a measurement is used to update parameters of the GM component only if the squared Mahalanobis distance is below gating threshold  $T_G$ . Mathematically

$$\|\mathbf{z}_k^j - \mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^{n,i}, \mathbf{x}_k^n)\|_{(\mathbf{S}_{k|k-1}^{n,i})^{-1}}^2 \leq T_G \quad (22)$$

where  $\mathbf{S}_{k|k-1}^{n,i} = \mathbf{G}_m \mathbf{P}_{k|k-1}^{n,i} \mathbf{G}_m^\top + \mathbf{R}_k$ ,  $\mathbf{G}_m$  denotes the Jacobian of  $\mathbf{g}$  with respect to  $\mathbf{m}^i$  evaluated at  $\hat{\mathbf{m}}_{k|k-1}^{n,i}$ , and we have used the notation  $\|\mathbf{e}\|_{\Omega}^2 = \mathbf{e}^\top \Omega \mathbf{e}$  above. Furthermore, measurements that are not used to update any landmark constitute  $\mathcal{Z}_k^B$ , which is then used to initialize new landmarks at the next time step. The updated map has at most  $M_{k|k}^n = M_{k|k-1}^n \times (|\mathcal{Z}_k| + 1)$  components if every component of  $v_{k|k-1}^n(\mathbf{m} | \mathbf{x}_{0:k-1}^n)$  is updated by a misdetection and by every measurement, but typically  $M_{k|k}^n < M_{k|k-1}^n \times (|\mathcal{Z}_k| + 1)$ .

### D. Robot Trajectory

The posterior of the robot trajectory,  $p(\mathbf{x}_{1:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$ , can be estimated using, for example, the sequential importance sampling PF (see, e.g., [43] and [44]), which recursively propagates the weights and support points. The recursion consists of two steps for every particle  $n$ ; the importance density is first computed and sampled from

$$\mathbf{x}_k^n \sim q(\mathbf{x}_k | \mathbf{x}_{0:k-1}^n, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}) \quad (23)$$

and thereafter, the weights are updated according to

$$w_k^n = w_{k-1}^n \frac{g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}^n) f(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n, \mathbf{u}_k)}{q(\mathbf{x}_k | \mathbf{x}_{0:k-1}^n, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k})}. \quad (24)$$

The optimal choice, in terms of minimizing the incremental particle weights, for the importance density in (23) is given by the OID [2], [44]

$$q(\mathbf{x}_k | \mathbf{x}_{0:k-1}^n, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}) = p(\mathbf{x}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k-1}^n, \mathbf{u}_{1:k}) \\ = \frac{g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}^n) f(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n, \mathbf{u}_k)}{p(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k-1}^n, \mathbf{u}_{1:k})}. \quad (25)$$

Unfortunately, sampling directly from this distribution is impossible in the general case; it does not even possess a closed form since we cannot express the right-hand side as a Gaussian in  $\mathbf{x}_k$ . However, an approximation of the OID can be attained if the measurement model is approximated by a linear function whereas the dynamic model may remain nonlinear [32]. Such an approximation is utilized in FastSLAM 2.0 [31] and a similar one in a recent PHD-SLAM work [20], but the solutions condition on the selected best DA assignment, and a wrong assignment can have significant ramifications to the approximated importance density. To address this problem, we propose sampling the poses under consideration of both the motion of the robot and the measurements, as well as DA uncertainty. The details of the improved approximation for (25) are given in Section V.

It is to be noted that in PHD-SLAM, a common choice for (23) is the transition density [13], [33], [34], [36], [45]

$$q(\mathbf{x}_k | \mathbf{x}_{0:k-1}^n, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}) = f(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n, \mathbf{u}_k) \quad (26)$$

simplifying the weight update in (24) to

$$w_k^n = w_{k-1}^n g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}^n). \quad (27)$$

Using the transition density as the importance density enables straightforward and efficient particle updates. However, if covariance of the process noise is large with respect to the measurement noise covariance, a downside of using such a proposal is that most of the sampled particles will have an insignificant weight after computing (24) and a large number of particles are required to accurately approximate the posterior. In Section VI, the performance of PHD-SLAM using different importance densities is presented and discussed.

## V. MULTIHYPOTHESES IMPORTANCE DENSITY

This section presents an improved approximation for the importance density given in (25) to better cope with DA uncertainty. To handle data ambiguities, an MH-ID is derived and a GMM representation is proposed, in which each component of the GM represents a single DA. Before presenting the MH-ID, we introduce a method that can be used to rank the DAs in the ascending order of cost. Then, the  $\Gamma$  best DAs are used for approximating the MH-ID. In the following, the importance density in (23) and weight in (24) are computed independently for each particle and dependence of  $\mathbf{x}_k^n$  and  $v_k^n(\cdot)$  on particle  $n$  is excluded from the notation for brevity.

### A. Computation of $\Gamma$ Best DAs

If the DA is unknown, the number of ways  $M_k$  landmarks can be assigned to  $J_k$  measurements increases very fast as  $M_k$  and  $J_k$  grow. Instead of exhaustively computing all the possible assignments,  $\Gamma_{\text{all}}$ , the goal is to rank the candidate DA assignments in the ascending order of cost, select the  $\Gamma$  best DAs, and truncate the rest. This problem can be cast as a ranked assignment problem and solved using Murty's algorithm [46].

The ranked assignment problem can be formulated using the optimal assignment problem, given by [47]

$$\begin{aligned} & \text{minimize} \quad \text{tr}(\Phi^T \mathbf{L}) \\ & \text{s.t.} \quad \{\Phi\}_{i,j} \in \{0, 1\} \quad \forall i, j \\ & \quad \sum_{j=1}^{M_k+J_k} \{\Phi\}_{i,j} = 1 \quad \forall i \\ & \quad \sum_{i=1}^{M_k} \{\Phi\}_{i,j} \in \{0, 1\} \quad \forall j \end{aligned} \quad (28)$$

where  $\Phi$  is the assignment matrix variable,  $\mathbf{L}$  is the cost matrix,  $i \in \{1, \dots, M_k\}$ , and  $j \in \{1, \dots, M_k + J_k\}$ . The cost matrix is defined as

$$\mathbf{L} = -\log \begin{bmatrix} \ell^{1,1} & \dots & \ell^{1,J_k} & | & \ell^{1,0} & \dots & 0 \\ \vdots & \ddots & \vdots & | & \vdots & \ddots & \vdots \\ \ell^{M_k,1} & \dots & \ell^{M_k,J_k} & | & 0 & \dots & \ell^{M_k,0} \end{bmatrix} \quad (29)$$

where the left  $M_k \times J_k$  submatrix corresponds to measurements that are associated with a landmark and the right  $M_k \times M_k$  diagonal submatrix corresponds to misdetections. The elements of the cost matrix are given by [48]

$$\ell^{i,j} = \int \frac{p_D(\mathbf{m}^i | \mathbf{x})}{c(\mathbf{z})} g(\mathbf{z}_k^j | \mathbf{m}^i, \mathbf{x}_{k|k-1}) p(\mathbf{m}^i | \mathbf{x}) d\mathbf{m}^i \quad (30)$$

$$\ell^{i,0} = \int (1 - p_D(\mathbf{m}^i | \mathbf{x})) p(\mathbf{m}^i | \mathbf{x}) d\mathbf{m}^i \quad (31)$$

in which  $\mathbf{x}_{k|k-1} = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$  denotes the predicted state,  $p(\mathbf{m}^i | \mathbf{x}) = v_{k|k-1}(\mathbf{m}^i | \mathbf{x}_{0:k})$  is the shorthand notation for the predicted density of the  $i$ th landmark, and the detection probability is approximated as  $p_D(\mathbf{m}^i | \mathbf{x}) \approx p_D(\hat{\mathbf{m}}_{k|k-1}^i | \mathbf{x}_{k|k-1})$  to make computation of the integrals feasible.

The optimal assignment problem seeks an assignment matrix  $\Phi$  that minimizes (28) and the ranked assignment problem seeks an enumeration of the least highest cost assignment matrices in nondecreasing order [46]. The cost of assignment  $\Phi$  is

$$\gamma_{k|k-1} = \exp(-\text{tr}(\Phi^T \mathbf{L})) \quad (32)$$

and Murty's algorithm [46] is used to find  $\tilde{\Gamma}$  assignment matrices such that  $\gamma_{k|k-1}^1 \leq \gamma_{k|k-1}^2 \leq \dots \leq \gamma_{k|k-1}^{\tilde{\Gamma}}$ . A fixed value could be used for  $\tilde{\Gamma}$  but to conserve computational resources,  $\tilde{\Gamma}$  is adaptively tuned to only account for DAs with a meaningful cost as proposed in [49]. In Murty's algorithm, we terminate hypotheses generation whenever the cost of the most likely hypotheses is below some fixed percentage of the cost of the next hypotheses, that is,  $\gamma_{k|k-1}^1 / \gamma_{k|k-1}^{\tilde{\Gamma}}$ . We have noticed that including the assignment,  $\{\Phi^{\tilde{\Gamma}+1}\}_{i,j} = 1 \quad \forall i \wedge j = J_k + i$ , increases robustness of the proposed algorithm in scenarios where a wrong DA leads to an inaccurate importance density. In such circumstances, it is beneficial to treat all of the measurements as clutter and use the transition density as the importance density. Thus, the total number of DAs is  $\Gamma = \tilde{\Gamma} + 1$ .

To make the treatment comprehensive, the mapping from  $\Phi$  to the association variable  $\phi_k^{i,j}$  in (6) is defined as follows.

Let  $\Phi^t$  denote the  $t$ th association matrix with cost  $\gamma_{k|k-1}^t$ . The association variable is defined as  $\phi_k^t = [\phi_k^{t,1}, \dots, \phi_k^{t,M_k}]$  for which the elements are given by

$$\phi_k^{t,i} = \begin{cases} j & \text{if } \{\Phi^t\}_{i,j} = 1 \text{ and } j \leq J_k \\ 0 & \text{otherwise.} \end{cases} \quad (33)$$

### B. MH-ID Approximation

The proposed method to compute parameters of the importance density is based on the IPL OID approximation introduced in [22]. The proposed solution exploits the measurement model structure and uses partitioned updates, that is, updates the importance density approximation using one measurement at a time, which yields an algorithm that scales according to  $\mathcal{O}(|\mathcal{M}|)$ . This efficiency makes it possible to include multiple DA hypotheses in the importance density approximation and still obtain a computationally efficient algorithm.

First, recall that the optimal choice (that minimizes the incremental particle weights) for the importance density  $q(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k})$  in (23) for the PHD-SLAM filter is given by [2], [44]

$$q(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}) \propto g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \times f(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \quad (34)$$

where [13, eq. (24)]

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) &= \int g(\mathcal{Z}_k, \mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \delta \mathcal{M}_k \\ &= \int g(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_k) p(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \delta \mathcal{M}_k \end{aligned} \quad (35)$$

with  $g(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_k)$  and  $p(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k})$  as in (4) and (12), respectively. Then, for  $\Gamma$  DA hypotheses, we have (see the Appendix for details of the derivation)

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) &= \sum_{t=1}^{\Gamma} \exp(-\lambda_c) \prod_{j=1}^{|\mathcal{Z}|} c(\mathbf{z}_k^j) \\ &\times \prod_{i:\phi_k^{t,i}=0} \eta_{k|k-1}^i \int [1 - p_D(\mathbf{m}^i | \mathbf{x}_k)] \\ &\times \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^i \\ &\times \prod_{i:\phi_k^{t,i}>0} \frac{\eta_{k|k-1}^i}{c(\mathbf{z}_k^{\phi_k^{t,i}})} \int p_D(\mathbf{m}^i | \mathbf{x}_k) g(\mathbf{z}_k^{\phi_k^{t,i}} | \mathbf{m}^i, \mathbf{x}_k) \\ &\times \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^i. \end{aligned} \quad (36)$$

Unfortunately, the detection probability makes the solution of the remaining integrals not feasible. However, using the predicted probability of detection  $\hat{p}_D^i \approx p_D(\hat{\mathbf{m}}_{k|k-1}^i | \mathbf{x}_{k|k-1})$  (obtained from the DA, Section V-A), we obtain the approximation

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) &\approx \sum_{t=1}^{\Gamma} \exp(-\lambda_c) \prod_{j=1}^{|\mathcal{Z}|} c(\mathbf{z}_k^j) \prod_{i:\phi_k^{t,i}=0} \eta_{k|k-1}^i [1 - \hat{p}_D^i] \\ &\times \prod_{i:\phi_k^{t,i}>0} \frac{\eta_{k|k-1}^i \hat{p}_D^i}{c(\mathbf{z}_k^{\phi_k^{t,i}})} \\ &\times \int g(\mathbf{z}_k^{\phi_k^{t,i}} | \mathbf{m}^i, \mathbf{x}_k) \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^i. \end{aligned} \quad (37)$$

Next, recall that PHD-SLAM (as well as FastSLAM) uses an affine approximation of the likelihood around the predicted landmark state  $\hat{\mathbf{m}}_{k|k-1}^i$  to facilitate the use of approximate Rao–Blackwellization in the PF, which makes the SLAM problem tractable. This yields the affine approximation of the measurement likelihood function (5)

$$\begin{aligned} g(\mathbf{z}_k^{\phi_k^{t,i}} | \mathbf{m}^i, \mathbf{x}_k) &\approx \mathcal{N}(\mathbf{z}_k^{\phi_k^{t,i}}; \mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^i, \mathbf{x}_k) + \mathbf{G}_{\mathbf{m},k}(\mathbf{m}^i - \hat{\mathbf{m}}_{k|k-1}^i), \mathbf{R}_k) \end{aligned}$$

where  $\mathbf{G}_{\mathbf{m},k}$  is the Jacobian of  $\mathbf{g}$  with respect to  $\mathbf{m}^i$  evaluated at  $\hat{\mathbf{m}}_{k|k-1}^i$ . Hence, the predicted likelihood is approximated as

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) &\approx \sum_{t=1}^{\Gamma} \exp(-\lambda_c) \prod_{j=1}^{|\mathcal{Z}|} c(\mathbf{z}_k^j) \prod_{i:\phi_k^{t,i}=0} \eta_{k|k-1}^i [1 - \hat{p}_D^i] \\ &\times \prod_{i:\phi_k^{t,i}>0} \frac{\eta_{k|k-1}^i \hat{p}_D^i}{c(\mathbf{z}_k^{\phi_k^{t,i}})} \\ &\times \mathcal{N}(\mathbf{z}_k^{\phi_k^{t,i}}; \mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^i, \mathbf{x}_k), \mathbf{G}_{\mathbf{m},k} \mathbf{P}_{k|k-1}^i \mathbf{G}_{\mathbf{m},k}^\top + \mathbf{R}_k). \end{aligned} \quad (38)$$

Then, the MH-ID can in turn be approximated as

$$\begin{aligned} p(\mathbf{x}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}) &\propto \sum_{t=1}^{\Gamma} \kappa_{k|k-1}^t f(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \\ &\times \prod_{i:\phi_k^{t,i}>0} \mathcal{N}(\mathbf{z}_k^{\phi_k^{t,i}}; \mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^i, \mathbf{x}_k), \\ &\mathbf{G}_{\mathbf{m},k} \mathbf{P}_{k|k-1}^i \mathbf{G}_{\mathbf{m},k}^\top + \mathbf{R}_k) \end{aligned} \quad (39)$$



where  $\propto$  denotes “approximately proportional to.” This is a mixture density with mixture weights

$$\kappa_{k|k-1}^t \propto \exp(-\lambda_c) \prod_{j=1}^{|Z|} c(\mathbf{z}_k^j) \prod_{i:\phi_k^{t,i}=0} \eta_{k|k-1}^i [1 - \hat{p}_D] \times \prod_{i:\phi_k^{t,i}>0} \frac{\eta_{k|k-1}^i \hat{p}_D^i}{c(\mathbf{z}_k^{\phi_k^{t,i}})}. \quad (40)$$

Unfortunately, this cannot be sampled from and instead, we propose to approximate the above mixture density using a GM of the form

$$p(\mathbf{x}_k \mid \mathcal{Z}_{1:k}, \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}) \approx \sum_{t=1}^{\Gamma} \kappa_k^t \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t) \quad (41)$$

instead. It is to be noted that the work in [20] uses a single Gaussian representing one DA to approximate the importance density, whereas in this work, we use a GM representing  $\Gamma$  DAs to approximate the importance density. As in [20], the mixture moments  $\boldsymbol{\mu}_k^t$  and  $\boldsymbol{\Sigma}_k^t$  are calculated using the IPL-based OID approximation approach [22]. However, we use a partitioned update strategy, which entails that the individual landmark measurements are used one at a time to update the proposal density, whereas the work in [20] uses a joint update strategy.

Algorithm 1 summarizes the resulting algorithm, which consists of three steps. First, the measurement likelihood is linearized, which requires calculating the moments

$$\tilde{\boldsymbol{\mu}}_l^z = E_{\pi}\{\mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^i, \mathbf{x}_k)\} \quad (42a)$$

$$\begin{aligned} \tilde{\boldsymbol{\Sigma}}_l^{zz} = & E_{\pi}\{(\mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^i, \mathbf{x}_k) - \tilde{\boldsymbol{\mu}}_l^z)(\mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^i, \mathbf{x}_k) - \tilde{\boldsymbol{\mu}}_l^z)^{\top}\} \\ & + E_{\pi}\{\mathbf{G}_{\mathbf{m},k} \mathbf{P}_{k|k-1}^i \mathbf{G}_{\mathbf{m},k}^{\top}\} + \mathbf{R}_k \end{aligned} \quad (42b)$$

$$\tilde{\boldsymbol{\Sigma}}_l^{zx} = E_{\pi}\{(\mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^i, \mathbf{x}_k) - \tilde{\boldsymbol{\mu}}_l^z)(\mathbf{x}_k - \tilde{\boldsymbol{\mu}}_l^x)^{\top}\} \quad (42c)$$

where  $E_{\pi}\{\cdot\}$  denotes the expectation with respect to the linearization density  $\pi(\mathbf{x}_k)$ . Note that the resulting integrals can typically not be solved in closed form, but Taylor series expansion or sigma points can be used to approximate them. In this article, we resort to local linearization using a first-order Taylor series expansion.

Second, based on the linearization moments (42), the moments of the linearized likelihood (linearized w.r.t.  $\pi(\mathbf{x}_k)$ ) are calculated according to [21], [22]

$$\mathbf{B}_l = \tilde{\boldsymbol{\Sigma}}_l^{zx} \left( \tilde{\boldsymbol{\Sigma}}_l^{xx} \right)^{-1} \quad (43a)$$

$$\boldsymbol{\mu}_l^z = \tilde{\boldsymbol{\mu}}_l^z + \mathbf{B}_l (\tilde{\boldsymbol{\mu}}_l^x - \tilde{\boldsymbol{\mu}}_{l-1}^x) \quad (43b)$$

$$\boldsymbol{\Sigma}_l^{zz} = \tilde{\boldsymbol{\Sigma}}_l^{zz} + \mathbf{B}_l \left( \tilde{\boldsymbol{\Sigma}}_l^{xx} - \tilde{\boldsymbol{\Sigma}}_{l-1}^{xx} \right) \mathbf{B}_l^{\top} \quad (43c)$$

$$\boldsymbol{\Sigma}_l^{xz} = \tilde{\boldsymbol{\Sigma}}_l^{zx} \mathbf{B}_l^{\top}. \quad (43d)$$

Third, the mean and covariance of the mixture component are updated using the standard Kalman filter update. Note, however, that the measurements are incorporated one at a time in a partitioned update scheme, which has computational cost

---

**Algorithm 1:** IPL Approximation for Mixture Component  $t$ .

---

- 1: Set  $\tilde{\boldsymbol{\mu}}_0^x = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$ ,  $\tilde{\boldsymbol{\Sigma}}_0^{xx} = \mathbf{Q}_{k-1}$
  - 2: **for**  $l = 1, \dots, L$  **do**
  - 3:   Set  $\tilde{\boldsymbol{\mu}}_l^x = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$ ,  $\tilde{\boldsymbol{\Sigma}}_l^{xx} = \mathbf{Q}_{k-1}$
  - 4:   **for**  $i : \phi_k^{t,i} > 0$  **do**
  - 5:     *Linearization:* Calculate the moments  $\tilde{\boldsymbol{\mu}}_l^z$ ,  $\tilde{\boldsymbol{\Sigma}}_l^{zz}$ , and  $\tilde{\boldsymbol{\Sigma}}_l^{zx}$  using  $\mathbf{z}_k^{\phi_k^{t,i}}$  w.r.t. the linearization density  $\pi(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \tilde{\boldsymbol{\mu}}_{l-1}^x, \tilde{\boldsymbol{\Sigma}}_{l-1}^{xx})$  according to (42)
  - 6:     *Moment matching:* Calculate the moments  $\boldsymbol{\mu}_l^z$ ,  $\boldsymbol{\Sigma}_l^{zz}$ , and  $\boldsymbol{\Sigma}_l^{xz}$  of the linearized measurement model according to (43)
  - 7:     *Measurement update for landmark  $i$ :*  

$$\mathbf{K}_l = \boldsymbol{\Sigma}_l^{xz} (\boldsymbol{\Sigma}_l^{zz})^{-1}$$

$$\tilde{\boldsymbol{\mu}}_l^x = \tilde{\boldsymbol{\mu}}_l^x + \mathbf{K}_l (\mathbf{z}_k^{\phi_k^{t,i}} - \boldsymbol{\mu}_l^z)$$

$$\tilde{\boldsymbol{\Sigma}}_l^{xx} = \tilde{\boldsymbol{\Sigma}}_l^{xx} - \mathbf{K}_l \boldsymbol{\Sigma}_l^{zz} \mathbf{K}_l^{\top}$$
  - 8:   **end for**
  - 9:   **end for**
  - 10: Set  $\boldsymbol{\mu}_k^t = \tilde{\boldsymbol{\mu}}_L^x$ ,  $\boldsymbol{\Sigma}_k^t = \tilde{\boldsymbol{\Sigma}}_L^{xx}$
- 

of  $\mathcal{O}(|\mathcal{M}_k|)$ , compared with a full joint update used in [20] that scales according to  $\mathcal{O}(|\mathcal{M}_k|^3)$ .

The choice of the linearization density is as follows. First, the linearization density is chosen to the dynamic model (i.e., the prior). Then, after each iteration, a new approximation of the posterior density is obtained, which is then used as the linearization density in the next iteration. The procedure is repeated either for a fixed number of iterations  $L$  or upon convergence, see [22]. Once the moments have been computed, weight of the mixture component is given by

$$\begin{aligned} \kappa_k^t \propto & \exp(-\lambda_c) \prod_{j=1}^{|Z|} c(\mathbf{z}_k^j) \prod_{i:\phi_k^{t,i}=0} \eta_{k|k-1}^i [1 - \hat{p}_D] \\ & \times \prod_{i:\phi_k^{t,i}>0} \frac{\eta_{k|k-1}^i \hat{p}_D^i}{c(\mathbf{z}_k^{\phi_k^{t,i}})} \mathcal{N}\left(\mathbf{z}_k^{\phi_k^{t,i}}; \mathbf{g}(\hat{\mathbf{m}}_{k|k-1}^i, \boldsymbol{\mu}_k^t), \mathbf{S}_k^{t,i}\right) \end{aligned} \quad (44)$$

where  $\mathbf{S}_k^{t,i} = \mathbf{G}_{\mathbf{x},k} \boldsymbol{\Sigma}_k^t \mathbf{G}_{\mathbf{x},k}^{\top} + \mathbf{G}_{\mathbf{m},k} \mathbf{P}_{k|k-1}^i \mathbf{G}_{\mathbf{m},k}^{\top} + \mathbf{R}_k$  and  $\mathbf{G}_{\mathbf{x},k}$  is the Jacobian of  $\mathbf{g}$  with respect to  $\mathbf{x}$  evaluated at  $\boldsymbol{\mu}_k^t$ .

### C. Sampling and Importance Weight

After computing parameters of the GM-ID approximation in (41), a new particle  $\mathbf{x}_k^n$  is drawn from the proposed importance density as follows. First, a component  $t$  is randomly drawn from the categorical distribution defined by the weights (44). Thereafter, the multivariate normal distribution with moments  $\boldsymbol{\mu}_k^t$  and  $\boldsymbol{\Sigma}_k^t$  is sampled from, that is,  $\mathbf{x}_k^n \sim \mathcal{N}(\boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t)$ . It is important to note that for each particle only one mixture component is sampled from the GM-ID and the multihypotheses structure is only partially preserved over time for a single particle. Since

**Algorithm 2:** Proposed Algorithm to Update Particle  $n$ .

- 
- 1: **Input:**  $\mathcal{Z}_k, \mathbf{u}_k, \{w_{k-1}^n, \mathbf{x}_{k-1}^n, v_{k|k-1}^n(\mathbf{m}|\mathbf{x}_{0:k}^n)\}_{n=1}^N$
  - 2: **Procedure** Compute GM-ID
  - 3: *Cost matrix:* Propagate particle  
 $\mathbf{x}_{k|k-1} = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$  and compute  $\mathbf{L}$  according to (29).
  - 4: *Rank DAs:* Compute  $\Gamma$  best DAs using  $\mathbf{L}$  and Murty's algorithm as presented in Section V-A.
  - 5: **for**  $t = 1, \dots, \Gamma$  **do**
  - 6:   Compute moments  $\boldsymbol{\mu}_k^t$  and  $\boldsymbol{\Sigma}_k^t$  using Algorithm 1 and weight  $\kappa_k^t$  according to (44).
  - 7: **end for**
  - 8: **end procedure**
  - 9: *Sample from GM-ID:*  $\mathbf{x}_k^n \sim \sum_{t=1}^{\Gamma} \kappa_k^t \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t)$ .
  - 10: *Incremental weight:* Update  $w_k^n$  according to (45).
  - 11: **Output:** Updated particle  $\{w_k^n, \mathbf{x}_k^n\}$  at time step  $k$
- 

the weights are normalized by the likelihood of the GM-ID, the multihypotheses structure is encoded in the weights of the particles. In addition, different particles, even with the same prior, can sample different mixture components from the GM-ID, and the PHD-SLAM density in (18) is able to capture  $N$  hypotheses overall.

Log of the importance weight in (24),  $\tilde{w}_k^n = \log(w_k^n)$ , is updated using

$$\begin{aligned} \tilde{w}_k^n &= \log \left( w_{k-1}^n \frac{g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}^n) f(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n, \mathbf{u}_k)}{q(\mathbf{x}_k | \mathbf{x}_{0:k-1}^n, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k})} \right) \\ &= \tilde{w}_{k-1}^n + \tilde{w}_{\text{meas.}}^n + \tilde{w}_{\text{prior}}^n - \tilde{w}_{\text{prop.}}^n. \end{aligned} \quad (45)$$

In literature, several methods for computing  $\tilde{w}_{\text{meas.}}^n$  have been proposed including an empty map update [13], single feature update [13], and multifeature update [33]. With a PPP prior and a point object measurement model, we use the exact expression for  $\tilde{w}_{\text{meas.}}^n$ , computed as [34]

$$\tilde{w}_{\text{meas.}}^n = \sum_{\mathbf{z} \in \mathcal{Z}_k} \log \left( c(\mathbf{z}) + \int \Lambda(\mathbf{m}, \mathbf{z} | \mathbf{x}_k^n) d\mathbf{m} \right) \quad (46)$$

where  $\Lambda(\cdot)$  is defined in (21). Log-likelihood of the prior is given by

$$\tilde{w}_{\text{prior}}^n = \log \mathcal{N}(\mathbf{x}_k^n; \mathbf{f}(\mathbf{x}_{k-1}^n, \mathbf{u}_k), \mathbf{Q}_{k-1}). \quad (47)$$

Respectively, log-likelihood of the proposal is computed using

$$\tilde{w}_{\text{prop.}}^n = \log \sum_{t=1}^{\Gamma} \tilde{\kappa}_k^t \mathcal{N}(\mathbf{x}_k^n; \boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t) \quad (48)$$

where  $\tilde{\kappa}_k^t = \kappa_k^t / \sum_{t=1}^{\Gamma} \kappa_k^t$  are the normalized weights. Algorithm 2 summarizes the proposed algorithm to approximate the importance density in (23), sample from it in closed form, and update the incremental weights in (24). In practice however, it is computationally more efficient to compute the incremental weight in two phases. The weight is updated with log-likelihoods of the prior and proposal during the weight update and  $\tilde{w}_{\text{meas.}}^n$  is added to the importance weight during the PHD update step.

**Algorithm 3:** Proposed PHD-SLAM Algorithm at Time Step  $k$ .

- 
- 1: **Input:**  $\mathcal{Z}_k, \mathbf{u}_k, \{w_{k-1}^n, \mathbf{x}_{k-1}^n, v_{k-1}^n(\mathbf{m}|\mathbf{x}_{0:k-1}^n)\}_{n=1}^N$
  - 2: **for**  $n = 1, \dots, N$  **do**
  - 3:   *PHD predict:* Compute  $v_{k|k-1}^n(\mathbf{m}|\mathbf{x}_{0:k}^n)$  using (19).
  - 4:   *Particle update:* Use Algorithm 2 to obtain  $\{w_k^n, \mathbf{x}_k^n\}$ .
  - 5:   *PHD update:* Compute  $v_k^n(\mathbf{m}|\mathbf{x}_{0:k}^n)$  using (20).
  - 6:   *Hypotheses reduction:* Prune and merge  $M_{k|k}^n$  GM-PHD components to obtain a reduced number of components,  $M_k^n \leq M_{k|k}^n$ , as presented in [40].
  - 7: **end for**
  - 8: *Estimate:* Find  $j = \arg \max_N \{w_k\}_{n=1}^N$  and compute the map  $\hat{\mathcal{M}}_k = \{\hat{\mathbf{m}}_k^{j,i} : \eta^{j,i} \geq T_\eta, i = 1, \dots, M_k^j\}$  and robot  $\hat{\mathbf{x}}_k = \mathbf{x}_k^j$  estimates.
  - 9: *Resample:* Normalize the weights  $w_k^n = w_k^n / \sum_{n=1}^N w_k^n$ , compute the effective sample size  $\text{ESS} = 1 / \sum_{n=1}^N w_k^n$ . and resample if  $\text{ESS} \leq T_{\text{ESS}}$ .
  - 10: **Output:**  $\hat{\mathcal{M}}_k, \hat{\mathbf{x}}_k, \{w_k^n, \mathbf{x}_k^n, v_k^n(\mathbf{m}|\mathbf{x}_{0:k}^n)\}_{n=1}^N$ .
- 

## VI. EXPERIMENTAL RESULTS

The development efforts of the article are evaluated with synthetic data [35] and the experimental Victoria Park dataset [50]. The experiments focus on vehicles operating in planar environments so that pose can be represented by the 2-D location  $(x, y)$  and heading  $\theta$ . Furthermore, the vehicle is controlled by speed  $v$  and steering  $\omega$  commands. Thus, the state and control input of the vehicle at time  $k$  are  $\mathbf{x}_k = [x_k, y_k, \theta_k]^\top$  and  $\mathbf{u}_k = [v_k, \omega_k]^\top$ , in respective order. The landmarks in the environment are static, and location of the  $i$ th landmark is  $\mathbf{m}^i = [x^i, y^i]^\top$ .

We compare the developed filter only with other PHD-SLAM filters [13], [20], [35], since it has already been demonstrated that they outperform more conventional SLAM filters (e.g., [11], [31], and [51]) in high clutter scenarios [13], [20], [33], [35]. We refer to the original PHD-SLAM filter [13] as PHD-SLAM 1.0 and the other two filters as PHD-SLAM 2.0a [35] and PHD-SLAM 2.0b [20] since both approximate the importance density considering both the motion of the robot and the measurements. This naming convention follows the FastSLAM algorithms, since the original algorithm that used the motion model as the importance density was coined as FastSLAM 1.0 [11] and the filter that used the improved sampling distribution was named as FastSLAM 2.0 [31]. Honoring this naming convention, the algorithm implemented in this article is coined as PHD-SLAM 3.0, since the presented MH-ID offers a notable advantage in cluttered scenarios where data ambiguities are common.

Overall, 100 Monte Carlo simulations (MCSs) are performed and the results are obtained by averaging over the independent simulations. The accuracy of the state estimates is evaluated using the root mean squared error (RMSE) and mapping accuracy is evaluated with the generalized optimal subpattern assignment (GOSPA) metric, which captures the localization error and penalizes for missed and false landmarks. Let  $\mathcal{M}_k =$

$\{\mathbf{m}_k^1, \dots, \mathbf{m}_k^{|\mathcal{M}|}\}$  and  $\hat{\mathcal{M}}_k = \{\hat{\mathbf{m}}_k^1, \dots, \hat{\mathbf{m}}_k^{|\hat{\mathcal{M}}|}\}$  denote the map and its estimate at time  $k$ , respectively. Now, GOSPA is defined as [52]

$$d_{k,p}^c(\mathcal{M}_k, \hat{\mathcal{M}}_k) = \left[ \min_{\xi \in \Xi} \sum_{(i,j) \in \xi} \|\mathbf{m}_k^i - \hat{\mathbf{m}}_k^j\|^p + \frac{c^p}{2} (|\mathcal{M}_k| + |\hat{\mathcal{M}}_k| - 2|\gamma|) \right]^{\frac{1}{p}}$$

in which  $\xi$  is the assignment set between  $\mathcal{M}$  and  $\hat{\mathcal{M}}$ ,  $\Xi$  is the set of all possible assignment sets  $\xi$ ,  $\|\cdot\|$  denotes the Euclidean norm, and  $c$  defines the maximum allowable localization error. Unless otherwise stated, GOSPA is computed only for the last time instant to capture the localization and cardinality errors of the final map. Parameters used in computing GOSPA are:  $c = 20$  m and  $p = 2$ .

As with all PHD-SLAM filters, pruning and merging operations are required to limit the exponential growth of Gaussian components in the PHD. These operations are carried out as in [40, Table II] using a pruning threshold of  $\log(10^{-6})$  and a merging threshold of 50. In addition, ellipsoidal gating with threshold  $T_G = 41.4465$  is used to lower computational complexity of the PHD update step in (20), as well as, computing the cost matrix in (29). Weight of newly created landmarks is initialized to  $\eta = \log(10^{-6})$ . The map estimate threshold is  $T_\eta = (1 - p_D)^2$  to allow two consecutive misdetections before the landmark is not considered as an estimate. The maximum number of ranked assignments computed by Murty's algorithm is limited to  $\bar{\Gamma} = 50$  and the threshold to terminate hypotheses generation is  $T_\gamma = \log(10^{-3})$ . The maximum number of IPL iterations is  $L = 5$  and the convergence threshold is  $\epsilon = 10^{-3}$  [22, eq. (11)]. The effective sample size (ESS) threshold to perform resampling is set to  $T_{\text{ESS}} = 0.2 \times N$ . The developed PHD-SLAM filter is summarized in Algorithm 3.

The PHD-SLAM filters are implemented using MATLAB and the core functions are written in C/C++ and compiled to MATLAB MEX-files to enable a highly efficient implementation of the algorithms. The simulations and experiments are run on a Lenovo ThinkPad P1 Gen 2, with a 2.6 GHz 6-Core Intel i7-9850H CPU and 64 GB of memory. Multithreading is not exploited and the simulations are run on a single CPU core.

#### A. Synthetic Dataset

The simulation scenario is illustrated in Fig. 1 in which a robot is exploring a 1 km<sup>2</sup> area that contains 160 landmarks along the robot trajectory. Kinematics of the robot is described by a velocity motion model [2]

$$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x_{k-1} - \frac{v_k}{\omega_k} \sin(\theta_{k-1}) + \frac{v_k}{\omega_k} \sin(\theta_{k-1} + \omega_k T) \\ y_{k-1} + \frac{v_k}{\omega_k} \cos(\theta_{k-1}) - \frac{v_k}{\omega_k} \cos(\theta_{k-1} + \omega_k T) \\ \theta_{k-1} + \omega_k T \end{bmatrix} \quad (49)$$

and disturbances enter the system via a noisy control input that is corrupted by zero-mean independent identically distributed (i.i.d.) Gaussian noise  $\boldsymbol{\vartheta}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ , with covariance  $\mathbf{Q} =$

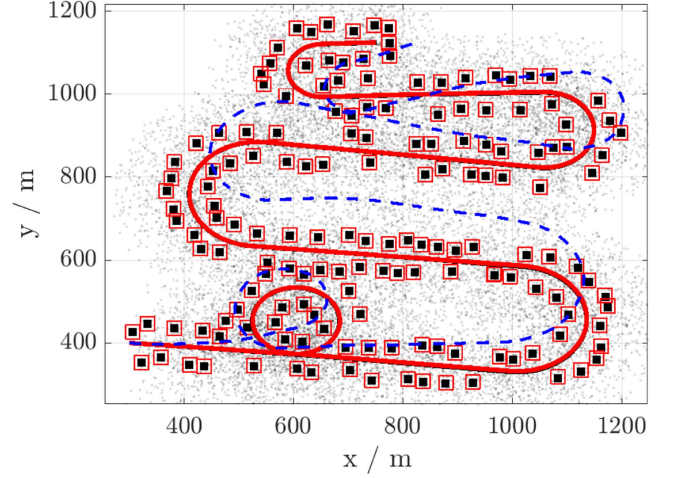


Fig. 1. Simulation scenario in which the landmarks are illustrated using (■), the robot trajectory with (—), and measurements as observed from the ground truth poses using (·). Trajectory estimate only using odometry information (---) and the trajectory (—) and map (□) estimates of PHD-SLAM 3.0 using one particle. In the example, rms positioning error is 1.30 m, rms heading error is 0.27°, and GOSPA of the final map is 21.94 m.

$\text{diag}(\sigma_v^2, \sigma_\omega^2)$ . The robot is equipped with a sensor that measures the range and bearing of the landmark relative to the robot's local coordinate frame. The measurement model for the  $i$ th landmark is given by [2]

$$\mathbf{g}(\mathbf{m}^i, \mathbf{x}_k) = \begin{bmatrix} \sqrt{(x^i - x_k)^2 + (y^i - y_k)^2} \\ \text{atan2}(y^i - y_k, x^i - x_k) - \theta_k \end{bmatrix} \quad (50)$$

and the measurements are corrupted by zero-mean i.i.d. Gaussian noise,  $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ , with covariance  $\mathbf{R} = \text{diag}(\sigma_r^2, \sigma_\phi^2)$ .

In the simulations, the robot speed is 1 m/s on average and the maximum angular velocity is approximately 1 °/s. Standard deviations of the control noises are  $\sigma_v = 0.8$  m/s and  $\sigma_\omega = \frac{0.5\pi}{180}$  rad/s. The range-bearing sensor outputs a measurement every  $T = 1$  s, and the sensor has a limited FOV with a  $\pm 90^\circ$  scanning angle and a maximum range of 150 m. The detection probability is  $p_D = 0.95$  within the FOV, zero outside the FOV, and the detection probability is approximated using  $E\{p_D(\mathbf{m} | \mathbf{x}_k)\}$  in which the expectation is evaluated with respect to the landmark density  $\mathcal{N}(\hat{\mathbf{m}}, \mathbf{P})$ . Standard deviations of measurement noise are  $\sigma_r = 0.8$  m and  $\sigma_\phi = \frac{0.3\pi}{180}$  rad. The clutter intensity inside the FOV is  $\lambda = 5$  so that the expected number of false measurements is five per time epoch. The simulation parameters are the same as used in [35].

1) *Particle Degeneracy*: We begin the evaluation by examining sample degeneracy of the filters. PFs utilize resampling to algorithmically limit degeneracy of the particles [32] but as a drawback, diversity in the sample set is decreased as some of the particles are neglected while others are duplicated. After resampling has been performed enough times, all of the particles will share a single mutual ancestor and diversity in the sample set beyond this point is lost. The distance to the common ancestor is crucial to the performance of RBPF-based SLAM algorithms, since it defines how large a loop can effectively be closed [53].



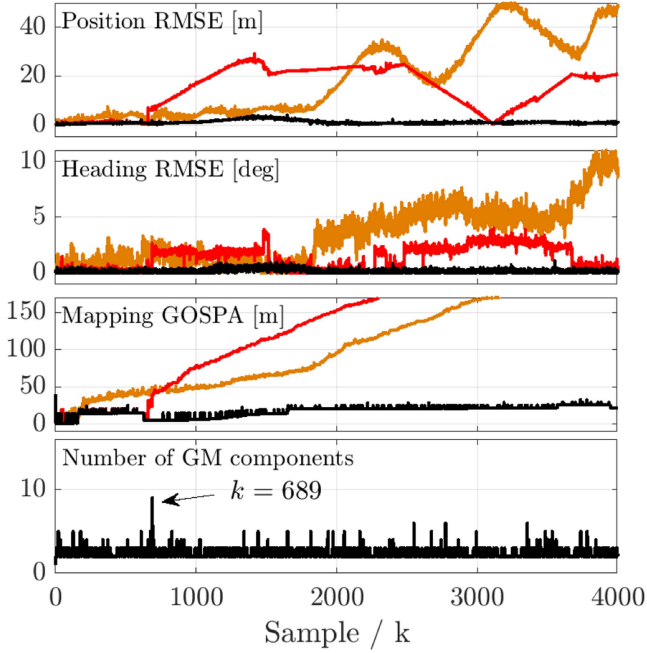


Fig. 2. Evaluation metrics as a function of time for the different filters, which are illustrated using: PHD 2.0a (—), PHD 2.0b (—), and PHD 3.0 (—) for which the estimated trajectory is shown in Fig. 1. On the bottom, the number of GM components used in the PHD 3.0 GM-ID approximation, and as shown, data ambiguity increases the number of GM components, which enables the proposed filter to account for DA uncertainty. Wrong DA can cause a filter to diverge as portrayed by PHD 2.0b at time instance  $k = 689$ .

Hence, resampling is disabled in the following analysis to emphasize sample degeneracy. Results for PHD-SLAM 1.0 are partly omitted in the following analysis since the filter exhibits severe particle degeneracy and requires very frequent resampling. For brevity, we refer to the filters simply as PHD X.0x from now on.

The evaluation metrics of an exemplar MCS and the number of GM components used by the proposed GM-ID approximation are illustrated in Fig. 2. In the example, the filters only use one particle and as visualized, the proposed filter is able to track the robot and map the environment accurately (see also Fig. 1). On the contrary, PHD 2.0a and PHD 2.0b diverge as indicated by the evaluation metrics. The importance density of PHD 2.0a is computed by conditioning the predicted robot state to the measurements independently to obtain  $|\mathcal{M}_k| \times |\mathcal{Z}_k|$  independent conditional distributions. Thereafter, the GMM is approximated as a single Gaussian using weighted averaging to obtain the mean and covariance of the proposal density. The method is suboptimal since the dependencies between the robot and landmarks are mostly ignored. PHD 2.0b approximates the importance density by first finding the joint approximation of the robot and map, and then by conditioning on the measurements of a single DA. Although this approach typically yields a more accurate approximation of the importance density, it is sensitive to data ambiguities, which can cause the filter to diverge, as illustrated in Fig. 2. The proposal of PHD 3.0 takes data ambiguity into account by computing the importance density approximation for multiple DAs resulting in a GMM proposal,

TABLE I  
POSITIONING AND MAPPING ACCURACY OF PHD-SLAM WITHOUT RESAMPLING

Filter	$N = 1$		$N = 10$		$N = 100$	
	Pos.	Map	Pos.	Map	Pos.	Map
PHD 1.0	254.2	229.5	258.7	230.1	237.8	230.4
PHD 2.0a	51.5	217.3	46.4	215.8	47.2	214.5
PHD 2.0b	29.7	189.4	26.7	185.9	23.3	175.3
PHD 3.0	3.5	46.8	2.4	38.7	2.5	38.4

The evaluation metrics are in meters.

which is then used for sampling. As illustrated in Fig. 2, the proposed filter is able to handle data ambiguities, at for example sample  $k = 689$ , resulting in good performance even when using just one particle.

In general, the accuracy of RBPF-based SLAM solutions can be improved using more particles, and Fig. 3 illustrates the trajectories of 100 independent particles of an exemplar MCS. As illustrated, PHD 3.0 is able to constrain uncertainty of the posterior approximation at a much lower level than the benchmark solutions, which is advantageous when closing large loops. The proposal used in PHD 2.0a is overly conservative and the filter is unable to propagate the posterior approximation accurately over time leading to satisfactory performance. The proposal of PHD 2.0b utilizes an importance density approximation similar to the OID improving the performance with respect to PHD 2.0a. However, wrong DAs increase dispersion of the particles and the filter requires a high number of particles to account for data ambiguities. On the contrary, PHD 3.0 is able to achieve high accuracy already with one particle since the posterior can be accurately approximated over extended periods of time and under DA uncertainty. The results with different number of particles are summarized in Table I and the results imply that PHD 3.0 has notable benefits with respect to the benchmark solutions.

The DAs used by PHD 2.0b and PHD 3.0 for approximating the importance density are explicitly defined so that we can quantify the accuracy of DA for these two filters. Assigning measurements to correct landmarks is measured using the true positive ratio (TPR), assigning measurements to clutter is measured using the true negative ratio (TNR), and value of one corresponds to perfect DA. For PHD 2.0b,  $\text{TPR} = 0.9223$  and  $\text{TNR} = 0.9980$ , and for PHD 3.0,  $\text{TPR} = 0.9829$  and  $\text{TNR} = 0.9984$ . It is important to note that for PHD 3.0, the DA assignments are ranked in nondecreasing order of cost, that is,  $\gamma_{k|k-1}^1 \leq \gamma_{k|k-1}^2 \leq \dots \leq \gamma_{k|k-1}^\Gamma$ . PHD 2.0b always uses the optimal assignment with cost  $\gamma_{k|k-1}^1$  for approximating the importance density. On the other hand, PHD 3.0 samples from the GMM according to the mixture weights  $\kappa_k^t$  and it is possible that the optimal assignment with cost  $\gamma_{k|k-1}^1$  does not result to the highest weight so that another mixture component is sampled instead. The main difference of computing  $\gamma_{k|k-1}^t$  and  $\kappa_k^t$  is the used prior,  $\gamma_{k|k-1}^t$  is computed using the predicted vehicle pose, whereas  $\kappa_k^t$  is computed using the IPL approximation. Thanks to this subtle but important difference, PHD 3.0 is able to make correct DAs more frequently, validating the research premise that the GM-ID is more suited for modeling DA uncertainty.



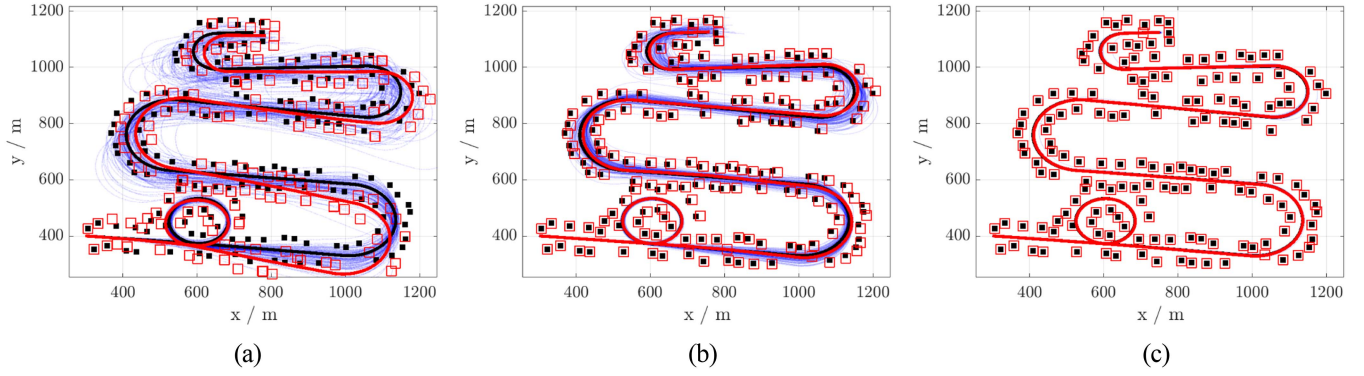


Fig. 3. In total, 100 particle trajectories estimated using PHD 2.0a, PHD 2.0b, and PHD 3.0 without resampling. In the figures, the landmarks are illustrated using (■) and the ground truth trajectory with (—). The independent particle trajectories shown with (—), the most likely hypotheses illustrated using (—) and its map estimate portrayed with (□). (a) PHD 2.0a. (b) PHD 2.0b. (c) PHD 3.0.

TABLE II  
PERFORMANCE SUMMARY USING SYNTHETIC DATA

Filter version	$N$	Pos. $\pm 1$ std. [m]	Head. $\pm 1$ std. [deg]	GOSPA $\pm 1$ std. [m]	ESS [%]	Resampling [%]	Time $\pm 1$ std. [ms]
PHD 1.0	1	$254.21 \pm 169.25$	$30.62 \pm 30.28$	$229.48 \pm 8.23$	N/A	0.00	$0.07 \pm 0.06$
PHD 2.0a	1	$51.45 \pm 34.77$	$6.12 \pm 6.12$	$217.32 \pm 17.29$	N/A	0.00	$0.10 \pm 0.03$
PHD 2.0b	1	$29.69 \pm 20.73$	$3.49 \pm 3.49$	$189.41 \pm 32.93$	N/A	0.00	$0.26 \pm 0.11$
PHD 3.0	1	$3.45 \pm 2.45$	$0.42 \pm 0.42$	$46.84 \pm 22.87$	N/A	0.00	$0.14 \pm 0.03$
Oracle RBPF	1	$2.59 \pm 1.68$	$0.33 \pm 0.33$	$29.36 \pm 16.47$	N/A	0.00	N/A
PHD 1.0	10	$21.51 \pm 14.72$	$2.62 \pm 2.61$	$169.62 \pm 41.14$	13.65	99.92	$0.59 \pm 0.09$
PHD 2.0a	10	$3.57 \pm 2.24$	$0.42 \pm 0.42$	$45.26 \pm 20.97$	18.62	99.38	$0.65 \pm 0.11$
PHD 2.0b	10	$3.61 \pm 2.76$	$0.43 \pm 0.43$	$41.97 \pm 25.65$	52.14	7.43	$2.28 \pm 1.06$
PHD 3.0	10	$2.66 \pm 1.83$	$0.34 \pm 0.34$	$38.27 \pm 15.42$	49.34	5.14	$1.17 \pm 0.28$
Oracle RBPF	10	$2.18 \pm 1.40$	$0.29 \pm 0.29$	$24.41 \pm 14.26$	47.72	2.43	N/A
PHD 1.0	100	$3.09 \pm 1.93$	$0.36 \pm 0.36$	$39.50 \pm 17.89$	5.25	99.95	$5.45 \pm 0.75$
PHD 2.0a	100	$2.64 \pm 1.70$	$0.30 \pm 0.30$	$33.96 \pm 15.80$	11.48	99.84	$5.95 \pm 0.78$
PHD 2.0b	100	$2.31 \pm 1.54$	$0.30 \pm 0.30$	$32.44 \pm 14.68$	50.94	11.59	$22.16 \pm 10.81$
PHD 3.0	100	$2.18 \pm 1.42$	$0.29 \pm 0.29$	$32.12 \pm 12.92$	49.21	9.42	$10.94 \pm 2.58$
Oracle RBPF	100	$2.17 \pm 1.45$	$0.29 \pm 0.29$	$23.52 \pm 15.14$	46.66	5.26	N/A

The last column indicates the time required to perform one recursion of the filter.

Comprehensive understanding of the underlying uncertainties supports inference, which in turn can be used to improve the overall performance of PHD-SLAM as we have demonstrated above.

2) *Filter Performance*: Resampling is a crucial step of most PFs to limit degeneracy of the algorithm [32], and as summarized in Table II, the performance of all filters improves when resampling is enabled. Most of the particles sampled from the proposals of PHD 1.0 and PHD 2.0a have an insignificant weight as indicated by the low ESS and as a result, these filters rely on resampling at nearly every time step to avoid sample degeneracy. ESS of PHD 2.0b and PHD 3.0 is significantly higher, and resampling is performed only at a fraction of times, which increases the probability to close larger loops effectively. In addition, more efficient use of particles enables the proposed system to achieve comparative performance and a reduced computational overhead, or higher accuracy and a comparative run time with respect to the benchmark PHD-SLAM filters.

To investigate RBPF-based SLAM performance with perfect DAs, we implement an Oracle RBPF algorithm that knows the correspondences between measurements and landmarks. The implemented algorithm is similar to FastSLAM 2.0 [31] with the difference that the used sampling distribution is replaced with the IPL-based OID approximation approach [22]. The reason

why we implement FastSLAM 2.0 is that PHD-SLAM does not offer any advantage over FastSLAM 2.0 if the correspondences are known since clutter, detections, misdetections, and DAs can be discarded. The results of the Oracle RBPF are tabulated in Table II and interestingly, PHD 3.0 nearly achieves the same performance and the difference between the two diminishes as  $N$  grows. Already with 100 particles, PHD 3.0 and Oracle RBPF yield comparative tracking accuracy but the mapping accuracy of PHD 3.0 is worse. The reason is that PHD 3.0 occasionally has a missed or false landmark in the final map and the GOSPA metric penalizes heavily for such cardinality errors. For Oracle RBPF, the cardinality error is always zero since the DAs are known. Since PHD 3.0 has comparative accuracy to the Oracle RBPF, it can be concluded that the PHD-SLAM filter combined with the proposed GM-ID fulfills the objectives of probabilistic SLAM, that is, accurately estimate the joint posterior density of the map and robot trajectory.

3) *Computational Complexity*: Let  $M = |\mathcal{M}_{\text{FOV}}|$  denote the number of landmarks within the FOV and let us assume each landmark is associated with a measurement for simplicity. The joint importance density approximation utilized by PHD 2.0b has complexity  $\mathcal{O}(M^3)$ , whereas the partitioned update proposed in this article scales linearly as  $\mathcal{O}(M)$ . Moreover, the joint approximation requires  $(\dim(\mathbf{x}) + \dim(\mathbf{m}) \times M)^2 =$

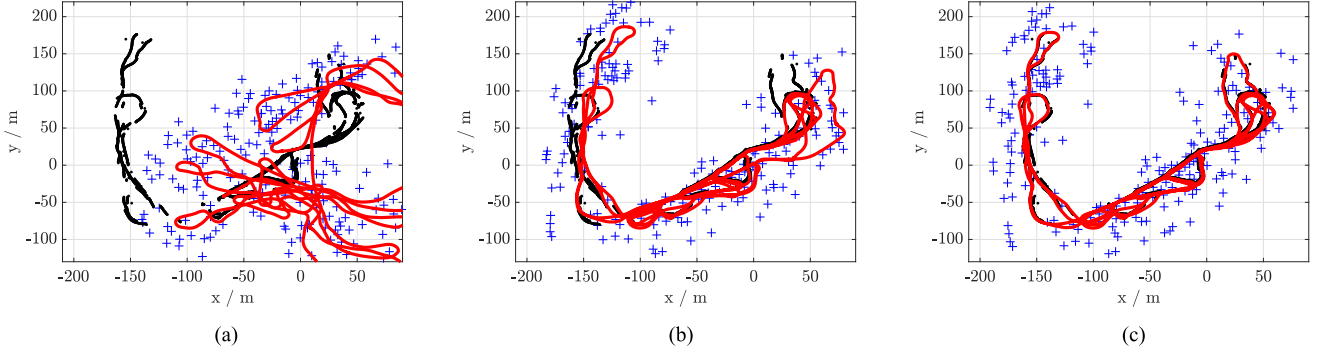


Fig. 4. Filter performance using one particle with the benchmark Victoria Park dataset. The GPS coordinates illustrated using (—), the estimated trajectory with (—), and the map estimate portrayed with (+). (a) PHD-SLAM 2.0a. (b) PHD-SLAM 2.0b. (c) PHD-SLAM 3.0.

$(3 + 2M)^2$  of memory, which must be allocated dynamically, whereas the partitioned update only requires  $3^2 + 2^2$  of static memory. The aforementioned memory requirements are only for the covariance matrix in (42) and similar memory requirements also hold for other parameters involved in the importance density approximation (see Algorithm 1). These beneficial features of the partitioned IPL importance density approximation yield to a notable reduction in computational complexity and on average, the partitioned approximation can be computed in  $14.0 \mu\text{s}$  for a single particle and GM component, whereas the joint approximation takes  $173.4 \mu\text{s}$ . As a result, even though the proposed approach considers multiple DAs in the GM-ID, the overall filtering algorithm is computationally more efficient than PHD 2.0b, as tabulated in Table II. With respect to PHD 1.0 and PHD 2.0a, the higher computational cost of PHD 3.0 can be easily justified since the number of particles can be reduced notably resulting in a much more efficient algorithm.

### B. Victoria Park Dataset

The algorithms are also tested on a benchmark SLAM dataset collected with an instrumented vehicle covering a distance of over 4 km in Victoria Park, Sydney [10]. The vehicle is equipped with a laser rangefinder, encoders, and GPS. The laser scans are used to obtain range and bearing measurements to nearby trees and the encoders are used to measure velocity and steering angle of the vehicle. The GPS is used for evaluation purposes only and it provides ground truth for the vehicle trajectory, while no ground truth data are available for the locations of the landmarks. The reader is referred to [10] and [50] for further details on the dataset and used models. To make the scenario more challenging, artificial clutter is added to the original dataset to make the detection conditions less ideal as in [33]. The clutter measurements follow a Poisson distribution with clutter intensity  $\lambda = 5$  and the false measurements are uniformly distributed inside the FOV. In the experiment, the FOV is limited to  $\pm 85^\circ$  scanning angle and 50 m scanning range, and the detection probability is approximated using  $p_D = 0.7(1 - \|\hat{\mathbf{m}} - \mathbf{x}_k\|/50)$  within the FOV. The filters use  $\mathbf{Q} = \text{diag}([1^2 \text{ [m/s]}^2, (\frac{4\pi}{180})^2 \text{ [rad/s]}^2])$  and  $\mathbf{R} = \text{diag}([1^2 \text{ [m]}^2, (\frac{1\pi}{180})^2 \text{ [rad]}^2])$  for the control and measurement covariances, respectively. To demonstrate the capability of

TABLE III  
PERFORMANCE SUMMARY USING THE VICTORIA PARK DATASET

Filter version	$N$	Pos. [m]	Resampling [%]	Time [ms]
PHD 1.0	1	154.49	0.00	0.23
PHD 2.0a	1	145.91	0.00	0.32
PHD 2.0b	1	5.90	0.00	0.54
PHD 3.0	1	3.57	0.00	0.40
PHD 1.0	5	6.06	40.54	0.67
PHD 2.0a	5	5.30	34.00	0.83
PHD 2.0b	5	3.68	0.52	2.11
PHD 3.0	5	3.38	0.49	1.79
PHD 1.0	10	5.41	25.18	1.17
PHD 2.0a	10	5.40	20.84	1.57
PHD 2.0b	10	3.61	0.41	4.06
PHD 3.0	10	3.36	0.39	3.41

The last column indicates the time of one filter recursion.

PHD 2.0b and PHD 3.0 to close large loops, the ESS threshold to perform resampling is set to  $T_{\text{ESS}} = 1$  for these two filters in the Victoria Park experiments.

Fig. 4 illustrates a typical estimation result of the filters, and Table III summarizes performance of the algorithms. The results are inline with the simulations, PHD 1.0 and PHD 2.0a<sup>3</sup> require a high number of particles to obtain satisfactory performance, and PHD 2.0b outperforms the two but is prone to diverge if a low number of particles are used. PHD 3.0 is robust to data ambiguities and results to the best overall performance. It is to be noted that correct DA is very challenging in the considered scenario and PHD 3.0 confronts this difficulty by using more components in the GM-ID approximation, which increases computational overhead of the algorithm. However, the small number of particles required by PHD-SLAM 3.0 translates to a highly efficient algorithm. While the data acquisition phase in the Victoria Park dataset required 1549 s, the proposed algorithm with one particle only takes 2.90 s to run, which is only 0.19% of the experiment duration.

<sup>3</sup>The reference implementation of PHD2.0a, which is available at [54], does not sample the particles according to [35, eq. (51)] but uses the mean of the importance density instead. In this article, we have implemented PHD2.0a as presented in [35] and how PFs typically operate, that is, sampling is performed by drawing random samples from the approximated importance density. In addition, we have added artificial clutter to the measurements and therefore, the presented results for PHD 2.0a differ significantly from the ones presented in [35].

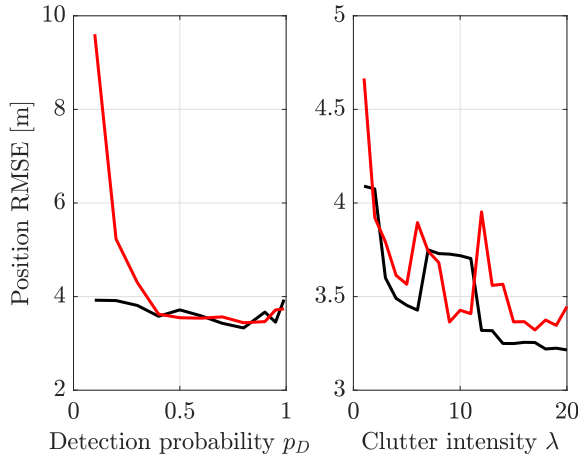


Fig. 5. Sensitivity analysis using the synthetic (—) and Victoria Park (—) datasets. (a) RMSE as a function of detection probability. (b) RMSE as a function of clutter intensity. In (b), the large changes in RMSE for  $\lambda \geq 5$  are caused by a single MCS, which has much higher RMSE with some specific  $\lambda$  values.

### C. Discussion

The proposed PHD filter has numerous tuning parameters that provide a tradeoff between accuracy and computational complexity. These tuning parameters include the gating threshold, hypotheses pruning parameters, and IPL parameters. We have set the parameters such that the values provide a good compromise between accuracy and computational complexity. For example, increasing  $\tilde{\Gamma}$  improves the accuracy up to some limit, but at the same time, the computational complexity increases. We have set the value such that it provides fairly low computational overhead and still high enough so that the performance cannot be improved using an even higher value ( $\tilde{\Gamma} > 50$ ). The other tuning parameters behave in the same manner. Either increasing or decreasing the parameter value improves the accuracy up to some limit but it also increases the computational overhead. The values of the other parameters were adjusted to achieve good filter performance and moderate computational overhead.

There are also models and parameters that are not precisely known in real-world experiments. As an example, the detection statistics are unknown for the Victoria Park dataset since it is a function of the used sensor as well as the environment. In this work, the detection statistics are approximated using the probability of detection and clutter intensity parameters. Since these two parameters are also very central for the PHD filter, we have analyzed sensitivity of the proposed algorithm to changes in these parameters for which the results are illustrated in Fig. 5. As shown, the algorithm is not very sensitive to changes in either of the parameters and good accuracy is achieved when  $p_D \geq 0.4$  and  $\lambda \geq 5$ . It is to be noted that in the Victoria Park experiment, the used model for the detection probability is sufficient to obtain good accuracy but it has certain limitations since it does not account for obstructions created by other landmarks and vegetation. Using more accurate detection statistics, which is able to take occlusions into account [55], it is possible to further improve the filter performance, which we will explore in future research.

The most significant weakness of the proposed algorithm is inconsistency of the filter, which is mostly caused by linearization errors combined with resampling. The problem is that whenever resampling is performed, an entire trajectory and map hypotheses are lost permanently for those particles that are not selected. This depletes the number of samples representing past poses and consequently erodes the statistics of the landmark estimates conditioned on these past poses. After resampling has been performed enough times, all of the particles will share a single mutual ancestor and estimate diversity beyond this point is lost forever. As a consequence, the loss of particle diversity can prevent a consistent long-term estimate of the joint RFS-SLAM density. The very same problem is encountered by FastSLAM [56] and, even though not reported in literature, other PHD-SLAM filters. To prolong the time-period over which RBPF-SLAM solutions are reasonably consistent, it is necessary to reduce the impact of resampling [56], [57]. In this regard, the proposed algorithm's ability to constrain uncertainty of the posterior approximation at a much lower level and the ability to achieve good performance even without resampling can be seen as a major advantage.

Most filtering-based SLAM methods have been shown to be inconsistent [56], [58]. Fundamentally, inconsistency is caused by the filter's inability to reflect the unobservable degrees of freedom of SLAM and the filter tends to erroneously acquire information along the directions spanned by these unobservable states [59]. These erroneous updates cannot be undone in the filtering framework and therefore, more robust SLAM solutions consider the smoothing version of the SLAM problem instead [7], [8], [27], [28], [29]. A smoothing approach to SLAM involves not just the most current robot location, but the entire robot trajectory up to the current time. A significant advantage of the smoothing approach is that the problem can be relinearized and wrong DAs can be corrected in the SLAM front-end. An interesting comparison between PHD-SLAM and GraphSLAM is provided in [30] in which it is shown that PHD-SLAM can outperform GraphSLAM in scenarios in which DA is very challenging. Moreover, the authors also present the smoothing version of PHD-SLAM called Loopy PHD-SLAM that combines beneficial properties of both SLAM methods. As the authors discuss in [30, Ch. 3.3], the PHD-SLAM method is not very efficient and it would benefit from a better importance density approximation and the GM-ID proposed in this article is a good alternative. Moreover, in the GraphSLAM front-end one needs to solve the DA problem [8] and the method proposed in this article presents one alternative to model DA uncertainty and for solving the DA problem.

## VII. CONCLUSION

This article presents an MH-ID approximation to improve the efficiency and robustness of landmark-based SLAM. By modeling the measurements and landmarks as RFSs, an MH-ID that incorporates DA uncertainties is derived and a tractable GM-ID approximation is presented that can be sampled from in closed form. In the proposed GM-ID, each mixture component represents a single DA and the GM representation allows

incorporating multiple DAs into the importance density approximation. An iterative method based on using generalized SLR combined with IPL is used to compute parameters of the GM components and a partitioned update strategy that exploits the measurement model structure is developed.

Analysis was carried out both in a simulated environment through MCS and an experimental outdoor Victoria Park SLAM dataset. Results demonstrated the accuracy and robustness of the proposed method, as well as, the efficiency of the developed PHD-SLAM filter. In fact, it is demonstrated that a single particle is sufficient to generate an accurate map of the environment even in high clutter scenarios. The newly proposed GM-ID approximation admits numerous possibilities of future research into other RBPF-based SLAM approaches or enhancements via other RFS-SLAM filters. For example, the PMBM filter is expected to improve the robustness to clutter and data ambiguities, improving the overall performance even more. Moreover, the proposed GM-ID approximation could be directly or with small modifications used in any particle filtering SLAM approach that utilize an importance density approximation similar to the one used in FastSLAM 1.0 and 2.0. As an example, Pareja [30] discussed that the PHD filter for visual SLAM is not very efficient and the method would benefit using a better importance density. In future work, the proposed method will be evaluated with other experimental datasets and the algorithm will be applied to other SLAM approaches, such as visual SLAM.

#### APPENDIX

This Appendix provides a more detailed derivation of the approximation of  $g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k})$  used in the approximation of the MH-ID. First, recall that

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \\ = \int g(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_k) p(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \delta \mathcal{M}_k. \end{aligned}$$

Next, we plug in (4) for  $g(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_k)$ ,  $p(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k})$  is approximated using (16) and (19), and since the cardinality is known for each DA, we get

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \\ = \int \sum_{t=1}^{\Gamma} \exp(-\lambda_c) \prod_{j=1}^{|\mathcal{Z}|} c(\mathbf{z}_k^j) \prod_{i:\phi_k^{t,i}=0} [1 - p_D(\mathbf{m}^i | \mathbf{x}_k)] \\ \times \prod_{i:\phi_k^{t,i}>0} \frac{p_D(\mathbf{m}^i | \mathbf{x}_k) g(\mathbf{z}_k^{\phi_k^{t,i}} | \mathbf{m}^i, \mathbf{x}_k)}{c(\mathbf{z}_k^{\phi_k^{t,i}})} \\ \times \prod_{i=1}^{M_k} \eta_{k|k-1}^i \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^1 \dots d\mathbf{m}^{M_k}. \end{aligned}$$

Regrouping the landmarks  $\mathbf{m}^i$  from the prior with the  $\mathbf{m}^i$  from the likelihood and moving terms independent of any  $\mathbf{m}^i$  out of the integral yields

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \\ = \sum_{t=1}^{\Gamma} \exp(-\lambda_c) \prod_{j=1}^{|\mathcal{Z}|} c(\mathbf{z}_k^j) \\ \times \int \prod_{i:\phi_k^{t,i}=0} [1 - p_D(\mathbf{m}^i | \mathbf{x}_k)] \\ \times \eta_{k|k-1}^i \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) \\ \times \prod_{i:\phi_k^{t,i}>0} \frac{p_D(\mathbf{m}^i | \mathbf{x}_k) g(\mathbf{z}_k^{\phi_k^{t,i}} | \mathbf{m}^i, \mathbf{x}_k)}{c(\mathbf{z}_k^{\phi_k^{t,i}})} \\ \times \eta_{k|k-1}^i \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^1 \dots d\mathbf{m}^{M_k}. \end{aligned}$$

Noting that any landmark  $\mathbf{m}^i$  is either detected ( $\phi_k^{t,i} > 0$ ) or misdetected ( $\phi_k^{t,i} = 0$ ), the integral over the products can be written as

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \\ = \sum_{t=1}^{\Gamma} \exp(-\lambda_c) \prod_{j=1}^{|\mathcal{Z}|} c(\mathbf{z}_k^j) \\ \times \prod_{i:\phi_k^{t,i}=0} \int [1 - p_D(\mathbf{m}^i | \mathbf{x}_k)] \\ \times \eta_{k|k-1}^i \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^i \\ \times \prod_{i:\phi_k^{t,i}>0} \int \frac{p_D(\mathbf{m}^i | \mathbf{x}_k) g(\mathbf{z}_k^{\phi_k^{t,i}} | \mathbf{m}^i, \mathbf{x}_k)}{c(\mathbf{z}_k^{\phi_k^{t,i}})} \\ \times \eta_{k|k-1}^i \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^i. \end{aligned}$$

Finally, rearranging the integrals yields

$$\begin{aligned} g(\mathcal{Z}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \\ = \sum_{t=1}^{\Gamma} \exp(-\lambda_c) \prod_{j=1}^{|\mathcal{Z}|} c(\mathbf{z}_k^j) \\ \times \prod_{i:\phi_k^{t,i}=0} \eta_{k|k-1}^i \int [1 - p_D(\mathbf{m}^i | \mathbf{x}_k)] \\ \times \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^i \\ \times \prod_{i:\phi_k^{t,i}>0} \frac{\eta_{k|k-1}^i}{c(\mathbf{z}_k^{\phi_k^{t,i}})} \int p_D(\mathbf{m}^i | \mathbf{x}_k) g(\mathbf{z}_k^{\phi_k^{t,i}} | \mathbf{m}^i, \mathbf{x}_k) \\ \times \mathcal{N}(\mathbf{m}^i; \hat{\mathbf{m}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) d\mathbf{m}^i \end{aligned}$$

which is (36).



## REFERENCES

- [1] R. Smith, M. Self, and P. Cheeseman, *Estimating Uncertain Spatial Relationships in Robotics*. New York, NY, USA: Springer, 1990.
- [2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Cambridge, MA, USA: MIT Press, 2005.
- [3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
- [4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.
- [5] C. Stachniss, D. Hahnel, and W. Burgard, "Exploration with active loop-closing for FastSLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, pp. 1505–1510.
- [6] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
- [7] S. Thrun and M. Montemerlo, "The Graph SLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Robot. Res.*, vol. 25, no. 5–6, pp. 403–429, 2006.
- [8] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Feb. 2011.
- [9] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.
- [10] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 242–257, Jun. 2001.
- [11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. 18th Nat. Conf. Artif. Intell.*, 2002, pp. 593–598.
- [12] J. Neira and J. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 890–897, Dec. 2001.
- [13] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to Bayesian SLAM," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 268–282, Apr. 2011.
- [14] J. Mullane, B.-N. Vo, M. D. Adams, and W. S. Wijesoma, "A random set formulation for Bayesian SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 1043–1049.
- [15] H. Deusch, S. Reuter, and K. Dietmayer, "The labeled multi-Bernoulli SLAM filter," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1561–1565, Oct. 2015.
- [16] D. Moratuwage, M. Adams, and F. Inostroza, " $\delta$ -generalised labelled multi-Bernoulli simultaneous localisation and mapping," in *Proc. Int. Conf. Control Autom. Inf. Sci.*, 2018, pp. 175–182.
- [17] Y. Ge, H. Kim, F. Wen, L. Svensson, S. Kim, and H. Wymeersch, "Exploiting diffuse multipath in 5G SLAM," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.
- [18] H. Kim, K. Granström, L. Svensson, S. Kim, and H. Wymeersch, "PMBM-based SLAM filters in 5G mmWave vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8646–8661, Aug. 2022.
- [19] K. P. Murphy, "Bayesian map learning in dynamic environments," in *Proc. 12th Int. Conf. Neural Inf. Process. Syst.*, 1999, pp. 1015–1021.
- [20] O. Kaltiokallio et al., "Towards real-time Radio-SLAM via optimal importance sampling," in *Proc. IEEE 23rd Int. Workshop Signal Process. Adv. Wireless Commun.*, 2022, pp. 1–5.
- [21] A. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior linearization filter: Principles and implementation using sigma points," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5561–5573, Oct. 2015.
- [22] R. Hostettler, F. Tronarp, A. F. García-Fernández, and S. Särkkä, "Importance densities for particle filtering using iterated conditional expectations," *IEEE Signal Process. Lett.*, vol. 27, pp. 211–215, Dec. 2020.
- [23] J. Nieto, J. Guivant, E. Nebot, and S. Thrun, "Real time data association for FastSLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 412–418.
- [24] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," *Int. J. Robot. Res.*, vol. 32, no. 7, pp. 826–840, 2013.
- [25] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1722–1729.
- [26] K. J. Doherty, D. P. Baxter, E. Schneeweiss, and J. J. Leonard, "Probabilistic data association via mixture models for robust semantic SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 1098–1104.
- [27] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.
- [28] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.
- [29] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3281–3288.
- [30] A. Falchetti Pareja, "Random finite sets in visual SLAM," Master's thesis, Elect. Eng. Dept., Universidad de Chile, Santiago, Chile, 2017.
- [31] M. Montemerlo, S. Thrun, D. Roller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. 18th Int. Joint Conf. Artif. Intell.*, 2003, pp. 1151–1156.
- [32] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.
- [33] K. Y. K. Leung, F. Inostroza, and M. Adams, "Multifeature-based importance weighting for the PHD SLAM filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 6, pp. 2697–2714, Dec. 2016.
- [34] H. Kim, K. Granström, L. Gao, G. Battistelli, S. Kim, and H. Wymeersch, "5G mmWave cooperative positioning and mapping using multi-model PHD filter and map fusion," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3782–3795, Jun. 2020.
- [35] L. Gao, G. Battistelli, and L. Chisci, "PHD-SLAM 2.0: Efficient SLAM in the presence of misdetections and clutter," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1834–1843, Oct. 2021.
- [36] M. Adams, B.-N. Vo, R. Mahler, and J. Mullane, "SLAM gets a PHD: New concepts in map estimation," *IEEE Robot. Automat. Mag.*, vol. 21, no. 2, pp. 26–37, Jun. 2014.
- [37] R. P. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, 2014.
- [38] R. P. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
- [39] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. dissertation, Dept. Elect. Eng. Comp. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2001.
- [40] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.
- [41] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- [42] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Advanced Applications*. Norwood, MA, USA: Artech House, 1990.
- [43] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [44] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Handbook of Nonlinear Filtering*, vol. 12, D. Crisan and B. Rozovskii, Eds., Oxford, U.K.: Oxford Univ. Press, 2011, pp. 656–704.
- [45] C. S. Lee, D. E. Clark, and J. Salvi, "SLAM with dynamic targets via single-cluster PHD filtering," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 543–552, Jun. 2013.
- [46] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Res.*, vol. 16, no. 3, pp. 682–687, 1968.
- [47] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, no. 1–2, pp. 83–97, 1955.
- [48] B.-N. Vo, B.-T. Vo, and D. Phung, "Labeled random finite sets and the Bayes multi-target tracking filter," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6554–6567, Dec. 2014.
- [49] I. Cox and M. Miller, "On finding ranked assignments with application to multitarget tracking and motion correspondence," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 1, pp. 486–489, Jan. 1995.
- [50] "Victoria park SLAM data set," J. Guivant, Australian centre for field robotics - The University of Sydney. Accessed: Aug. 11, 2022. [Online]. Available: [http://www-personal.acfr.usyd.edu.au/nebot/victoria\\_park.htm](http://www-personal.acfr.usyd.edu.au/nebot/victoria_park.htm)

- [51] C. Kim, R. Sakthivel, and W. K. Chung, "Unscented FastSLAM: A robust and efficient solution to the SLAM problem," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 808–820, Aug. 2008.
- [52] A. S. Rahmattullah, A. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *Proc. IEEE 20th Int. Conf. Inf. Fusion*, 2017, pp. 1–8.
- [53] M. Montemerlo, "FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, Jul. 2003.
- [54] "Matlab implementation of PHD-SLAM 2.0," Lin Gao, *School of Information and Communication Engineering, Univ. Electron. Sci. Technol. China*, Chengdu, China. Accessed: Jun. 27, 2023. [Online]. Available: <https://github.com/gao12345ab/PHD-SLAM-2.0>
- [55] F. Inostroza, M. Adams, and K. Leung, "Modeling detection statistics in feature-based robotic navigation for range sensors," *Navigation*, vol. 65, no. 3, pp. 297–318, 2018.
- [56] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 424–429.
- [57] L. Zhang, X. jiong Meng, and Y. wu Chen, "Convergence and consistency analysis for FastSLAM," in *Proc. IEEE Intell. Veh. Symp.*, 2009, pp. 447–452.
- [58] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 3562–3568.
- [59] M. Brossard, A. Barrau, and S. Bonnabel, "Exploiting symmetries to design EKFs with consistency properties for navigation and SLAM," *IEEE Sensors J.*, vol. 19, no. 4, pp. 1572–1579, Feb. 2019.



**Ossi Kaltio kallio** (Member, IEEE) received the Ph.D. degree in electrical engineering from Aalto University, Espoo, Finland, in 2017.

He has held Postdoctoral Fellow position with the University of Utah, Aalto University, and Tampere University. He is currently a Senior Research Fellow with the Electrical Engineering Unit, Tampere University, Tampere, Finland. His current research interests include the intersection of statistical signal processing and wireless networking for improving radio-based positioning and sensing technologies.

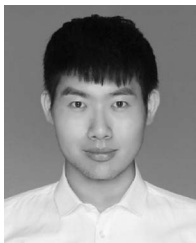


**Roland Hostettler** (Member, IEEE) received the Dipl. Ing. degree in electrical and communication engineering from the Bern University of Applied Sciences, Bern, Switzerland, in 2007, and the M.Sc. degree in electrical engineering and the Ph.D. degree in automatic control from the Luleå University of Technology, Luleå, Sweden, in 2009 and 2014, respectively.

He has held Postdoctoral Researcher positions with the Luleå University of Technology, Sweden, and Aalto University, Finland. He is currently an Associate

Professor with the Department of Electrical Engineering, Uppsala University, Uppsala, Sweden. His research interests include statistical signal processing and sensor fusion and their applications.

Dr. Hostettler is a Member of the IEEE Machine Learning for Signal Processing Technical Committee.



**Yu Ge** (Graduate Student Member, IEEE) received the B.E. degree from Zhejiang University, Hangzhou, China, in 2017, and the M.Sc. degree from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2019, both in electrical engineering. He is currently working toward the Ph.D. degree in electrical engineering with the Department of Electrical and Engineering, Chalmers University of Technology, Gothenburg, Sweden.

His research interests include integrated communication and sensing, wireless positioning systems, simultaneous localization and mapping, and multi-object tracking, particularly in 5G and beyond 5G scenarios.



**Hyowon Kim** (Member, IEEE) received the Ph.D. degree in department of electronics engineering from the Department of Electronic Engineering, Hanyang University, Seoul, South Korea, in 2021.

From 2019 to 2020, he was a Visiting Researcher with the Department of Electrical Engineering, Chalmers University of Technology, Sweden, where from 2021 to 2023, he was a Marie Skłodowska-Curie Fellow/Postdoctoral Researcher. He is currently an Assistant Professor with the Department of Electronics Engineering, Chungnam National University, Daejeon, South Korea. His main research interests include wireless communications and integrated sensing/localization/communications, in 5G and beyond 5G communication systems.



**Jukka Talvitie** (Member, IEEE) received the M.Sc. degree in automation engineering and the Ph.D. degree in computing and electrical engineering from the Tampere University of Technology, Tampere, Finland, in 2008 and 2016, respectively.

He is currently a University Lecturer with the Unit of Electrical Engineering, Tampere University, Tampere. His research interests include signal processing for wireless communications, radio-based positioning and sensing, radio link waveform design, and radio system design, particularly concerning 5G and beyond mobile technologies.



**Henk Wymeersch** (Fellow, IEEE) received the Ph.D. degree in electrical engineering/applied sciences from Ghent University, Ghent, Belgium, in 2005.

He is currently a Professor of communication systems with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. His current research focuses on the convergence of communication and sensing, in a 5G and beyond 5G context.

Dr. Wymeersch is currently a Senior Member of the IEEE Signal Processing Magazine Editorial Board.

During 2019–2021, he was an IEEE Distinguished Lecturer with the Vehicular Technology Society.



**Mikko Valkama** (Fellow, IEEE) received the M.Sc. (Tech.) and D.Sc. (Tech.) degrees (both with honors) in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 2000 and 2001, respectively.

In 2003, he was with the Communications Systems and Signal Processing Institute, SDSU, San Diego, CA, as a Visiting Research Fellow. He is currently a Full Professor and the Head of the Unit of Electrical Engineering with the newly formed Tampere University, Tampere. His research interests include radio communications, radio localization, and radio-based sensing, with particular emphasis on 5G and 6G mobile radio networks.