A Belief Propagation Algorithm for Multipath-Based SLAM

Erik Leitinger, *Member, IEEE*, Florian Meyer, *Member, IEEE*, Franz Hlawatsch, *Fellow, IEEE*, Klaus Witrisal, *Member, IEEE*, Fredrik Tufvesson, *Fellow, IEEE*, and Moe Z. Win, *Fellow, IEEE*

Abstract

We present a simultaneous localization and mapping (SLAM) algorithm that is based on radio signals and the association of specular multipath components (MPCs) with geometric features. Especially in indoor scenarios, *robust* localization from radio signals is challenging due to diffuse multipath propagation, unknown MPC-feature association, and limited visibility of features. In our approach, specular reflections at flat surfaces are described in terms of virtual anchors (VAs) that are mirror images of the physical anchors (PAs). The positions of these VAs and possibly also of the PAs are unknown. We develop a Bayesian model of the SLAM problem and represent it by a factor graph, which enables the use of belief propagation (BP) for efficient marginalization of the joint posterior distribution. The resulting BP-based SLAM algorithm detects the VAs associated with the PAs and estimates jointly the time-varying position of the mobile agent and the positions of the VAs and possibly also of the PAs, thereby leveraging the MPCs in the radio signal for improved accuracy and robustness of agent localization. The algorithm has a low computational complexity and scales well in all relevant system parameters. Experimental results using both synthetic measurements and real ultra-wideband radio signals demonstrate the excellent performance of the algorithm in challenging indoor environments.

Index Terms

Simultaneous localization and mapping, SLAM, multipath channel, data association, factor graph, message passing, sum-product algorithm.

I. Introduction

The goal of simultaneous localization and mapping (SLAM) [1], [2] is to estimate the time-varying pose of a mobile agent—including the agent's position—and a map of the surrounding

E. Leitinger and F. Tufvesson are with the Department of Electrical and Information Technology, Lund University, Lund, Sweden (e-mail: {erik.leitinger, fredrik.tufvesson}@eit.lth.se). F. Meyer and M. Win are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA (e-mail: {fmeyer,moewin}@mit.edu). F. Hlawatsch is with the Institute of Telecommunications, TU Wien, Vienna, Austria and with Brno University of Technology, Brno, Czech Republic (e-mail: franz.hlawatsch@tuwien.ac.at). K. Witrisal is with the Christian Doppler Laboratory for location-aware electronic systems, Graz University of Technology, Graz, Austria, (e-mail: witrisal@tugraz.at). This work was supported in part by the Austrian Science Fund (FWF) under grants J 4027, J 3886, and P27370-N30, by the U.S. Department of Commerce, National Institute of Standards and Technology under Grant 70NANB17H17, and by the Czech Science Foundation (GAČR) under grant 17-19638S. Parts of this paper were presented at IEEE ICC, ANLN Workshop 2017, Paris, France, June 2017.

environment, from measurements provided by one or multiple sensors. SLAM is important in many fields including robotics [1], autonomous driving [3], location-aware communication [4], and robust indoor localization [5]–[9]. Achieving a required level of accuracy robustly is still elusive in indoor environments characterized by harsh multipath channel conditions. Therefore, most existing systems supporting multipath channels either use sensing technologies that mitigate multipath effects [10] or fuse multiple information sources [11], [12].

In multipath-assisted indoor localization [5], [13], [14], the relation of multipath components (MPCs) with the local geometry potentially turns multipath propagation from an impairment into an advantage. This paper presents a SLAM algorithm for robust indoor localization based on radio signals. The radio signals are transmitted from a mobile agent to base stations, called physical anchors (PAs). MPCs due to specular reflections are modeled by virtual anchors (VAs), which are mirror images of the PAs [15]. Our algorithm detects the VAs associated with the PAs and estimates the VA positions and possibly also the PA positions jointly with the time-varying position of the mobile agent. The algorithm is designed to cope with harsh multipath channel conditions, which tend to lead to measurements with a high level of false alarms and missed detections. While MPCs can be generated by various propagation phenomena such as specular reflections, scattering, and diffraction, our model focuses on PA/VA-related MPCs; all the other MPCs are treated as interference, even if they contain geometric information. We note that MPCs associated with scatter points are also considered in the feature model used in [13].

A. Feature-based SLAM

The proposed algorithm follows the feature-based approach to SLAM [2], [16], [17]. The map is represented by an unknown number of *features* with unknown spatial positions, whose states are estimated in a sequential (time-recursive) way. In our model, the features are given by the PAs and VAs. Prominent feature-based SLAM algorithms are extended Kalman filter SLAM (EKF-SLAM) [16], Rao-Blackwellized (RB)-SLAM (dubbed FastSLAM) [2], [18], variational-inference-based SLAM [19], [20], and set-based SLAM [17], [21], [22]. Recently, feature-based SLAM methods that exploit position-related information in radio signals were introduced [13], [23], [24]. Most of these methods operate on estimated parameters related to MPCs, such as distances (which are proportional to delays), angles-of-arrival (AoAs), or angles-of-departure (AoDs) [25]–[29]. These parameters are estimated from the signal in a preprocessing stage and are considered as "measurements" by the SLAM method. An important aspect is the *data*

association (DA) between these measurements and the PAs or VAs.

Feature-based SLAM is closely related to multitarget tracking (MTT), and MTT methods have been adapted to feature-based SLAM [17], [21], [30]. MTT methods that are applicable to SLAM include the joint probabilistic DA (JPDA) filter [31] and the joint integrated probabilistic DA (JIPDA) filter [32]. An approach similar to the JIPDA filter is taken by the methods presented in [33], [34]. More recently, the use of belief propagation (BP) [35], [36] was introduced for probabilistic DA within MTT in [37] and for multisensor MTT in [38]–[40]. In particular, the BP algorithms in [39], [40] are based on a factor graph representation of the multisensor MTT problem and have a computational complexity that scales only quadratically with the number of objects (targets) and linearly with the number of sensors. MTT methods that are based on random finite sets and embed a BP algorithm for probabilistic DA were presented in [41], [42]. We finally note that our approach to feature-based SLAM is also related to multisensor target tracking with uncertain sensor locations using Bayesian methods [43].

B. Contributions and Organization of the Paper

Here, we propose a BP-based, Bayesian detection and estimation algorithm for SLAM using radio signals. Our algorithm jointly performs probabilistic DA and sequential estimation of the states of a mobile agent and of "potential features" (PFs) characterizing the map. We use a probabilistic model for feature existence where each PF state is augmented by a binary existence variable and associated with a *probability of existence*, which is also estimated. The proposed algorithm is inspired by the BP algorithms for multisensor MTT presented in [39], [40], and will hence be briefly referred to as *BP-SLAM* algorithm. Probabilistic DA and state estimation are performed by running BP on a factor graph [35], [36] representing the statistical structure of the SLAM problem. The BP approach leverages conditional statistical independencies to achieve low complexity and high scalability. In fact, in contrast to conventional SLAM algorithms such as EKF-SLAM and RB-SLAM, the proposed BP-SLAM algorithm assumes that in each time step the agent state and all the feature states are a priori independent.

Our probabilistic model for DA and feature existence uncertainty allows the BP-SLAM algorithm to succeed in the particularly challenging range-only SLAM problem [44], which cannot be addressed straightforwardly by established SLAM techniques [1], [2]. Performing SLAM only from range measurements is challenging because (i) when new features are initialized, the probability distributions of the PAs/VAs are annularly shaped and thus cannot be well represented

by a Gaussian distribution; and (ii) DA is more difficult compared to classical SLAM problems in robotics since PAs/VAs at widely different positions can generate almost identical range measurements. In particular, the algorithm of [44] is able to cope with a range-only measurement model, but not with DA uncertainty. To the best of our knowledge, our algorithm—along with a preliminary version presented in [30]—is the first BP algorithm for feature-based SLAM with probabilistic DA that is also suitable for range-based SLAM.

Key innovative contributions of this paper include the following:

- We establish a Bayesian model for feature-based SLAM that uses MPC parameters extracted from radio signals as input measurements and models probabilistically the appearance and disappearance of PAs/VAs as well as the DA uncertainty.
- Based on a factor graph representation of this model, we develop a scalable BP algorithm that estimates the state of the mobile agent and the numbers and positions of PAs/VAs.
- We evaluate the performance of the proposed algorithm on synthetic and real data. Our experimental results demonstrate the algorithm's high accuracy and robustness.

We apply the proposed BP-SLAM algorithm to the challenging setup of a range-only measurement model. However, bearing information (AoA and AoD of MPCs) or information derived from inertial measurement unit sensors can be easily incorporated in the BP-SLAM algorithm, and this would lead to a significant performance gain. For simplicity, we assume time synchronization between the PAs and the mobile agent. However, the BP-SLAM algorithm can be extended to nonsynchronized PA-agent links along the lines of [13] (based on the fact that the relevant geometric information is also contained in the time differences of the MPCs [14]) or to joint SLAM and synchronization along the lines of [45]. Furthermore, we assume that the probabilities with which the preliminary signal analysis stage (producing measurements) detects features in the radio signals are known; however, an adaptive extension to unknown and time-varying detection probabilities can be obtained along the lines of [30], [40], [46].

This paper advances over our conference paper [30] in that it replaces the heuristic used therein for determining the initial distribution of new PFs by an improved Bayesian technique. Furthermore, the factor graph and BP algorithm of [30] are extended by the introduction of "new PFs," i.e., features that generate measurements for the first time. The remainder of this paper is organized as follows. Section II considers the received radio signals and the MPC parameters. Section III describes the system model and provides a statistical formulation of the

SLAM problem. The posterior distribution of all states and the corresponding factor graph are derived in Section IV. In Section V, the proposed BP-SLAM algorithm is described. The results of numerical experiments are reported in Section VI. Section VII concludes the paper.

II. Radio Signal and MPC Parameters

Radio signal based SLAM [5], [13], [23], [24] associates specular MPC parameters estimated from the received radio signals with "geometrically expected" parameters, such as distances (corresponding to delays), AoAs, and AoDs. These parameters are modeled in terms of the positions of the mobile agent and of the PAs or VAs. The VA positions are mirror images of the PA positions that are induced by reflections at flat surfaces (typically walls) and thus depend on the surrounding environment (floor plan) [15]. For each reflection path, the length from the PA (via the flat surface) to the mobile agent is equal to the length from the VA to the mobile agent. Even though the mobile agent moves, the VAs remain static as long as the PAs and the flat surfaces (floor plan) are static. Note that the VA positions are unknown because the floor plan is unknown. An example is shown in Fig. 1, which depicts two PAs and some of the corresponding first-order VAs belonging to some larger flat surfaces. Also shown are two reflection paths (for the two PAs) for mobile agent position $\boldsymbol{p}_{350}$.

We consider a mobile agent with unknown time-varying position $\boldsymbol{p}_n \in \mathbb{R}^2$ and J PAs with possibly unknown positions $\boldsymbol{a}_{1,n}^{(j)} \in \mathbb{R}^2$, $j = 1, \ldots, J$, where J is assumed to be known. Associated with the jth PA, there are $L_n^{(j)} - 1$ VAs at unknown positions $\boldsymbol{a}_{l,n}^{(j)} \in \mathbb{R}^2$, $l = 2, \ldots, L_n^{(j)}$. The PAs and VAs will also be referred to as *features*. The number of features, $L_n^{(j)}$, is unknown and time-varying, and it depends on the agent position $\boldsymbol{p}_n$. We note that the PA and VA positions are allowed to be time-varying for the sake of generality; in an indoor scenario, they are typically static. In each discrete time slot n, the mobile agent transmits a radio signal $s(t)$ and the PAs act as receivers. However the proposed algorithm can be easily reformulated for the case where the PAs act as transmitters and the mobile agent acts as a receiver.

The baseband signal received by the jth PA is modeled as [14]

$$r_n^{(j)}(t) = \sum_{l=1}^{L_n^{(j)}} \alpha_{l,n}^{(j)} s\big(t - \tau_{l,n}^{(j)}\big) + d_n^{(j)}(t) + w(t). \tag{1}$$

Here, the first term on the right-hand side describes the contribution of $L_n^{(j)}$ specular MPCs with complex amplitudes $\alpha_{l,n}^{(j)}$ and delays $\tau_{l,n}^{(j)}$, where $l \in \mathcal{L}_n^{(j)} \triangleq \big\{1, \ldots, L_n^{(j)}\big\}$. These MPCs potentially correspond to features (PAs or VAs). The delays $\tau_{l,n}^{(j)}$ are proportional to the distances (ranges)

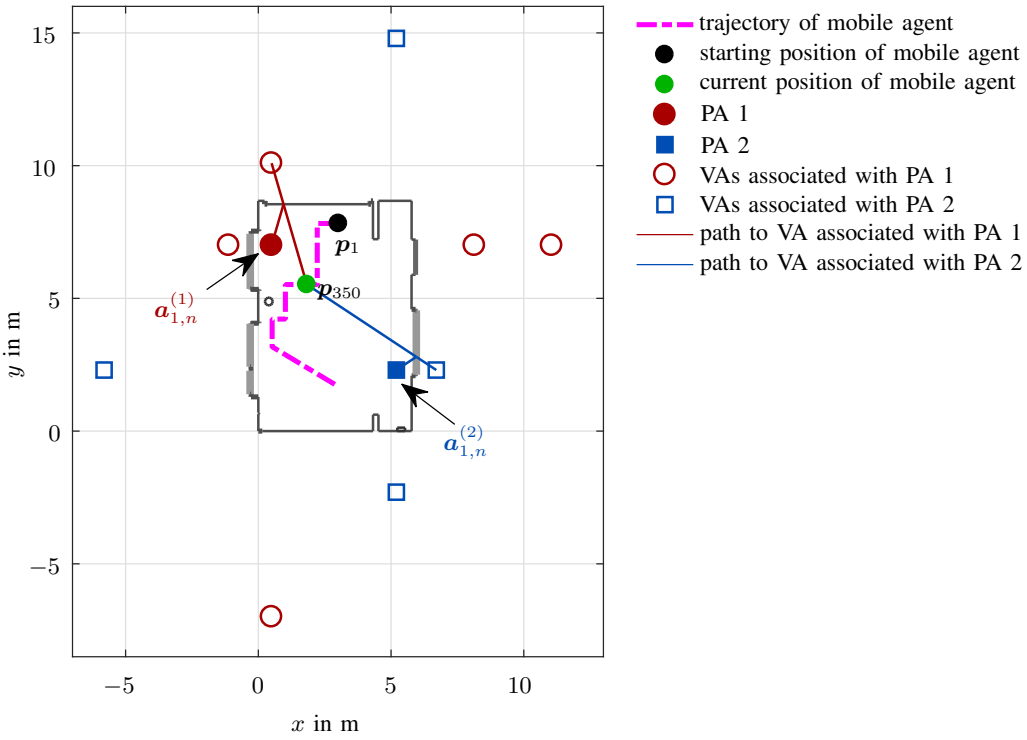


Fig. 1: Example of an environment map (floor plan). The PAs at fixed positions $\boldsymbol{a}_{1,n}^{(1)}$ (PA 1) and $\boldsymbol{a}_{1,n}^{(2)}$ (PA 2) are indicated by, respectively, a red bullet and a blue box within the floor plan. The magenta dashed-dotted line represents the trajectory of the mobile agent. The starting position of the mobile agent, $\boldsymbol{p}_1$, and the position $\boldsymbol{p}_{350}$ (at discrete time $n=350$) are indicated by a black and a green bullet, respectively. The red circles and blue squares outside the floor plan indicate some of the geometrically expected VAs associated with PA 1 and PA 2, respectively. Two exemplary reflection paths between the mobile agent at position $\boldsymbol{p}_{350}$ and two VAs associated with the two PAs are shown by red and blue lines.

between the agent and either the jth PA (for $l=1$) or the associated VAs (for $l \in \{2, \ldots, L_n^{(j)}\}$). That is, $\tau_{l,n}^{(j)} = \left\|\boldsymbol{p}_n - \boldsymbol{a}_{l,n}^{(j)}\right\|/c$, where c is the speed of light. The second term in (1), $d_n^{(j)}(t)$, represents the diffuse multipath, which interferes with the MPC term. The third term in (1), $w(t)$, is additive white Gaussian noise. We note that expression (1) presupposes a common time reference at the mobile agent and at the PAs. However, our algorithm can be extended to an unsynchronized system along the lines of [13], [47], exploiting the fact that the relevant geometric information is preserved in the time differences of the MPC delays [14]. Furthermore, our algorithm can also be extended to the case where the MPC parameters include AoAs and/or AoDs in addition to the delays $\tau_{l,n}^{(j)}$. An extension that uses the complex MPC amplitudes to directly estimate the detection probability of each feature has been proposed in [46].

In each time slot n and for each PA $j \in \{1, \ldots, J\}$, a parametric radio channel estimator [25]–[29], [48] processes the radio signals $r_n^{(j)}(t)$ and produces $M_n^{(j)}$ MPC parameter estimates along

with estimates of the corresponding complex amplitudes $\hat{\alpha}_{m,n}^{(j)}$, with $m \in \mathcal{M}_n^{(j)} \triangleq \{1, \ldots, M_n^{(j)}\}$. The set of estimated MPC parameters $\mathcal{M}_n^{(j)}$ is related to the set of specular MPCs $\mathcal{L}_n^{(j)}$ as follows. It is possible that some specular MPCs are not "detected" by the radio channel estimator and thus do not produce an MPC parameter estimate, and it is also possible that some estimates do not correspond to MPCs. Accordingly, $M_n^{(j)} = \big|\mathcal{M}_n^{(j)}\big|$ may be smaller than, equal to, or larger than $L_n^{(j)} = \big|\mathcal{L}_n^{(j)}\big|$. (Here, $|\cdot|$ denotes the cardinality of a set.) Note also that $M_n^{(j)}$ depends on the agent position $\boldsymbol{p}_n$ and on the environment. The amplitude estimates are used to calculate estimates of the MPC parameter variances $\hat{\sigma}_{m,n}^{(j)2}$, if such estimates are not provided by the channel estimator (cf. Section VI-C). We denote by $\boldsymbol{z}_{m,n}^{(j)}$ with $m \in \mathcal{M}_n^{(j)}$ the estimated parameters of the mth MPC of PA j. The stacked vectors $\boldsymbol{z}_n^{(j)} \triangleq \big[\boldsymbol{z}_{1,n}^{(j)\mathrm{T}} \cdots \boldsymbol{z}_{M_n^{(j)},n}^{(j)\mathrm{T}}\big]^{\mathrm{T}}$ are used as noisy "measurements" by the proposed BP-SLAM algorithm.

III. System Model and Statistical Formulation

A. Agent State and PF States

The state of the mobile agent at time n is $\boldsymbol{x}_n \triangleq [\boldsymbol{p}_n^{\mathrm{T}}\ \boldsymbol{v}_n^{\mathrm{T}}]^{\mathrm{T}}$, where $\boldsymbol{v}_n$ is the agent's velocity. The PFs are indexed by the tuple (j,k), where $j \in \{1, \ldots, J\}$ and $k \in \mathcal{K}_n^{(j)} \triangleq \{1, \ldots, K_n^{(j)}\}$ (recall that there are $K_n^{(j)}$ PFs for each PA j). The number of PAs J is known, but the number of PFs $K_n^{(j)}$ (for PA j) is unknown and random. The existence of the (j,k)th PF as an actual *feature* is indicated by the variable $r_{k,n}^{(j)} \in \{0,1\}$, where $r_{k,n}^{(j)} = 0$ ($r_{k,n}^{(j)} = 1$) means that the PF does not exist (exists) at time n. The state of PF (j,k) is the PF's position $\boldsymbol{a}_{k,n}^{(j)}$, and the *augmented state* is $\boldsymbol{y}_{k,n}^{(j)} \triangleq \big[\boldsymbol{a}_{k,n}^{(j)\mathrm{T}}\ r_{k,n}^{(j)}\big]^{\mathrm{T}}$ [39]. We also define $\boldsymbol{y}_n^{(j)} \triangleq \big[\boldsymbol{y}_{1,n}^{(j)\mathrm{T}} \cdots \boldsymbol{y}_{K_n^{(j)},n}^{(j)\mathrm{T}}\big]^{\mathrm{T}}$ and $\boldsymbol{y}_n \triangleq \big[\boldsymbol{y}_n^{(1)\mathrm{T}} \cdots \boldsymbol{y}_n^{(J)\mathrm{T}}\big]^{\mathrm{T}}$. We will formally consider PF states also for the nonexisting PFs (case $r_{k,n}^{(j)} = 0$); however, their values are obviously irrelevant. Therefore, the probability density function (pdf) of an augmented state, $f\big(\boldsymbol{y}_{k,n}^{(j)}\big) = f\big(\boldsymbol{a}_{k,n}^{(j)}, r_{k,n}^{(j)}\big)$, is such that for $r_{k,n}^{(j)} = 0$, $f\big(\boldsymbol{a}_{k,n}^{(j)}, 0\big) = f_{k,n}^{(j)} f_{\mathrm{D}}\big(\boldsymbol{a}_{k,n}^{(j)}\big)$, where $f_{\mathrm{D}}\big(\boldsymbol{a}_{k,n}^{(j)}\big)$ is an arbitrary "dummy pdf" and $f_{k,n}^{(j)} \geqslant 0$ can be interpreted as the probability of PF nonexistence [39]. We note that the joint augmented PF state, described here by the random vector $\boldsymbol{y}_n^{(j)}$, can also be modeled by a multi-Bernoulli random finite set [41]. However, the unordered nature of random finite sets would complicate the development of a factor graph and a BP algorithm.

At any time n, each PF is either a *legacy PF*, which was already established in the past, or a *new PF*, which is established for the first time. The augmented states of legacy PFs and new PFs for PA j will be denoted by $\tilde{\boldsymbol{y}}_{k,n}^{(j)} \triangleq \big[\tilde{\boldsymbol{a}}_{k,n}^{(j)\mathrm{T}}\ \tilde{r}_{k,n}^{(j)}\big]^{\mathrm{T}}$, $k \in \mathcal{K}_{n-1}^{(j)}$ and $\breve{\boldsymbol{y}}_{m,n}^{(j)} \triangleq \big[\breve{\boldsymbol{a}}_{m,n}^{(j)\mathrm{T}}\ \breve{r}_{m,n}^{(j)}\big]^{\mathrm{T}}$, $m \in \mathcal{M}_n^{(j)}$,

respectively. Thus, the number of new PFs equals the number of measurements, $M_n^{(j)}$. The set and number of legacy PFs are updated according to

$$\mathcal{K}_n^{(j)} = \mathcal{K}_{n-1}^{(j)} \cup \mathcal{M}_n^{(j)}, \qquad K_n^{(j)} = K_{n-1}^{(j)} + M_n^{(j)}, \tag{2}$$

where the first relation is understood to include a suitable reindexing of the elements of $\mathcal{M}_n^{(j)}$. (The number of PFs does not actually grow by $M_n^{(j)}$ because the set of PFs is pruned as explained in Section V-A.) We also define the following state-related vectors. For the legacy PFs for PA j, $\tilde{\boldsymbol{a}}_n^{(j)} \triangleq \left[\tilde{\boldsymbol{a}}_{1,n}^{(j)\mathrm{T}} \cdots \tilde{\boldsymbol{a}}_{K_{n-1},n}^{(j)\mathrm{T}}\right]^{\mathrm{T}}$, $\tilde{\boldsymbol{r}}_n^{(j)} \triangleq \left[\tilde{r}_{1,n}^{(j)} \cdots \tilde{r}_{K_{n-1},n}^{(j)}\right]^{\mathrm{T}}$, and $\tilde{\boldsymbol{y}}_n^{(j)} \triangleq \left[\tilde{\boldsymbol{y}}_{1,n}^{(j)\mathrm{T}} \cdots \tilde{\boldsymbol{y}}_{K_{n-1},n}^{(j)\mathrm{T}}\right]^{\mathrm{T}}$. For the new PFs for PA j, $\breve{\boldsymbol{a}}_n^{(j)} \triangleq \left[\breve{\boldsymbol{a}}_{1,n}^{(j)\mathrm{T}} \cdots \breve{\boldsymbol{a}}_{M_n^{(j)},n}^{(j)\mathrm{T}}\right]^{\mathrm{T}}$, $\breve{\boldsymbol{r}}_n^{(j)} \triangleq \left[\breve{r}_{1,n}^{(j)} \cdots \breve{r}_{M_n^{(j)},n}^{(j)}\right]^{\mathrm{T}}$, and $\breve{\boldsymbol{y}}_n^{(j)} \triangleq \left[\breve{\boldsymbol{y}}_{1,n}^{(j)\mathrm{T}} \cdots \breve{\boldsymbol{y}}_{M_n^{(j)},n}^{(j)\mathrm{T}}\right]^{\mathrm{T}}$. For the combination of legacy PFs and new PFs for PA j, $\boldsymbol{y}_n^{(j)} \triangleq \left[\tilde{\boldsymbol{y}}_n^{(j)\mathrm{T}}\ \breve{\boldsymbol{y}}_n^{(j)\mathrm{T}}\right]^{\mathrm{T}}$; note that the vector entries (subvectors) of $\boldsymbol{y}_n^{(j)}$ are given by $\boldsymbol{y}_{k,n}^{(j)}$ for $k \in \mathcal{K}_n^{(j)} = \mathcal{K}_{n-1}^{(j)} \cup \mathcal{M}_n^{(j)}$. For all the legacy PFs, $\tilde{\boldsymbol{y}}_n \triangleq \left[\tilde{\boldsymbol{y}}_n^{(1)\mathrm{T}} \cdots \tilde{\boldsymbol{y}}_n^{(J)\mathrm{T}}\right]^{\mathrm{T}}$, and for all the new PFs, $\breve{\boldsymbol{y}}_n \triangleq \left[\breve{\boldsymbol{y}}_n^{(1)\mathrm{T}} \cdots \breve{\boldsymbol{y}}_n^{(J)\mathrm{T}}\right]^{\mathrm{T}}$.

The number of new PFs at time n is known only after the current measurements have been observed. Features that are observed for the first time will be referred to as *newly detected features*. Before the current measurements are observed, only prior information about the newly detected features is available (namely, their prior distribution and mean number). After the current measurements are observed, newly detected features are represented by new PFs.

The agent state $\boldsymbol{x}_n$ and the augmented states of the legacy PFs, $\tilde{\boldsymbol{y}}_{k,n}^{(j)}$, are assumed to evolve independently according to Markovian state dynamics, i.e.,

$$f\left(\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n | \boldsymbol{x}_{n-1}, \boldsymbol{y}_{n-1}\right) = f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\tilde{\boldsymbol{y}}_n|\boldsymbol{y}_{n-1}) = f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) \prod_{j=1}^{J} \prod_{k=1}^{K_{n-1}^{(j)}} f\left(\tilde{\boldsymbol{y}}_{k,n}^{(j)} | \boldsymbol{y}_{k,n-1}^{(j)}\right), \tag{3}$$

where $f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$ and $f\left(\tilde{\boldsymbol{y}}_{k,n}^{(j)} | \boldsymbol{y}_{k,n-1}^{(j)}\right)$ are the state-transition pdfs of the agent and of legacy PF (j,k), respectively. Note that $\tilde{\boldsymbol{y}}_{k,n}^{(j)}$ depends on both $\tilde{\boldsymbol{y}}_{k,n-1}^{(j)}$ and $\breve{\boldsymbol{y}}_{m,n-1}^{(j)}$. If PF (j,k) exists at time $n-1$, i.e., $r_{k,n-1}^{(j)} = 1$, it either dies, i.e., $\tilde{r}_{k,n}^{(j)} = 0$, or survives, i.e., $\tilde{r}_{k,n}^{(j)} = 1$; in the latter case, it becomes a legacy PF at time n. The probability of survival is denoted by P_s. If the PF survives, its new state $\tilde{\boldsymbol{a}}_{k,n}^{(j)}$ is distributed according to the state-transition pdf $f\left(\tilde{\boldsymbol{a}}_{k,n}^{(j)} | \boldsymbol{a}_{k,n-1}^{(j)}\right)$. Therefore, $f\left(\tilde{\boldsymbol{y}}_{k,n}^{(j)} | \boldsymbol{y}_{k,n-1}^{(j)}\right) = f\left(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)} | \boldsymbol{a}_{k,n-1}^{(j)}, r_{k,n-1}^{(j)}\right)$ in (3) is given for $r_{k,n-1}^{(j)} = 1$ by

$$f\left(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)} | \boldsymbol{a}_{k,n-1}^{(j)}, r_{k,n-1}^{(j)} = 1\right) = \begin{cases} (1-P_\mathrm{s})\, f_\mathrm{D}\left(\tilde{\boldsymbol{a}}_{k,n}^{(j)}\right), & \tilde{r}_{k,n}^{(j)} = 0 \\ P_\mathrm{s} f\left(\tilde{\boldsymbol{a}}_{k,n}^{(j)} | \boldsymbol{a}_{k,n-1}^{(j)}\right), & \tilde{r}_{k,n}^{(j)} = 1. \end{cases} \tag{4}$$

If PF (j,k) does not exist at time $n-1$, i.e., $r_{k,n-1}^{(j)} = 0$, it cannot exist as a legacy PF at time n either. Therefore,

$$f\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)} \big| \boldsymbol{a}_{k,n-1}^{(j)}, r_{k,n-1}^{(j)} = 0\big) = \begin{cases} f_{\mathrm{D}}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}\big), & \tilde{r}_{k,n}^{(j)} = 0 \\ 0, & \tilde{r}_{k,n}^{(j)} = 1. \end{cases} \tag{5}$$

B. Association Vectors

For each PA j, the measurements (MPC parameter estimates) $\boldsymbol{z}_{m,n}^{(j)}$, $m \in \mathcal{M}_n^{(j)}$ described in Section II are subject to a measurement origin uncertainty, also known as DA uncertainty. That is, it is not known which measurement $\boldsymbol{z}_{m,n}^{(j)}$ is associated with which PF $k \in \mathcal{K}_n^{(j)}$, or if a measurement $\boldsymbol{z}_{m,n}^{(j)}$ did not originate from any PF (this is known as a *false alarm* or *clutter*), or if a PF did not give rise to any measurement (this is known as a *missed detection*). The probability that a PF is "detected" in the sense that it generates a measurement $\boldsymbol{z}_{m,n}^{(j)}$ in the MPC parameter estimation stage is denoted by $P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \boldsymbol{a}_{k,n}^{(j)}\big)$. The distribution of false alarm measurements is described by the pdf $f_{\mathrm{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big)$. The functions $P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \boldsymbol{a}_{k,n}^{(j)}\big) \in (0, 1]$ and $f_{\mathrm{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big) \geq 0$ are supposed known. Following [31], we assume that at any time n, each PF can generate at most one measurement, and each measurement can be generated by at most one PF.

The associations between the measurements $m \in \mathcal{M}_n^{(j)}$ and the legacy PF states (j, k), $k \in \mathcal{K}_{n-1}^{(j)}$ at time n can be described by the $K_{n-1}^{(j)}$-dimensional *feature-oriented DA vector* $\boldsymbol{c}_n^{(j)} = \big[c_{1,n}^{(j)} \cdots c_{K_{n-1}^{(j)},n}^{(j)}\big]^{\mathrm{T}}$, whose kth entry is defined to be $c_{k,n}^{(j)} \triangleq m \in \mathcal{M}_n^{(j)}$ if legacy PF (j, k) generates measurement $\boldsymbol{z}_{m,n}^{(j)}$, and $c_{k,n}^{(j)} \triangleq 0$ if it does not generate any measurement. In addition, following [37], [39], we consider the $M_n^{(j)}$-dimensional *measurement-oriented DA vector* $\boldsymbol{b}_n^{(j)} = \big[b_{1,n}^{(j)} \cdots b_{M_n^{(j)},n}^{(j)}\big]^{\mathrm{T}}$, whose mth entry is defined to be $b_{m,n}^{(j)} \triangleq k \in \mathcal{K}_{n-1}^{(j)}$ if measurement $\boldsymbol{z}_{m,n}^{(j)}$ is generated by legacy PF (j, k), and $b_{m,n}^{(j)} \triangleq 0$ if it is not generated by any legacy PF We also define $\boldsymbol{c}_n \triangleq \big[\boldsymbol{c}_n^{(1)\mathrm{T}} \cdots \boldsymbol{c}_n^{(J)\mathrm{T}}\big]^{\mathrm{T}}$ and $\boldsymbol{b}_n \triangleq \big[\boldsymbol{b}_n^{(1)\mathrm{T}} \cdots \boldsymbol{b}_n^{(J)\mathrm{T}}\big]^{\mathrm{T}}$. The two DA vectors $\boldsymbol{c}_n$ and $\boldsymbol{b}_n$ are unknown and modeled as random. They are equivalent since one can be determined from the other. However, the redundant formulation of DA uncertainty in terms of both $\boldsymbol{c}_n$ and $\boldsymbol{b}_n$ is key to obtaining the scalability properties of the BP algorithm to be presented in Section V-B. Furthermore, as will be discussed in Section IV, it facilitates the establishment of a factor graph for the problem of jointly inferring the agent state and the states of the legacy PFs and new PFs.

C. Prior Distributions

For a given PA j, there are $M_n^{(j)} = |\mathcal{M}_n^{(j)}|$ new PFs at time n. The number of false alarms and the number of newly detected features are assumed Poisson distributed with mean $\mu_{\mathrm{FA}}^{(j)}$ and $\mu_{\mathrm{n},n}^{(j)}$, respectively [31], [49]. Then, one can derive the following expression of the joint conditional

prior probability mass function (pmf) of the DA vector $\boldsymbol{c}_n^{(j)}$, the vector of existence indicators of new PFs, $\breve{\boldsymbol{r}}_n^{(j)}$, and the number of measurements or equivalently new PFs, $M_n^{(j)}$, given the state of the mobile agent, $\boldsymbol{x}_n$, and the vector of augmented states of legacy PFs, $\tilde{\boldsymbol{y}}_n^{(j)}$ (cf. [39])

$$p\big(\boldsymbol{c}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big) = \chi_{\boldsymbol{c}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}}\, \psi\big(\boldsymbol{c}_n^{(j)}\big) \left(\prod_{m \in \mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}^{(j)}} \Gamma_{\boldsymbol{c}_n^{(j)}}\big(\breve{r}_{m,n}^{(j)}\big) \right) \left(\prod_{k \in \mathcal{D}_{\boldsymbol{c}_n^{(j)}, \tilde{\boldsymbol{r}}_n^{(j)}}^{(j)}} P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \boldsymbol{a}_{k,n}^{(j)}\big) \right) \times \prod_{k' \in \bar{\mathcal{D}}_{\boldsymbol{c}_n^{(j)}, \tilde{\boldsymbol{r}}_n^{(j)}}^{(j)}} \big[1\big(c_{k',n}^{(j)}\big) - \tilde{r}_{k',n}^{(j)} P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k',n}^{(j)}\big)\big], \tag{6}$$

with $\chi_{\boldsymbol{c}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}} \triangleq e^{-(\mu_{\mathrm{FA}}^{(j)} + \mu_{\mathrm{n},n}^{(j)})} \big(\mu_{\mathrm{n},n}^{(j)}\big)^{\left|\mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}\right|} \big(\mu_{\mathrm{FA}}^{(j)}\big)^{M_n^{(j)} - \left|\mathcal{D}_{\boldsymbol{c}_n^{(j)}, \tilde{\boldsymbol{r}}_n^{(j)}}\right| - \left|\mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}\right|} / M_n^{(j)}!$. Here, $\mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}$ denotes the set of existing new PFs, i.e., $\mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}} \triangleq \big\{m \in \mathcal{M}_n^{(j)} : \breve{r}_{m,n}^{(j)} = 1\big\}$; $\mathcal{D}_{\boldsymbol{c}_n^{(j)}, \tilde{\boldsymbol{r}}_n^{(j)}}$ denotes the set of existing legacy PFs for PA j, i.e., $\mathcal{D}_{\boldsymbol{c}_n^{(j)}, \tilde{\boldsymbol{r}}_n^{(j)}} \triangleq \big\{k \in \mathcal{K}_{n-1}^{(j)} : \tilde{r}_{k,n}^{(j)} = 1, c_{k,n}^{(j)} \neq 0\big\}$; and $\bar{\mathcal{D}}_{\boldsymbol{c}_n^{(j)}, \tilde{\boldsymbol{r}}_n^{(j)}} \triangleq \mathcal{K}_{n-1}^{(j)} \big\backslash \mathcal{D}_{\boldsymbol{c}_n^{(j)}, \tilde{\boldsymbol{r}}_n^{(j)}}$. Furthermore, $\psi\big(\boldsymbol{c}_n^{(j)}\big)$ is defined to be 0 if there exist $k, k' \in \mathcal{K}_{n-1}^{(j)}$ with $k \neq k'$ such that $c_{k,n}^{(j)} = c_{k',n}^{(j)} \neq 0$, and to be 1 otherwise; and $\Gamma_{\boldsymbol{c}_n^{(j)}}\big(\breve{r}_{m,n}^{(j)}\big)$ is defined to be 0 if $\breve{r}_{m,n}^{(j)} = 1$ and there exists $k \in \mathcal{K}_{n-1}^{(j)}$ such that $c_{k,n}^{(j)} = m$, and to be 1 otherwise. Finally, $1(c)$ denotes the indicator function of the event $c = 0$ (i.e., $1(c) = 1$ if $c = 0$ and 0 otherwise). The functions $\psi\big(\boldsymbol{c}_n^{(j)}\big)$ and $\Gamma_{\boldsymbol{c}_n}^{(j)}\big(\breve{r}_{m,n}^{(j)}\big)$ enforce our DA assumption from Section III-B, i.e., $\psi\big(\boldsymbol{c}_n^{(j)}\big)$ enforces $p\big(\boldsymbol{c}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big) = 0$ if any measurement is associated with more than one legacy PF, and $\Gamma_{\boldsymbol{c}_n}^{(j)}\big(\breve{r}_{m,n}^{(j)}\big)$ enforces $p\big(\boldsymbol{c}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big) = 0$ if a new PF is associated with a measurement m that is also associated with a legacy PF. At time $n = 1$, prior information on PAs and VAs can be incorporated by introducing legacy PFs. In the opposite case, there are no legacy PFs at $n = 1$, i.e., $\tilde{\boldsymbol{y}}_1^{(j)}$ is an empty vector, and thus $p\big(\boldsymbol{c}_1^{(j)}, \breve{\boldsymbol{r}}_1^{(j)}, M_1^{(j)} \big| \boldsymbol{x}_1, \tilde{\boldsymbol{y}}_1^{(j)}\big) = p\big(\boldsymbol{c}_1^{(j)}, \breve{\boldsymbol{r}}_1^{(j)}, M_1^{(j)} \big| \boldsymbol{x}_1\big)$. An expression of this pmf can be obtained by replacing in (6) (for $n = 1$) all factors involving $\tilde{\boldsymbol{y}}_1^{(j)}$ (or, equivalently, $\tilde{\boldsymbol{a}}_1^{(j)}$ and $\tilde{\boldsymbol{r}}_1^{(j)}$) by 1.

The states of newly detected features are assumed to be a priori independent and identically distributed (iid) according to some pdf $f_{\mathrm{n},n}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)} \big| \boldsymbol{x}_n\big)$, which will be discussed in Section VI-A3. The prior pdf of the states of new PFs for PA j, $\breve{\boldsymbol{a}}_n^{(j)}$, conditioned on $\boldsymbol{x}_n$, $\breve{\boldsymbol{r}}_n^{(j)}$, and $M_n^{(j)}$ is then obtained as

$$f\big(\breve{\boldsymbol{a}}_n^{(j)} \big| \boldsymbol{x}_n, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big) = \left(\prod_{m \in \mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}^{(j)}} f_{\mathrm{n},n}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)} \big| \boldsymbol{x}_n\big) \right) \prod_{m' \in \bar{\mathcal{N}}_{\breve{\boldsymbol{r}}_n^{(j)}}^{(j)}} f_{\mathrm{D}}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}\big), \tag{7}$$

where $\bar{\mathcal{N}}_{\breve{\boldsymbol{r}}_n^{(j)}} \triangleq \mathcal{M}_n^{(j)} \big\backslash \mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}$. Note that before the measurements are obtained, $M_n^{(j)}$ and, thus, the length of the vectors $\breve{\boldsymbol{a}}_n^{(j)}$ and $\breve{\boldsymbol{r}}_n^{(j)}$ (which is $M_n^{(j)}$) is random.

We assume that for the agent state at time $n=1$, $\boldsymbol{x}_1$, an informative prior pdf $f(\boldsymbol{x}_1)$ is available. We also assume that the new PF state vector $\breve{\boldsymbol{a}}_n^{(j)}$ and the DA vectors $\boldsymbol{c}_n^{(j)}$ are conditionally independent given the legacy PF state vector $\tilde{\boldsymbol{a}}_n^{(j)}$. Let us next consider $\boldsymbol{x}_{n'}$, $\boldsymbol{y}_{n'}$, $\boldsymbol{c}_{n'}$, and $\boldsymbol{m}_{n'} \triangleq \big[M_{n'}^{(1)} \cdots M_{n'}^{(J)}\big]^{\mathrm{T}}$ for all time steps $n' = 1, \ldots, n$, and accordingly define the vector $\boldsymbol{x}_{1:n} \triangleq \big[\boldsymbol{x}_1^{\mathrm{T}} \cdots \boldsymbol{x}_n^{\mathrm{T}}\big]^{\mathrm{T}}$ and similarly $\boldsymbol{y}_{1:n}$, $\boldsymbol{c}_{1:n}$, and $\boldsymbol{m}_{1:n}$. We then obtain the joint prior pdf as

$$f\big(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}, \boldsymbol{c}_{1:n}, \boldsymbol{m}_{1:n}\big) = f(\boldsymbol{x}_1)\left(\prod_{j'=1}^{J} p\big(\boldsymbol{c}_1^{(j')}, \breve{\boldsymbol{r}}_1^{(j')}, M_1^{(j')} \big| \boldsymbol{x}_1, \tilde{\boldsymbol{y}}_1^{(j')}\big)\right) \prod_{n'=2}^{n} f(\boldsymbol{x}_{n'}|\boldsymbol{x}_{n'-1}) \\ \times \prod_{j=1}^{J} f\big(\tilde{\boldsymbol{y}}_{n'}^{(j)} \big| \boldsymbol{y}_{n'-1}^{(j)}\big) f\big(\breve{\boldsymbol{a}}_{n'}^{(j)} \big| \boldsymbol{x}_{n'}, \breve{\boldsymbol{r}}_{n'}^{(j)}, M_{n'}^{(j)}\big) p\big(\boldsymbol{c}_{n'}^{(j)}, \breve{\boldsymbol{r}}_{n'}^{(j)}, M_{n'}^{(j)} \big| \boldsymbol{x}_{n'}, \tilde{\boldsymbol{y}}_{n'}^{(j)}\big), \tag{8}$$

where the last three factors are given by (4) and (5); (7); and (6), respectively.

D. Likelihood Function

The conditional pdf $f\big(\boldsymbol{z}_{m,n}^{(j)} \big| \boldsymbol{x}_n, \boldsymbol{a}_{k,n}^{(j)}\big)$ characterizing the statistical relation between the measurements $\boldsymbol{z}_{m,n}^{(j)}$ and the states $\boldsymbol{x}_n$ and $\boldsymbol{a}_{k,n}^{(j)}$ depends on the concrete measurement model; an example will be considered in Section VI. This pdf is a central element in the conditional pdf of the total measurement vector $\boldsymbol{z}_n \triangleq \big[\boldsymbol{z}_n^{(1)\mathrm{T}} \cdots \boldsymbol{z}_n^{(J)\mathrm{T}}\big]^{\mathrm{T}}$ given $\boldsymbol{x}_n$, $\tilde{\boldsymbol{y}}_n$, $\breve{\boldsymbol{y}}_n$, $\boldsymbol{c}_n$, and $\boldsymbol{m}_n$. Assuming that the $\boldsymbol{z}_n^{(j)}$ are conditionally independent across j given $\boldsymbol{x}_n$, $\tilde{\boldsymbol{y}}_n$, $\breve{\boldsymbol{y}}_n$, $\boldsymbol{c}_n$, and $\boldsymbol{m}_n$ [31], we obtain $f(\boldsymbol{z}_n|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n, \breve{\boldsymbol{y}}_n, \boldsymbol{c}_n, \boldsymbol{m}_n) = \prod_{j=1}^{J} f\big(\boldsymbol{z}_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}, \breve{\boldsymbol{y}}_n^{(j)}, \boldsymbol{c}_n^{(j)}, M_n^{(j)}\big)$, where [31]

$$f\big(\boldsymbol{z}_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}, \breve{\boldsymbol{y}}_n^{(j)}, \boldsymbol{c}_n^{(j)}, M_n^{(j)}\big) = \left(\prod_{m=1}^{M_n^{(j)}} f_{\mathrm{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big)\right) \left(\prod_{k \in \mathcal{D}_{\boldsymbol{c}_n^{(j)}, \tilde{\boldsymbol{r}}_n^{(j)}}} \frac{f\Big(\boldsymbol{z}_{c_{k,n}^{(j)},n}^{(j)} \Big| \boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}\Big)}{f_{\mathrm{FA}}\Big(\boldsymbol{z}_{c_{k,n}^{(j)},n}^{(j)}\Big)}\right) \\ \times \prod_{m' \in \mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}} \frac{f\big(\boldsymbol{z}_{m',n}^{(j)} \big| \boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m',n}^{(j)}\big)}{f_{\mathrm{FA}}\big(\boldsymbol{z}_{m',n}^{(j)}\big)}. \tag{9}$$

In particular, at $n=1$, $f\big(\boldsymbol{z}_1^{(j)} \big| \boldsymbol{x}_1, \tilde{\boldsymbol{y}}_1^{(j)}, \breve{\boldsymbol{y}}_1^{(j)}, \boldsymbol{c}_1^{(j)}, M_1^{(j)}\big) = f\big(\boldsymbol{z}_1^{(j)} \big| \boldsymbol{x}_1, \breve{\boldsymbol{y}}_1^{(j)}, \boldsymbol{c}_1^{(j)}, M_1^{(j)}\big)$. An expression of this pdf can be obtained by replacing in (9) (for $n=1$) all factors involving $\tilde{\boldsymbol{y}}_1^{(j)}$ (or, equivalently, $\tilde{\boldsymbol{a}}_1^{(j)}$ and $\tilde{\boldsymbol{r}}_1^{(j)}$) by 1.

Let us consider $f\big(\boldsymbol{z}_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}, \breve{\boldsymbol{y}}_n^{(j)}, \boldsymbol{c}_n^{(j)}, M_n^{(j)}\big)$ as a *likelihood function*, i.e., a function of $\boldsymbol{x}_n$, $\tilde{\boldsymbol{y}}_n^{(j)}$, $\breve{\boldsymbol{y}}_n^{(j)}$, $\boldsymbol{c}_n^{(j)}$, and $M_n^{(j)}$, for observed $\boldsymbol{z}_n^{(j)}$. If $\boldsymbol{z}_n^{(j)}$ is observed and therefore fixed, also $M_n^{(j)}$ is fixed, and we can rewrite (9), up to a constant normalization factor, as

$$f\big(\boldsymbol{z}_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}, \breve{\boldsymbol{y}}_n^{(j)}, \boldsymbol{c}_n^{(j)}, M_n^{(j)}\big) \propto \left(\prod_{k=1}^{K_n^{(j)}} g_1\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big)\right) \prod_{m \in \mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}} \frac{f\big(\boldsymbol{z}_{m,n}^{(j)} \big| \boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m,n}^{(j)}\big)}{f_{\mathrm{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big)}. \tag{10}$$

Here, the factors $g_1\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big)$ are defined as

$$g_1\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) \triangleq \begin{cases} \dfrac{f\big(\boldsymbol{z}_{m,n}^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}\big)}{f_{\text{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big)}, & c_{k,n}^{(j)} \in \mathcal{M}_n^{(j)} \\ 1, & c_{k,n}^{(j)} = 0 \end{cases} \tag{11}$$

and $g_1\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, 0, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) \triangleq 1$. Finally, the likelihood function for $\boldsymbol{z}_{1:n} \triangleq \big[\boldsymbol{z}_1^{\text{T}} \cdots \boldsymbol{z}_n^{\text{T}}\big]^{\text{T}}$, involving the measurements $\boldsymbol{z}_{m,n'}^{(j)}$ of all PAs $j = 1, \ldots, J$ and all time steps $n' = 1, \ldots, n$, can be derived similarly to (8); one obtains

$$f(\boldsymbol{z}_{1:n}|\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}, \boldsymbol{c}_{1:n}, \boldsymbol{m}_{1:n}) \propto \prod_{j=1}^{J} \Bigg(\prod_{m \in \mathcal{N}_{\breve{\boldsymbol{r}}_1^{(j)}}} \frac{f\big(\boldsymbol{z}_{m,1}^{(j)}\big|\boldsymbol{x}_1, \breve{\boldsymbol{a}}_{m,1}^{(j)}\big)}{f_{\text{FA}}\big(\boldsymbol{z}_{m,1}^{(j)}\big)} \Bigg) \prod_{n'=2}^{n} \Bigg(\prod_{m' \in \mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}} \frac{f\big(\boldsymbol{z}_{m',n'}^{(j)}\big|\boldsymbol{x}_{n'}, \breve{\boldsymbol{a}}_{m',n'}^{(j)}\big)}{f_{\text{FA}}\big(\boldsymbol{z}_{m',n'}^{(j)}\big)} \Bigg)$$
$$\times \prod_{k=1}^{K_{n'}^{(j)}} g_1\big(\boldsymbol{x}_{n'}, \tilde{\boldsymbol{a}}_{k,n'}^{(j)}, \tilde{r}_{k,n'}^{(j)}, c_{k,n'}^{(j)}; \boldsymbol{z}_{n'}^{(j)}\big). \tag{12}$$

IV. Joint Posterior pdf and Factor Graph

A. Redundant Formulation of the Exclusion Constraint

The proposed BP algorithm is based on a redundant formulation of probabilistic DA in terms of the vectors $\boldsymbol{c}_n^{(j)}$ and $\boldsymbol{b}_n^{(j)}$ [37], [39]. To obtain a probabilistic description and, in turn, a factor graph, we formally replace the exclusion constraint factor $\psi\big(\boldsymbol{c}_n^{(j)}\big)$ involved in the prior pmf in (6) by

$$\psi\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}\big) \triangleq \prod_{k=1}^{K_n^{(j)}} \prod_{m=1}^{M_n^{(j)}} \psi\big(c_{k,n}^{(j)}, b_{m,n}^{(j)}\big), \tag{13}$$

where $\psi\big(c_{k,n}^{(j)}, b_{m,n}^{(j)}\big)$ is defined to be 0 if either $c_{k,n}^{(j)} = m$ and $b_{m,n}^{(j)} \neq k$ or $b_{m,n}^{(j)} = k$ and $c_{k,n}^{(j)} \neq m$, and 1 otherwise. The resulting modified prior pmf $p\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big)$ is related to the original prior pmf $p\big(\boldsymbol{c}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big)$ in (6) acording to $p\big(\boldsymbol{c}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big) = \sum_{\boldsymbol{b}_n^{(j)}} p\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big)$, where the summation is over all $\boldsymbol{b}_n^{(j)} \in \big\{0, 1, \ldots, K_n^{(j)}\big\}^{M_n^{(j)}}$.

Let us now consider the product of the likelihood function $f\big(\boldsymbol{z}_n^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}, \breve{\boldsymbol{y}}_n^{(j)}, \boldsymbol{c}_n^{(j)}, M_n^{(j)}\big)$ in (10), the prior pdf of the new PF states $f\big(\breve{\boldsymbol{a}}_n^{(j)}\big|\boldsymbol{x}_n, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big)$ in (7), and the modified prior pmf $p\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big)$. We obtain

$$f\big(\boldsymbol{z}_n^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}, \breve{\boldsymbol{y}}_n^{(j)}, \boldsymbol{c}_n^{(j)}, M_n^{(j)}\big)\, f\big(\breve{\boldsymbol{a}}_n^{(j)}\big|\boldsymbol{x}_n, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big)\, p\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big|\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big)$$
$$\propto \psi\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}\big) \Bigg(\prod_{k=1}^{K_n^{(j)}} g\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) \Bigg) \Bigg(\prod_{m' \in \bar{\mathcal{N}}_{\breve{\boldsymbol{r}}_n^{(j)}}} f_{\text{D}}\big(\breve{\boldsymbol{a}}_{m',n}^{(j)}\big) \Bigg)$$

$$\times \prod_{m \in \mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}} \frac{\mu_{\mathrm{n},n}^{(j)} f_{\mathrm{n},n}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)} \big| \boldsymbol{x}_n\big)\, \Gamma_{\boldsymbol{c}_n^{(j)}}\big(\breve{r}_{m,n}^{(j)}\big)\, f\big(\boldsymbol{z}_{m,n}^{(j)} \big| \boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m,n}^{(j)}\big)}{f_{\mathrm{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big)}. \tag{14}$$

Here, $g\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) \triangleq g_1\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) g_2\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; M_n^{(j)}\big)$, where $g_1\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big)$ was defined in Section III-D and $g_2\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; M_n^{(j)}\big)$ is defined as

$$g_2\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1, c_{k,n}^{(j)}; M_n^{(j)}\big) \triangleq \begin{cases} \dfrac{P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}\big)}{\mu_{\mathrm{FA}}^{(j)}}, & c_{k,n}^{(j)} \in \mathcal{M}_n^{(j)} \\ 1 - P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}\big), & c_{k,n}^{(j)} = 0 \end{cases}$$

and $g_2\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, 0, c_{k,n}^{(j)}; M_n^{(j)}\big) \triangleq 1\big(c_{k,n}^{(j)}\big)$. One thus obtains for $g\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big)$

$$g\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) = \begin{cases} \dfrac{P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}\big)\, f\big(\boldsymbol{z}_{m,n}^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}\big)}{\mu_{\mathrm{FA}}^{(j)} f_{\mathrm{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big)}, & c_{k,n}^{(j)} \in \mathcal{M}_n^{(j)} \\ 1 - P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}\big), & c_{k,n}^{(j)} = 0 \end{cases}$$

and $g\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, 0, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) = 1\big(c_{k,n}^{(j)}\big)$. For a choice of $\boldsymbol{c}_n^{(j)}$ and $\boldsymbol{b}_n^{(j)}$ that is valid in the sense that $p\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big) \neq 0$, the exclusion constraint expressed by $\Gamma_{\boldsymbol{c}_n^{(j)}}\big(\breve{r}_{m,n}^{(j)}\big)$ is satisfied, i.e., $\Gamma_{\boldsymbol{c}_n^{(j)}}\big(\breve{r}_{m,n}^{(j)}\big) = 1$, if and only if both $b_{m,n}^{(j)} = 0$ and $\breve{r}_{m,n}^{(j)} = 1$. Therefore, in (14), we can summarize the products over all $m \in \mathcal{N}_{\breve{\boldsymbol{r}}_n^{(j)}}$ and over all $m' \in \bar{\mathcal{N}}_{\breve{\boldsymbol{r}}_n^{(j)}}$ by a product over all $m \in \mathcal{M}_n^{(j)} = \{1, \ldots, M_n^{(j)}\}$. More specifically, we can rewrite (14) as

$$\begin{aligned} &f\big(\boldsymbol{z}_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}, \breve{\boldsymbol{y}}_n^{(j)}, \boldsymbol{c}_n^{(j)}, M_n^{(j)}\big)\, f\big(\breve{\boldsymbol{a}}_n^{(j)} \big| \boldsymbol{x}_n, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)}\big)\, p\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}, \breve{\boldsymbol{r}}_n^{(j)}, M_n^{(j)} \big| \boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n^{(j)}\big) \\ &\quad \propto \psi\big(\boldsymbol{c}_n^{(j)}, \boldsymbol{b}_n^{(j)}\big) \left(\prod_{k=1}^{K_n^{(j)}} g\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) \right) \prod_{m=1}^{M_n^{(j)}} h\big(\boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m,n}^{(j)}, \breve{r}_{m,n}^{(j)}, b_{m,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big), \end{aligned} \tag{15}$$

where $h\big(\boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m,n}^{(j)}, \breve{r}_{m,n}^{(j)}, b_{m,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big)$ is defined as

$$h\big(\boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m,n}^{(j)}, 1, b_{m,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) \triangleq \begin{cases} 0, & b_{m,n}^{(j)} \in \mathcal{K}_n^{(j)} \\ \dfrac{\mu_{\mathrm{n},n}^{(j)} f_{\mathrm{n},n}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)} \big| \boldsymbol{x}_n\big)\, f\big(\boldsymbol{z}_{m,n}^{(j)} \big| \boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m,n}^{(j)}\big)}{\mu_{\mathrm{FA}}^{(j)} f_{\mathrm{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big)}, & b_{m,n}^{(j)} = 0 \end{cases}$$

and $h\big(\boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m,n}^{(j)}, 0, b_{m,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big) \triangleq f_{\mathrm{D}}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}\big)$. The mean number $\mu_{\mathrm{n},n}^{(j)}$ and conditional pdf $f_{\mathrm{n},n}\big(\boldsymbol{a}_{m,n}^{(j)} \big| \boldsymbol{x}_n\big)$ of newly detected features can either be pre-specified (e.g., constant with respect to time n and with a spatial distribution that is uniform in $\boldsymbol{a}_{m,n}^{(j)}$) or inferred online by means of a probability hypothesis density (PHD) filter, as described in [34].

B. Joint Posterior pdf

Using Bayes' rule and independence assumptions related to the state-transition pdfs (see Section III-A), the prior pdfs (see Section III-C), and the likelihood model (see Section III-D), the joint posterior pdf of $\boldsymbol{x}_{n'}$, $\tilde{\boldsymbol{y}}_{n'}$, $\breve{\boldsymbol{y}}_{n'}$, $\boldsymbol{c}_{n'}$, and $\boldsymbol{b}_{n'}$ for all $n' = 1, \ldots, n$ is obtained as

$$
\begin{aligned}
&f(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}, \boldsymbol{c}_{1:n}, \boldsymbol{b}_{1:n}, \boldsymbol{m}_{1:n} | \boldsymbol{z}_{1:n}) \\
&\propto f(\boldsymbol{z}_{1:n} | \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}, \boldsymbol{c}_{1:n}, \boldsymbol{m}_{1:n}) f(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}, \boldsymbol{c}_{1:n}, \boldsymbol{m}_{1:n}) \\
&= f(\boldsymbol{x}_1) \left(\prod_{j'=1}^{J} f\big(\boldsymbol{z}_1^{(j')} \big| \boldsymbol{x}_1, \breve{\boldsymbol{y}}_1^{(j')}, \boldsymbol{c}_1^{(j')}, M_1^{(j')}\big) f\big(\breve{\boldsymbol{y}}_1^{(j')}, \boldsymbol{c}_1^{(j')}, M_1^{(j')}\big) \right) \prod_{n'=2}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) \prod_{j=1}^{J} f\big(\tilde{\boldsymbol{y}}_{n'}^{(j)} \big| \boldsymbol{y}_{n'-1}^{(j)}\big) \\
&\quad \times f\big(\boldsymbol{z}_{n'}^{(j)} \big| \boldsymbol{x}_{n'}, \tilde{\boldsymbol{y}}_{n'}^{(j)}, \breve{\boldsymbol{y}}_{n'}^{(j)}, \boldsymbol{c}_{n'}^{(j)}, M_{n'}^{(j)}\big) f\big(\breve{\boldsymbol{a}}_{n'}^{(j)} \big| \boldsymbol{x}_{n'}, \breve{\boldsymbol{r}}_{n'}^{(j)}, M_{n'}^{(j)}\big) p\big(\boldsymbol{c}_{n'}^{(j)}, \boldsymbol{b}_{n'}^{(j)}, \breve{\boldsymbol{r}}_{n'}^{(j)}, M_{n'}^{(j)} \big| \boldsymbol{x}_{n'}, \tilde{\boldsymbol{y}}_{n'}^{(j)}\big) . \quad (16)
\end{aligned}
$$

Inserting expression (15) yields

$$
\begin{aligned}
&f(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}, \boldsymbol{c}_{1:n}, \boldsymbol{b}_{1:n}, \boldsymbol{m}_{1:n} | \boldsymbol{z}_{1:n}) \\
&\quad \propto f(\boldsymbol{x}_1) \left(\prod_{j'=1}^{J} \prod_{m'=1}^{M_1^{(j')}} h\big(\boldsymbol{x}_1, \breve{\boldsymbol{a}}_{m',1}^{(j')}, \breve{r}_{m',1}^{(j')}, b_{m',1}^{(j')}; \boldsymbol{z}_1^{(j')}\big) \right) \prod_{n'=2}^{n} f(\boldsymbol{x}_{n'} | \boldsymbol{x}_{n'-1}) \prod_{j=1}^{J} \psi\big(\boldsymbol{c}_{n'}^{(j)}, \boldsymbol{b}_{n'}^{(j)}\big) \\
&\qquad \times \left(\prod_{m=1}^{M_{n'}^{(j)}} h\big(\boldsymbol{x}_{n'}, \breve{\boldsymbol{a}}_{m,n'}^{(j)}, \breve{r}_{m,n'}^{(j)}, b_{m,n'}^{(j)}; \boldsymbol{z}_{n'}^{(j)}\big) \right) \prod_{k=1}^{K_{n'-1}^{(j)}} f\big(\tilde{\boldsymbol{y}}_{k,n'}^{(j)} \big| \boldsymbol{y}_{k,n'-1}^{(j)}\big) g\big(\boldsymbol{x}_{n'}, \tilde{\boldsymbol{a}}_{k,n'}^{(j)}, \tilde{r}_{k,n'}^{(j)}, c_{k,n'}^{(j)}; \boldsymbol{z}_{n'}^{(j)}\big) .
\end{aligned} \quad (17)
$$

This factorization of the joint posterior pdf is represented by the factor graph [35], [36] shown in Fig. 2.

V. The BP-SLAM Algorithm

The proposed BP-SLAM algorithm performs Bayesian detection and estimation of the relevant states from all the measurements acquired so far. The posterior distributions required for detection and estimation are calculated in an efficient, time-recursive manner by using BP message passing [36] on the factor graph in Fig. 2.

A. Detection and Estimation

Our goal is to estimate the agent state $\boldsymbol{x}_n$ and to detect and estimate the PF states $\boldsymbol{a}_{k,n}^{(j)}$ from the total measurement vector $\boldsymbol{z}_{1:n}$. For estimating $\boldsymbol{x}_n$, we will develop an approximate calculation of the minimum mean-square error (MMSE) estimator [50]

$$
\hat{\boldsymbol{x}}_n^{\mathrm{MMSE}} \triangleq \int \boldsymbol{x}_n f(\boldsymbol{x}_n | \boldsymbol{z}_{1:n}) \mathrm{d}\boldsymbol{x}_n . \quad (18)
$$

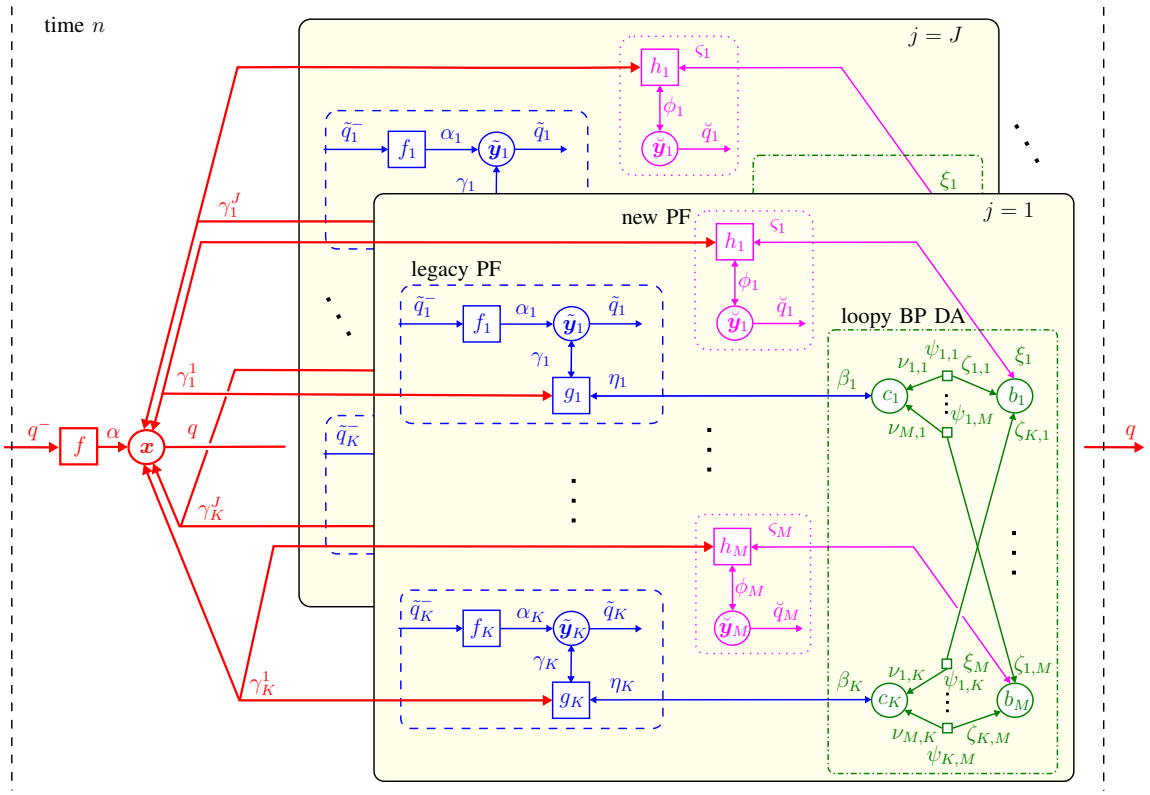


Fig. 2: Factor graph representing the factorization of the joint posterior pdf in (17). The subgraphs corresponding to individual PAs are indicated by boxes with light yellow background color. All the factor nodes, variable nodes, and messages related to the agent state are represented in red boldface style, those related to the legacy PF states are represented by the blue parts contained in dashed boxes, those related to the new PF states are represented by the magenta parts contained in dotted boxes, and those related to loopy BP DA are represented by the green parts contained in dashed-dotted boxes. The following short notations are used: $K \triangleq K_n^{(j)}$, $M \triangleq M_n^{(j)}$, $\boldsymbol{x} \triangleq \boldsymbol{x}_n$, $\tilde{\boldsymbol{y}}_k \triangleq \tilde{\boldsymbol{y}}_{k,n}^{(j)}$, $\breve{\boldsymbol{y}}_m \triangleq \breve{\boldsymbol{y}}_{m,n}^{(j)}$, $b_m \triangleq b_{m,n}^{(j)}$, $c_k \triangleq c_{k,n}^{(j)}$, $\psi_{k,m} \triangleq \psi\big(c_{k,n}^{(j)}, b_{m,n}^{(j)}\big)$, $g_k \triangleq g\big(\boldsymbol{x}_n, \tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}, c_{k,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big)$, $h_m \triangleq h\big(\boldsymbol{x}_n, \breve{\boldsymbol{a}}_{m,n}^{(j)}, \breve{r}_{m,n}^{(j)}, b_{m,n}^{(j)}; \boldsymbol{z}_n^{(j)}\big)$, $f \triangleq f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$, $f_k \triangleq f\big(\tilde{\boldsymbol{y}}_{k,n}^{(j)}\big|\boldsymbol{y}_{k,n-1}^{(j)}\big)$, $\alpha \triangleq \alpha(\boldsymbol{x}_n)$, $\alpha_k \triangleq \alpha_k\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}\big)$, $q^- \triangleq q(\boldsymbol{x}_{n-1})$, $q \triangleq q(\boldsymbol{x}_n)$, $\tilde{q}_k^- \triangleq \tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n-1}^{(j)}, \tilde{r}_{k,n-1}^{(j)}\big)$, $\tilde{q}_k \triangleq \tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}\big)$, $\breve{q}_m \triangleq \breve{q}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, \breve{r}_{m,n}^{(j)}\big)$, $\beta_k \triangleq \beta\big(c_{k,n}^{(j)}\big)$, $\xi_m \triangleq \xi\big(b_{m,n}^{(j)}\big)$, $\eta_k \triangleq \eta\big(c_{k,n}^{(j)}\big)$, $\varsigma_m \triangleq \varsigma\big(b_{m,n}^{(j)}\big)$, $\nu_{m,k} \triangleq \nu_{m\to k}^{(p)}\big(c_{k,n}^{(j)}\big)$, $\zeta_{k,m} \triangleq \zeta_{k\to m}^{(p)}\big(b_{m,n}^{(j)}\big)$, $\gamma_k^j \triangleq \gamma_k^{(j)}(\boldsymbol{x}_n)$, $\gamma_k \triangleq \gamma\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}\big)$, and $\phi_m \triangleq \phi\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, \breve{r}_{m,n}^{(j)}\big)$.

This estimator involves the posterior pdf $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$. Furthermore, detecting (i.e., determining the existence of) PF $k \in \mathcal{K}_n^{(j)}$ at time n is based on the posterior existence probability $p\big(r_{k,n}^{(j)} = 1\big|\boldsymbol{z}_{1:n}\big)$. This probability can be obtained from the posterior pdf of the augmented PF state, $f\big(\boldsymbol{y}_{k,n}^{(j)}\big|\boldsymbol{z}_{1:n}\big) = f\big(\boldsymbol{a}_{k,n}^{(j)}, r_{k,n}^{(j)}\big|\boldsymbol{z}_{1:n}\big)$, by a marginalization, i.e.,

$$p\big(r_{k,n}^{(j)}=1\big|\boldsymbol{z}_{1:n}\big) = \int f\big(\boldsymbol{a}_{k,n}^{(j)}, r_{k,n}^{(j)}=1\big|\boldsymbol{z}_{1:n}\big)\,\mathrm{d}\boldsymbol{a}_{k,n}^{(j)}. \tag{19}$$

Then PF k is defined to be detected at time n if $p\big(r_{k,n}^{(j)}=1\big|\boldsymbol{z}_{1:n}\big) > P_{\text{det}}$, where P_{det} is a detection threshold. The states $\boldsymbol{a}_{k,n}^{(j)}$ of the detected PFs are finally estimated as

$$\hat{\boldsymbol{a}}_{k,n}^{(j)\mathrm{MMSE}} \triangleq \int \boldsymbol{a}_{k,n}^{(j)} f\big(\boldsymbol{a}_{k,n}^{(j)} \big| r_{k,n}^{(j)}=1, \boldsymbol{z}_{1:n}\big)\, \mathrm{d}\boldsymbol{a}_{k,n}^{(j)}, \tag{20}$$

where

$$f\big(\boldsymbol{a}_{k,n}^{(j)} \big| r_{k,n}^{(j)}=1, \boldsymbol{z}_{1:n}\big) = \frac{f\big(\boldsymbol{a}_{k,n}^{(j)}, r_{k,n}^{(j)}=1 \big| \boldsymbol{z}_{1:n}\big)}{p\big(r_{k,n}^{(j)}=1 \big| \boldsymbol{z}_{1:n}\big)}. \tag{21}$$

The posterior existence probabilities $p\big(r_{k,n}^{(j)}=1\big|\boldsymbol{z}_{1:n}\big)$ in (19) are also used in a different context. To prevent an indefinite increase of the total number of PFs for PA j due to (2), i.e., $K_n^{(j)} = K_{n-1}^{(j)} + M_n^{(j)}$, a pruning of the PFs is employed. More specifically, PF k is retained only if $p\big(r_{k,n}^{(j)}=1\big|\boldsymbol{z}_{1:n}\big)$ exceeds a suitably chosen pruning threshold P_{prun}.

B. Message Passing Algorithm

The posterior pdfs $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$ and $f\big(\boldsymbol{a}_{k,n}^{(j)}, r_{k,n}^{(j)}\big|\boldsymbol{z}_{1:n}\big)$ involved in (18)–(21) are marginal pdfs of the joint posterior pdf $f(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}, \boldsymbol{c}_{1:n}, \boldsymbol{b}_{1:n}, \boldsymbol{m}_{1:n}|\boldsymbol{z}_{1:n})$ in (17). Because direct marginalization of the joint posterior pdf is infeasible, we calculate approximations of the marginal pdfs by using loopy (iterative) BP message passing [36] on the factor graph in Fig. 2. Since the factor graph contains loops, the resulting beliefs are only approximations of the respective posterior pdfs, and there is no canonical order in which the messages should be computed [36]. In our method, we choose the order according to the following rules: (i) Messages are not passed backward in time; (ii) *iterative* message passing is only performed for DA, and only for each time step and for each PA separately (i.e., in particular, for the loops connecting different PAs, we only perform a single message passing iteration); (iii) along an edge connecting an agent state variable node and a new PF state variable node, messages are only sent from the former to the latter. The resulting BP message passing algorithm is presented in what follows; cf. also the underlying factor graph in Fig. 2. Similarly to the "dummy pdfs" introduced in Section III-A, we consider BP messages $\phi\big(\boldsymbol{y}_{k,n}^{(j)}\big) = \phi\big(\boldsymbol{a}_{k,n}^{(j)}, r_{k,n}^{(j)}\big)$ also for the non-existing PF states, i.e., for $r_{k,n}^{(j)}=0$. We define these messages by setting $\phi\big(\boldsymbol{a}_{k,n}^{(j)}, 0\big) = \phi_{k,n}^{(j)}$ (note that these BP messages are not pdfs and thus are not required to integrate to 1).

First, a prediction step is performed. The prediction message for the agent state is obtained as

$$\alpha(\boldsymbol{x}_n) = \int f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})\, q(\boldsymbol{x}_{n-1})\, \mathrm{d}\boldsymbol{x}_{n-1}, \tag{22}$$

and the prediction message for the legacy PFs is obtained as

$$\alpha_k\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)}\big) = \sum_{r_{k,n-1}^{(j)} \in \{0,1\}} \int f\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, \tilde{r}_{k,n}^{(j)} \big| \boldsymbol{a}_{k,n-1}^{(j)}, r_{k,n-1}^{(j)}\big)\, \tilde{q}\big(\boldsymbol{a}_{k,n-1}^{(j)}, r_{k,n-1}^{(j)}\big)\, \mathrm{d}\boldsymbol{a}_{k,n-1}^{(j)}, \tag{23}$$

where the beliefs of the mobile agent state, $q(\boldsymbol{x}_{n-1})$, and of the PF states, $\tilde{q}\big(\boldsymbol{a}_{k,n-1}^{(j)},1\big)$, were calculated at the preceding time $n-1$. Inserting (4) and (5) for $f\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},\tilde{r}_{k,n}^{(j)}\big|\boldsymbol{a}_{k,n-1}^{(j)},r_{k,n-1}^{(j)}=1\big)$ and $f\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},\tilde{r}_{k,n}^{(j)}\big|\boldsymbol{a}_{k,n-1}^{(j)},r_{k,n-1}^{(j)}=0\big)$, respectively, we obtain for $r_{k,n}^{(j)}=1$

$$\alpha_k\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},1\big)=P_{\mathrm{s}}\int f\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}\big|\boldsymbol{a}_{k,n-1}^{(j)}\big)\,\tilde{q}\big(\boldsymbol{a}_{k,n-1}^{(j)},1\big)\,\mathrm{d}\boldsymbol{a}_{k,n-1}^{(j)}, \tag{24}$$

and for $r_{k,n}^{(j)}=0$

$$\alpha_{k,n}^{(j)}=(1-P_{\mathrm{s}})\int\tilde{q}\big(\boldsymbol{a}_{k,n-1}^{(j)},1\big)\,\mathrm{d}\boldsymbol{a}_{k,n-1}^{(j)}+\tilde{q}_{k,n-1}^{(j)}, \tag{25}$$

where $\alpha_{k,n}^{(j)}\triangleq\int\alpha_k\big(\boldsymbol{a}_{k,n}^{(j)},0\big)\,\mathrm{d}\boldsymbol{a}_{k,n}^{(j)}$ and $\tilde{q}_{k,n-1}^{(j)}\triangleq\int\tilde{q}\big(\boldsymbol{a}_{k,n-1}^{(j)},0\big)\,\mathrm{d}\boldsymbol{a}_{k,n-1}^{(j)}$.

After the prediction step, the following calculations are performed for all legacy PFs $k\in\mathcal{K}_{n-1}^{(j)}$ and for all new PFs $m\in\mathcal{M}_n^{(j)}$ for all PAs $j\in\{1,\ldots,J\}$ in parallel:

1) *Measurement evaluation for legacy PFs*: The messages $\beta\big(c_{k,n}^{(j)}\big)$ passed to the variable nodes corresponding to the feature-oriented DA variables $c_{k,n}^{(j)}$ (cf. Fig. 2) are calculated as

$$\beta\big(c_{k,n}^{(j)}\big)=\iint\alpha_k\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},1\big)\,\alpha(\boldsymbol{x}_n)\,g\big(\boldsymbol{x}_n,\tilde{\boldsymbol{a}}_{k,n}^{(j)},1,c_{k,n}^{(j)};\boldsymbol{z}_n^{(j)}\big)\,\mathrm{d}\boldsymbol{x}_n\,\mathrm{d}\tilde{\boldsymbol{a}}_{k,n}^{(j)}+1\big(c_{k,n}^{(j)}\big)\,\alpha_{k,n}^{(j)}. \tag{26}$$

2) *Measurement evaluation for new PFs*: The messages $\xi\big(b_{m,n}^{(j)}\big)$ passed to the variable nodes corresponding to the measurement-oriented DA variables $b_{m,n}^{(j)}$ are calculated as

$$\xi\big(b_{m,n}^{(j)}\big)=\sum_{\breve{r}_{m,n}^{(j)}\in\{0,1\}}\iint h\big(\boldsymbol{x}_n,\breve{\boldsymbol{a}}_{m,n}^{(j)},\breve{r}_{m,n}^{(j)},b_{m,n}^{(j)};\boldsymbol{z}_n^{(j)}\big)\,\alpha(\boldsymbol{x}_n)\,\mathrm{d}\boldsymbol{x}_n\,\mathrm{d}\breve{\boldsymbol{a}}_{m,n}^{(j)}. \tag{27}$$

Using the expression of $h\big(\boldsymbol{x}_n,\breve{\boldsymbol{a}}_{m,n}^{(j)},\breve{r}_{m,n}^{(j)},b_{m,n}^{(j)};\boldsymbol{z}_n^{(j)}\big)$ found in Section IV-A, Eq. (27) is easily seen to simplify to $\xi\big(b_{m,n}^{(j)}\big)=1$ for $b_{m,n}^{(j)}\in\mathcal{K}_{n-1}^{(j)}$, and for $b_{m,n}^{(j)}=0$ it becomes

$$\xi\big(b_{m,n}^{(j)}\big)=1+\frac{\mu_{\mathrm{n},n}^{(j)}}{\mu_{\mathrm{FA}}^{(j)}\,f_{\mathrm{FA}}\big(\boldsymbol{z}_{m,n}^{(j)}\big)}\iint\alpha(\boldsymbol{x}_n)\,f\big(\boldsymbol{z}_{m,n}^{(j)}\big|\boldsymbol{x}_n,\breve{\boldsymbol{a}}_{m,n}^{(j)}\big)\,f_{\mathrm{n},n}\big(\breve{\boldsymbol{a}}_n^{(j)}\big|\boldsymbol{x}_n\big)\,\mathrm{d}\boldsymbol{x}_n\,\mathrm{d}\breve{\boldsymbol{a}}_{m,n}^{(j)}. \tag{28}$$

3) *Iterative data association*: Next, from $\beta\big(c_{k,n}^{(j)}\big)$ and $\xi\big(b_{m,n}^{(j)}\big)$, messages $\eta\big(c_{k,n}^{(j)}\big)$ and $\varsigma\big(b_{m,n}^{(j)}\big)$ are obtained by means of loopy (iterative) BP. First, for each measurement $m\in\mathcal{M}_n^{(j)}$, messages $\nu_{m\to k}^{(p)}\big(c_{k,n}^{(j)}\big)$ and $\zeta_{k\to m}^{(p)}\big(b_{m,n}^{(j)}\big)$ are calculated iteratively according to [37], [40]

$$\nu_{m\to k}^{(p)}\big(c_{k,n}^{(j)}\big)=\sum_{b_{m,n}^{(j)}=0}^{K_{n-1}^{(j)}}\xi\big(b_{m,n}^{(j)}\big)\,\psi\big(c_{k,n}^{(j)},b_{m,n}^{(j)}\big)\prod_{k'\in\mathcal{K}_{n-1}^{(j)}\setminus\{k\}}\zeta_{k'\to m}^{(p-1)}\big(b_{m,n}^{(j)}\big) \tag{29}$$

$$\zeta_{k\to m}^{(p)}\big(b_{m,n}^{(j)}\big)=\sum_{c_{k,n}^{(j)}=0}^{M_n^{(j)}}\beta\big(c_{k,n}^{(j)}\big)\,\psi\big(c_{k,n}^{(j)},b_{m,n}^{(j)}\big)\prod_{m'\in\mathcal{M}_n^{(j)}\setminus\{m\}}\nu_{m'\to k}^{(p)}\big(c_{k,n}^{(j)}\big), \tag{30}$$

for $k=1,\ldots,K_{n-1}^{(j)}$, $m=1,\ldots,M_n^{(j)}$, and iteration index $p=1,\ldots,P$. The recursion defined by (29) and (30) is initialized (for $p=0$) by $\zeta_{k\to m}^{(0)}\big(b_{m,n}^{(j)}\big)=\sum_{c_{k,n}^{(j)}=0}^{M_n^{(j)}}\beta\big(c_{k,n}^{(j)}\big)\psi\big(c_{k,n}^{(j)},b_{m,n}^{(j)}\big)$. Then, after the last iteration $p=P$, the messages $\eta\big(c_{k,n}^{(j)}\big)$ and $\varsigma\big(b_{m,n}^{(j)}\big)$ are calculated as

$$\eta\big(c_{k,n}^{(j)}\big)=\prod_{m\in\mathcal{M}_n^{(j)}}\nu_{m\to k}^{(P)}\big(c_{k,n}^{(j)}\big),\qquad \varsigma\big(b_{m,n}^{(j)}\big)=\prod_{k\in\mathcal{K}_{n-1}^{(j)}}\zeta_{k\to m}^{(P)}\big(b_{m,n}^{(j)}\big). \tag{31}$$

4) *Measurement update for the agent*: From $\eta\big(c_{k,n}^{(j)}\big)$, $\alpha_k\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},1\big)$, and $\alpha_{k,n}^{(j)}$, the message $\gamma_k^{(j)}(\boldsymbol{x}_n)$ related to the agent is calculated as

$$\gamma_k^{(j)}(\boldsymbol{x}_n)=\sum_{c_{k,n}^{(j)}=0}^{M_n^{(j)}}\eta\big(c_{k,n}^{(j)}\big)\int\alpha_k\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},1\big)\,g\big(\boldsymbol{x}_n,\tilde{\boldsymbol{a}}_{k,n}^{(j)},1,c_{k,n}^{(j)};\boldsymbol{z}_n^{(j)}\big)\,\mathrm{d}\tilde{\boldsymbol{a}}_{k,n}^{(j)}+\eta\big(c_{k,n}^{(j)}=0\big)\,\alpha_{k,n}^{(j)}. \tag{32}$$

5) *Measurement update for legacy PFs*: Similarly, the messages $\gamma\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},\tilde{r}_{k,n}^{(j)}\big)$ related to the legacy PFs are calculated as

$$\gamma\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},1\big)=\sum_{c_{k,n}^{(j)}=0}^{M_n^{(j)}}\eta\big(c_{k,n}^{(j)}\big)\int\alpha(\boldsymbol{x}_n)\,g\big(\boldsymbol{x}_n,\tilde{\boldsymbol{a}}_{k,n}^{(j)},1,c_{k,n}^{(j)};\boldsymbol{z}_n^{(j)}\big)\,\mathrm{d}\boldsymbol{x}_n \tag{33}$$

$$\gamma_{k,n}^{(j)}\triangleq\gamma\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},0\big)=\eta\big(c_{k,n}^{(j)}=0\big). \tag{34}$$

6) *Measurement update for new PFs*: Finally, the messages $\phi\big(\breve{\boldsymbol{a}}_{m,n}^{(j)},\breve{r}_{m,n}^{(j)}\big)$ related to the new PFs are calculated as

$$\phi\big(\breve{\boldsymbol{a}}_{m,n}^{(j)},1\big)=\varsigma\big(b_{m,n}^{(j)}=0\big)\int\alpha(\boldsymbol{x}_n)\,h\big(\boldsymbol{x}_n,\breve{\boldsymbol{a}}_{m,n}^{(j)},1,0;\boldsymbol{z}_n^{(j)}\big)\,\mathrm{d}\boldsymbol{x}_n \tag{35}$$

$$\phi_{m,n}^{(j)}\triangleq\phi\big(\breve{\boldsymbol{a}}_{k,n}^{(j)},0\big)=\sum_{b_{m,n}^{(j)}=0}^{K_{n-1}^{(j)}}\varsigma\big(b_{m,n}^{(j)}\big). \tag{36}$$

Once these messages are available, the beliefs approximating the desired marginal posterior pdfs are obtained. The belief for the agent state is given, up to a normalization factor, by

$$q(\boldsymbol{x}_n)\propto\alpha(\boldsymbol{x}_n)\prod_{j=1}^{J}\prod_{k\in\mathcal{K}_{n-1}^{(j)}}\gamma_k^{(j)}(\boldsymbol{x}_n). \tag{37}$$

This belief (after normalization) provides an approximation of the marginal posterior pdf $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$, and it is used instead of $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$ in (18). Furthermore, the beliefs $\tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},\tilde{r}_{k,n}^{(j)}\big)$ for the augmented states of the legacy PFs, $\tilde{\boldsymbol{y}}_{k,n}^{(j)}=\big[\tilde{\boldsymbol{a}}_{k,n}^{(j)\mathrm{T}}\ \tilde{r}_{k,n}^{(j)}\big]^{\mathrm{T}}$, are calculated as

$$\tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},1\big)\propto\alpha_k\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},1\big)\gamma\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},1\big),\qquad \tilde{q}_{k,n}^{(j)}\triangleq\tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)},0\big)\propto\alpha_{k,n}^{(j)}\gamma_{k,n}^{(j)}, \tag{38}$$

and the beliefs $\breve{q}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, \breve{r}_{m,n}^{(j)}\big)$ for the augmented states of the new PFs, $\breve{\boldsymbol{y}}_{m,n}^{(j)} = \big[\breve{\boldsymbol{a}}_{m,n}^{(j)\mathrm{T}}\ \breve{r}_{m,n}^{(j)}\big]^{\mathrm{T}}$, as

$$\breve{q}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, 1\big) \propto \phi\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, 1\big), \qquad \breve{q}_{m,n}^{(j)} \triangleq \breve{q}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, 0\big) \propto \phi_{m,n}^{(j)}. \tag{39}$$

In particular, $\tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1\big)$ and $\breve{q}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, 1\big)$ approximate the marginal posterior pdf $f\big(\boldsymbol{a}_{k',n}^{(j)}, r_{k',n}^{(j)} = 1 \big| \boldsymbol{z}_{1:n}\big)$, where $k' \in \mathcal{K}_{n-1}^{(j)} \cup \mathcal{M}_n^{(j)}$ (assuming an appropriate index mapping between k and m on the one hand and k' on the other), and they are used in (19)–(21).

A summary of the overall proposed BP-SLAM algorithm is provided in Algorithm 1, and a flowchart is available online at https://www2.spsc.tugraz.at/people/eriklei/BP-SLAM/. A computationally feasible approximate calculation of the various messages and beliefs can be based on the sequential Monte Carlo (particle-based) implementation approach introduced in [39], [51]. In our case, the sequential Monte Carlo implementation uses a "stacked state" [51] comprising the agent state and the PF states. The resulting complexity scales only linearly in the number of particles. MATLAB code for this particle-based implementation is available online at https://www2.spsc.tugraz.at/people/eriklei/BP-SLAM/.

ALGORITHM 1: BP-SLAM

***Initialization*:**

The recursive algorithm is initialized at time $n=1$ by $f(\boldsymbol{x}_1)$ and $\mathcal{K}_1^{(j)} = \emptyset$, $j \in \{1, \ldots, J\}$.

***Step 1—Prediction*:**

Calculate $\alpha(\boldsymbol{x}_n)$ from $q(\boldsymbol{x}_{n-1})$ according to (22), and calculate $\alpha_k\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1\big)$ and $\alpha_{k,n}^{(j)}$ from $\tilde{q}\big(\boldsymbol{a}_{k,n-1}^{(j)}, r_{k,n-1}^{(j)}\big)$, $k \in \mathcal{K}_{n-1}^{(j)}$, $j \in \{1, \ldots, J\}$ according to (24), (25).

***Step 2—Measurement evaluation*:**

For legacy PFs: Calculate $\beta\big(c_{k,n}^{(j)}\big)$, $k \in \mathcal{K}_{n-1}^{(j)}$, $j \in \{1, \ldots, J\}$ according to (26).

For new PFs: Calculate $\xi\big(b_{m,n}^{(j)}\big)$, $m \in \mathcal{M}_n^{(j)}$, $j \in \{1, \ldots, J\}$ according to (27), (28).

***Step 3—Iterative data association*:**

Calculate $\eta\big(c_{k,n}^{(j)}\big)$, $k \in \mathcal{K}_{n-1}^{(j)}$, $j \in \{1, \ldots, J\}$ and $\varsigma\big(b_{m,n}^{(j)}\big)$, $m \in \mathcal{M}_n^{(j)}$, $j \in \{1, \ldots, J\}$ according to (29)–(31).

***Step 4—Measurement update*:**

For the mobile agent: Calculate $\gamma_k^{(j)}(\boldsymbol{x}_n)$, $k \in \mathcal{K}_{n-1}^{(j)}$, $j \in \{1, \ldots, J\}$ according to (32).

For legacy PFs: Calculate $\gamma\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1\big)$ and $\gamma_{k,n}^{(j)}$, $k \in \mathcal{K}_{n-1}^{(j)}$, $j \in \{1, \ldots, J\}$ according to (33), (34).

For new PFs: Calculate $\phi\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, 1\big)$ and $\phi_{m,n}^{(j)}$, $m \in \mathcal{M}_n^{(j)}$, $j \in \{1, \ldots, J\}$ according to (35), (36).

***Step 5—Belief calculation*:**

For the mobile agent: Calculate $q(\boldsymbol{x}_n)$ according to (37).

For legacy PFs: Calculate $\tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1\big)$ and $\tilde{q}_{k,n}^{(j)}$, $k \in \mathcal{K}_{n-1}^{(j)}$, $j \in \{1, \ldots, J\}$ according to (38).

For new PFs: Calculate $\breve{q}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, 1\big)$ and $\breve{q}_{m,n}^{(j)}$, $m \in \mathcal{M}_n^{(j)}$, $j \in \{1, \ldots, J\}$ according to (39).

***Step 6—Pruning*:**

Determine the set $\tilde{\mathcal{K}}_n^{(j)} = \mathcal{K}_{n-1}^{(j)} \cup \mathcal{M}_n^{(j)}$.

For all $j \in \{1, \ldots, J\}$, reinterpret/reindex the beliefs $\tilde{q}\big(\tilde{\boldsymbol{a}}_{k',n}^{(j)}, 1\big)$ and $\tilde{q}_{k',n}^{(j)}$, $k' \in \tilde{\mathcal{K}}_{n-1}^{(j)}$ and the beliefs $\breve{q}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, 1\big)$ and $\breve{q}_{m,n}^{(j)}$, $m \in \mathcal{M}_n^{(j)}$ as beliefs $\tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1\big)$ and $\tilde{q}_{k,n}^{(j)}$ of the legacy PFs $k \in \tilde{\mathcal{K}}_n^{(j)}$.

For all $j \in \{1, \ldots, J\}$, calculate estimates $\hat{p}\big(r_{k,n}^{(j)}=1 \big| \boldsymbol{z}_{1:n}\big)$ of the existence probabilities $p\big(r_{k,n}^{(j)}=1 \big| \boldsymbol{z}_{1:n}\big)$, $k \in \tilde{\mathcal{K}}_n^{(j)}$ according to (19) with $f\big(\boldsymbol{a}_{k,n}^{(j)}, r_{k,n}^{(j)}=1 \big| \boldsymbol{z}_{1:n}\big)$ replaced by $\tilde{q}\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)}, 1\big)$ or $\breve{q}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}, 1\big)$.

For all $j \in \{1, \ldots, J\}$, determine the set $\hat{\mathcal{K}}_n^{(j)}$ of legacy PFs k for which $\hat{p}\big(r_{k,n}^{(j)}=1 \big| \boldsymbol{z}_{1:n}\big) > P_{\text{prun}}$.

***Step 7—Detection and estimation*:**

Calculate an agent state estimate $\hat{\boldsymbol{x}}_n$ according to (18) with $f(\boldsymbol{x}_n|\boldsymbol{z}_{1:n})$ replaced by the normalized $q(\boldsymbol{x}_n)$.

For all $j \in \{1, \ldots, J\}$, determine the set $\hat{\mathcal{K}}_n^{(j)}$ of PFs k for which $\hat{p}\big(r_{k,n}^{(j)}=1 \big| \boldsymbol{z}_{1:n}\big) > P_{\text{det}}$.

For all $j \in \{1, \ldots, J\}$ and $k \in \hat{\mathcal{K}}_n^{(j)}$, calculate a PF state estimate $\hat{\boldsymbol{a}}_{k,n}^{(j)}$ according to (20) and (21).

VI. Experimental Results

To analyze the performance of the proposed BP-SLAM algorithm, we apply it to synthetic and real measurement data within two-dimensional (2-D) scenarios. The parameters involved in the algorithm and those used to generate the synthetic measurements are listed in Table I.

A. Analysis Setup

1) State-Evolution Model: The agent's state-transition pdf $f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$, with $\boldsymbol{x}_n = [\boldsymbol{p}_n^{\mathrm{T}} \; \boldsymbol{v}_n^{\mathrm{T}}]^{\mathrm{T}}$, is defined by a linear, near constant-velocity motion model [52, Sec. 6.3.2], i.e.,

$$\boldsymbol{x}_n = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{n-1} + \begin{bmatrix} \Delta T^2/2 & 0 \\ 0 & \Delta T^2/2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix} \boldsymbol{w}_n \,. \tag{40}$$

Here, $\Delta T = 1\,\mathrm{s}$ and the driving process $\boldsymbol{w}_n$ is independent across n, zero-mean, and Gaussian with covariance matrix $\boldsymbol{Q}_w = \sigma_w^2 \boldsymbol{I}_2$, where $\boldsymbol{I}_2$ denotes the 2×2 identity matrix. The PFs are static, i.e., the state-transition pdfs are given by $f\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)} \big| \boldsymbol{a}_{k,n-1}^{(j)}\big) = \delta\big(\tilde{\boldsymbol{a}}_{k,n}^{(j)} - \boldsymbol{a}_{k,n-1}^{(j)}\big)$, where $\delta(\cdot)$ is the Dirac delta function. However, in our implementation of the BP-SLAM algorithm, we introduced a small driving process in the PF state-evolution model for the sake of numerical stability. Accordingly, the state evolution is modeled as $\tilde{\boldsymbol{a}}_{k,n}^{(j)} = \boldsymbol{a}_{k,n-1}^{(j)} + \boldsymbol{\omega}_{k,n}^{(j)}$, where $\boldsymbol{\omega}_{k,n}^{(j)}$ is independent across k, n, and j, zero-mean, and Gaussian with covariance matrix $\boldsymbol{\Omega}_{k,n}^{(j)} = \sigma_a^2 \boldsymbol{I}_2$.

2) Measurement Model: In contrast to usual SLAM setups [1], our measurement model is solely based on MPC ranges, i.e., it does not exploit bearing information or information derived from inertial measurement unit sensors. The scalar range measurements $z_{m,n}^{(j)}$ are modeled as

$$z_{m,n}^{(j)} = \big\| \boldsymbol{p}_n - \boldsymbol{a}_{k,n}^{(j)} \big\| + \nu_{m,n}^{(j)} \,, \tag{41}$$

TABLE I: Simulation parameters.

Parameters involved in the proposed algorithm												
σ_w	σ_a	$\sigma_{a,1}$	σ_m	$\mu_{\text{FA}}^{(j)}$	μ_{b}	$\mu_{\text{n},1}$	P_{s}	P_{d}	P_{det}	P_{prun}	#particles	#simu.runs
$0.01\,\text{m/s}^2$	$10^{-4}\,\text{m}$ or $0.5\cdot 10^{-2}\,\text{m}$	$10^{-3}\,\text{m}$	$0.15\,\text{m}$	1 or 2	10^{-4}	6	0.999	0.95 or 0.5	0.5	10^{-4}	10^5 or $3\cdot 10^4$	100 or 30

Parameters used to generate the synthetic measurements		
$\mu_{\text{FA}}^{(j)}$	P_{d}	$\sigma_{m,n}^{(j)}$
1 or 2	0.95 or 0.5	0.1m

where the measurement noise $\nu_{m,n}^{(j)}$ is independent across m, n, and j, zero-mean, and Gaussian with variance $\sigma_{m,n}^{(j)2}$. The measurement model (41) determines the likelihood function factors $f\big(\boldsymbol{z}_{m,n}^{(j)}\big|\boldsymbol{x}_n,\tilde{\boldsymbol{a}}_{k,n}^{(j)}\big)$ and $f\big(\boldsymbol{z}_{m,n}^{(j)}\big|\boldsymbol{x}_n,\breve{\boldsymbol{a}}_{m,n}^{(j)}\big)$ in (9). However, we emphasize that the BP-SLAM algorithm can be extended to measurement models involving bearing measurements (AoAs and/or AoDs) or measurements from inertial measurement unit sensors. Such an extension would further improve the robustness of our approach.

3) Common Simulation Parameters: The following implementation parameters are used for both synthetic and real measurements, see also Table I. (The other parameters will be described in Sections VI-B and VI-C.) We use the floor plan shown in Fig. 1, with two static PAs at positions $\boldsymbol{a}_{1,n}^{(1)}$ and $\boldsymbol{a}_{1,n}^{(2)}$. The false alarm pdf $f_{\text{FA}}\big(z_{m,n}^{(j)}\big)$ is uniform on $[0\text{m},30\text{m}]$. The conditional pdf $f_{\text{n},n}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}\big|\boldsymbol{x}_n\big)$ and the mean number $\mu_{\text{n},n}$ of newly detected features are inferred online by a PHD filter using the birth pdf $f_{\text{b},n}\big(\breve{\boldsymbol{a}}_{m,n}^{(j)}\big|\boldsymbol{x}_n\big)$ and the mean number of newborn features μ_{b} (see [34]). The birth pdf is uniform on the region of interest (ROI), which is a circular disk of radius 30m around the center of the floor plan shown in Fig. 1. At time $n=1$, newly detected features are initialized by $\mu_{\text{n},1}^{(j)}=6$ and an initial pdf $f_{\text{n},1}\big(\breve{\boldsymbol{a}}_{m,1}^{(j)}\big|\boldsymbol{x}_1\big)$ that is uniform on the ROI. The detection probability is set to a constant value for all PFs and PAs, i.e. $P_{\text{d}}^{(j)}\big(\boldsymbol{x}_n,\boldsymbol{a}_{k,n}^{(j)}\big)=P_{\text{d}}$. The values of the parameters μ_{b}, $\mu_{\text{n},1}^{(j)}$, σ_w, σ_a, P_{s}, P_{det}, and P_{prun} are given in Table I.

Our implementation of the BP-SLAM algorithm uses a particle-based representation of messages and beliefs similarly to [39], [51]. The particles for the initial PA states are drawn from the 2-D Gaussian distributions $\mathcal{N}\big(\boldsymbol{a}_{1,1}^{(j)},\sigma_{a,1}^2\boldsymbol{I}_2\big)$, where $\boldsymbol{a}_{1,1}^{(j)}$ is the position of PA $j\in\{1,2\}$ and $\sigma_{a,1}=0.001\text{m}$. This highly informative prior implies that the initial PA positions are effectively known. The particles for the initial agent state are drawn from a 4-D uniform distribution with center $\boldsymbol{x}_1=[\boldsymbol{p}_1^{\text{T}}\ 0\ 0]^{\text{T}}$, where $\boldsymbol{p}_1$ is the starting position of the agent trajectory, and with the support of each component about the respective center given by $[-\lambda,\lambda]$. Here, λ is 0.5 for

our simulations based on synthetic data in Section VI-B1 and for our experiments using real measurements in Section VI-C, while it is 0.1 for our simulations based on synthetic data in Section VI-B2; the physical dimension of λ is m (position) or m/s (velocity). We note that the BP-SLAM algorithm performs well even without any prior information about the initial states of the mobile agent and the PAs. However, because only relative range measurements are used, the estimated feature map and agent trajectory would then contain an arbitrary translation and rotation relative to the true positions. Finally, the number P of message passing iterations for DA is limited by the termination condition $\big[\sum_{k\in\mathcal{K}_{n-1}^{(j)}}\sum_{m\in\mathcal{M}_n^{(j)}}\big(\nu_{m\to k}^{(p)}(c_{k,n}^{(j)})-\nu_{m\to k}^{(p-1)}(c_{k,n}^{(j)})\big)^2\big]^{1/2}<10^{-7}$ (cf. (29)) or by the maximum number $P_{\max}=1000$.

B. Results for Synthetic Measurements

For our simulations based on synthetic measurements, we used the common simulation parameters described above. The range measurements were generated with noise standard deviation $\sigma_{m,n}^{(j)}=0.1\text{m}$. However, for numerical robustness, the BP-SLAM algorithm used noise standard deviation $\sigma_m=1.5\cdot\sigma_{m,n}^{(j)}$. The standard deviation of the driving process in the PF state-evolution model was $\sigma_a=10^{-4}\,\text{m}$. We performed 100 simulation runs. In each run, we generated with detection probability P_d noisy ranges $z_{m,n}^{(j)}$ according to (41). Evaluation of (41) was based on the fixed PA positions $\boldsymbol{a}_{1,n}^{(1)}$, $\boldsymbol{a}_{1,n}^{(2)}$ and the fixed VA positions $\boldsymbol{a}_{l,n}^{(j)}\in\mathbb{R}^2$, $l=2,\ldots,L_n^{(j)}$ for $j=1,2$ (shown in Fig. 1), where $L_n^{(1)}=6$ and $L_n^{(2)}=5$ are the numbers of features (PA plus VAs). In addition, false alarm measurements $z_{m,n}^{(j)}$ were generated as described below.

1) Comparison of Different Parameter Settings: We considered three different parameter settings dubbed SLAM 1, SLAM 2, and SLAM 3. In SLAM 1 and SLAM 2, we used detection probability $P_\text{d}=0.95$ and mean number of false alarms $\mu_\text{FA}^{(j)}=1$. The posterior pdfs of the agent state, of the legacy PF states, and of the new PF states were each represented by 100.000 (SLAM 1) or by 30.000 (SLAM 2) particles.[1] In SLAM 3, we used $P_\text{d}=0.5$ and $\mu_\text{FA}^{(j)}=2$ to analyze the robustness of the BP-SLAM algorithm to extremely poor radio signal conditions, i.e., to scenarios characterized by a high probability that existing MPCs are not detected or nonexisting MPCs are detected; furthermore, the agent state and the PF states were each represented by 100.000 particles. For one exemplary simulation run, Fig. 3 illustrates the convergence of the

[1]The number of particles could be strongly reduced if, e.g., the MPCs' AoAs were used in addition to the range measurements, since this would decrease the effective support regions of the posterior pdfs. The number of particles could also be reduced by performing an adaptive adjustment of the detection probability P_d as in [30]. However, since the complexity of the BP-SLAM algorithm scales only linearly with the number of particles, the runtime of a MATLAB implementation on an Intel i7-6820HQ CPU is still below 0.4s per time step n even for the high numbers of particles we are using.

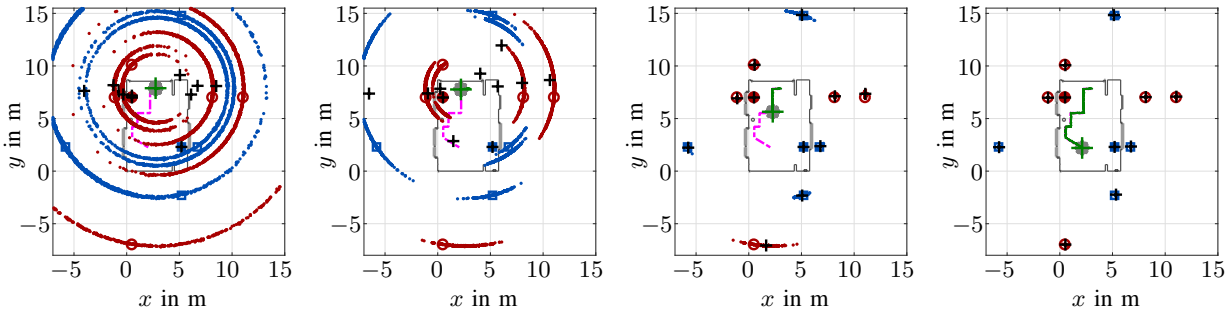


Fig. 3: Particle convergence. Particles representing the posterior pdfs of the state of the mobile agent (gray) and of the states of the detected PFs (red for PA 1, blue for PA 2) are shown at (a) $n=30$, (b) $n=90$, (c) $n=300$, and (d) $n=900$. The positions of PA 1 and PA 2 are indicated by a red bullet and a blue box, respectively, and the corresponding geometrically expected VA positions by red circles and blue squares. The green line represents the MMSE estimates of the mobile agent position, and the green cross the MMSE estimate of the currently last mobile agent position. The black crosses indicate the MMSE estimates of the positions of the detected PFs.

posterior pdfs of the PF positions to the true feature (PA and VA) positions by displaying the respective particles at times $n=30, 90, 300, 900$. These results demonstrate that the BP-SLAM algorithm is able to cope with highly multimodal distributions and with measurements conveying only very limited information at each time step (since only range measurements are used).

Fig. 4 shows the root mean square error (RMSE) of the estimated time-varying agent position, the average numbers of detected PFs for the two PAs, and the mean optimal subpattern assignment (MOSPA)[2] errors [53] for the two PAs and the associated VAs, all versus time n. These results were obtained by averaging over the 100 simulation runs. The MOSPA errors are based on the Euclidean metric and use cutoff parameter 5m and order 1 [53]. It can be seen in Fig. 4(a) that the average agent position RMSE is mostly below 0.12m for all parameter settings and below 0.073m for SLAM 1 and SLAM 2. The average numbers of detected PFs in Figs. 4(b), (c) are effectively equal to the respective true numbers of features $L_n^{(1)}=6$ and $L_n^{(2)}=5$ for SLAM 1 and SLAM 2. For SLAM 3, the average numbers are slightly below the true numbers, which can be explained by the low detection probability $P_{\mathrm{d}}=0.5$ and the high mean number of false alarms $\mu_{\mathrm{FA}}^{(j)}=2$ relative to the true numbers of features. However, the performance of the BP-SLAM algorithm is still good, which suggests that the algorithm has a high level of robustness. The MOSPA errors for SLAM 1 are seen in Figs. 4(d), (e) to decrease with time until they are

[2]We remark that an alternative to the "classical" OSPA metric [53] is provided by the generalized OSPA metric proposed in [54], which does not normalize the OSPA error by the cardinality of the larger set and penalizes the cardinality error differently.

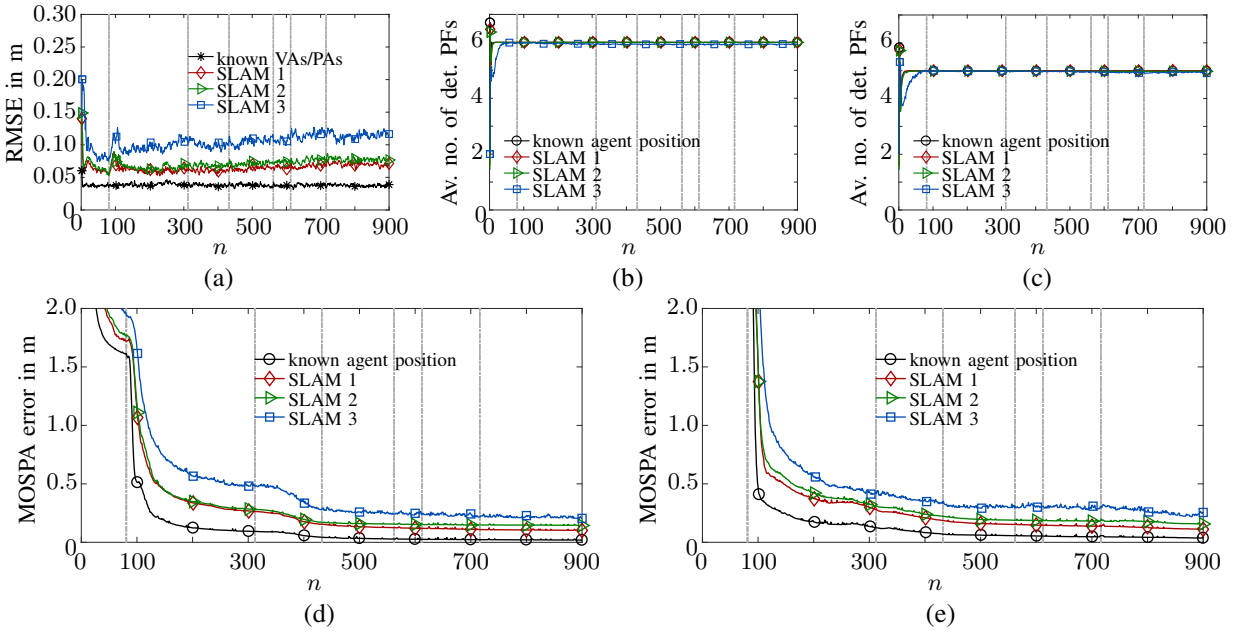


Fig. 4: Results for synthetic measurements: (a) Average agent position RMSE, (b) and (c) average number of detected PFs associated with PA 1 and 2, respectively, (d) and (e) MOSPA error for, respectively, PA 1 and 2 and the associated VAs. The curves labeled "known agent position" show the performance for the case where the agent position is known. Note that in (b) and (c) the curves labeled "known agent position," "SLAM 1," and "SLAM 2" coincide. The vertical dash-dotted lines indicate the times around which the mobile agent performs turns.

ultimately below 0.11m for both PAs. The position RMSEs of the individual PFs are at most 0.25m and in many cases around 0.05m. In general, the RMSE and MOSPA errors for SLAM 2 and SLAM 3 are slightly larger (ultimately 0.15m for SLAM 2 and 0.20m for SLAM 3).

As a performance benchmark for the accuracy of agent localization, we compare in Fig. 4(a) the average agent position RMSE of the BP-SLAM algorithm to that of the algorithm in [9], which also uses BP-based probabilistic DA but assumes knowledge of the feature map, i.e., of the PA/VA positions. In addition, as a performance benchmark for the accuracy of feature map estimation, we compare in Figs. 4(b)–(e) the average number of detected PFs and the MOSPA errors to those that would be obtained for the SLAM 1 parameter setting if the agent position was known at all times. These benchmarks provide bounds on the two main performance aspects of SLAM, i.e., accuracy in localization and mapping.

Finally, we considered the use of measurement gating, which is commonly employed in practical implementations to reduce computational complexity [31, Sec. 2.3.2]. With measurement gating, DA is performed only on those measurements that fall into given "gates" around predicted measurement values. We chose the gating threshold as $\gamma = 6.635$, which implies that the probability that feature-originated measurements are outside their corresponding gate is 10^{-2}

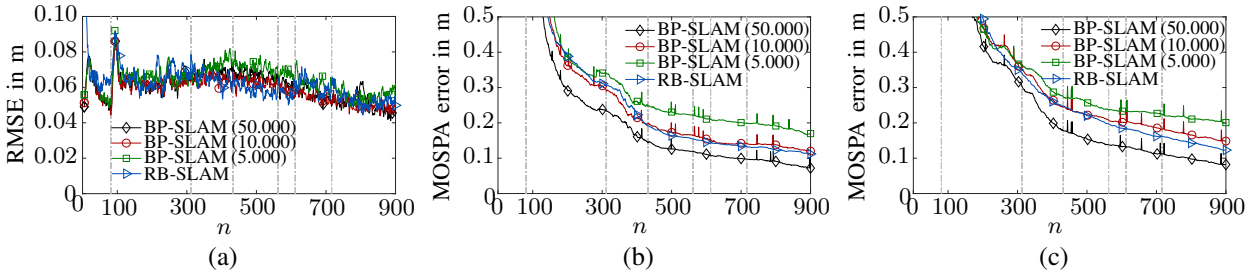


Fig. 5: Results for synthetic measurements; comparison between BP-SLAM and RB-SLAM [13]: (a) Average agent position RMSE, (b) and (c) MOSPA error for, respectively, PA 1 and 2 and the associated VAs. The vertical dash-dotted lines indicate the times around which the mobile agent performs turns.

[31, Table 2.3.2-1]. For SLAM 2, we measured the average runtime per time step n, averaged over the 900 time steps and 100 simulation runs, as 0.0929s without measurement gating and 0.0735s with measurement gating. Thus, measurement gating yielded a reduction of the average runtime by 20.1%, and this did not come at the cost of a noticeable performance loss in terms of agent position RMSE or MOSPA error. These results were obtained by using a MATLAB implementation on an Intel i5-4690 CPU. We note that besides measurement gating, another potential means of reducing complexity is the use of geometric data structures such as kd-trees [55]. For example, kd-trees were used in the RB-SLAM (FastSLAM) algorithm reported in [18]. However, in the particle-based implementation of the proposed BP-SLAM algorithm, due to the necessity of performing probabilistic DA, the weights representing the agent state and the feature states are updated by a weighted sum of measurements instead of a single measurement (as is done in established particle-based RB-SLAM algorithms, even when Monte Carlo-based DA is used [1], [18]). Due to this difference, to the best of our knowledge, geometric data structures such as kd-trees [55] cannot be directly applied to SLAM with probabilistic DA.

2) Comparison with Rao-Blackwellized SLAM: Next, we compare the accuracy and complexity (runtime) of our BP-SLAM algorithm with those of RB-SLAM [2], [18], which is an important state-of-the-art method. This comparison is interesting also because it analyzes the impact of neglecting the dependencies between the feature states and the dependencies between the feature states and the agent state, which is done in BP-SLAM but not in RB-SLAM. We note that RB-SLAM can be motivated by the idea of stacking the agent state and all the feature states into one high-dimensional joint state and using a particle filter to track the joint state. A direct implementation of this idea is typically infeasible due to the curse of dimensionality [1], which is further exacerbated by our range-only measurement model. Indeed, even the 100.000

particles used in simulation setup SLAM 1 would be totally insufficient to represent the joint state vector. The RB-SLAM algorithm avoids the curse of dimensionality by exploiting the *conditional* independencies between the agent state and the feature states and applying a Rao-Blackwellization to the joint state [18].

Because our measurement model is highly nonlinear, we adopt the RB-SLAM implementation proposed in [13], which employs particle filters instead of EKFs to estimate the feature states. As is commonly done in classical RB-SLAM, we perform Monte Carlo-based DA for each particle separately [2], [18]. The cassical RB-SLAM algorithm is very sensitive to missed detections and clutter measurements, especially for our range-only measurement model, which implies a strong DA uncertainty. Therefore, to avoid the risk of divergence in RB-SLAM, we assumed in our simulation that there are no missed detections or false alarms, i.e., we chose $P_{\mathrm{d}} = 1$ and $\mu_{\mathrm{FA}}^{(j)} = 0$. In BP-SLAM, each posterior state pdf (for the agent state, each legacy PF state, and each new PF state) was represented by 50.000, 10.000, or 5.000 particles. In RB-SLAM, the posterior pdfs of the agent state and of each feature state were represented by 50 and 10.000 particles, respectively.[3]

Fig. 5 shows the RMSE of the estimated agent position and the MOSPA errors for the two PAs and the associated VAs. It can be seen that the agent position RMSEs obtained with BP-SLAM and with RB-SLAM are generally quite similar. Also, the number of particles used in BP-SLAM (50.000, 10.000, or 5.000) does not influence the RMSE significantly. On the other hand, as may be expected, the MOSPA error of BP-SLAM is smaller for a larger number of particles. The MOSPA error of RB-SLAM is seen to be only slightly smaller than that of BP-SLAM using 10.000 particles (which is a good basis for a comparison, because also RB-SLAM uses 10.000 particles for each feature state). This is remarkable, in view of the fact that BP-SLAM assumes a priori conditional independence of the agent state and all the feature states. BP-SLAM using 50.000 particles even has a significantly smaller MOSPA error than RB-SLAM.

Table II shows the average runtimes per time step n (averaged over 900 time steps and 100 simulation runs) of MATLAB implementations of the various algorithms on an Intel i7-6820HQ CPU. It is seen that BP-SLAM is significantly less complex than RB-SLAM; in particular, the

[3]This choice of the number of particles differs significantly from that in [13]. Indeed, we used 10.000 particles—more than in [13]—for each feature state because of the challenging range-only measurement model. On the other hand, we used only 50 particles for the agent state to avoid an excessive complexity. (We note that the complexity of particle-based RB-SLAM scales linearly in the *product* of the number of agent particles and the total number of feature particles.) Our experimental results for the agent position RMSE shown in Fig. 5(a) demonstrate that 50 particles for the agent state are sufficient for good results.

TABLE II: Average runtimes of BP-SLAM and RB-SLAM per time step.

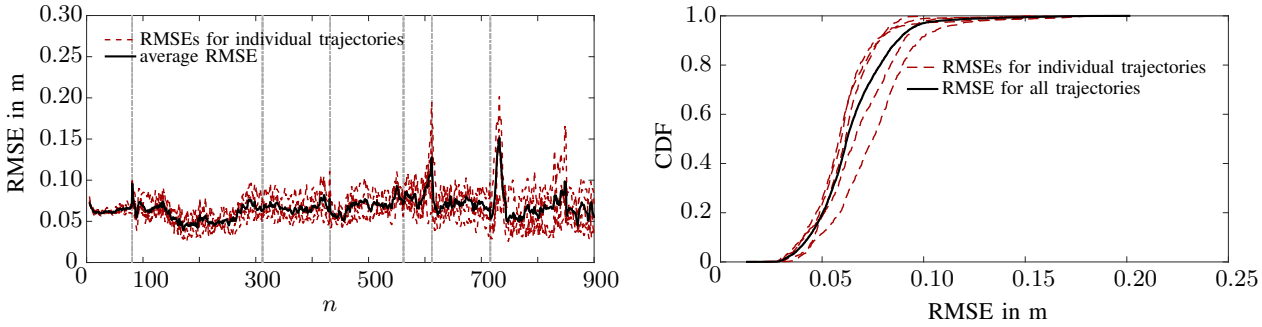
RB-SLAM	BP-SLAM (50.000)	BP-SLAM (10.000)	BP-SLAM (5.000)
0.49 s	0.146 s	0.037 s	0.028 s

runtime of BP-SLAM using 10.000 particles—i.e., the same number of particles as in RB-SLAM, leading to only slightly poorer performance as observed above—is smaller by a factor of more than 10. The significantly higher complexity of RB-SLAM is due to the fact that RB-SLAM calculates the distances of each agent state particle to each feature state particle. Consequently, the complexity of RB-SLAM scales linearly in the *product* of the number of agent particles and the total number of feature particles; by contrast, the complexity of BP-SLAM scales linearly in the number of all particles. We can thus conclude that the slightly improved accuracy of RB-SLAM relative to BP-SLAM (using 10.000 particles) comes at the expense of a significantly increased computational complexity.

C. Results for Real Measurements

For an evaluation of the performance of the proposed BP-SLAM algorithm using real measurements, we chose $P_{\mathrm{d}}^{(j)}\big(\boldsymbol{x}_n, \boldsymbol{a}_{k,n}^{(j)}\big) = P_{\mathrm{d}} = 0.6$ and $\mu_{\mathrm{FA}}^{(j)} = 2$. This accounts for the lower detection probability and higher false alarm probability exhibited by the preliminary signal analysis stage due to the diffuse multipath existing in indoor environments. We no longer consider RB-SLAM because, as noted above, it would perform poorly for such challenging values of P_{d} and $\mu_{\mathrm{FA}}^{(j)}$.

The measurements were taken from the seminar room scenario previously used in [30], [56]. They correspond to five closely spaced parallel trajectories each consisting of 900 agent positions with a spacing of 0.01 m, resulting in a total number of 4500 agent positions. The magenta line in Fig. 1 represents one of the five trajectories. Note that in our simulations, each individual trajectory is processed independently, i.e., the estimated PF positions of a trajectory are not used as prior knowledge for another trajectory. More details about the measurements can be found in [56]. At each agent position, the agent transmitted an ultra-wideband signal, which was received by the two static PAs. This signal was measured using an M-sequence correlative channel sounder with frequency range 3–10 GHz and antennas with an approximately uniform radiation pattern in the azimuth plane and zeros in the floor and ceiling directions. Within the measured band, the actual signal band was selected by a filter with raised-cosine impulse response $s(t)$ with a roll-off factor of 0.5, a two-sided 3-dB bandwidth of 2 GHz, and a center frequency of 7 GHz.



(a) (b)

Fig. 6: Results for real measurements: (a) Agent position RMSEs for the five individual trajectories and average RMSE (averaged over the five trajectories), (b) empirical CDFs of the RMSEs for the five individual trajectories and empirical CDF of the five RMSEs taken together.

From the measured signals, the range measurements $z_{m,n}^{(j)} = c\hat{\tau}_{m,n}^{(j)}$ constituting the input to the proposed algorithm were derived by means of a snapshot-based SISO SAGE algorithm [25] for estimating the delays $\hat{\tau}_{m,n}^{(j)}$ and complex amplitudes $\hat{\alpha}_{m,n}^{(j)}$ of the MPCs (cf. (1)). In this method, the maximum number of estimated MPCs for each PA j was defined as $M_{n,\text{max}}^{(j)} = 20$. Estimates of the range variances $\hat{\sigma}_{m,n}^{(j)2}$ (cf. (41)) were determined from the estimated complex amplitudes as described in [30], [56]. The standard deviation of the driving process in the PF state-evolution model was $\sigma_a = 0.5 \cdot 10^{-2}\,\text{m}$. We performed 30 simulation runs. The pdfs of the states were represented by 30.000 particles each.

Fig. 6(a) shows the agent position RMSEs of BP-SLAM obtained individually for the five trajectories versus time n, along with the overall RMSE averaged over the five trajectories. Fig. 6(b) shows the empirical cumulative distribution function (CDF) of the individual RMSEs and the empirical CDF of the five RMSEs taken together. It can be seen that the individual CDFs are very close to 1 already at RMSE equal to 0.12m or even less. The maximum of all the individual RMSEs is below 0.2m in all cases and below 0.083m in 90% of all cases.[4]

For an exemplary simulation run, Fig. 7 depicts the particles representing the posterior pdfs of the mobile agent and detected PF states as well as the MMSE position estimates of the detected PFs for the five trajectories. Almost all estimated PF positions can be associated with geometrically expected VA positions. This shows that the BP-SLAM algorithm is able to leverage position-related information contained in the radio signals for accurate and robust localization.

[4]The lower values of RMSE (below 0.072m in all cases and below 0.035m in 90% of all cases) obtained by the SLAM algorithm in [30] is due to an adaptive adjustment of the detection probability P_d.

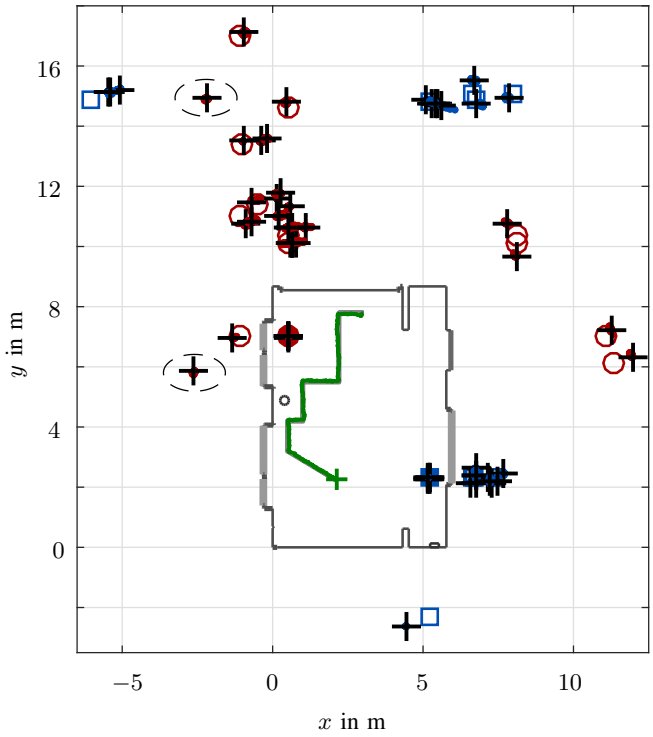


Fig. 7: Results for real measurements: Particles representing the posterior pdfs of the states of the detected PFs (red for PA 1, blue for PA 2). The positions of PA 1 and PA 2 are indicated by a red bullet and a blue box, respectively, and the corresponding geometrically expected VA positions by red circles and blue squares. The green line represents the MMSE estimates of the mobile agent position, and the green cross the MMSE estimate of the currently last mobile agent position. The black crosses indicate the MMSE estimates of the positions of the detected PFs. The dashed ellipses indicate PF positions that cannot be associated with geometrically expected VA positions.

VII. Conclusions and Future Perspectives

We proposed a radio signal based SLAM algorithm with probabilistic DA. The underlying system model describes specular MPCs in terms of VAs with unknown and possibly time-varying positions. To tackle the DA problem, i.e., the unknown association of MPCs with VAs, we modeled the entire SLAM problem including probabilistic DA in a Bayesian framework. We then represented the factorization of the joint posterior distribution by a factor graph and applied BP for approximate marginalization of the joint posterior distribution. This approach allowed the incorporation of an efficient BP algorithm for probabilistic DA that was originally proposed for MTT [37], [39]. Our factor graph extends that of [30] by the states of new potential features.

Simulation results using synthetic data showed that the proposed BP-SLAM algorithm estimates the time-varying agent position and the feature map with high accuracy and robustness, even in conditions of strong clutter and low probability of detection. Moreover, an extensive

experimental analysis using real ultra-wideband radio signals in an indoor environment showed that the BP-SLAM algorithm performs similarly well in real-world scenarios; the agent position error was observed to be below 0.2m for 100% and below 0.083m for 90% of all measurements.

A promising direction for future research is an extension of our BP-SLAM algorithm that exploits further MPC parameters, such as AoAs and AoDs. Other possible extensions are to track additional types of features, such as scatter points, and to redefine the features to be extended objects. In the latter case, the BP-based DA needs to be adapted accordingly. Finally, operation in an unsynchronized sensor network and a distributed (decentralized) mode of operation would be theoretically and practically interesting.

REFERENCES

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

[2] 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.

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

[4] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location-aware communications for 5G networks: How location information can improve scalability, latency, and robustness of 5G," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 102–112, Nov. 2014.

[5] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, and M. Z. Win, "High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 59–70, Mar. 2016.

[6] S. Bartoletti, A. Conti, A. Giorgetti, and M. Z. Win, "Sensor radar networks for indoor tracking," *IEEE Wireless Commun. Letters*, vol. 3, no. 2, pp. 157–160, Apr. 2014.

[7] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1263–1278, Apr. 2015.

[8] F. Guidi, A. Guerra, and D. Dardari, "Personal mobile radars with millimeter-wave massive arrays for indoor mapping," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1471–1484, Jun. 2016.

[9] E. Leitinger, F. Meyer, P. Meissner, K. Witrisal, and F. Hlawatsch, "Belief propagation based joint probabilistic data association for multipath-assisted indoor navigation and tracking," in *Proc. ICL-GNSS-16*, Barcelona, Spain, Jun. 2016.

[10] H. Wymeersch, S. Marano, W. Gifford, and M. Win, "A machine learning approach to ranging error mitigation for UWB localization," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1719–1728, Jun. 2012.

[11] Y. Shen, S. Mazuelas, and M. Win, "Network Navigation: Theory and Interpretation," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 9, pp. 1823–1834, Oct. 2012.

[12] M. Z. Win, Y. Shen, and W. Dai, "A theoretical foundation of network localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1136–1165, Jul. 2018, special issue on *Foundations and Trends in Localization Technologies*.

[13] C. Gentner, T. Jost, W. Wang, S. Zhang, A. Dammann, and U. C. Fiebig, "Multipath assisted positioning with simultaneous localization and mapping," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6104–6117, Sep. 2016.

[14] E. Leitinger, P. Meissner, C. Rudisser, G. Dumphart, and K. Witrisal, "Evaluation of position-related information in multipath components for indoor positioning," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2313–2328, Nov. 2015.

[15] J. Borish, "Extension of the image model to arbitrary polyhedra," *JASA*, vol. 75, no. 6, pp. 1827–1836, Mar. 1984.

[16] 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.

[17] J. Mullane, B.-N. Vo, M. 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.

[18] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI-02*, Edmonton, Canda, Jul. 2002, pp. 593–598.

[19] M. Lundgren, L. Svensson, and L. Hammarstrand, "Variational Bayesian expectation maximization for radar map estimation," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1391–1404, Mar. 2016.

[20] M. Fatemi, L. Svensson, L. Hammarstrand, and M. Lundgren, "Variational Bayesian EM for SLAM," in *Proc. IEEE CAMSAP-15*, Cancun, Mexico, Dec. 2015, pp. 501–504.

[21] 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.

[22] M. Fatemi, K. Granström, L. Svensson, F. J. R. Ruiz, and L. Hammarstrand, "Poisson multi-Bernoulli mapping using Gibbs sampling," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2814–2827, Jun. 2017.

[23] E. Leitinger, P. Meissner, M. Lafer, and K. Witrisal, "Simultaneous localization and mapping using multipath channel information," in *Proc. IEEE ICCW-15*, London, UK, Jun. 2015, pp. 754–760.

[24] M. Zhu, J. Vieira, Y. Kuang, K. Astrom, A. Molisch, and F. Tufvesson, "Tracking and positioning using phase information from estimated multi-path components," in *Proc. IEEE ICCW-15*, London, UK, Jun. 2015, pp. 712–717.

[25] B. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. I. Pedersen, "Channel parameter estimation in mobile radio environments using the SAGE algorithm," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 3, pp. 434–450, Mar. 1999.

[26] A. Richter, "Estimation of Radio Channel Parameters," Ph.D. dissertation, Ilmenau University of Technology, 2005.

[27] J. Salmi and A. Molisch, "Propagation parameter estimation, modeling and measurements for ultrawideband MIMO radar," *IEEE Trans. Antennas Propag*, vol. 59, no. 11, pp. 4257–4267, Nov. 2011.

[28] D. Shutin, W. Wang, and T. Jost, "Incremental sparse Bayesian learning for parameter estimation of superimposed signals," in *Proc. SampTA-13*, Bremen, Germany, Jul. 2013.

[29] M. A. Badiu, T. L. Hansen, and B. H. Fleury, "Variational Bayesian inference of line spectra," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2247–2261, May 2017.

[30] E. Leitinger, F. Meyer, F. Tufvesson, and K. Witrisal, "Factor graph based simultaneous localization and mapping using multipath channel information," in *Proc. IEEE ICC-17*, Paris, France, Jun. 2017.

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

[32] D. Musicki and R. Evans, "Joint integrated probabilistic data association: JIPDA," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 3, pp. 1093–1099, Jul. 2004.

[33] P. Horridge and S. Maskell, "Searching for, initiating and tracking multiple targets using existence probabilities," in *Proc. FUSION-09*, Seattle, WA, USA, Jul. 2009, pp. 611–617.

[34] ——, "Using a probabilistic hypothesis density filter to confirm tracks in a multi-target environment," in *Proc. INFORMATIK-11*, Berlin, Germany, Jul. 2011.

[35] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

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

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

[38] F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

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

[40] F. Meyer, P. Braca, F. Hlawatsch, M. Micheli, and K. LePage, "Scalable adaptive multitarget tracking using multiple sensors," in *Proc. IEEE GLOBECOM-16*, Washington DC, USA, Dec. 2016.

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

[42] T. Kropfreiter, F. Meyer, and F. Hlawatsch, "Sequential Monte Carlo implementation of the track-oriented marginal multi-Bernoulli/Poisson filter," in *Proc. FUSION-16*, Heidelberg, Germany, Jul. 2016, pp. 972–979.

[43] A. D. Marrs, "Asynchronous multi-sensor tracking in clutter with uncertain sensor locations using bayesian sequential monte carlo methods," in *Proc. IEEE Aerosp. Conf.*, vol. 5, Big Sky, MT, USA, 2001, pp. 2171–2178.

[44] J. Blanco, J. Gonzalez, and J. Fernandez-Madrigal, "A pure probabilistic approach to range-only SLAM," in *Proc. IEEE ICRA-08*, Pasadena CA, USA, May 2008, pp. 1436–1441.

[45] B. Etzlinger, F. Meyer, F. Hlawatsch, A. Springer, and H. Wymeersch, "Cooperative simultaneous localization and synchronization in mobile agent networks," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3587–3602, Jul. 2017.

[46] E. Leitinger, S. Grebien, X. Li, F. Tufvesson, and K. Witrisal, "On the use of MPC amplitude information in radio signal based SLAM," in *Proc. IEEE SSP-18*, Freiburg im Breisgau, Germany, Jun. 2018.

[47] C. Gentner and T. Jost, "Indoor positioning using time difference of arrival between multipath components," in *Proc. IPIN-13*, Montbeliard-Belfort, France, Oct. 2013, pp. 1–10.

[48] T. Jost, W. Wang, U. Fiebig, and F. Perez-Fontan, "Detection and tracking of mobile propagation channel paths," *IEEE Trans. Antennas Propag.*, vol. 60, no. 10, pp. 4875–4883, Oct. 2012.

[49] J. Vermaak, S. J. Godsill, and P. Perez, "Monte Carlo filtering for multi target tracking and data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 309–332, Jan. 2005.

[50] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice Hall, 1993.

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

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

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

[54] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *Proc. FUSION-17*, Xi'an, China, Jul. 2017, pp. 1–8.

[55] A. W. Moore, "Very fast EM-based mixture model clustering using multiresolution kd-trees," in *Proc. NIPS-98*, Denver, CO, USA, Dec. 1998, pp. 543–549.

[56] P. Meissner, E. Leitinger, and K. Witrisal, "UWB for robust indoor tracking: Weighting of multipath components for efficient estimation," *IEEE Wireless Comm. Lett.*, vol. 3, no. 5, pp. 501–504, Oct. 2014.