Message Passing Algorithms for Scalable Multitarget Tracking

Florian Meyer, *Member, IEEE*, Thomas Kropfreiter, Jason L. Williams, *Senior Member, IEEE*, Roslyn A. Lau, *Student Member, IEEE*, Franz Hlawatsch, *Fellow, IEEE*, Paolo Braca, *Senior Member, IEEE*, and Moe Z. Win, *Fellow, IEEE*

Superior scalability and performance of message passing algorithms enable new applications involving a large number of inexpensive sensors.

***Abstract*—Situation-aware technologies enabled by multitarget tracking will lead to new services and applications in fields such as autonomous driving, indoor localization, robotic networks, and crowd counting. In this tutorial paper, we advocate a recently proposed paradigm for scalable multitarget tracking that is based on message passing or, more concretely, the loopy sum-product algorithm. This approach has advantages regarding estimation accuracy, computational complexity, and implementation flexibility. Most importantly, it provides a highly effective, efficient, and scalable solution to the probabilistic data association problem, a major challenge in multitarget tracking. This fact makes it attractive for emerging applications requiring real-time operation on resource-limited devices. In addition, the message passing approach is intuitively appealing and suited to nonlinear and non-Gaussian models. We present message passing based multitarget tracking methods for single-sensor and multiple-sensor scenarios, and for a known and unknown number of targets. The presented methods can cope with clutter, missed detections, and an unknown association between targets and measurements. We also discuss the integration of message passing based probabilistic data association into existing multitarget tracking methods. The superior performance, low complexity, and attractive scaling properties of the presented methods are verified numerically. In addition to simulated data, we use measured data captured by two radar stations with overlapping fields-of-view observing a large number of targets simultaneously.**

***Index Terms*—Multitarget tracking, data association, message passing, sum-product algorithm, factor graph, data fusion.**

Manuscript received XX, 2017; revised XX, 201Y. This work was supported in part by the Austrian Science Fund (FWF) under grants J3886-N31 and P27370-N30, by the NATO Supreme Allied Command Transformation (ACT) under projects SAC000601 and SAC000608, by the Czech Science Foundation (GAČR) under grant 17-19638S, and by the Office of Naval Research (ONR) under Grant N00014-16-1-2141.

F. Meyer and Moe Z. Win are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA (e-mail: fmeyer@mit.edu, moewin@mit.edu). T. Kropfreiter and F. Hlawatsch are with the Institute of Telecommunications, TU Wien, Vienna, Austria and with Brno University of Technology, Brno, Czech Republic (e-mail: thomas.kropfreiter@tuwien.ac.at, franz.hlawatsch@tuwien.ac.at). J. L. Williams is with the National Security, Intelligence, Surveillance and Reconnaissance Division, Defence Science and Technology Group, Edinburgh, Australia and with the School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, Australia (e-mail: jason.williams@dst.defence.gov.au). R. A. Lau is with the Maritime Division, Defence Science and Technology Group, Edinburgh, Australia and with the Research School of Computer Science, Australian National University, Canberra, Australia (e-mail: roslyn.lau@dst.defence.gov.au). P. Braca is with the NATO Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy (e-mail: paolo.braca@cmre.nato.int).

I. INTRODUCTION

A new era of real-time pervasive situational awareness will be enabled by emerging inexpensive sensors and scalable data fusion algorithms. Situation-aware technologies are key to innovative products and services that are profoundly changing various aspects of our daily life. For example, safe autonomous navigation is based on a clear understanding of static and mobile objects in the environment. Multitarget tracking (MTT) can infer the states of these objects (or "targets") from measurements provided by one or more sensors even if their number is unknown [1]–[4]. The state of a target usually includes its position and possibly other quantities such as its velocity. A complicating factor in many MTT applications is the measurement origin uncertainty or data association (DA) problem, i.e., the unknown association of measurements with targets. Since the first MTT methods were introduced for aerospace surveillance some 40 years ago [5], [6], the field has experienced an intense development, and MTT methods have proven useful in further surveillance scenarios [7]–[9] as well as in biomedical analytics [10]–[12]. Emerging MTT applications such as autonomous driving [13]–[15], indoor localization [16]–[18], robotic networks [19]–[21], and crowd counting [22]–[24] are especially challenging since they require real-time operation on resource-limited devices.

A. The Message Passing Approach to Multitarget Tracking

The existing MTT methods can be grouped in two broad categories that may be referred to as "vector-type" and "set-type" methods. The former describe the multitarget states and measurements as random vectors [1], whereas the latter describe the multitarget states and measurements by random finite sets (RFSs) [3]. Many of these MTT methods have a high complexity and do not scale well in relevant system parameters. Thus, they are often impractical for applications involving a large number of inexpensive sensors and/or targets.

Here, we demonstrate that MTT methods with low complexity and good scalability are obtained by using the methodology of message passing, also known as belief propagation, or, more concretely, the *sum-product algorithm* (SPA). The SPA provides a principled approximation of optimum Bayesian inference that achieves an attractive performance-complexity compromise [25]–[33]. It is intuitively appealing, suited to nonlinear and non-Gaussian system models, and able to cope with unknown and time-varying hyperparameters.

In the last 25 years, message passing has been successfully used in many applications including iterative decoding of channel codes [34]–[37], communication receivers [38]–[40], and cooperative localization [41]–[43]. Early work on message passing in the context of MTT [44]–[47] focused on the calculation of "hard" measurement-to-target associations using the max-product algorithm. Only recently it was discovered that the "loopy" SPA is particularly attractive for MTT [48]–[55]. More specifically, the loopy SPA approach to MTT has the following advantages:

- it enables an efficient calculation of association probabilities for "soft" measurement-target associations;
- it provides a principled, general technique for deriving MTT methods within the Bayesian inference framework;
- it generalizes previously proposed MTT methods such as the joint probabilistic data-association (JPDA) filter; and
- it enables the development of scalable MTT methods that are suitable for scenarios involving a large number of targets and/or sensors and/or measurements.

B. Contributions, Paper Organization, Notation

This tutorial paper presents SPA-based methods for MTT and discusses their properties and advantages in relation to other MTT methods. We will present three distinct types of SPA-based MTT methods: (i) vector-type methods for a known, fixed number of targets; (ii) vector-type methods for an unknown, time-varying number of targets; and (iii) set-type methods for an unknown, time-varying number of targets.

The paper's main contributions and organization are as follows. In the remainder of this section, we introduce some basic notation and formulate certain assumptions that are common to all presented methods. In Section II, we survey existing MTT methods. An introduction to factor graphs and the SPA is given in Section III. Section IV describes a vector-type system model and a corresponding statistical formulation for the case where the number of targets is fixed and known. The corresponding posterior distributions and their graphical representation using factor graphs are derived in Section V. Section VI discusses the solution of the probabilistic DA problem using the loopy SPA. In Section VII, SPA-based methods for vector-type MTT are presented for the case of a known, fixed number of targets. Section VIII describes a vector-type system model, statistical formulation, and factor graph for the case where the number of targets is time-varying and unknown. For that case, new SPA-based methods for vector-type MTT are developed in Section IX. Section X commences our treatment of set-type MTT methods by giving an introduction to RFSs. In Section XI, we describe a set-type system model and a corresponding statistical formulation. Section XII develops multitarget state propagation relations for the set-type system model. Building on these relations, Section XIII derives SPA-based methods for set-type MTT. In Section XIV, we evaluate the performance and complexity of the presented SPA-based MTT methods in comparison to state-of-the-art methods. In addition to simulated data, we use measurement data acquired by two radar stations observing a large number of targets simultaneously. Our results demonstrate substantial advantages—regarding both performance and complexity—of SPA-based methods over other methods.

We will use the following basic notation. Random variables are displayed in sans serif, upright fonts; their realizations in serif, italic fonts. Vectors and matrices are denoted by bold lowercase and uppercase letters, respectively. For example, x is a random variable and x is its realization, and $\mathbf{x}$ is a random vector and $\boldsymbol{x}$ is its realization. Random sets and their realizations are denoted by upright sans serif and calligraphic font, respectively. For example, X is a random set and $\mathcal{X}$ is its realization. The expectation of a random variable or random vector is denoted by $\mathbb{E}\{\cdot\}$. The probability of an event is denoted by $\mathbb{P}\{\cdot\}$. We denote probability density functions (pdfs) as $f(\cdot)$ and probability mass functions (pmfs) as $p(\cdot)$. We write k for the discrete time index, i for the index of a target, m for the index of a measurement, and s for the index of a sensor. Integrals are over the entire space of the integration variable. The symbol $\propto$ indicates equality up to a constant factor. A consecutive list of i integers, i.e., $1, 2, \ldots, i$, is briefly denoted as $1, \ldots, i$.

C. Basic Assumptions

We consider scenarios where in order to reduce the data flow of the sensors and the overall complexity, measurements are produced by a detector performing a thresholding and, possibly, some further preparatory processing of the raw sensor data [1]. The measurement corresponding to a target can be missed by the detector if its signal-to-noise ratio is below the detection threshold. Another premise is that both the targets and the measurements are "points" and, at any particular time, at most one detection can be due to any particular target [1]. There is a *measurement origin uncertainty*, i.e., it is unknown which measurement originated from which target, or if a measurement is a false alarm (due to clutter). This leads to a nontrivial *DA problem*.

In the following, we state some basic assumptions underlying both the vector-type and set-type MTT methods to be presented. At (discrete) time k, there are i_k targets $i \in \{1, \ldots, i_k\}$. The state $\mathbf{x}_k^{(i)}$ of the ith target at time k consists of the position of the target and possibly further parameters, and it is a random vector of dimension d_x, i.e., $\mathbf{x}_k^{(i)} \in \mathbb{R}^{d_x}$. There are n_s sensors $s \in \{1, \ldots, n_\mathrm{s}\}$. At time k, sensor s produces $\mathsf{m}_{k,s}$ measurements $\mathbf{z}_{k,s}^{(m)}$, $m \in \{1, \ldots, \mathsf{m}_{k,s}\}$. Each measurement is a random vector of dimension d_z, i.e., $\mathbf{z}_{k,s}^{(m)} \in \mathbb{R}^{d_z}$. We make the following further assumptions [1]:

- A1: The *multitarget state* (defined as an ordered or unordered list of all the single-target states $\mathbf{x}_k^{(i)}$) evolves according to a first-order Markov process.
- A2: The single-target states $\mathbf{x}_k^{(i)}$ evolve independently. The evolution of state $\mathbf{x}_k^{(i)}$ is described by the single-target state-transition pdf $f\big(\boldsymbol{x}_k^{(i)} \big| \boldsymbol{x}_{k-1}^{(i)}\big)$.
- A3: The measurements $\mathbf{z}_{k,s}^{(m)}$ produced by sensor s at time k have no order, i.e., they are randomly shuffled.
- A4: Each measurement $\mathbf{z}_{k,s}^{(m)}$ originates from a target or from clutter (false alarm), and it cannot originate from more than one target simultaneously. Conversely, at time k,

one target can generate at most one measurement $\mathbf{z}_{k,s}^{(m)}$ at sensor s.

A5: The a priori probability that a measurement originates from target i (i.e., that sensor s "detects target i" at time k) is independent across the targets i and a known function of the target state $\boldsymbol{x}_k^{(i)}$, denoted as $p_{\mathrm{d}}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big)$. Consequently, the probability that no measurement originates from target i (i.e., that sensor s "misses target i" at time k) is $1-p_{\mathrm{d}}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big)$. We assume that $0 \leq p_{\mathrm{d}}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big) < 1$.

A6: The number of clutter measurements at sensor s and time k is Poisson distributed with mean $\mu_{\mathrm{c}}^{(s)}$. It is furthermore independent across the sensors s and independent of the number of targets that are detected at sensor s.

A7: At sensor s and time k, the clutter measurements are independent of the target-originated measurements.

A8: The clutter measurements at sensor s and time k are independent and identically distributed (iid) with pdf $f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big) \neq 0$.

A9: At sensor s and time k, given all the target states $\boldsymbol{x}_k^{(i)}$, the target-originated measurements are conditionally independent of each other and also conditionally independent of all the clutter measurements. Furthermore, a target-originated measurement $\mathbf{z}_{k,s}^{(m)}$, given the respective target state $\boldsymbol{x}_k^{(i)}$, is distributed according to $f\big(\boldsymbol{z}_{k,s}^{(m)}|\boldsymbol{x}_k^{(i)}\big)$.

A10: At time k, the measurements of different sensors s are conditionally independent given all the target states $\boldsymbol{x}_k^{(i)}$. (Here, the measurement of sensor s at time k is defined as a randomly ordered list or a set of all single measurements $\mathbf{z}_{k,s}^{(m)}$.)

A11: At time k, given all the target states $\boldsymbol{x}_k^{(i)}$, the measurements $\mathbf{z}_{k,s}^{(m)}$ are conditionally independent of all the past and future measurements $\mathbf{z}_{k',s}^{(m)}$ and target states $\mathbf{x}_{k'}^{(i)}$, $k' \neq k$.

II. Survey of State-of-the-art Methods

The basic setting of most MTT methods is sequential Bayesian estimation (or "filtering") [56], which consists of a prediction step based on the Chapman-Kolmogorov equation,

$$f\big(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}\big) = \int f\big(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\big) f\big(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}\big)\,\mathrm{d}\boldsymbol{x}_{k-1}\,, \quad (1)$$

and an update step based on Bayes' rule,

$$f\big(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}\big) \propto f\big(\boldsymbol{z}_k|\boldsymbol{x}_k\big) f\big(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}\big)\,. \quad (2)$$

Here, $\boldsymbol{x}_k$ is the state of a single target and $\boldsymbol{z}_k$ is a corresponding measurement, both at time k, and $\boldsymbol{z}_{1:k}$ is short for $\big[\boldsymbol{z}_1^{\mathrm{T}} \cdots \boldsymbol{z}_k^{\mathrm{T}}\big]^{\mathrm{T}}$. Based on the posterior pdf $f\big(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}\big)$, Bayesian state estimation can be performed, e.g., by means of the minimum mean-square error (MMSE) estimator [57]

$$\hat{\boldsymbol{x}}_k^{\mathrm{MMSE}} \triangleq \int \boldsymbol{x}_k\, f\big(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}\big)\,\mathrm{d}\boldsymbol{x}_k. \quad (3)$$

The Kalman filter [58] exploits the fact that for linear-Gaussian system models, the operations (1) and (2) can be performed in closed form. For nonlinear/non-Gaussian system models, computationally feasible approximate methods for sequential Bayesian estimation include the extended Kalman filter [58], the unscented Kalman filter [59], the Gaussian sum filter [60], and the particle filter [56], [61]. However, all these methods assume that the number of targets and the association between targets and measurements are perfectly known. Next, we discuss Bayesian methods for MTT that do not require knowledge of the association between targets and measurements.

A. Vector-Type MTT Methods

Vector-type MTT methods describe the multitarget states and measurements by random vectors. They are able to explicitly maintain track continuity, i.e., they associate a state estimate with a previous state estimate or declare the appearance of a new target. Many vector-type methods use heuristics to take into account the appearance and disappearance of targets [1]. Others model the discovery of new targets but not target disappearance [6], [62], or use target existence states to capture target disappearance while lacking a fully Bayesian approach to initiating tracks based on measurements [63]–[65].

1) Methods Based on Probabilistic DA: Probabilistic DA (PDA) methods for single-target tracking treat the DA parameter as a nuisance variable that is "marginalized out" [1, Sec. 3.4]. The PDA filter subsequently fits a Gaussian pdf to the resulting posterior target state pdf, whereas other methods (e.g., [66]) retain a more detailed representation. Joint PDA (JPDA) extends PDA to the case of multiple targets [1, Sec. 6.2]. The joint association parameters are marginalized out under the constraint that each measurement relates to at most one target. Like PDA, JPDA then fits a Gaussian pdf to the posterior state pdf for each target, whereas methods such as [67] retain a Gaussian mixture distribution.

JPDA assumes that the number of targets is known. This limitation is removed in joint integrated PDA (JIPDA) [2], [64], which extends JPDA to incorporate a probability of target existence. Target existence is modeled as a Markov chain, and new target tracks are initialized based on measurements that are not in the neighborhood (the "gate") of an existing track. A similar model along with a particle filter implementation was presented in [65]. A second limitation of JPDA, the high computational complexity of summing over all the joint associations, has been addressed by various heuristic approximations [68]–[70], by efficient hypothesis management [71], [72], or by using likely measurement-target associations to find approximations of the marginal posterior state pdfs [73], [74]. A third limitation, known as coalescence, arises when targets become closely spaced and the filter is no longer aware which target is in which position. As a result, the marginal pdfs become strongly multimodal. Possible solutions include a pruning of the association hypotheses [75], [76] and approximate calculation of marginal pdfs via optimization of information-theoretic measures [77], [78] or the use of mean-field formulations [79], [80].

Multisensor extensions of MTT filters based on JPDA were presented in [81], [82]. The measurements of different sensors are either processed in a sequential manner—known as the "iterated-corrector" approach—or in parallel. The performance of sequential processing may strongly depend on the order in which the measurements are processed.

2) MHT Methods: Multiple hypothesis tracking (MHT) methods seek to find the most likely measurement-target association over a sliding window of consecutive time steps [6]. A decision is made only on the association at the oldest time step. The original MHT formulation [6]—known as hypothesis-oriented MHT—forms an expanding tree of association hypotheses, where each leaf represents a partitioning of all the measurements acquired until the current time into subsets believed to correspond to the same target. The computational complexity is problematic but can be reduced by considering only likely hypotheses, which are identified by an efficient m-best assignment algorithm [83]–[86]. The more efficient track-oriented MHT methods [87]–[89] represent global association hypotheses—i.e., for all targets and measurements—implicitly via a series of tree structures. Each tree structure represents the possible association histories for a single target. The most likely hypothesis is found by choosing a leaf node for each single-target tree. An enumeration of hypotheses is avoided through combinatorial optimization techniques [90]–[93].

The original MHT formulation in [6] assumes a Poisson distribution of the number of newly detected targets. Calculations are simplified by disregarding targets that have not been detected so far. This idea is refined in [62], which assumes a fixed but unknown number of targets, with a Poisson prior, and in [94], which accommodates an unknown, time-varying number of targets. We note that MHT methods can also be derived in the set framework [95].

Multisensor MHT methods include iterated-corrector versions performing a sequential processing of the measurements [96], and versions that perform a parallel processing of the measurements by solving a multidimensional assignment problem [91]. In the multistage MHT approach [96], a first MHT stage is used for each sensor individually to reject clutter measurements, and a second MHT stage processes all the measurements using the iterated-corrector principle.

A key challenge in MHT methods is to retain sufficiently diverse association hypotheses until clarifying information is received. Graph-based methods have been proposed as an alternative solution to this problem in circumstances where association likelihoods are well approximated as Markovian [97], [98]. In [99], [100], these methods have been adapted to problems in which sporadic identity information is available.

B. Set-Type MTT Methods

Set-type MTT methods describe the multitarget states and measurements by RFSs [3]. (An introduction to RFSs will be given in Section X.) The RFS approach facilitates the modeling of target appearance and disappearance in a Bayesian setting, and introduces tools for handling complex, hybrid continuous/discrete distributions. Some set-type methods are unable to maintain track continuity [3], [78], [101]–[106], others maintain it implicitly [51], [107], and others maintain it explicitly by augmenting the states by distinct labels [108], [109].

1) PHD Filter and CPHD Filter: The probability hypothesis density (PHD) filter [106], [110] approximates the predicted posterior pdf of the target states, $f(\mathcal{X}_k|\mathcal{Z}_{1:k-1})$, by a Poisson RFS pdf. The intensity function (aka PHD) of that Poisson RFS is chosen such that the RFS Kullback-Leibler divergence relative to $f(\mathcal{X}_k|\mathcal{Z}_{1:k-1})$ is minimized [78]. The cardinalized PHD (CPHD) filter [111] uses a similar principle but approximates $f(\mathcal{X}_k|\mathcal{Z}_{1:k-1})$ by the pdf of an iid cluster RFS [78]. Multisensor extensions of the PHD and CPHD filters [110] use the iterated-corrector approach, i.e., they sequentially perform a separate (C)PHD approximation for each sensor [112], or they perform a single (C)PHD approximation for all sensors jointly [104]. The performance of the latter strategy is superior and moreover invariant to the ordering of the sensors. (C)PHD filters cannot maintain track continuity.

2) Multi-Bernoulli Filters: Multi-Bernoulli (MB) filters approximate the posterior pdf $f(\mathcal{X}_k|\mathcal{Z}_{1:k})$ by an MB RFS pdf. The original MB filter (abbreviated as MeMBer filter) [3] uses approximations of an RFS representation known as the probability generating functional. The MeMBer filter was found to have a cardinality bias, which was compensated in [103]. The track-oriented marginal MB/Poisson (TOMB/P) filter and the measurement-oriented marginal MB/Poisson (MOMB/P) filter [51], [113] are two MB filter variants that are based on the observation that the exact pdf $f(\mathcal{X}_k|\mathcal{Z}_{1:k})$ involves an MB mixture. Each MB pdf in this MB mixture corresponds to one of the global association hypotheses in MHT methods. Similar to [62], the TOMB/P and MOMB/P filters model undetected targets by a Poisson RFS, so that the overall multitarget state RFS is the union of independent Poisson and MB mixture RFSs. However, in contrast to [62] and other MHT variants, the TOMB/P and MOMB/P filters also model target appearance and disappearance. In the case of the TOMB/P filter, a computationally feasible MB filtering algorithm is obtained by approximating the DA pmf by the product of its marginals and by calculating the marginals using a scalable SPA-based algorithm. Although the TOMB/P filter is based on Bernoulli components without an explicit order or label, continuity of targets over successive time steps can be obtained implicitly, similar to the JIPDA filter. A sequential Monte Carlo implementation of the TOMB/P filter was presented in [107] and applied to static source localization in [114]. The MeMBer and MOMB/P filters cannot maintain track continuity. A multisensor extension of the MB filter was presented in [105].

3) Labeled RFS-Based Methods: Using labels allows RFS-based MTT filters to explicitly maintain track continuity, i.e., to obtain entire trajectories of consecutive target states. In the MTT context, the elements of a labeled RFS are the target state vectors augmented by a distinct label that identifies the respective target. The δ-GLMB filter [108] models the multitarget state by a generalized labeled MB (GLMB) RFS, where each GLMB component corresponds to one possible target-measurement association history. In contrast to the TOMB/P and MOMB/P filters, the number of hypothesized targets per GLMB component is deterministic, similarly to traditional MHT methods. To avoid an exponential increase of the number of GLMB components with time, components with a low weight are pruned and the generation of new components is limited by an m-best assignment algorithm. The LMB filter [109] is a reduced-complexity approximation of the δ-GLMB filter. The GLMB RFS is approximated by a labeled MB

(LMB) RFS, which is chosen such that its PHD matches the PHD of the GLMB RFS. This approximation is analogous to that underlying the TOMB/P filter. However, in [109], the DA problem is solved by means of the m-best assignment algorithm rather than a scalable SPA-based algorithm.

C. MTT Methods Using Message Passing

In the context of MTT, message passing techniques were first studied for DA in sensor networks where each sensor has a narrow field of view [44]–[47]. Association variables were used to hypothesize joint association events for all targets and measurements within certain nonoverlapping regions and to perform "hard" DA by means of the max-product algorithm. For such multisensor problems, calculating the marginal DA probabilities using exhaustive hypothesis enumeration is typically infeasible. Related techniques were studied in [115], [116], and a tree-based approximation of DA messages was developed in [117]. The efficient hypothesis management method proposed in [71], [72] exploits the redundancy present in many cases to achieve an exact evaluation with reduced complexity. This method is effectively a highly tailored version of the junction tree algorithm [25], [32]. Still, the complexity remains exponential in some extreme scenarios.

Message passing methods provide approximate marginal probabilities (using the SPA) and maximum a posteriori (MAP) estimates (using the max-product algorithm) with reduced complexity [25], [26]. The max-product algorithm is guaranteed to converge to the optimum MAP solution in single-scan DA problems (i.e., DA problems where the measurements of a single sensor and a single time step are considered) [118]. Max-product algorithms for multiple-scan DA were considered for MTT [49], [119], [120] and track association [121]. Dual decomposition methods [122], [123] provide a convergent alternative to the max-product algorithm, and were applied to multiple-scan DA in [119].

For estimation of marginal DA probabilities, an SPA-based method involving so-called mutual exclusion constraints [124], [32, Box 12.D] was shown in [49] to be outperformed by the bipartite SPA-based algorithm proposed in [125], [126] and subsequently studied in [48], [127]. This algorithm was developed independently and evaluated in the MTT context in [49], [128]. It was used in set-type MTT methods [51], [78], [107], [129], in vector-type multisensor MTT methods [52]–[55], [79], [80], and in vector-type methods for indoor localization [130]–[132]. This bipartite SPA-based algorithm for probabilistic DA will be discussed in Section VI. The underlying bipartite graphical model was also used for group tracking [133] and batch filtering performing MTT [134] (both based on the expectation-maximization algorithm), and for multipath tracking [135], [136].

In [52]–[54], the SPA was used not merely for probabilistic DA but for the entire multisensor MTT problem. These "total-SPA" MTT methods, which will be discussed in Sections VII and IX, integrate the bipartite SPA-based DA algorithm in the overall SPA formulation. In [53], a total-SPA MTT method was proposed in which target appearance and disappearance are modeled by "augmented target states" including a binary target existence indicator, and a particle implementation of the method was developed. An adaptive extension in which the probabilities of detection for the individual sensors are estimated jointly with the augmented target states was presented in [54]. A reformulation of [53] was used in SPA-based methods for indoor localization [130] and for simultaneous localization and mapping (SLAM) [131], [132]; these methods exploit the information provided by multipath components in ultra-wideband signals.

III. Introduction to the Sum-Product Algorithm

Many important algorithms—such as sequential Bayesian estimation, the Kalman filter, the particle filter, the forward-backward algorithm, the Viterbi algorithm, the turbo decoding algorithm, and fast Fourier transform algorithms—can be viewed as instances of the SPA. In addition, the SPA has been used to develop numerous new, high-performing algorithms in a wide range of applications [34]–[43], [137], [138].

In detection and estimation problems, the SPA can be used for an efficient calculation of marginal posterior pdfs [25], [27]. Consider the estimation of random parameter vectors x_i, $i = 1, \ldots, n_\mathrm{t}$ from an observation $\boldsymbol{z}$ of a random measurement vector z. Most Bayesian estimation methods require the posterior pdfs $f(\boldsymbol{x}_i|\boldsymbol{z})$ [57]. These pdfs are marginal pdfs of the joint posterior pdf $f(\boldsymbol{x}|\boldsymbol{z})$, where $\mathsf{x} \triangleq \left[\mathsf{x}_1^\mathrm{T} \cdots \mathsf{x}_{n_\mathrm{t}}^\mathrm{T}\right]^\mathrm{T}$. In many cases, the SPA is able to calculate the marginal posterior pdfs $f(\boldsymbol{x}_i|\boldsymbol{z})$, or accurate approximations thereof, at a small fraction of the computational cost of direct marginalizations. In particular, the computational cost of the SPA typically scales significantly better with relevant system parameters than that of direct marginalizations. This advantage makes an SPA-based solution feasible for many large-scale problems where other solutions are infeasible, including MTT problems with a large number of targets, sensors, and/or measurements.

A. Factor Graphs

The use of the SPA for marginalizing the joint posterior pdf $f(\boldsymbol{x}|\boldsymbol{z})$ presupposes that $f(\boldsymbol{x}|\boldsymbol{z})$ is the product of lower-dimensional factors, i.e.,

$$f(\boldsymbol{x}|\boldsymbol{z}) \propto \prod_l \psi_l\big(\boldsymbol{x}^{(l)}\big). \tag{4}$$

Here, each argument $\boldsymbol{x}^{(l)}$ comprises certain parameter vectors $\boldsymbol{x}_i$, where each $\boldsymbol{x}_i$ can appear in several $\boldsymbol{x}^{(l)}$. Note that the factors $\psi_l\big(\boldsymbol{x}^{(l)}\big)$ generally depend also on $\boldsymbol{z}$. The factorization (4) can be represented by a *factor graph* [25], [28]. (Alternative graphical models include Markov random fields [139] and Bayesian networks [140].) In a factor graph, each parameter variable $\boldsymbol{x}_i$ is represented by a variable node, and each factor $\psi_l(\cdot)$ by a factor node. Variable node "$\boldsymbol{x}_i$" and factor node "ψ_l" are adjacent, i.e., connected by an edge, if $\boldsymbol{x}_i$ is an argument of $\psi_l(\cdot)$. This is visualized in Fig. 1 for the factorization $f(\boldsymbol{x}|\boldsymbol{z}) \propto \psi_1(\boldsymbol{x}_1)\psi_2(\boldsymbol{x}_1, \boldsymbol{x}_2)\psi_3(\boldsymbol{x}_2)$, where $\boldsymbol{x} = \left[\boldsymbol{x}_1^\mathrm{T}\ \boldsymbol{x}_2^\mathrm{T}\right]^\mathrm{T}$.

A useful modification of factor graphs is provided by the principle of "stretching" or "opening" factors [25], [40]. Here, one introduces additional variables that may depend deterministically on certain variables in the original factor

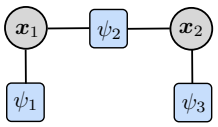


Fig. 1. Factor graph representing the factorization $f(\boldsymbol{x}|\boldsymbol{z}) \propto \psi_1(\boldsymbol{x}_1)\psi_2(\boldsymbol{x}_1,\boldsymbol{x}_2)\psi_3(\boldsymbol{x}_2)$. Variable nodes and factor nodes are represented by circles and squares, respectively.

graph. This causes certain factor nodes to be "stretched" into a larger number of factor nodes. The variable nodes adjacent to these new factor nodes tend to have lower dimensions than those adjacent to the original factor nodes. Accordingly, some of the messages in the SPA are replaced with lower-dimensional messages, which results in reduced computational complexity and improved scalability.

B. The SPA

The SPA [25], [29] is a message passing algorithm that calculates certain messages for each node and passes each of these messages to one of the adjacent nodes. Messages leaving or entering a variable node are functions of the associated variable. Let $\mathcal{V}_l$ be the set of the indices i of all variable nodes "$\boldsymbol{x}_i$" adjacent to factor node "ψ_l", or equivalently, of all variables $\boldsymbol{x}_i$ that appear in the argument of $\psi_l\big(\boldsymbol{x}^{(l)}\big)$ in (4). Furthermore, let $\mathcal{F}_i$ be the set of the indices l of all factor nodes "ψ_l" adjacent to variable node "$\boldsymbol{x}_i$", or equivalently, of all factors $\psi_l\big(\boldsymbol{x}^{(l)}\big)$ that involve $\boldsymbol{x}_i$ in their argument. Then, the message passed from factor node "ψ_l" to an adjacent variable node "$\boldsymbol{x}_i$", $i \in \mathcal{V}_l$ is given by

$$\zeta_{\psi_l \to \boldsymbol{x}_i}(\boldsymbol{x}_i) = \int \psi_l\big(\boldsymbol{x}^{(l)}\big) \prod_{i' \in \mathcal{V}_l \setminus \{i\}} \eta_{\boldsymbol{x}_{i'} \to \psi_l}(\boldsymbol{x}_{i'}) \, \mathrm{d}\boldsymbol{x}_{\sim i} \,. \quad (5)$$

Here, $\int \ldots \mathrm{d}\boldsymbol{x}_{\sim i}$ denotes integration with respect to all vectors $\boldsymbol{x}_{i'}$, $i' \in \mathcal{V}_l$ except $\boldsymbol{x}_i$. For factorizations involving discrete variables, the integration is replaced with a summation. Furthermore, the message $\eta_{\boldsymbol{x}_i \to \psi_l}(\boldsymbol{x}_i)$ passed from variable node "$\boldsymbol{x}_i$" to an adjacent factor node "ψ_l", $l \in \mathcal{F}_i$ is given by

$$\eta_{\boldsymbol{x}_i \to \psi_l}(\boldsymbol{x}_i) = \prod_{l' \in \mathcal{F}_i \setminus \{l\}} \zeta_{\psi_{l'} \to \boldsymbol{x}_i}(\boldsymbol{x}_i) \,. \quad (6)$$

In the case of a tree-structured factor graph, such as in Fig. 1, the message updates (5) and (6) are performed once for each variable node and factor node. This process starts at the variable and factor nodes with only one edge (which pass a constant message and the corresponding factor, respectively), and proceeds with any node whose incoming messages are already available. After all messages have been calculated, a *belief* $\tilde{f}(\boldsymbol{x}_i)$ is calculated for each variable node "$\boldsymbol{x}_i$" as

$$\tilde{f}(\boldsymbol{x}_i) = C_i \prod_{l \in \mathcal{F}_i} \zeta_{\psi_l \to \boldsymbol{x}_i}(\boldsymbol{x}_i) \,, \quad (7)$$

with C_i such that $\int \tilde{f}(\boldsymbol{x}_i) \mathrm{d}\boldsymbol{x}_i = 1$. For a tree-structured factor graph, it can be shown [25], [32] that each belief $\tilde{f}(\boldsymbol{x}_i)$ is exactly equal to the respective marginal posterior pdf $f(\boldsymbol{x}_i|\boldsymbol{z})$. In our example (see Fig. 1), the message passed from "ψ_2" to "$\boldsymbol{x}_2$" is $\zeta_{\psi_2 \to \boldsymbol{x}_2}(\boldsymbol{x}_2) = \int \psi_2(\boldsymbol{x}_1, \boldsymbol{x}_2) \eta_{\boldsymbol{x}_1 \to \psi_2}(\boldsymbol{x}_1) \mathrm{d}\boldsymbol{x}_1$ (here, $\boldsymbol{x}^{(2)} = \big[\boldsymbol{x}_1^{\mathrm{T}} \, \boldsymbol{x}_2^{\mathrm{T}}\big]^{\mathrm{T}}$), the message passed from "$\boldsymbol{x}_1$" to "ψ_2" is $\eta_{\boldsymbol{x}_1 \to \psi_2}(\boldsymbol{x}_1) = \zeta_{\psi_1 \to \boldsymbol{x}_1}(\boldsymbol{x}_1) = \psi_1(\boldsymbol{x}_1)$, and the belief for "$\boldsymbol{x}_2$" is $\tilde{f}(\boldsymbol{x}_2) = C_2 \, \zeta_{\psi_2 \to \boldsymbol{x}_2}(\boldsymbol{x}_2) \zeta_{\psi_3 \to \boldsymbol{x}_2}(\boldsymbol{x}_2) = C_2 \, \zeta_{\psi_2 \to \boldsymbol{x}_2}(\boldsymbol{x}_2) \psi_3(\boldsymbol{x}_2)$, where $C_2 = 1/\int \zeta_{\psi_2 \to \boldsymbol{x}_2}(\boldsymbol{x}_2) \psi_3(\boldsymbol{x}_2) \times \mathrm{d}\boldsymbol{x}_2$. With $\psi_1(\boldsymbol{x}_1) = f(\boldsymbol{x}_1|\boldsymbol{z}_1)$, $\psi_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = f(\boldsymbol{x}_2|\boldsymbol{x}_1)$, and $\psi_3(\boldsymbol{x}_2) = f(\boldsymbol{z}_2|\boldsymbol{x}_2)$, it can be verified that $\tilde{f}(\boldsymbol{x}_2) = f(\boldsymbol{x}_2|\boldsymbol{z}_{1:2})$. Thus, running the SPA on the factor graph in Fig. 1 is equivalent to performing prediction step (1) and update step (2).

In the case of a factor graph with loops, the SPA is applied in an iterative manner, i.e., the entire message update process is repeated several times. The resulting beliefs $\tilde{f}(\boldsymbol{x}_i)$ are then generally only approximations of the marginal posterior pdfs $f(\boldsymbol{x}_i|\boldsymbol{z})$. Theoretical analysis [26], [30], [31], [33] showed that this "loopy" SPA can be interpreted as a variational approach to approximate inference that corresponds to a constrained optimization problem, and that the iterative message updating process converges to a stationary point of that optimization problem. Although the optimization problem is typically non-convex, the approximation of the $f(\boldsymbol{x}_i|\boldsymbol{z})$ provided by the loopy SPA has been observed to be very accurate in many applications [34]–[36], [38]–[43], [137], [138]. Intuitively, the loopy SPA converges and provides a good approximation of the marginal posterior pdfs if the optimization problem is locally convex in a region containing the starting point and the optimal solution. Alternative optimization problems that are convex can be constructed [141], [142]. The resulting iterative message passing algorithms converge to a global optimum and may lead to more accurate beliefs than the loopy SPA. However, they are typically significantly more complex.

In an iterative loopy SPA, there is no canonical order of message calculation, and different orders may lead to different beliefs. Specifying the order (schedule) and using the "stretching factors" principle discussed in Section III-A gives a certain freedom in the design of message passing algorithms. We will take advantage of this design freedom in our development of SPA-based MTT methods in later sections.

For linear-Gaussian system models, the message passing equations (5)–(7) can be evaluated in closed form (see [27] for details). The resulting equations generalize the Kalman filter recursion [58] to more complex factorization structures. For nonlinear/non-Gaussian models, computationally feasible approximate implementations of (5)–(7) include nonparametric belief propagation [41] and sigma point belief propagation [143], which generalize the particle filter [56], [61] and the unscented Kalman filter [59], respectively.

IV. VECTOR-TYPE SYSTEM MODEL FOR A KNOWN, FIXED NUMBER OF TARGETS

In this section, following [1], we present a system model and a basic statistical formulation for the vector-type DA and MTT methods considered in Sections VI and VII, respectively. We assume that the number of targets is fixed and known. A vector-type MTT system model for an unknown, time-varying number of targets will be presented in Section VIII.

A. State-Transition pdf and Prior Distribution

The vector-type system model is based on the assumptions A1–A11 listed in Section I-C and on the following additional assumptions [1]:

Vk1: The number of targets, $i_k = n_\mathrm{t}$, is fixed and known.

Vk2: The target states are ordered (arbitrarily) according to their arrangement in a joint state vector $\mathsf{x}_k \triangleq \big[\mathsf{x}_k^{(1)\mathrm{T}} \cdots \mathsf{x}_k^{(n_\mathrm{t})\mathrm{T}}\big]^\mathrm{T}$.

Vk3: At time $k = 1$, the target states $\mathsf{x}_1^{(i)}$ are independent and distributed according to the prior pdfs $f\big(\boldsymbol{x}_1^{(i)}\big)$.

Using Vk1 and Vk2 along with A1 and A2, the joint state-transition pdf is obtained as

$$f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \prod_{i=1}^{n_\mathrm{t}} f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{x}_{k-1}^{(i)}\big) \quad k = 2, 3, \ldots. \tag{8}$$

Similarly, using Vk3, the joint prior pdf at time $k = 1$ is

$$f(\boldsymbol{x}_1) = \prod_{i=1}^{n_\mathrm{t}} f\big(\boldsymbol{x}_1^{(i)}\big). \tag{9}$$

B. Likelihood Function

The measurements produced by sensor s at time k are described by the vector $\mathsf{z}_{k,s} \triangleq \big[\mathsf{z}_{k,s}^{(1)\mathrm{T}} \cdots \mathsf{z}_{k,s}^{(\mathsf{m}_{k,s})\mathrm{T}}\big]^\mathrm{T}$, where the components (subvectors) $\mathsf{z}_{k,s}^{(m)}$, $m = 1, \ldots, \mathsf{m}_{k,s}$ have a random order (cf. A3). The (unknown) association between measurements and targets can be described by the DA vector $\mathsf{a}_{k,s} = \big[\mathsf{a}_{k,s}^{(1)} \cdots \mathsf{a}_{k,s}^{(n_\mathrm{t})}\big]^\mathrm{T}$ with entries [1]

$$\mathsf{a}_{k,s}^{(i)} \triangleq \begin{cases} m \in \{1, \ldots, \mathsf{m}_{k,s}\}, & \text{if at time } k, \text{ target } i \text{ generates measurement } m \text{ at sensor } s \\ 0, & \text{if at time } k, \text{ target } i \text{ does not generate a measurement at sensor } s. \end{cases} \tag{10}$$

If $\mathsf{a}_{k,s}$ was known, we would be able to identify clutter measurements and to associate measurements with targets. However, $\mathsf{a}_{k,s}$ is unknown and thus considered as a latent random variable in the inference problem. It will be convenient to express the constraint A4 by the indicator function

$$\psi(\boldsymbol{a}_{k,s}) \triangleq \begin{cases} 0, & \exists i, i' \in \{1, \ldots, n_\mathrm{t}\} \\ & \quad \text{such that } i \neq i' \text{ and } a_{k,s}^{(i)} = a_{k,s}^{(i')} \neq 0 \\ 1, & \text{otherwise.} \end{cases}$$

Using A3–A6, the prior pmf of the DA vector $\mathsf{a}_{k,s}$ and the number of measurements $\mathsf{m}_{k,s}$ conditioned on the multitarget state x_k is obtained as [1]

$$\begin{aligned} &p(\boldsymbol{a}_{k,s}, m_{k,s}|\boldsymbol{x}_k) \\ &= \psi(\boldsymbol{a}_{k,s}) \frac{e^{-\mu_\mathrm{c}^{(s)}} (\mu_\mathrm{c}^{(s)})^{m_{k,s}}}{m_{k,s}!} \left(\prod_{i=1}^{n_\mathrm{t}} \big(1 - p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big)\big) \right) \\ &\qquad \times \prod_{j \in \mathcal{D}_{\boldsymbol{a}_{k,s}}} \frac{p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(j)}\big)}{\mu_\mathrm{c}^{(s)} \big(1 - p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(j)}\big)\big)}, \end{aligned} \tag{11}$$

where $\mathcal{D}_{\boldsymbol{a}_{k,s}}$ is the set of "detected targets" corresponding to $\boldsymbol{a}_{k,s}$, i.e., $\mathcal{D}_{\boldsymbol{a}_{k,s}} \triangleq \big\{i \in \{1, \ldots, n_\mathrm{t}\} : a_{k,s}^{(i)} \neq 0\big\}$. The detailed derivation of expression (11) is reviewed in [144]. Note that the factor $\psi(\boldsymbol{a}_{k,s})$ guarantees that $p(\boldsymbol{a}_{k,s}, m_{k,s}|\boldsymbol{x}_k) = 0$ for vectors $\boldsymbol{a}_{k,s}$ violating A4 (e.g., if one measurement is associated with two targets). In addition, $\psi(\boldsymbol{a}_{k,s})$ introduces a coupling of the different DA variables $a_{k,s}^{(i)}$, which implies that performing inference independently on single-target states is suboptimum. Note that $p(\boldsymbol{a}_{k,s}, m_{k,s}|\boldsymbol{x}_k)$ in (11) is a valid pmf in the sense that $\sum_{m_{k,s}=0}^{\infty} \sum_{\boldsymbol{a}_{k,s}} p(\boldsymbol{a}_{k,s}, m_{k,s}|\boldsymbol{x}_k) = 1$ for arbitrary $\boldsymbol{x}_k$; here, $\sum_{\boldsymbol{a}_{k,s}}$ is short for $\sum_{\boldsymbol{a}_{k,s} \in \{0,1,\ldots,m_{k,s}\}^{n_\mathrm{t}}}$.

To gain further insights into the structure of $p(\boldsymbol{a}_{k,s}, m_{k,s}|\boldsymbol{x}_k)$, we next investigate three special cases. In the *no-detections, no-clutter case*, i.e., $m_{k,s} = 0$ and $\boldsymbol{a}_{k,s} = \boldsymbol{0}$, expression (11) reduces to

$$p(\boldsymbol{a}_{k,s} = \boldsymbol{0}, m_{k,s} = 0|\boldsymbol{x}_k) = e^{-\mu_\mathrm{c}^{(s)}} \prod_{i=1}^{n_\mathrm{t}} \big(1 - p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big)\big).$$

Here, $e^{-\mu_\mathrm{c}^{(s)}}$ is the Poisson pmf of the number of clutter measurements evaluated at 0, and $\prod_{i=1}^{n_\mathrm{t}} \big(1 - p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big)\big)$ is the probability that no target is detected by sensor s. In the *no-detections, all-clutter case*, i.e., $m_{k,s} > 0$ and $\boldsymbol{a}_{k,s} = \boldsymbol{0}$,

$$p(\boldsymbol{a}_{k,s} = \boldsymbol{0}, m_{k,s}|\boldsymbol{x}_k) = \frac{e^{-\mu_\mathrm{c}^{(s)}} (\mu_\mathrm{c}^{(s)})^{m_{k,s}}}{m_{k,s}!} \prod_{i=1}^{n_\mathrm{t}} \big(1 - p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big)\big),$$

where $e^{-\mu_\mathrm{c}^{(s)}} (\mu_\mathrm{c}^{(s)})^{m_{k,s}}/m_{k,s}!$ is the Poisson pmf of the number of clutter measurements evaluated at $m_{k,s}$. Finally, in the *all-detections, no-clutter* case, i.e., $m_{k,s} = n_\mathrm{t}$ and $\boldsymbol{a}_{k,s} = \boldsymbol{a}_{k,s}^\mathrm{d}$, where $\boldsymbol{a}_{k,s}^\mathrm{d}$ is any DA vector that assigns exactly one measurement to each target,

$$p(\boldsymbol{a}_{k,s} = \boldsymbol{a}_{k,s}^\mathrm{d}, m_{k,s} = n_\mathrm{t}|\boldsymbol{x}_k) = \frac{1}{n_\mathrm{t}!} e^{-\mu_\mathrm{c}^{(s)}} \prod_{i=1}^{n_\mathrm{t}} p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big).$$

Here, $e^{-\mu_\mathrm{c}^{(s)}}$ is again the Poisson pmf of the clutter measurements evaluated at 0, $\prod_{i=1}^{n_\mathrm{t}} p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big)$ is the probability that all targets are detected by sensor s, and the factor $1/n_\mathrm{t}!$ arises because there are $n_\mathrm{t}!$ different measurement-target associations and, thus, $n_\mathrm{t}!$ different $\boldsymbol{a}_{k,s}^\mathrm{d}$.

Next, using A7–A9, the dependence of the measurement vector $\mathsf{z}_{k,s}$ on x_k, $\mathsf{a}_{k,s}$, and $\mathsf{m}_{k,s}$ is described by the pdf [1]

$$\begin{aligned} f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k, \boldsymbol{a}_{k,s}, m_{k,s}) &= \left(\prod_{m=1}^{m_{k,s}} f_\mathrm{c}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big) \right) \\ &\quad \times \prod_{i \in \mathcal{D}_{\boldsymbol{a}_{k,s}}} f^{(s)}\Big(\boldsymbol{z}_{k,s}^{(a_{k,s}^{(i)})}\Big|\boldsymbol{x}_k^{(i)}\Big), \end{aligned} \tag{12}$$

with $f^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big|\boldsymbol{x}_k^{(i)}\big) \triangleq f\big(\boldsymbol{z}_{k,s}^{(m)}\big|\boldsymbol{x}_k^{(i)}\big)/f_\mathrm{c}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big)$. (Note that expression (12) presupposes that $\boldsymbol{a}_{k,s}$ is consistent with A4.) Again, we investigate $f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k, \boldsymbol{a}_{k,s}, m_{k,s})$ for the three special cases considered earlier. In the *no-detections, no-clutter case*, i.e., $m_{k,s} = 0$ and $\boldsymbol{a}_{k,s} = \boldsymbol{0}$, $\boldsymbol{z}_{k,s}$ is not defined; however, an expression of $f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k, \boldsymbol{a}_{k,s}, m_{k,s})$ replacing (12) can be formally introduced as $f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k, \boldsymbol{a}_{k,s} = \boldsymbol{0}, m_{k,s} = 0) = 1$. In the *no-detections, all-clutter case*, i.e., $m_{k,s} > 0$ and $\boldsymbol{a}_{k,s} = \boldsymbol{0}$, expression (12) reduces to

$$f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k, \boldsymbol{a}_{k,s} = \boldsymbol{0}, m_{k,s}) = \prod_{m=1}^{m_{k,s}} f_\mathrm{c}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big).$$

Here, since all the measurements are clutter measurements, $\boldsymbol{z}_{k,s}$ is independent of $\boldsymbol{x}_k$. Finally, in the *all-detections, no-*

clutter case, i.e., $m_{k,s}=n_\mathrm{t}$ and $\boldsymbol{a}_{k,s}=\boldsymbol{a}_{k,s}^\mathrm{d}$,

$$f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k,\boldsymbol{a}_{k,s}=\boldsymbol{a}_{k,s}^\mathrm{d},m_{k,s}=n_\mathrm{t})=\prod_{i=1}^{n_\mathrm{t}} f\Big(\boldsymbol{z}_{k,s}^{(a_{k,s}^{\mathrm{d}(i)})}\Big|\boldsymbol{x}_k^{(i)}\Big).$$

Here, each target state $\boldsymbol{x}_k^{(i)}$ generates one measurement $\boldsymbol{z}_{k,s}$, which is distributed according to $f\Big(\boldsymbol{z}_{k,s}^{(a_{k,s}^{\mathrm{d}(i)})}\Big|\boldsymbol{x}_k^{(i)}\Big)$.

The pdf of $\mathsf{z}_{k,s}$, $\mathsf{a}_{k,s}$, and $\mathsf{m}_{k,s}$ conditioned on x_k is now obtained as

$$f(\boldsymbol{z}_{k,s},\boldsymbol{a}_{k,s},m_{k,s}|\boldsymbol{x}_k)=f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k,\boldsymbol{a}_{k,s},m_{k,s})\times p\big(\boldsymbol{a}_{k,s},m_{k,s}\big|\boldsymbol{x}_k\big),$$

and further, inserting (11) and (12),

$$f(\boldsymbol{z}_{k,s},\boldsymbol{a}_{k,s},m_{k,s}|\boldsymbol{x}_k)=\frac{e^{-\mu_\mathrm{c}^{(s)}}}{m_{k,s}!}\left(\prod_{m=1}^{m_{k,s}}\mu_\mathrm{c}^{(s)}f_\mathrm{c}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big)\right)\times\psi(\boldsymbol{a}_{k,s})\prod_{i=1}^{n_\mathrm{t}}g\big(\boldsymbol{x}_k^{(i)},a_{k,s}^{(i)};\boldsymbol{z}_{k,s}\big),\tag{13}$$

with

$$g\big(\boldsymbol{x}_k^{(i)},a_{k,s}^{(i)};\boldsymbol{z}_{k,s}\big)\triangleq\begin{cases}\frac{1}{\mu_\mathrm{c}^{(s)}}p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big)f^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big|\boldsymbol{x}_k^{(i)}\big), & a_{k,s}^{(i)}=m\in\{1,\dots,m_{k,s}\}\\ 1-p_\mathrm{d}^{(s)}\big(\boldsymbol{x}_k^{(i)}\big), & a_{k,s}^{(i)}=0.\end{cases}\tag{14}$$

In (13) and hereafter, it is assumed that the value of $m_{k,s}$ is consistent with $\boldsymbol{z}_{k,s}$, i.e., equal to the number of subvectors $\boldsymbol{z}_{k,s}^{(m)}$ in $\boldsymbol{z}_{k,s}$.

Finally, we use A10 and the fact that since the measurements $\mathsf{z}_{k,s}$ of different sensors s are conditionally independent given $\boldsymbol{x}_k$, also the DA vectors $\mathsf{a}_{k,s}$ are conditionally independent. We thus obtain for the joint pdf of the all-sensors vectors $\mathsf{z}_k\triangleq\big[\mathsf{z}_{k,1}^\mathrm{T}\cdots\mathsf{z}_{k,n_\mathrm{s}}^\mathrm{T}\big]^\mathrm{T}$, $\mathsf{a}_k\triangleq\big[\mathsf{a}_{k,1}^\mathrm{T}\cdots\mathsf{a}_{k,n_\mathrm{s}}^\mathrm{T}\big]^\mathrm{T}$, and $\mathsf{m}_k\triangleq\big[\mathsf{m}_{k,1}\cdots\mathsf{m}_{k,n_\mathrm{s}}\big]^\mathrm{T}$ conditioned on x_k:

$$f(\boldsymbol{z}_k,\boldsymbol{a}_k,\boldsymbol{m}_k|\boldsymbol{x}_k)=\prod_{s=1}^{n_\mathrm{s}}f(\boldsymbol{z}_{k,s},\boldsymbol{a}_{k,s},m_{k,s}|\boldsymbol{x}_k).\tag{15}$$

The global (all-sensors) likelihood function $f(\boldsymbol{z}_k,\boldsymbol{m}_k|\boldsymbol{x}_k)$ follows via marginalization, i.e.,

$$f(\boldsymbol{z}_k,\boldsymbol{m}_k|\boldsymbol{x}_k)=\sum_{\boldsymbol{a}_k}f(\boldsymbol{z}_k,\boldsymbol{a}_k,\boldsymbol{m}_k|\boldsymbol{x}_k),\tag{16}$$

where the sum is over all $\boldsymbol{a}_k\in\{0,1,\dots,m_{k,1}\}^{n_\mathrm{t}}\times\cdots\times\{0,1,\dots,m_{k,n_\mathrm{s}}\}^{n_\mathrm{t}}$ (note that there are $\big(\prod_{s=1}^{n_\mathrm{s}}m_{k,s}\big)^{n_\mathrm{t}}$ different DA vectors $\boldsymbol{a}_k$).

For completeness and future reference, we also present the single-sensor likelihood function without $\boldsymbol{a}_{k,s}$:

$$\begin{aligned}f(\boldsymbol{z}_{k,s},m_{k,s}|\boldsymbol{x}_k)&=\sum_{\boldsymbol{a}_{k,s}}f(\boldsymbol{z}_{k,s},\boldsymbol{a}_{k,s},m_{k,s}|\boldsymbol{x}_k)\\&=\frac{e^{-\mu_\mathrm{c}^{(s)}}}{m_{k,s}!}\left(\prod_{m=1}^{m_{k,s}}\mu_\mathrm{c}^{(s)}f_\mathrm{c}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big)\right)\\&\quad\times\sum_{\boldsymbol{a}_{k,s}}\psi(\boldsymbol{a}_{k,s})\prod_{i=1}^{n_\mathrm{t}}g\big(\boldsymbol{x}_k^{(i)},a_{k,s}^{(i)};\boldsymbol{z}_{k,s}\big),\end{aligned}\tag{17}$$

where (13) was used. The sum is over all possible DA vectors $\boldsymbol{a}_{k,s}\in\{0,1,\dots,m_{k,s}\}^{n_\mathrm{t}}$, and thus the computational complexity of evaluating $f(\boldsymbol{z}_{k,s},m_{k,s}|\boldsymbol{x}_k)$ scales exponentially with the number of targets n_t. The expression (17) is invariant to a permutation of the subvectors $\boldsymbol{z}_{k,s}^{(m)}$ in $\boldsymbol{z}_{k,s}$ and of the subvectors $\boldsymbol{x}_k^{(i)}$ in $\boldsymbol{x}_k$. This independence of an assumed order of the targets and the measurements motivates the set-type approach considered in Sections XI–XIII. Note that $f(\boldsymbol{z}_{k,s},m_{k,s}|\boldsymbol{x}_k)$ in (17) is a valid hybrid pdf/pmf in the sense that $\sum_{m_{k,s}=0}^{\infty}\int f(\boldsymbol{z}_{k,s},m_{k,s}|\boldsymbol{x}_k)\,\mathrm{d}\boldsymbol{z}_{k,s}=1$.

V. Joint Posterior pdfs and Factor Graphs for a Known, Fixed Number of Targets

Bayesian estimation of the target states $\boldsymbol{x}_k^{(i)}$ typically relies on the posterior pdfs $f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$, where $\boldsymbol{z}_{1:k}\triangleq\big[\boldsymbol{z}_1^\mathrm{T}\cdots\boldsymbol{z}_k^\mathrm{T}\big]^\mathrm{T}$. The $f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ are marginals of the joint posterior pdf $f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})$, with $\boldsymbol{x}_{1:k}\triangleq\big[\boldsymbol{x}_1^\mathrm{T}\cdots\boldsymbol{x}_k^\mathrm{T}\big]^\mathrm{T}$. However, direct marginalization of $f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})$ is infeasible since the dimension of $\boldsymbol{x}_{1:k}$ grows linearly with the number of time steps k and the number of targets n_t, and thus the computational complexity scales exponentially in k and n_t. The exponential scaling in k can be avoided by exploiting statistical independencies across the time steps: using Bayes' rule on $f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})=f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k},\boldsymbol{m}_{1:k})$, where $\boldsymbol{m}_{1:k}\triangleq\big[\boldsymbol{m}_1^\mathrm{T}\cdots\boldsymbol{m}_k^\mathrm{T}\big]^\mathrm{T}$, we obtain

$$\begin{aligned}f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})&=\frac{f(\boldsymbol{z}_{1:k},\boldsymbol{m}_{1:k}|\boldsymbol{x}_{1:k})f(\boldsymbol{x}_{1:k})}{f(\boldsymbol{z}_{1:k},\boldsymbol{m}_{1:k})}\\&\propto f(\boldsymbol{z}_{1:k},\boldsymbol{m}_{1:k}|\boldsymbol{x}_{1:k})f(\boldsymbol{x}_{1:k}),\end{aligned}$$

since $\boldsymbol{z}_{1:k}$ (and, hence, also $\boldsymbol{m}_{1:k}$) are observed and thus considered fixed. (Note that knowledge of $\boldsymbol{m}_{1:k}$ is implied by knowledge of $\boldsymbol{z}_{1:k}$.) Using A1 and A11, we obtain further

$$f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})\propto\prod_{k'=1}^{k}f(\boldsymbol{x}_{k'}|\boldsymbol{x}_{k'-1})f(\boldsymbol{z}_{k'},\boldsymbol{m}_{k'}|\boldsymbol{x}_{k'}).\tag{18}$$

Here, $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ and $f(\boldsymbol{z}_k,\boldsymbol{m}_k|\boldsymbol{x}_k)$ are given by (8) and (16), respectively, and we formally introduced $f(\boldsymbol{x}_1|\boldsymbol{x}_0)\triangleq f(\boldsymbol{x}_1)$ (cf. (9)). The factorization (18) underlies sequential Bayesian estimation (filtering) using the prediction step (1) and the update step (2), whose complexity per time step is constant. The factor graph representing the factorization (18) is shown in Fig. 2(a). We note that sequential Bayesian estimation and related methods such as the Kalman filter and the particle filter can be interpreted as running the SPA on this simple tree-structured factor graph.

A. First Stretching Step: Introducing $\mathsf{a}_{1:k}$

The complexity of (1) and (2) still scales exponentially with n_t. To address this issue and obtain scalable SPA-based MTT methods, we use the stretching principle from Section III-A to introduce the DA vector $\boldsymbol{a}_{1:k}\triangleq\big[\boldsymbol{a}_1^\mathrm{T}\cdots\boldsymbol{a}_k^\mathrm{T}\big]^\mathrm{T}$ and formally replace the joint posterior pdf $f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})$ by $f(\boldsymbol{x}_{1:k},\boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k})$. Note that $\sum_{\boldsymbol{a}_{1:k}}f(\boldsymbol{x}_{1:k},\boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k})=f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})$, and thus the marginal posterior pdfs $f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ calculated from $f(\boldsymbol{x}_{1:k},\boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k})$ are equal to

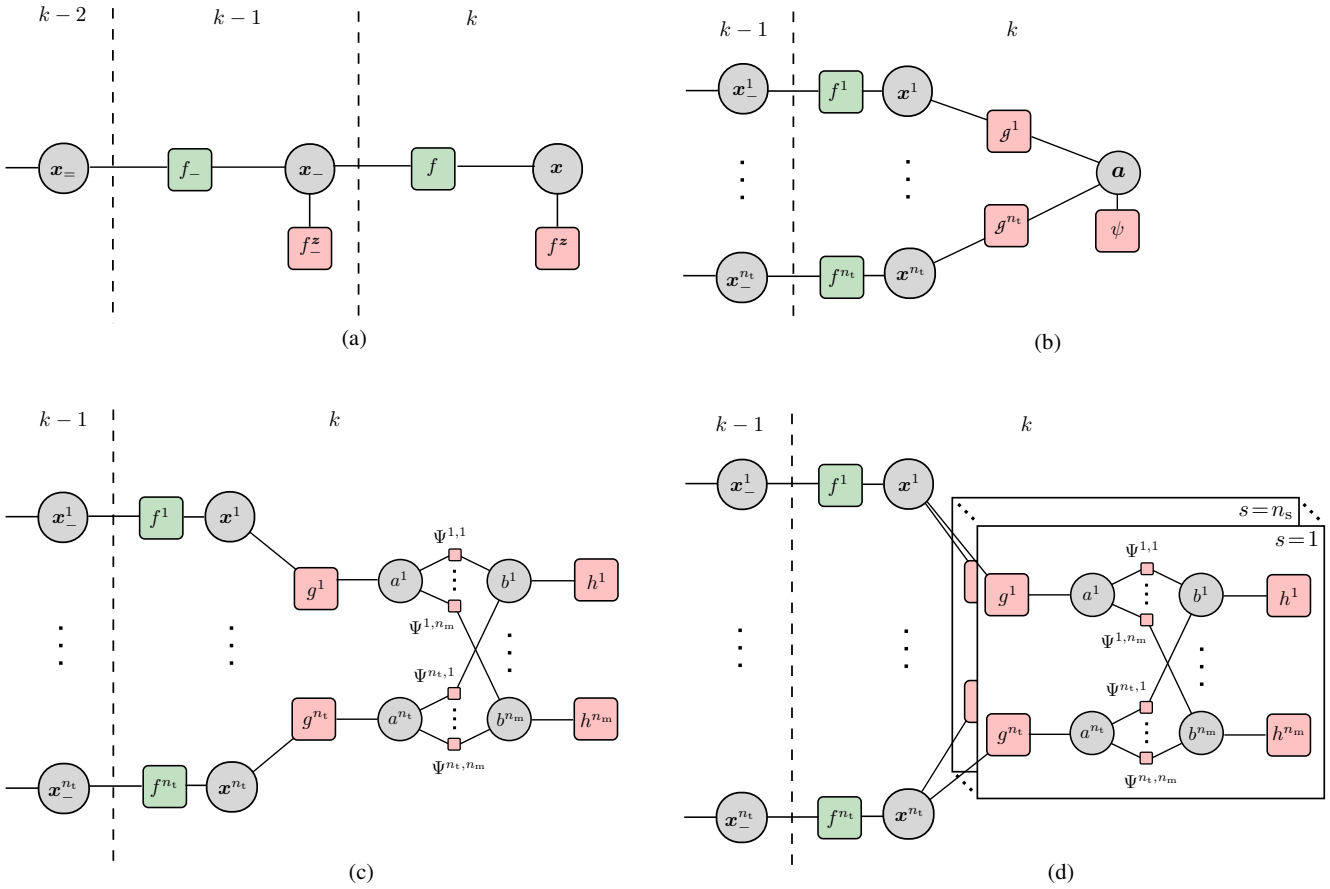


Fig. 2. Factor graphs for single-sensor and multisensor MTT, assuming a known, fixed number of targets: (a) Factor graph for sequential Bayesian estimation, corresponding to the factorization (18), (b) factor graph for the single-sensor MTT problem corresponding to the factorization (19) with $n_\mathrm{s}=1$, (c) stretched factor graph for the single-sensor MTT problem corresponding to the factorization (24) with $n_\mathrm{s}=1$, (d) factor graph for the multisensor MTT problem corresponding to the factorization (24). Two complete consecutive sections of the factor graph (for times $k-1$ and k) are shown in (a), and one complete section (for time k) in (b)–(d). Factor nodes in green represent factors related to the state-transition function and factor nodes in red represent factors related to the likelihood function. The time index k and the sensor index s are omitted, and the following short notations are used: $\boldsymbol{x}^i \triangleq \boldsymbol{x}_k^{(i)}$, $\boldsymbol{a}^i \triangleq \boldsymbol{a}_{k,s}^{(i)}$, $\boldsymbol{b}^m \triangleq \boldsymbol{b}_{k,s}^{(m)}$, $n_\mathrm{m} \triangleq m_{k,s}$, $\boldsymbol{x}_= \triangleq \boldsymbol{x}_{k-2}$, $\boldsymbol{x}_- \triangleq \boldsymbol{x}_{k-1}$, $\boldsymbol{x}_-^i \triangleq \boldsymbol{x}_{k-1}^{(i)}$, $f_- \triangleq f(\boldsymbol{x}_{k-1}|\boldsymbol{x}_{k-2})$, $f \triangleq f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$, $f^i \triangleq f(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})$, $f_-^{\boldsymbol{z}} \triangleq f(\boldsymbol{z}_{k-1}, \boldsymbol{m}_{k-1}|\boldsymbol{x}_{k-1})$, $f^{\boldsymbol{z}} \triangleq f(\boldsymbol{z}_k, \boldsymbol{m}_k|\boldsymbol{x}_k)$, $g^i \triangleq g(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s})$, $g^i \triangleq g(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s})$, $h^m \triangleq h(b_{k,s}^{(m)}; \boldsymbol{z}_{k,s}^{(m)})$, $\psi \triangleq \psi(\boldsymbol{a}_{k,s})$, and $\Psi^{i,m} \triangleq \Psi_{i,m}(a_{k,s}^{(i)}, b_{k,s}^{(m)})$.

the ones calculated from $f(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})$. Suitable modification of (18) yields the new factorization

$$\begin{aligned} &f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k}) \\ &\quad\propto \prod_{k'=1}^{k} f(\boldsymbol{x}_{k'}|\boldsymbol{x}_{k'-1}) f(\boldsymbol{z}_{k'}, \boldsymbol{a}_{k'}, \boldsymbol{m}_{k'}|\boldsymbol{x}_{k'}) \\ &\quad= \prod_{k'=1}^{k} f(\boldsymbol{x}_{k'}|\boldsymbol{x}_{k'-1}) \prod_{s=1}^{n_\mathrm{s}} f(\boldsymbol{z}_{k',s}, \boldsymbol{a}_{k',s}, m_{k',s}|\boldsymbol{x}_{k'}), \end{aligned}$$

where (15) was used in the last step. Inserting (8), (9), and (13), we obtain further

$$\begin{aligned} f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k}) \propto \prod_{k'=1}^{k} \left(\prod_{i=1}^{n_\mathrm{t}} f\big(\boldsymbol{x}_{k'}^{(i)}\big|\boldsymbol{x}_{k'-1}^{(i)}\big) \right) \prod_{s=1}^{n_\mathrm{s}} \psi(\boldsymbol{a}_{k',s}) \\ \times \prod_{i'=1}^{n_\mathrm{t}} g\big(\boldsymbol{x}_{k'}^{(i')}, a_{k',s}^{(i')}; \boldsymbol{z}_{k',s}\big), \quad (19) \end{aligned}$$

where $f\big(\boldsymbol{x}_1^{(i)}\big|\boldsymbol{x}_0^{(i)}\big) \triangleq f\big(\boldsymbol{x}_1^{(i)}\big)$. The factor graph representing this factorization is shown in Fig. 2(b) for the single-sensor case ($n_\mathrm{s}=1$). This factor graph has loops, which are however not visible because Fig. 2(b) essentially shows only the part of the factor graph corresponding to time step k.

B. Second Stretching Step: Introducing $\mathbf{b}_{1:k}$

Running the SPA on the factor graph in Fig. 2(b) is significantly less complex than running it on the factor graph in Fig. 2(a), because the marginalizations with respect to the multitarget state (the $\boldsymbol{x}_{k-1}$ integration in (1)) and with respect to the DA vector (the $\boldsymbol{a}_{k,s}$ summation in (17)) are avoided. However, there are still high-dimensional discrete marginalizations, whose complexity scales exponentially with n_t and $m_{k,s}$. This is because the messages related to the variable nodes $\boldsymbol{a}_{k,s}$ are functions of all the variables $a_{k,s}^{(i)}$, $i = 1, \ldots, n_\mathrm{t}$, where $a_{k,s}^{(i)} \in \{0, \ldots, m_{k,s}\}$. A computationally feasible SPA-based MTT method can be obtained by a modification of the factor graph in Fig. 2(b). We again invoke the stretching principle, this time to stretch the factor node "$\psi(\boldsymbol{a}_{k,s})$". Following [49], [50], we introduce the DA vector

$\mathbf{b}_{k,s} = \big[\mathsf{b}_{k,s}^{(1)} \cdots \mathsf{b}_{k,s}^{(m_{k,s})}\big]^{\mathrm{T}}$ with entries

$$\mathsf{b}_{k,s}^{(m)} \triangleq \begin{cases} i \in \{1,\dots,n_{\mathrm{t}}\}, & \text{if at time } k \text{, measurement } m \\ & \text{at sensor } s \text{ is generated by} \\ & \text{target } i \\ 0, & \text{if at time } k \text{, measurement } m \\ & \text{at sensor } s \text{ is not generated} \\ & \text{by a target.} \end{cases} \tag{20}$$

Note that $\mathbf{b}_{k,s}$ carries the same information as $\mathbf{a}_{k,s}$ but in a different form. If $\mathbf{b}_{k,s}$ was known, $\mathbf{a}_{k,s}$ would be known as well and the DA problem would be solved. However, $\mathbf{b}_{k,s}$ is unknown and thus considered as a latent random variable in the inference problem. The admissibility of a DA hypothesis according to the constraint A4 can now be expressed by the indicator function

$$\psi(\boldsymbol{a}_{k,s}, \boldsymbol{b}_{k,s}) \triangleq \prod_{i=1}^{n_{\mathrm{t}}} \prod_{m=1}^{m_{k,s}} \Psi_{i,m}\big(a_{k,s}^{(i)}, b_{k,s}^{(m)}\big), \tag{21}$$

with

$$\Psi_{i,m}\big(a_{k,s}^{(i)}, b_{k,s}^{(m)}\big) \triangleq \begin{cases} 0, & a_{k,s}^{(i)} = m,\ b_{k,s}^{(m)} \neq i \\ & \text{or } b_{k,s}^{(m)} = i,\ a_{k,s}^{(i)} \neq m \\ 1, & \text{otherwise.} \end{cases} \tag{22}$$

We note that $\sum_{\boldsymbol{b}_{k,s}} \psi(\boldsymbol{a}_{k,s}, \boldsymbol{b}_{k,s}) = \psi(\boldsymbol{a}_{k,s})$. Replacing $\psi(\boldsymbol{a}_{k,s})$ in (19) by $\psi(\boldsymbol{a}_{k,s}, \boldsymbol{b}_{k,s})$ and using (21) results in the new joint posterior pdf

$$\begin{aligned} f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k} | \boldsymbol{z}_{1:k}) \propto & \prod_{k'=1}^{k} \prod_{i=1}^{n_{\mathrm{t}}} f\big(\boldsymbol{x}_{k'}^{(i)} \big| \boldsymbol{x}_{k'-1}^{(i)}\big) \\ & \times \prod_{s=1}^{n_{\mathrm{s}}} g\big(\boldsymbol{x}_{k'}^{(i)}, a_{k',s}^{(i)}; \boldsymbol{z}_{k',s}\big) \\ & \times \prod_{m=1}^{m_{k',s}} \Psi_{i,m}\big(a_{k',s}^{(i)}, b_{k',s}^{(m)}\big). \end{aligned} \tag{23}$$

Finally, we perform a last modification of the factorization. This modification does not further reduce the complexity, but it is the basis for a general SPA-based DA algorithm, to be presented in Section VI, which can be used in many different MTT methods. First, we multiply (23) by the constant $c(\boldsymbol{z}_{1:k}) \triangleq \prod_{k'=1}^{k} \prod_{s=1}^{n_{\mathrm{s}}} \prod_{m=1}^{m_{k,s}} \mu_{\mathrm{c}}^{(s)} f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k',s}^{(m)}\big)$. For given k and s, let $\mathcal{M}_{k,s}^{0} \subseteq \{1,\dots,m_{k,s}\}$ be the set of all m with $b_{k,s}^{(m)} = 0$, i.e., all m indexing clutter measurements. Because of A4 expressed by $\prod_{i=1}^{n_{\mathrm{t}}} \prod_{m=1}^{m_{k,s}} \Psi_{i,m}\big(a_{k,s}^{(i)}, b_{k,s}^{(m)}\big)$, for each $m \in \{1,\dots,m_{k,s}\} \backslash \mathcal{M}_{k,s}^{0}$, the joint posterior pdf in (23) is nonzero only if there is exactly one $a_{k,s}^{(i)}$ that equals m. This means that for $a_{k,s}^{(i)} = m \in \{1,\dots,m_{k,s}\}$, the denominator $\mu_{\mathrm{c}}^{(s)} f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k,s}^{(a_{k,s}^{(i)})}\big)$ of $g\big(\boldsymbol{x}_{k}^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$ (see (14), recalling that $f^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)} \big| \boldsymbol{x}_{k}^{(i)}\big) = f\big(\boldsymbol{z}_{k,s}^{(m)} \big| \boldsymbol{x}_{k}^{(i)}\big) / f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big)$) is canceled by the corresponding factor $\mu_{\mathrm{c}}^{(s)} f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big)$ in $c(\boldsymbol{z}_{1:k})$. Equation (23) thus becomes

$$\begin{aligned} & f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k} | \boldsymbol{z}_{1:k}) \\ & \propto \prod_{k'=1}^{k} \left(\prod_{i'=1}^{n_{\mathrm{t}}} f\big(\boldsymbol{x}_{k'}^{(i')} \big| \boldsymbol{x}_{k'-1}^{(i')}\big) \right) \prod_{s=1}^{n_{\mathrm{s}}} \left(\prod_{i=1}^{n_{\mathrm{t}}} g\big(\boldsymbol{x}_{k'}^{(i)}, a_{k',s}^{(i)}; \boldsymbol{z}_{k',s}\big) \right. \\ & \qquad \left. \times \prod_{m'=1}^{m_{k',s}} \Psi_{i,m'}\big(a_{k',s}^{(i)}, b_{k',s}^{(m')}\big) \right) \prod_{m \in \mathcal{M}_{k',s}^{0}} \mu_{\mathrm{c}}^{(s)} f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k',s}^{(m)}\big), \end{aligned}$$

where for $a_{k,s}^{(i)} \in \{1,\dots,m_{k,s}\}$,

$$\begin{aligned} g\big(\boldsymbol{x}_{k}^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big) &\triangleq g\big(\boldsymbol{x}_{k}^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big) \mu_{\mathrm{c}}^{(s)} f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k,s}^{(a_{k,s}^{(i)})}\big) \\ &= p_{\mathrm{d}}^{(s)}\big(\boldsymbol{x}_{k}^{(i)}\big) f\big(\boldsymbol{z}_{k,s}^{(a_{k,s}^{(i)})} \big| \boldsymbol{x}_{k}^{(i)}\big), \end{aligned}$$

and for $a_{k,s}^{(i)} = 0$, $g\big(\boldsymbol{x}_{k}^{(i)}, 0; \boldsymbol{z}_{k,s}\big) \triangleq g\big(\boldsymbol{x}_{k}^{(i)}, 0; \boldsymbol{z}_{k,s}\big) = \big(1 - p_{\mathrm{d}}^{(s)}\big(\boldsymbol{x}_{k}^{(i)}\big)\big)$ (cf. (14)). Introducing

$$h\big(b_{k,s}^{(m)}; \boldsymbol{z}_{k,s}^{(m)}\big) \triangleq \begin{cases} \mu_{\mathrm{c}}^{(s)} f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big), & b_{k,s}^{(m)} = 0 \\ 1, & b_{k,s}^{(m)} > 1, \end{cases}$$

we obtain the final factorization of $f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k} | \boldsymbol{z}_{1:k})$ as

$$\begin{aligned} & f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k} | \boldsymbol{z}_{1:k}) \\ & \propto \prod_{k'=1}^{k} \left(\prod_{i'=1}^{n_{\mathrm{t}}} f\big(\boldsymbol{x}_{k'}^{(i')} \big| \boldsymbol{x}_{k'-1}^{(i')}\big) \right) \prod_{s=1}^{n_{\mathrm{s}}} \left(\prod_{i=1}^{n_{\mathrm{t}}} g\big(\boldsymbol{x}_{k'}^{(i)}, a_{k',s}^{(i)}; \boldsymbol{z}_{k',s}\big) \right. \\ & \qquad \left. \times \prod_{m'=1}^{m_{k',s}} \Psi_{i,m'}\big(a_{k',s}^{(i)}, b_{k',s}^{(m')}\big) \right) \prod_{m=1}^{m_{k',s}} h\big(b_{k',s}^{(m)}; \boldsymbol{z}_{k',s}^{(m)}\big), \end{aligned} \tag{24}$$

with $f\big(\boldsymbol{x}_{1}^{(i)} \big| \boldsymbol{x}_{0}^{(i)}\big) = f\big(\boldsymbol{x}_{1}^{(i)}\big)$. Note that $f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k} | \boldsymbol{z}_{1:k})$ is again consistent with the original joint posterior pdf $f(\boldsymbol{x}_{1:k} | \boldsymbol{z}_{1:k})$ in (18) in the sense that $\sum_{\boldsymbol{a}_{1:k}} \sum_{\boldsymbol{b}_{1:k}} f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k} | \boldsymbol{z}_{1:k}) = f(\boldsymbol{x}_{1:k} | \boldsymbol{z}_{1:k})$.

The factor graph representing the factorization (24) is shown in Fig. 2(c) for the single-sensor case ($n_{\mathrm{s}} = 1$) and in Fig. 2(d) for the multisensor case. These factor graphs contain one or multiple "bipartite" substructures that are associated with probabilistic DA. An interesting interpretation of these substructures is that information on target-originated measurements in the form of $g\big(\boldsymbol{x}_{k}^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$ is connected to target-oriented DA variables and information on clutter-originated measurements in the form of $h\big(b_{k,s}^{(m)}; \boldsymbol{z}_{k,s}^{(m)}\big)$ is connected to measurement-oriented DA variables. The two factor graphs will be used in Sections VI and VII as a basis for developing scalable SPA-based methods for MTT.

VI. SPA-BASED PROBABILISTIC DA

Next, we describe an efficient loopy SPA-based algorithm for probabilistic DA [48]–[50]. This algorithm will constitute an important building block of the SPA-based MTT methods developed in Sections VII, IX, and XIII. We consider n_{t} targets, where n_{t} is assumed known, and a single sensor (thus, we drop the sensor index s). The assumptions of a fixed, known number of targets and of a single sensor will be lifted in later sections.

The goal of probabilistic DA is to calculate the posterior DA pmfs $p\big(a_{k}^{(i)} \big| \boldsymbol{z}_{1:k}\big)$, where $a_{k}^{(i)}$ was defined in (10). These pmfs are marginals of the joint posterior pdf $f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k} | \boldsymbol{z}_{1:k})$ in (24). We now use the loopy SPA on the factor graph in Fig. 2(c) to calculate accurate approximations of these marginal pmfs. We choose a message calculation schedule such that each time k is considered individually; this is

achieved by passing messages only forward in time[1] [42]. According to (5), the messages passed from the factor nodes "$f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{x}_{k-1}^{(i)}\big)$" to the variable nodes "$\boldsymbol{x}_k^{(i)}$" are given by

$$\alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big) = \int f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{x}_{k-1}^{(i)}\big)\, \tilde{f}\big(\boldsymbol{x}_{k-1}^{(i)}\big)\, \mathrm{d}\boldsymbol{x}_{k-1}^{(i)}. \tag{25}$$

Here, $\tilde{f}\big(\boldsymbol{x}_{k-1}^{(i)}\big)$ is the belief approximating the marginal posterior pdf of $\boldsymbol{x}_{k-1}^{(i)}$; this belief was calculated at time k–1. The message calculation in (25) is analogous to the prediction step in sequential filtering (cf. (1)). Similarly, the messages passed from the factor nodes "$g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)$" to the variable nodes "$a_k^{(i)}$" are given by

$$\beta_k^{(i)}\big(a_k^{(i)}\big) = \int g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)\, \alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)\, \mathrm{d}\boldsymbol{x}_k^{(i)},$$

and the messages passed from the factor nodes "$h\big(b_k^{(m)}; \boldsymbol{z}_k^{(m)}\big)$" to the variable nodes "$b_k^{(m)}$" are given by $\xi_k^{(m)}\big(b_k^{(m)}\big) = h\big(b_k^{(m)}; \boldsymbol{z}_k^{(m)}\big)$. The $\beta_k^{(i)}\big(a_k^{(i)}\big)$ and $\xi_k^{(m)}\big(b_k^{(m)}\big)$ are referred to as association weights [50], [51], [107].

For future reference, we note that the belief of the joint DA vector $[\mathbf{a}_k^{\mathrm{T}}\ \mathbf{b}_k^{\mathrm{T}}]^{\mathrm{T}}$ can be obtained by interpreting all variable nodes related to the DA variables $\mathbf{a}_k^{(i)}$, $i = 1, \ldots, n_{\mathrm{t}}$ and $\mathbf{b}_k^{(m)}$, $m = 1, \ldots, m_{k,s}$ on the factor graph in Fig. 2(c) as one "supernode" and calculating the belief of that supernode according to (7). This belief is obtained as

$$\tilde{p}(\boldsymbol{a}_k, \boldsymbol{b}_k) \propto \psi\big(\boldsymbol{a}_k, \boldsymbol{b}_k\big) \left(\prod_{i=1}^{n_{\mathrm{t}}} \beta_k^{(i)}\big(a_k^{(i)}\big) \right) \prod_{m=1}^{m_k} \xi_k^{(m)}\big(b_k^{(m)}\big), \tag{26}$$

where $\psi\big(\boldsymbol{a}_k, \boldsymbol{b}_k\big)$ is given by (21).

A. SPA-Based DA

Once $\alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$ and $\beta_k^{(i)}\big(a_k^{(i)}\big)$ have been calculated, the iterative SPA involving the factor nodes "$\Psi_{i,m}\big(a_k^{(i)}, b_k^{(m)}\big)$" and the variable nodes "$a_k^{(i)}$" and "$b_k^{(m)}$" is performed for all states $i = 1, \ldots, n_{\mathrm{t}}$ and all measurements $m = 1, \ldots, m_k$ in parallel. More specifically, at message passing iteration $\ell \in \{1, \ldots, n_{\mathrm{it}}\}$, a message $\varphi^{[\ell]}_{\Psi_{i,m} \to b_k^{(m)}}\big(b_k^{(m)}\big)$ is passed from "$\Psi_{i,m}\big(a_k^{(i)}, b_k^{(m)}\big)$" to "$b_k^{(m)}$", and a message $\nu^{[\ell]}_{\Psi_{i,m} \to a_k^{(i)}}\big(a_k^{(i)}\big)$ is passed from "$\Psi_{i,m}\big(a_k^{(i)}, b_k^{(m)}\big)$" to "$a_k^{(i)}$". Since each factor node "$\Psi_{i,m}\big(a_k^{(i)}, b_k^{(m)}\big)$" is connected to only two variable nodes, an outgoing message from such a factor node can be obtained from the incoming message by inserting (6) into the discrete counterpart of (5). In this way, one obtains the following recursion for the messages φ and ν:

$$\begin{aligned}
&\varphi^{[\ell]}_{\Psi_{i,m} \to b_k^{(m)}}\big(b_k^{(m)}\big) \\
&= \sum_{a_k^{(i)}=0}^{m_k} \beta_k^{(i)}\big(a_k^{(i)}\big)\, \Psi_{i,m}\big(a_k^{(i)}, b_k^{(m)}\big) \prod_{\substack{m'=1 \\ m' \neq m}}^{m_k} \nu^{[\ell]}_{\Psi_{i,m'} \to a_k^{(i)}}\big(a_k^{(i)}\big)
\end{aligned} \tag{27}$$

and

$$\begin{aligned}
&\nu^{[\ell]}_{\Psi_{i,m} \to a_k^{(i)}}\big(a_k^{(i)}\big) \\
&= \sum_{b_k^{(m)}=0}^{n_{\mathrm{t}}} \xi_k^{(m)}\big(b_k^{(m)}\big)\, \Psi_{i,m}\big(a_k^{(i)}, b_k^{(m)}\big) \prod_{\substack{i'=1 \\ i' \neq i}}^{n_{\mathrm{t}}} \varphi^{[\ell-1]}_{\Psi_{i',m} \to b_k^{(m)}}\big(b_k^{(m)}\big),
\end{aligned} \tag{28}$$

for $i = 1, \ldots, n_{\mathrm{t}}$ and $m = 1, \ldots, m_k$. This iterative algorithm is initialized by the messages

$$\varphi^{[0]}_{\Psi_{i,m} \to b_k^{(m)}}\big(b_k^{(m)}\big) = \sum_{a_k^{(i)}=0}^{m_k} \beta_k^{(i)}\big(a_k^{(i)}\big)\, \Psi_{i,m}\big(a_k^{(i)}, b_k^{(m)}\big). \tag{29}$$

After the last iteration $\ell = n_{\mathrm{it}}$, approximations $\tilde{p}\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ of the marginal posterior DA pmfs $p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ are obtained according to (7), i.e.,

$$\tilde{p}\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big) = A_k^{(i)}\, \beta_k^{(i)}\big(a_k^{(i)}\big) \prod_{m=1}^{m_k} \nu^{[n_{\mathrm{it}}]}_{\Psi_{i,m} \to a_k^{(i)}}\big(a_k^{(i)}\big),$$

for $i = 1, \ldots, n_{\mathrm{t}}$, where the $A_k^{(i)}$ are normalization factors.

B. A Scalable SPA-Based DA Algorithm

The messages (27) and (28) can be simplified [50], [118], [128]. Because of the binary consistency constraints expressed by $\Psi_{i,m}\big(a_k^{(i)}, b_k^{(m)}\big)$, each message comprises only two different values: $\varphi^{[\ell]}_{\Psi_{i,m} \to b_k^{(m)}}\big(b_k^{(m)}\big)$ in (27) takes on one value for $b_k^{(m)} = i$ and another for all $b_k^{(m)} \neq i$, and $\nu^{[\ell]}_{\Psi_{i,m} \to a_k^{(i)}}\big(a_k^{(i)}\big)$ in (28) takes on one value for $a_k^{(i)} = m$ and another for all $a_k^{(i)} \neq m$. Thus, each message can be represented (up to an irrelevant constant factor) by the ratio of the first value and the second value, hereafter denoted as $\varphi_k^{[\ell](i \to m)}$ or $\nu_k^{[\ell](m \to i)}$. The recursion (27), (28) can then be reformulated as

$$\varphi_k^{[\ell](i \to m)} = \frac{\beta_k^{(i)}(m)}{\beta_k^{(i)}(0) + \sum_{\substack{m'=1 \\ m' \neq m}}^{m_k} \beta_k^{(i)}(m')\, \nu_k^{[\ell](m' \to i)}} \tag{30}$$

$$\nu_k^{[\ell](m \to i)} = \frac{\xi_k^{(m)}(i)}{\xi_k^{(m)}(0) + \sum_{\substack{i'=1 \\ i' \neq i}}^{n_{\mathrm{t}}} \xi_k^{(m)}(i')\, \varphi_k^{[\ell-1](i' \to m)}}, \tag{31}$$

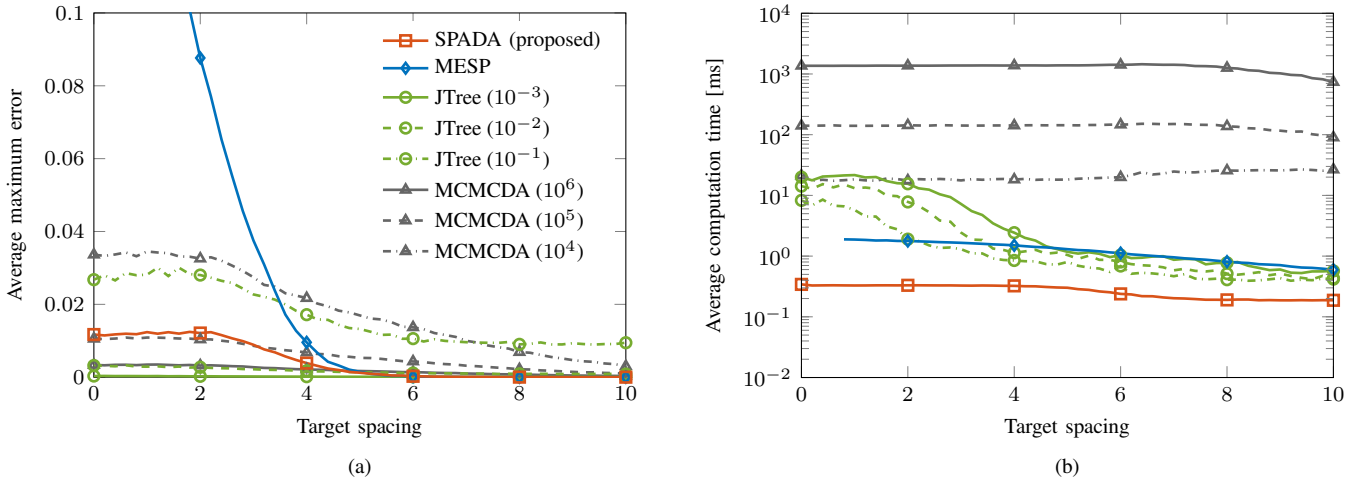
for $i = 1, \ldots, n_{\mathrm{t}}$ and $m = 1, \ldots, m_k$. This iterative algorithm is initialized by $\varphi_k^{[0](i \to m)} = \beta_k^{(i)}(m)/\beta_k^{(i)}(0)$. Note that this reformulation exploits the fact that, after replacing $\varphi^{[\ell]}_{\Psi_{i,m} \to b_k^{(m)}}\big(b_k^{(m)}\big)$ in (27) by $\varphi_k^{[\ell](i \to m)}$ and $\nu^{[\ell]}_{\Psi_{i,m} \to a_k^{(i)}}\big(a_k^{(i)}\big)$ in (28) by $\nu_k^{[\ell](m \to i)}$, for each term in the sum only one factor in the product of messages is different from 1. After the last iteration $\ell = n_{\mathrm{it}}$, approximations of the $p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ are obtained as

$$\tilde{p}\big(a_k^{(i)} = m\big|\boldsymbol{z}_{1:k}\big) = \frac{\beta_k^{(i)}(m)\, \nu_k^{[n_{\mathrm{it}}](m \to i)}}{\beta_k^{(i)}(0) + \sum_{m'=1}^{m_k} \beta_k^{(i)}(m')\, \nu_k^{[n_{\mathrm{it}}](m' \to i)}},$$

for $m = 0, 1, \ldots, m_k$. Here, for $m = 0$, $\nu_k^{[n_{\mathrm{it}}](0 \to i)} \triangleq 1$. In what follows, this efficient algorithm will be referred to as *Sum-Product Algorithm for Data Association* (SPADA). Note that one can also obtain approximations of the "measurement-oriented" DA pmfs $p\big(b_k^{(m)}\big|\boldsymbol{z}_{1:k}\big)$ as

$$\tilde{p}\big(b_k^{(m)} = i\big|\boldsymbol{z}_{1:k}\big) = \frac{\xi_k^{(m)}\big(i\big)\, \varphi_k^{[n_{\mathrm{it}}](i \to m)}}{\xi_k^{(m)}(0) + \sum_{i'=1}^{n_{\mathrm{t}}} \xi_k^{(m)}\big(i'\big)\, \varphi_k^{[n_{\mathrm{it}}](i' \to m)}},$$

[1] We note that SPA-based algorithms for probabilistic DA that consider multiple time steps jointly were introduced in [145].



(a) (b)

Fig. 3. Experimental comparison of SPADA with three alternative DA algorithms: (a) Average maximum error of the estimated marginal posterior pmfs $\tilde{p}\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ versus the target spacing, (b) average computation time versus the target spacing.

for $i = 0, 1, \ldots, n_\mathrm{t}$. Here, for $i=0$, $\varphi_k^{[n_\mathrm{it}](0\to m)} \triangleq 1$.

It was shown in [50], [128] that the recursion (30), (31) is a contraction, which is guaranteed to converge. In addition, it was shown in [127] and in [145, Appendix A-A] that the recursion (30), (31) solves a convex optimization problem, and thus it ultimately converges to the corresponding global optimum. Moreover, the number of iterations required to meet a specific convergence criterion is bounded [50]. Finally, the complexity of (30), (31) is significantly lower than that of (27), (28). In each iteration ℓ, $n_\mathrm{t} m_k$ messages $\varphi_k^{[\ell](i\to m)}$ and $n_\mathrm{t} m_k$ messages $\nu_k^{[\ell](m\to i)}$ are calculated. The associated complexity is essentially determined by that of calculating the n_t sums $\sum_{m'=1,m'\neq m}^{m_k} \beta_k^{(i)}(m')\,\nu_k^{[\ell](m'\to i)}$ and the m_k sums $\sum_{i'=1,i'\neq i}^{n_\mathrm{t}} \xi_k^{(m)}(i')\,\varphi_k^{[\ell-1](i'\to m)}$, which scales as $O(n_\mathrm{t} m_k)$. Hence, the overall complexity of SPADA per iteration is $O(n_\mathrm{t} m_k)$. This is much smaller than the complexity of (27) and (28), which is $O(n_\mathrm{t}^2 m_k^3 + n_\mathrm{t}^3 m_k^2)$. We note that MATLAB code for the recursion (30), (31) is provided in [128].

C. Simulation Results

We demonstrate the performance of SPADA for a two-dimensional (2-D) scenario with $n_\mathrm{t} = 6$ static targets arranged on a regular 2×3 grid [50]. The spacing of the targets is varied between 0 and 10. The measurements are the target positions plus circularly symmetric 2-D Gaussian noise of zero mean and variance 1. The clutter measurements are uniformly distributed over the region of interest with intensity $\mu_\mathrm{c} f_\mathrm{c}\big(\boldsymbol{z}_k^{(m)}\big) = 0.01$. The region of interest consists of the gates [1] of the six targets with gate threshold 18.4, which means that target-originated measurements are "gated out" with probability 10^{-4}. The probability that a target is detected by the sensor is $p_\mathrm{d}\big(\boldsymbol{x}_k^{(i)}\big) = 0.6$. Prior information on the $\boldsymbol{x}_k^{(i)}$ (replacing $\alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$ in (25)) is obtained through the procedure described in [50]. The number of message passing iterations n_it is chosen adaptively according to [50].

We compare SPADA with the following three alternative algorithms (cf. Section II-C): the *MESP* algorithm, which performs the SPA on a factor graph that is obtained with a different stretching of the factor node "$\psi(\boldsymbol{a}_k)$" based on elementary mutual exclusion constraints [49], [124] [32, Box 12.D]; the *JTree* algorithm, which performs the junction tree algorithm [25], [32] on that alternative factor graph, with association weights thresholded to 10^{-3}, 10^{-2}, or 10^{-1} to induce sparsity; and the *MCMCDA* algorithm [146] using 10^4, 10^5, or 10^6 Markov chain Monte Carlo steps. We consider only a single time step, $k = 1$, and accordingly, e.g., $\tilde{p}\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big) = \tilde{p}\big(a_1^{(i)}\big|\boldsymbol{z}_1\big)$. Fig. 3 shows the average maximum error of $\tilde{p}\big(a_1^{(i)}\big|\boldsymbol{z}_1\big)$ and the average computation time versus the target spacing. The average maximum error is calculated by maximizing $\big|\tilde{p}\big(a_k^{(i)}{=}m\big|\boldsymbol{z}_1\big) - p\big(a_1^{(i)}{=}m\big|\boldsymbol{z}_1\big)\big|$ over all m and averaging the result over the six targets ($i = 1, \ldots, 6$) and over 1000 simulation runs. (We do not plot the results of the MESP algorithm for very small target spacings because the MESP algorithm occasionally failed to converge in that case.) SPADA is seen to outperform the MESP algorithm when the targets are closely spaced, to outperform the JTree (10^{-1}) algorithm and the MCMCDA (10^4) algorithm, and to perform similarly to the MCMCDA (10^5) algorithm. The computation time of SPADA is seen to be always lower—partly by several orders of magnitude—than that of the other algorithms. This is true even though the MESP and JTree algorithms were implemented using a C++ library [147] whereas SPADA was implemented in MATLAB. Further experiments presented in [50] confirm the high accuracy and efficiency of SPADA.

VII. SPA-Based Vector-Type MTT Methods for a Known, Fixed Number of Targets

In this section, we consider the application of the SPA to MTT within the vector-type system model described in Section IV, i.e., assuming a known, fixed number n_t of targets. First, we formulate two "total-SPA" methods for single-sensor MTT, where the second method is based on SPADA. Next, we review the JPDA filter and develop a "SPADA-embedded JPDA filter," which is a JPDA filter using SPADA for an efficient approximative calculation of the marginal DA pmfs.

Remarkably, the first total-SPA method is equal to the JPDA filter and the second total-SPA method is equal to the SPADA-embedded JPDA filter. The second total-SPA method is finally extended to obtain scalable MTT methods for multiple sensors. New variants of these methods for multisensor MTT that allow for an unknown, time-varying number of targets will be presented in Section IX.

A. SPA-Based Methods for Single-Sensor MTT

We will now show that the SPA can be used not only for an efficient calculation of the marginal DA pmfs but for the entire MTT problem. The number of targets, n_t, is still assumed fixed and known. Furthermore, we still consider the single-sensor case ($n_\mathrm{s} = 1$), and thus again drop the sensor index s.

1) Total-SPA Method for Single-Sensor MTT: In contrast to Section VI, where we calculated the marginal DA pmfs $p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ by running the SPA on the bipartite part of the factor graph in Fig. 2(c), we now calculate the marginal posterior target state pdfs $f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ by running the SPA on the factor graph in Fig. 2(b). We again use a message calculation schedule such that messages are passed only forward in time and hence a noniterative SPA is obtained.

First, the messages $\alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$ passed from the factor nodes "$f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{x}_{k-1}^{(i)}\big)$" to the variable nodes "$\boldsymbol{x}_k^{(i)}$" are calculated according to (25), and the messages $\beta_k^{(i)}\big(a_k^{(i)}\big)$ passed from the factor nodes "$g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)$" to the variable node "$\boldsymbol{a}_k$" are calculated according to[2] (26) (with $g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)$ replaced by $g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)$). Then, the messages passed from "$\boldsymbol{a}_k$" to "$g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)$" are calculated according to (6), i.e.,

$$\omega_k^{(i)}(\boldsymbol{a}_k) = \psi(\boldsymbol{a}_k) \prod_{\substack{i'=1 \\ i' \neq i}}^{n_\mathrm{t}} \beta_k^{(i')}\big(a_k^{(i')}\big).$$

Next, the messages passed from "$g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)$" to "$\boldsymbol{x}_k^{(i)}$" are calculated according to the discrete counterpart of (5), i.e.,

$$\begin{aligned}\gamma_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big) &= \sum_{\boldsymbol{a}_k} g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)\, \omega_k^{(i)}(\boldsymbol{a}_k) \\ &= \sum_{a_k^{(i)}=0}^{m_k} g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)\, \kappa_k^{(i)}\big(a_k^{(i)}\big), \end{aligned} \tag{32}$$

with

$$\kappa_k^{(i)}\big(a_k^{(i)}\big) \triangleq \sum_{\sim a_k^{(i)}} \omega_k^{(i)}(\boldsymbol{a}_k), \tag{33}$$

where $\sum_{\sim a_k^{(i)}}$ denotes the summation over all $a_k^{(i')} \in \{0, \ldots, m_k\}$ for all $i' \in \{1, \cdots, n_\mathrm{t}\} \backslash \{i\}$. Finally, the beliefs for the $\boldsymbol{x}_k^{(i)}$ are obtained according to (7), i.e,

$$\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big) \propto \alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big) \gamma_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big). \tag{34}$$

These beliefs approximate the marginal posterior pdfs $f\big(\boldsymbol{x}_k^{(i)}|\boldsymbol{z}_{1:k}\big)$. They can be used for Bayesian state estimation, e.g., by means of the MMSE estimator in (3).

2) A Scalable Total-SPA Method for Single-Sensor MTT: The complexity of the total-SPA method presented above is exponential in n_t due to the summation in (33). This exponential scaling can be avoided by running the SPA on the *entire* factor graph in Fig. 2(c) rather than on the factor graph in Fig. 2(b) (or on the bipartite part of the factor graph in Fig. 2(c) as done for DA in Section VI). The resulting method will also be used in Section VII-C to develop scalable MTT methods for multiple sensors. Due to the additional stretching step underlying the factor graph in Fig. 2(c), the summation in (33) can be avoided by using SPADA.

More specifically, after calculating the messages $\alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$ and $\beta_k^{(i)}\big(a_k^{(i)}\big)$ according to (25) and (26), respectively, the iterative SPA (27)–(29) is executed, which yields the messages $\nu_{\Psi_{i,m} \to a_k^{(i)}}^{[n_\mathrm{it}]}\big(a_k^{(i)}\big)$ in (28). Then, the messages passed from "$a_k^{(i)}$" to "$g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)$" are calculated according to (6), i.e.,

$$\kappa_k^{(i)}\big(a_k^{(i)}\big) = \prod_{m=1}^{m_k} \nu_{\Psi_{i,m} \to a_k^{(i)}}^{[n_\mathrm{it}]}\big(a_k^{(i)}\big). \tag{35}$$

Next, the messages $\gamma_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$ passed from "$g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)$" to "$\boldsymbol{x}_k^{(i)}$" are calculated according to (32) with $\kappa_k^{(i)}\big(a_k^{(i)}\big)$ replaced by $\kappa_k^{(i)}\big(a_k^{(i)}\big)$. Thereby, the high-dimensional summation in (33) is avoided. Finally, the iterative SPA (27)–(29) is replaced by SPADA. For this, the messages $\nu_k^{[n_\mathrm{it}](m \to i)}$ in (31) are converted into the messages $\nu_{\Psi_{i,m} \to a_k^{(i)}}^{[n_\mathrm{it}]}\big(a_k^{(i)}\big)$ used in (35) according to $\nu_{\Psi_{i,m} \to a_k^{(i)}}^{[n_\mathrm{it}]}\big(a_k^{(i)}\big) = \nu_k^{[n_\mathrm{it}](m \to i)}$ for $a_k^{(i)} = m$ and $\nu_{\Psi_{i,m} \to a_k^{(i)}}^{[n_\mathrm{it}]}\big(a_k^{(i)}\big) = 1$ for $a_k^{(i)} \neq m$. Finally, evaluation of (34) yields the beliefs[3] $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big)$. This total-SPA method for single-sensor MTT has excellent scalability (namely, $O(n_\mathrm{t} m_k)$) and is thus suitable also for large tracking scenarios with closely spaced targets.

B. JPDA Filter with SPA-Based Probabilistic DA

The JPDA filter is a vector-type MTT method that approximates the joint posterior DA pmf by the product of its marginals. While the original JPDA filter additionally employs a Gaussian approximation for the target state pdfs [1], we here use the term JPDA more broadly to refer to a method that propagates a marginal pdf (without a parametric representation) separately for each single-target state [82]. In this subsection, we first review the JPDA filter and then show how to embed SPADA.

1) Prediction Step: In the prediction step at time k, the Chapman-Kolmogorov equation (1) is used to convert the approximate posterior pdf $\tilde{f}(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ calculated at time $k-$

[2]Note that this is still consistent with our general notation defined by (5), because the message $\beta_k^{(i)}\big(a_k^{(i)}\big)$ can be interpreted as a function in $\boldsymbol{a}_k$ (which is constant in all $a_k^{(i')}$ except $a_k^{(i)}$).

[3]These beliefs are not the same approximations of the marginal posterior pdfs $f\big(\boldsymbol{x}_k^{(i)}|\boldsymbol{z}_{1:k}\big)$ as in (34) because they are based on loopy SPA for DA. For simplicity, we do not indicate this difference by our notation.

1 into an approximate predicted posterior pdf $\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$. We assume that $\tilde{f}(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ factors into its marginals, i.e.,

$$\tilde{f}(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) = \prod_{i=1}^{n_\mathrm{t}} \tilde{f}\big(\boldsymbol{x}_{k-1}^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)\,. \tag{36}$$

The validity of this assumption will be verified and discussed in Section VII-B2. Note also that for $k-1=1$, (36) is equivalent to (9). Inserting (36) and (8) into (1) yields

$$\begin{aligned}\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) &= \prod_{i=1}^{n_\mathrm{t}} \int f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{x}_{k-1}^{(i)}\big)\, \tilde{f}\big(\boldsymbol{x}_{k-1}^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)\,\mathrm{d}\boldsymbol{x}_{k-1}^{(i)} \\ &= \prod_{i=1}^{n_\mathrm{t}} \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)\,, \end{aligned} \tag{37}$$

with

$$\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k-1}\big) = \int f\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{x}_{k-1}^{(i)}\big)\, \tilde{f}\big(\boldsymbol{x}_{k-1}^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)\,\mathrm{d}\boldsymbol{x}_{k-1}^{(i)}\,, \tag{38}$$

for $i = 1,\ldots,n_\mathrm{t}$. Thus, the prediction step (1)—which required an $n_\mathrm{t}d_x$-dimensional integration—reduces to n_t separate predictions of single-target states, each requiring only a d_x-dimensional integration.

2) Update Step: In the update step at time k, the approximate posterior pdf $\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is calculated from the approximate predicted posterior pdf $\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$ in (37) and the current measurement vector $\boldsymbol{z}_k$ (cf. (2)). To derive the update rule, we first expand the approximate posterior pdf as

$$\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \sum_{\boldsymbol{a}_k} \tilde{f}(\boldsymbol{x}_k|\boldsymbol{a}_k, \boldsymbol{z}_{1:k})\, p(\boldsymbol{a}_k|\boldsymbol{z}_{1:k})\,, \tag{39}$$

where the summation is over all $\boldsymbol{a}_k \in \{0,\ldots,m_k\}^{n_\mathrm{t}}$. Using A11, $\tilde{f}(\boldsymbol{x}_k|\boldsymbol{a}_k, \boldsymbol{z}_{1:k})$ is obtained as [144]

$$\tilde{f}(\boldsymbol{x}_k|\boldsymbol{a}_k, \boldsymbol{z}_{1:k}) = \prod_{i=1}^{n_\mathrm{t}} \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|a_k^{(i)}, \boldsymbol{z}_{1:k}\big)\,, \tag{40}$$

where

$$\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|a_k^{(i)}, \boldsymbol{z}_{1:k}\big) = \frac{g\big(\boldsymbol{x}_k^{(i)}, a_k^{(i)}; \boldsymbol{z}_k\big)\, \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)}{\int g\big(\boldsymbol{x}_k^{(i)\prime}, a_k^{(i)}; \boldsymbol{z}_k\big)\, \tilde{f}\big(\boldsymbol{x}_k^{(i)\prime}\big|\boldsymbol{z}_{1:k-1}\big)\,\mathrm{d}\boldsymbol{x}_k^{(i)\prime}}\,, \tag{41}$$

with $g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$ defined in (14). We note that the original derivation of the JPDA filter [1] assumed that the probability of detection does not depend on the target state $\boldsymbol{x}_k^{(i)}$, i.e., $p_\mathrm{d}\big(\boldsymbol{x}_k^{(i)}\big) = p_\mathrm{d}$. In that case, (41) simplifies as follows: for $a_k^{(i)} = m \in \{1,\ldots,m_k\}$,

$$\tilde{f}\big(\boldsymbol{x}_k^{(i)}|m, \boldsymbol{z}_{1:k}\big) = \frac{f\big(\boldsymbol{z}_k^{(m)}\big|\boldsymbol{x}_k^{(i)}\big)\, \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)}{\int f\big(\boldsymbol{z}_k^{(m)}\big|\boldsymbol{x}_k^{(i)\prime}\big)\, \tilde{f}\big(\boldsymbol{x}_k^{(i)\prime}\big|\boldsymbol{z}_{1:k-1}\big)\,\mathrm{d}\boldsymbol{x}_k^{(i)\prime}}\,,$$

and for $a_k^{(i)} = 0$, $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|a_k^{(i)}=0, \boldsymbol{z}_{1:k}\big) = \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)$.

To avoid computations involving functions of the high-dimensional multitarget state $\boldsymbol{x}_k$, the JPDA filter updates each single-target state $\boldsymbol{x}_k^{(i)}$ individually. This is achieved by approximating the joint posterior DA pmf $p(\boldsymbol{a}_k|\boldsymbol{z}_{1:k})$ by the product of its marginals, i.e.,

$$p(\boldsymbol{a}_k|\boldsymbol{z}_{1:k}) \approx \prod_{i=1}^{n_\mathrm{t}} p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)\,, \tag{42}$$

with

$$p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big) = \sum_{\sim a_k^{(i)}} p(\boldsymbol{a}_k|\boldsymbol{z}_{1:k})\,. \tag{43}$$

Because the summation in (43) is with respect to all $a_k^{(i')}$ for $i' \in \{1,\ldots,n_\mathrm{t}\}\backslash\{i\}$, the computational complexity of this summation scales exponentially with n_t. We note that (42), (43) is the optimum approximation of $p(\boldsymbol{a}_k|\boldsymbol{z}_{1:k})$ by a fully factorizing pmf, where optimality is defined as minimum Kullback-Leibler divergence [32, Proposition 8.3]. Inserting (40) and (42) into (39), we obtain

$$\begin{aligned}\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) &\approx \sum_{\boldsymbol{a}_k} \prod_{i=1}^{n_\mathrm{t}} \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|a_k^{(i)}, \boldsymbol{z}_{1:k}\big)\, p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big) \\ &= \prod_{i=1}^{n_\mathrm{t}} \sum_{a_k^{(i)}=0}^{m_k} \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|a_k^{(i)}, \boldsymbol{z}_{1:k}\big)\, p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big) \qquad (44) \\ &= \prod_{i=1}^{n_\mathrm{t}} \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)\,. \qquad (45)\end{aligned}$$

Hence, the calculation of $\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ simplifies to n_t separate calculations of the approximate marginal posterior pdfs $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$. According to (44), these approximate marginal posterior pdfs are given by

$$\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big) = \sum_{a_k^{(i)}=0}^{m_k} \tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|a_k^{(i)}, \boldsymbol{z}_{1:k}\big)\, p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)\,, \tag{46}$$

with $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|a_k^{(i)}, \boldsymbol{z}_{1:k}\big)$ given by (41) and $p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ given by (43). The $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ can be used for Bayesian state estimation, e.g., by means of the MMSE estimator in (3).

It can be shown that approximating the joint posterior DA pmfs $p(\boldsymbol{a}_k|\boldsymbol{z}_{1:k})$ according to (42), as is done in the JPDA filter, results in approximate marginal posterior state pdfs $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ in (46) that are equal to the beliefs $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big)$ in (34). Thus, the total-SPA method presented in Section VII-A1 can be viewed as an SPA-based reformulation of the JPDA filter. Furthermore, we note that the factorization (45) is consistent with our initial factorization assumption in (36). In fact, together with (9), the above derivation provides an inductive proof that the factorization postulated in (42) entails (36), i.e., the factorization of $\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ into the approximate marginal posterior pdfs $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$, for all $k = 1, 2, \ldots$.

A closed-form implementation of the JPDA filter for linear-Gaussian models is presented in [1]. A sequential Monte Carlo (particle-based) implementation for nonlinear and non-Gaussian models can be found in [82].

3) Embedding SPADA: While the approximation (42) makes it possible to update the target states $\boldsymbol{x}_k^{(i)}$ individually, the calculation of the marginal DA pmfs $p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ in (43) still scales exponentially with n_t. The complexity can be reduced by gating [1], which is an additional approximation, but it remains exponential in the case of closely spaced targets. This problem can be solved by using SPADA for an approximative calculation of the $p\big(a_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$. As an input to SPADA, the approximate marginal posterior pdfs at the preceding time, $\tilde{f}\big(\boldsymbol{x}_{k-1}^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)$, are used, i.e., the $\tilde{f}\big(\boldsymbol{x}_{k-1}^{(i)}\big|\boldsymbol{z}_{1:k-1}\big)$ are substituted for the $\tilde{f}\big(\boldsymbol{x}_{k-1}^{(i)}\big)$ in (25). Remarkably, the resulting

"SPADA-embedded JPDA filter" is equivalent to the scalable total-SPA method developed in Section VII-A2. Indeed, it can be shown that the approximate marginal posterior pdfs produced by the former method are equal to the beliefs produced by the latter method.

C. SPA-Based Methods for Multisensor MTT

Next, we discuss the extension of the total-SPA reformulation of the JPDA filter from Section VII-A1 to the case of $n_\mathrm{s} \geq 2$ sensors. Let us consider the factorization (24) of the joint posterior pdf $f(\boldsymbol{x}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k} | \boldsymbol{z}_{1:k})$. The corresponding factor graph is shown in Fig. 2(d); it is a generalization of the single-sensor factor graph in Fig. 2(c) featuring an additional "outer loop" across the sensors. For this factor graph, the associated variational inference problem can be shown to be a nonconvex optimization problem [145], and thus the iterative SPA is not guaranteed to converge. Indeed, performing multiple message passing iterations over the outer loop does not necessarily lead to improved performance. In the following, we discuss two message passing schedules that avoid convergence issues and have been observed to result in good performance in many MTT scenarios. Alternative, iterative approaches based on convexification are studied in forthcoming work [145].

1) Sequential Processing: Following the iterated-corrector strategy, the target state beliefs are updated sequentially with respect to the sensors. This corresponds to a message passing schedule consisting of n_s successive "sensor update" steps, where in each step the result of the preceding step is used (except for the first step) and messages are passed to and from a part of the factor graph related to one sensor.

First, the messages $\alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$ are calculated according to (25). Then, iterated beliefs $\tilde{f}_s\big(\boldsymbol{x}_k^{(i)}\big)$ are calculated sequentially for each sensor $s = 1, \ldots, n_\mathrm{s}$. At the sth sensor update step, the messages passed from "$g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$" to "$a_{k,s}^{(i)}$" are calculated as (cf. (26))

$$\beta_{k,s}^{(i)}\big(a_{k,s}^{(i)}\big) = \int g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)\, \tilde{f}_{s-1}\big(\boldsymbol{x}_k^{(i)}\big)\, \mathrm{d}\boldsymbol{x}_k^{(i)}, \tag{47}$$

where the $\tilde{f}_{s-1}\big(\boldsymbol{x}_k^{(i)}\big)$ were calculated at the preceding sensor update step (cf. (49) below) or, for $s = 1$, $\tilde{f}_0\big(\boldsymbol{x}_k^{(i)}\big) = \alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$. Note that expression (47) involves the measurement $\boldsymbol{z}_{k,s}$ of sensor s. Using the $\beta_{k,s}^{(i)}\big(a_{k,s}^{(i)}\big)$ as input, the iterative SPA (27)–(29) or SPADA (30), (31) is executed, which yields the messages $\nu^{[n_\mathrm{it}]}_{\Psi_{i,m} \to a_{k,s}^{(i)}}\big(a_{k,s}^{(i)}\big)$. Then, the messages passed from "$a_{k,s}^{(i)}$" to "$g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$" are calculated according to (35), i.e., $\kappa_{k,s}^{(i)}\big(a_{k,s}^{(i)}\big) = \prod_{m=1}^{m_{k,s}} \nu^{[n_\mathrm{it}]}_{\Psi_{i,m} \to a_{k,s}^{(i)}}\big(a_{k,s}^{(i)}\big)$, and the messages passed from "$g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$" to "$\boldsymbol{x}_k^{(i)}$" are calculated as (cf. (32))

$$\gamma_{k,s}^{(i)}\big(\boldsymbol{x}_k^{(i)}\big) = \sum_{a_{k,s}^{(i)}=0}^{m_{k,s}} g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)\, \kappa_{k,s}^{(i)}\big(a_{k,s}^{(i)}\big). \tag{48}$$

Finally, iterated beliefs are obtained according to (7), i.e.,

$$\tilde{f}_s\big(\boldsymbol{x}_k^{(i)}\big) \propto \alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big) \prod_{s'=1}^{s} \gamma_{k,s'}^{(i)}\big(\boldsymbol{x}_k^{(i)}\big), \tag{49}$$

which generalizes (34) and can be calculated recursively. Note that $\tilde{f}_s\big(\boldsymbol{x}_k^{(i)}\big)$ incorporates the sensor measurements $\boldsymbol{z}_{k,s'}$ for $s' = 1, \ldots, s$. According to Fig. 2(d) and (6), $\tilde{f}_s\big(\boldsymbol{x}_k^{(i)}\big)$ is also the message passed from "$\boldsymbol{x}_k^{(i)}$" to "$g\big(\boldsymbol{x}_k^{(i)}, a_{k,s+1}^{(i)}; \boldsymbol{z}_{k,s+1}\big)$" at sensor update step $s+1$. The final beliefs are given as $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big) \triangleq \tilde{f}_{n_\mathrm{s}}\big(\boldsymbol{x}_k^{(i)}\big)$; they take into account the measurements of all sensors. These beliefs are used for state estimation (e.g., using the MMSE estimator in (3) with $f(\boldsymbol{x}_k | \boldsymbol{z}_{1:k})$ replaced by $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big)$) and to calculate the messages $\alpha_{k+1}^{(i)}\big(\boldsymbol{x}_{k+1}^{(i)}\big)$ at the next time step $k+1$ according to (25).

This sequential message passing algorithm is simple and its computational complexity is linear in the number of sensors n_s. On the other hand, it is not well suited to a parallel or distributed implementation, and the final beliefs $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big)$ depend on the chosen sensor order. We note that a sensor-sequential processing is also used by the sequential multisensor JPDA filter [81] and by iterated-corrector multisensor extensions of set-type filters (see Section XIII-B).

2) Parallel Processing: An alternative SPA-based method for multisensor MTT is provided by the following "parallel" message passing schedule. First, the messages $\alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$ are calculated according to (25). Then, the following operations are carried out separately for each sensor $s \in \{1, \ldots, n_\mathrm{s}\}$. The messages $\beta_{k,s}^{(i)}\big(a_{k,s}^{(i)}\big)$ passed from "$g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$" to "$a_{k,s}^{(i)}$" are calculated according to (26), i.e.,

$$\beta_{k,s}^{(i)}\big(a_{k,s}^{(i)}\big) = \int g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)\, \alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)\, \mathrm{d}\boldsymbol{x}_k^{(i)}. \tag{50}$$

Note that in contrast to (47), the expression (50) does not involve a result related to a different sensor. The iterative SPA (27)–(29) or SPADA (30), (31) is now executed, and the messages passed from "$a_{k,s}^{(i)}$" to "$g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$" are calculated according to (35), i.e., $\kappa_{k,s}^{(i)}\big(a_{k,s}^{(i)}\big) = \prod_{m=1}^{m_{k,s}} \nu^{[n_\mathrm{it}]}_{\Psi_{i,m} \to a_{k,s}^{(i)}}\big(a_{k,s}^{(i)}\big)$. Next, the messages $\gamma_{k,s}^{(i)}\big(\boldsymbol{x}_k^{(i)}\big)$ passed from "$g\big(\boldsymbol{x}_k^{(i)}, a_{k,s}^{(i)}; \boldsymbol{z}_{k,s}\big)$" to "$\boldsymbol{x}_k^{(i)}$" are calculated according to (48). After these operations have been done for all $s \in \{1, \ldots, n_\mathrm{s}\}$ separately, beliefs are calculated as

$$\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big) \propto \alpha_k^{(i)}\big(\boldsymbol{x}_k^{(i)}\big) \prod_{s=1}^{n_\mathrm{s}} \gamma_{k,s}^{(i)}\big(\boldsymbol{x}_k^{(i)}\big).$$

This message passing schedule allows for a parallel implementation and facilitates a distributed implementation, and the results do not depend on a chosen sensor ordering as in sequential processing. However, in some scenarios, the independent processing of the individual sensor measurements may result in a reduced estimation accuracy. We note that a sensor-parallel processing is also used by the parallel multisensor JPDA filter [81] and by the multisensor Monte Carlo JPDA filter [82].

VIII. Vector-Type System Model and Factor Graph for an Unknown, Time-Varying Number

OF TARGETS

Next, we consider the practically more relevant case of an unknown, time-varying number i_k of targets. Extending [53], [80], [148], we first present a vector-type system model and an associated factor graph for this case. Corresponding total-SPA vector-type methods for MTT will be developed in Section IX. The presented vector-type approach models only detected targets, i.e., targets that have so far generated at least one measurement at any of the sensors. By contrast, the set-type approach presented in Section XI models also undetected targets, i.e., targets that potentially exist but did not generate any measurement yet.

Our model is based on an arbitrary ordering $s = 1, 2, \ldots, n_\mathrm{s}$ of the sensors. At time k and sensor s, we distinguish between the following two types of detected targets:

- *newly detected targets*, which exist at time k and have been detected for the first time at time k and sensor s;
- *survived targets*, which exist at time k and have been detected previously, either at a previous time $k' < k$ or at the current time k but at a previous sensor $s' < s$.

The numbers of newly detected targets and survived targets are unknown. To account for this fact, we introduce the notion of *potential targets* (PTs). The number of PTs at time k, denoted j_k, is the *maximum possible* number of targets that have generated a measurement at any of the sensors up to time k. This number depends on the number of measurements observed at time k, as explained in Section VIII-B. The existence/nonexistence of PT $j \in \{1, \ldots, j_k\}$ is modeled by a binary variable $\mathsf{r}_k^{(j)} \in \{0, 1\}$, i.e., PT j exists at time k if and only if $\mathsf{r}_k^{(j)} = 1$. The state of PT j is denoted $\mathsf{x}_k^{(j)}$, and is formally considered also if $\mathsf{r}_k^{(j)} = 0$. *The augmented PT state* is defined as $\mathsf{y}_k^{(j)} \triangleq \big[\mathsf{x}_k^{(j)\mathrm{T}} \; \mathsf{r}_k^{(j)}\big]^\mathrm{T}$, and the joint state vector as $\mathbf{y}_k \triangleq \big[\mathsf{y}_k^{(1)\mathrm{T}} \cdots \mathsf{y}_k^{(j_k)\mathrm{T}}\big]^\mathrm{T}$. The states $\mathsf{x}_k^{(j)}$ of nonexisting PTs are obviously irrelevant. Therefore, all pdfs defined for PT states, $f\big(\boldsymbol{y}_k^{(j)}\big) = f\big(\boldsymbol{x}_k^{(j)}, r_k^{(j)}\big)$, are such that

$$f\big(\boldsymbol{x}_k^{(j)}, r_k^{(j)} = 0\big) = f_k^{(j)} f_\mathrm{D}\big(\boldsymbol{x}_k^{(j)}\big), \tag{51}$$

where $f_k^{(j)} \in [0, 1]$ is a constant and $f_\mathrm{D}\big(\boldsymbol{x}_k^{(j)}\big)$ is an arbitrary "dummy pdf."

A. Assumptions

We will use the following assumptions, which replace Vk1–Vk3:

- Vu1: The number of targets, i_k, is time-varying and unknown.
- Vu2: The PT states are ordered (arbitrarily) according to their arrangement in the joint augmented state vector $\mathbf{y}_k = \big[\mathsf{y}_k^{(1)\mathrm{T}} \cdots \mathsf{y}_k^{(j_k)\mathrm{T}}\big]^\mathrm{T}$.
- Vu3: A PT j that exists at time $k-1$ survives (i.e., still exists at time k) with survival probability $p_\mathrm{s}\big(\boldsymbol{x}_{k-1}^{(j)}\big)$ and disappears with probability $1 - p_\mathrm{s}\big(\boldsymbol{x}_{k-1}^{(j)}\big)$.
- Vu4: The number of newly detected targets at time k and sensor s is a priori (i.e., before the measurements are observed) Poisson distributed with mean $\mu_\mathrm{n}^{(s)}$. It is furthermore independent of the number of clutter measurements and of the number of survived targets.
- Vu5: The states of newly detected targets at time k and sensor s are a priori iid and distributed according to[4] $f_\mathrm{n}(\boldsymbol{x}_k)$.
- Vu6: At time k, the states of newly detected targets are independent of the states of survived targets.
- Vu7: At time $k = 0$, there are no PTs, i.e., $\mathbf{y}_0$ is an empty vector and $j_0 = 0$.

Based on these assumptions and the common assumptions in Section I-C, we will next establish a system model and a statistical formulation and, subsequently, derive a "stretched" version of the joint posterior pdf $f(\boldsymbol{y}_{1:k} | \boldsymbol{z}_{1:k})$.

B. Legacy PTs and New PTs

Each PT at time k and sensor s is either a "legacy" PT, i.e., a PT that was already established in the past, or a "new" PT. The corresponding states will be denoted by $\underline{\mathbf{y}}_{k,s}^{(j)}$ and $\overline{\mathbf{y}}_{k,s}^{(m)}$, respectively. The new PTs and legacy PTs are related to, respectively, the newly detected targets and survived targets mentioned earlier, as described in what follows.

- *New PTs*: To incorporate in the state space targets that are newly detected by sensor s at time k, $m_{k,s}$ new PT states $\overline{\mathbf{y}}_{k,s}^{(m)}$, $m = 1, \ldots, m_{k,s}$—one for each measurement $\boldsymbol{z}_{k,s}^{(m)}$—are introduced. Accordingly, $\overline{r}_{k,s}^{(m)} = 1$ means that measurement m produced by sensor s at time k was generated by a newly detected target. We denote by $\overline{\mathbf{y}}_{k,s} \triangleq \big[\overline{\mathbf{y}}_{k,s}^{(1)\mathrm{T}} \cdots \overline{\mathbf{y}}_{k,s}^{(m_{k,s})\mathrm{T}}\big]^\mathrm{T}$ the joint vector of all new PT states. Before the current measurements $\boldsymbol{z}_{k,s}^{(m)}$ are observed, the number $\mathsf{m}_{k,s}$ of new PT states $\overline{\mathbf{y}}_{k,s}^{(m)}$ is random.
- *Legacy PTs*: The legacy PT states $\underline{\mathbf{y}}_{k,s}^{(j)}$ represent survived targets, i.e., targets that have been detected previously, either at a previous time $k' < k$ or at the current time k but at a previous sensor $s' < s$. The target represented by the new PT state $\overline{\mathbf{y}}_{k',s'}^{(m')}$ introduced due to measurement m' of sensor s' at time $k' \leqslant k$ is represented by the legacy PT state $\underline{\mathbf{y}}_{k,s}^{(j)}$ at time k, with $j = j_{k'-1} + \sum_{s''=1}^{s'-1} m_{k,s''} + m'$. (Note that here, either $k' < k$ and s, s' arbitrary or $k' = k$ and $s' < s$.) Accordingly, $\underline{r}_{k,s}^{(j)} = 1$ means that the target that was detected the first time via measurement m' of sensor s' at time k' still exists when the measurements of sensor s at time k are incorporated. We denote by $\underline{\mathbf{y}}_{k,s} \triangleq \big[\underline{\mathbf{y}}_{k,s}^{(1)\mathrm{T}} \cdots \underline{\mathbf{y}}_{k,s}^{(j_{k,s})\mathrm{T}}\big]^\mathrm{T}$ the joint vector of all legacy PT states a time k and sensor s. (The relation between $j_{k,s}$ and j_k will be explained shortly.)

New PTs become legacy PTs when the next measurements—either of the next sensor or at the next time step—are incorporated. In particular, at time k, when the measurements of the next sensor, s, are incorporated, the number of legacy PTs is updated as

$$j_{k,s} = j_{k,s-1} + m_{k,s-1}\,,$$

with $j_{k,1} = j_{k-1}$. Here, $j_{k,s}$ is equal to the number of all measurements collected at time k up to sensor s. Note that

[4] Here and hereafter, with an abuse of notation, $\boldsymbol{x}_k$ denotes a generic single-target state vector (whereas in Section VII, it denoted the multitarget state vector).

the vector of all legacy PT states at time k up to sensor s can be written as $\underline{\mathbf{y}}_{k,s} = \big[\underline{\mathbf{y}}_{k,s-1}^{\mathrm{T}}\ \overline{\mathbf{y}}_{k,s-1}^{\mathrm{T}}\big]^{\mathrm{T}}$. The vector of all legacy PT states at time k, before any sensor measurements at time k are incorporated, is denoted by $\underline{\mathbf{y}}_k$. (Note that $\underline{\mathbf{y}}_{k,1} = \underline{\mathbf{y}}_k$ and thus for $j \in \{1,\ldots,j_{k-1}\}$, $\underline{\mathbf{y}}_{k,s}^{(j)} = \underline{\mathbf{y}}_k^{(j)}$ for all $s \in \{1,\ldots,n_{\mathrm{s}}\}$.) We also introduce the joint state of all new PTs introduced at time k as $\overline{\mathbf{y}}_k \triangleq \big[\overline{\mathbf{y}}_{k,1}^{\mathrm{T}} \cdots \overline{\mathbf{y}}_{k,n_{\mathrm{s}}}^{\mathrm{T}}\big]^{\mathrm{T}}$. After the measurements of all sensors $s \in \{1,\ldots,n_{\mathrm{s}}\}$ have been incorporated at time k, the total number of PT states is

$$j_k = j_{k,n_{\mathrm{s}}} + m_{k,n_{\mathrm{s}}} = j_{k-1} + \sum_{s=1}^{n_{\mathrm{s}}} m_{k,s}\,, \tag{52}$$

and the vector of all PT states at time k is given by $\mathbf{y}_k = \big[\underline{\mathbf{y}}_k^{\mathrm{T}}\ \overline{\mathbf{y}}_k^{\mathrm{T}}\big]^{\mathrm{T}} = \big[\underline{\mathbf{y}}_{k,n_{\mathrm{s}}}^{\mathrm{T}}\ \overline{\mathbf{y}}_{k,n_{\mathrm{s}}}^{\mathrm{T}}\big]^{\mathrm{T}}$. This comprises j_{k-1} PTs that have been introduced at previous time steps and $\sum_{s=1}^{n_{\mathrm{s}}} m_{k,s}$ PTs introduced at time k. Since new PTs are introduced as new measurements are incorporated, the number of PT states would grow indefinitely. Thus, for the development of a feasible MTT method, a suboptimum pruning step removing PTs is employed; this will be further discussed in Section IX-A6.

With a vector-type model in which the number of targets is unknown, the derivation of a likelihood function of the form $f(\boldsymbol{z}_{k,s}, m_{k,s}|\boldsymbol{y}_k)$ is complicated by the fact that the number of PT states depends on the number of measurements[5] $m_{k,s}$. Thus, contrary to the case of a known number of targets, described in Sections IV and V, we will use a derivation of the joint posterior pdf $f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k}|\boldsymbol{z}_{1:k})$ that does not involve a likelihood function. The joint posterior pdf and the corresponding factor graph will be the basis for the development of scalable MTT methods in Section IX. We first establish some pdfs and pmfs to be used in the derivation of the joint posterior pdf.

C. State-Transition pdf

For each PT state $\boldsymbol{y}_{k-1}^{(j)}$, $j \in \{1,\ldots,j_{k-1}\}$ at time $k-1$, there is one legacy PT state $\underline{\boldsymbol{y}}_k^{(j)}$ at time k. According to A1 and A2, the state-transition pdf for legacy PT state $\underline{\mathbf{y}}_k \triangleq \big[\underline{\mathbf{y}}_k^{(1)\mathrm{T}} \cdots \underline{\mathbf{y}}_k^{(j_{k-1})\mathrm{T}}\big]^{\mathrm{T}}$ factorizes as

$$f(\underline{\boldsymbol{y}}_k|\boldsymbol{y}_{k-1}) = \prod_{j=1}^{j_{k-1}} f\big(\underline{\boldsymbol{y}}_k^{(j)}\big|\boldsymbol{y}_{k-1}^{(j)}\big)\,, \tag{53}$$

where the single-target augmented state-transition pdf $f\big(\underline{\boldsymbol{y}}_k^{(j)}\big|\boldsymbol{y}_{k-1}^{(j)}\big) = f\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big|\boldsymbol{x}_{k-1}^{(j)}, r_{k-1}^{(j)}\big)$ is given as follows. If PT j does not exist at time $k-1$, i.e., $r_{k-1}^{(j)} = 0$, then it does not exist at time k either, i.e., $\underline{r}_k^{(j)} = 0$, and thus its state pdf is $f_{\mathrm{D}}\big(\underline{\boldsymbol{x}}_k^{(j)}\big)$. This means that

$$f\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big|\boldsymbol{x}_{k-1}^{(j)}, r_{k-1}^{(j)}=0\big) = \begin{cases} f_{\mathrm{D}}\big(\underline{\boldsymbol{x}}_k^{(j)}\big), & \underline{r}_k^{(j)}=0 \\ 0, & \underline{r}_k^{(j)}=1. \end{cases} \tag{54}$$

On the other hand, if PT j exists at time $k-1$, i.e., $r_{k-1}^{(j)} = 1$, then, using Vu3, the probability that it still exists at time k, i.e., $\underline{r}_k^{(j)} = 1$, is given by the survival probability $p_{\mathrm{s}}\big(\boldsymbol{x}_{k-1}^{(j)}\big)$, and if it still exists at time k, its state $\underline{\boldsymbol{x}}_k^{(j)}$ is distributed according to the state-transition pdf $f\big(\underline{\boldsymbol{x}}_k^{(j)}\big|\boldsymbol{x}_{k-1}^{(j)}\big)$. Thus,

$$f\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big|\boldsymbol{x}_{k-1}^{(j)}, r_{k-1}^{(j)}=1\big) = \begin{cases} \big(1-p_{\mathrm{s}}\big(\boldsymbol{x}_{k-1}^{(j)}\big)\big)\, f_{\mathrm{D}}\big(\underline{\boldsymbol{x}}_k^{(j)}\big), & \underline{r}_k^{(j)}=0 \\ p_{\mathrm{s}}\big(\boldsymbol{x}_{k-1}^{(j)}\big)\, f\big(\underline{\boldsymbol{x}}_k^{(j)}\big|\boldsymbol{x}_{k-1}^{(j)}\big), & \underline{r}_k^{(j)}=1. \end{cases} \tag{55}$$

We note that the difference of the state-transition model (53)–(55) from the state-transition model for a known number of targets in (8) is due to the fact that targets may disappear.

According to Vu7, $\mathbf{y}_0$ is empty. Thus, no state transition (53) can be performed at time $k=1$ and, consequently, $\underline{\mathbf{y}}_1$ is empty as well. For future use, we formally introduce $f(\underline{\boldsymbol{y}}_1^{(j)}|\boldsymbol{y}_0^{(j)}) \triangleq 1$ and $f(\underline{\boldsymbol{y}}_1|\boldsymbol{y}_0) \triangleq 1$.

D. Conditional pdf of DA Vector, New PT State, and Number of Measurements

Using A3–A6 and Vu4, the prior pmf of the DA vector $\mathbf{a}_{k,s} \triangleq \big[\mathsf{a}_{k,s}^{(1)} \cdots \mathsf{a}_{k,s}^{(j_{k,s})}\big]^{\mathrm{T}}$, the vector of binary existence variables for new PTs $\overline{\mathbf{r}}_{k,s} \triangleq \big[\overline{\mathsf{r}}_{k,s}^{(1)} \cdots \overline{\mathsf{r}}_{k,s}^{(m_{k,s})}\big]^{\mathrm{T}}$, and the number of measurements $\mathsf{m}_{k,s}$ conditioned on the vector of legacy PT states $\underline{\mathbf{y}}_{k,s}$ is obtained as (see [144] for a derivation)

$$\begin{aligned} &p\big(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}\big|\underline{\mathbf{y}}_{k,s}\big) \\ &= \frac{1}{m_{k,s}!}\, e^{-\mu_{\mathrm{n}}^{(s)}} \big(\mu_{\mathrm{n}}^{(s)}\big)^{|\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}|}\, e^{-\mu_{\mathrm{c}}^{(s)}} \big(\mu_{\mathrm{c}}^{(s)}\big)^{m_{k,s}-|\mathcal{D}_{\boldsymbol{a}_{k,s}}|-|\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}|} \\ &\quad \times \psi(\boldsymbol{a}_{k,s}) \left(\prod_{m\in\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}} \Gamma_{\boldsymbol{a}_{k,s}}^{(m)}\right) \left(\prod_{j\in\mathcal{D}_{\boldsymbol{a}_{k,s}}} \underline{r}_{k,s}^{(j)}\, p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}\big)\right) \\ &\quad \times \prod_{j'\notin\mathcal{D}_{\boldsymbol{a}_{k,s}}} \big(1 - \underline{r}_{k,s}^{(j)} p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j')}\big)\big)\,. \end{aligned} \tag{56}$$

Here, $\psi(\boldsymbol{a}_{k,s})$ and $\mathcal{D}_{\boldsymbol{a}_{k,s}}$ are defined as in Section IV-B but with i, i', and n_{t} replaced by j, j', and $j_{k,s}$, respectively; $j \notin \mathcal{D}_{\boldsymbol{a}_{k,s}}$ is short for $j \in \{1,\ldots,j_{k,s}\}\setminus\mathcal{D}_{\boldsymbol{a}_{k,s}}$; $\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}$ is the set of new PTs that exist at time k, i.e., $\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}} \triangleq \big\{m \in \{1,\ldots,m_{k,s}\} : \overline{r}_{k,s}^{(m)} = 1\big\}$; and $\Gamma_{\boldsymbol{a}_{k,s}}^{(m)}$ is defined as

$$\Gamma_{\boldsymbol{a}_{k,s}}^{(m)} \triangleq \begin{cases} 0, & \exists j \in \{1,\ldots,j_{k,s}\} \text{ such that } a_{k,s}^{(j)} = m \\ 1, & \text{otherwise.} \end{cases}$$

Together, the factors $\psi(\boldsymbol{a}_{k,s})$ and $\prod_{m\in\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}} \Gamma_{\boldsymbol{a}_{k,s}}^{(m)}$ enforce A4, i.e., $p(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}|\underline{\mathbf{y}}_{k,s}) \neq 0$ only if each measurement is associated either with a legacy PT or with a new PT or with clutter, and no measurement is associated with more than one PT. Note that $p\big(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}\big|\underline{\mathbf{y}}_{k,s}\big)$ in (56) is a valid pmf in the sense that $\sum_{m_{k,s}=0}^{\infty} \sum_{\overline{\boldsymbol{r}}_{k,s}\in\{0,1\}^{m_{k,s}}} \sum_{\boldsymbol{a}_{k,s}} p\big(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}\big|\underline{\mathbf{y}}_{k,s}\big) = 1$ for arbitrary $\underline{\mathbf{y}}_{k,s}$. The vector of legacy PTs $\underline{\boldsymbol{y}}_{k,s}$ may be empty, i.e., $j_{k,s} = 0$. In that case, $\mathcal{D}_{\boldsymbol{a}_{k,s}}$ and $\{1,\ldots,j_{k,s}\}\setminus\mathcal{D}_{\boldsymbol{a}_{k,s}}$ are empty, and an expression of $p(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}|\underline{\mathbf{y}}_{k,s}) = p(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s})$ can be obtained by setting the two products in (56) involving $\mathcal{D}_{\boldsymbol{a}_{k,s}}$ to 1.

[5] This issue can be addressed by using a set-type model as discussed in Section XI.

To better understand the structure of $p(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}|\underline{\boldsymbol{y}}_{k,s})$, we next consider four special cases. In the *no-new-detections, no-legacy-detections, no-clutter case*, i.e., $m_{k,s}=0$, $\overline{\boldsymbol{r}}_{k,s}=\mathbf{0}$, and $\boldsymbol{a}_{k,s}=\mathbf{0}$, expression (56) reduces to

$$p\big(\boldsymbol{a}_{k,s}=\mathbf{0}, \overline{\boldsymbol{r}}_{k,s}=\mathbf{0}, m_{k,s}=0\big|\underline{\boldsymbol{y}}_{k,s}\big) = e^{-\mu_{\mathrm{n}}^{(s)}} e^{-\mu_{\mathrm{c}}^{(s)}} \prod_{j=1}^{j_{k,s}} \big(1-\underline{r}_{k,s}^{(j)} p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}\big)\big).$$

Here, $e^{-\mu_{\mathrm{n}}^{(s)}}$ is the Poisson pmf of the number of newly detected targets evaluated at 0, $e^{-\mu_{\mathrm{c}}^{(s)}}$ is the Poisson pmf of the number of clutter measurements evaluated at 0, and $\prod_{j=1}^{j_{k,s}}\big(1-\underline{r}_{k,s}^{(j)} p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(i)}\big)\big)$ is the probability that no survived target is detected by sensor s. In the *no-new-detections, no-legacy-detections, all-clutter case*, i.e., $m_{k,s}>0$, $\overline{\boldsymbol{r}}_{k,s}=\mathbf{0}$, and $\boldsymbol{a}_{k,s}=\mathbf{0}$,

$$p\big(\boldsymbol{a}_{k,s}=\mathbf{0}, \overline{\boldsymbol{r}}_{k,s}=\mathbf{0}, m_{k,s}\big|\underline{\boldsymbol{y}}_{k,s}\big) = e^{-\mu_{\mathrm{n}}^{(s)}} \frac{e^{-\mu_{\mathrm{c}}^{(s)}} (\mu_{\mathrm{c}}^{(s)})^{m_{k,s}}}{m_{k,s}!} \prod_{j=1}^{j_{k,s}} \big(1-\underline{r}_{k,s}^{(j)} p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}\big)\big),$$

where $e^{-\mu_{\mathrm{c}}^{(s)}} (\mu_{\mathrm{c}}^{(s)})^{m_{k,s}}/m_{k,s}!$ is the Poisson pmf of the number of clutter measurements evaluated at $m_{k,s}$. In the *all-new-detections, no-legacy-detections, no-clutter case*, i.e., $\overline{\boldsymbol{r}}_{k,s}=\mathbf{1}$ and $\boldsymbol{a}_{k,s}=\mathbf{0}$,

$$p\big(\boldsymbol{a}_{k,s}=\mathbf{0}, \overline{\boldsymbol{r}}_{k,s}=\mathbf{1}, m_{k,s}\big|\underline{\boldsymbol{y}}_{k,s}\big) = \frac{e^{-\mu_{\mathrm{n}}^{(s)}} (\mu_{\mathrm{n}}^{(s)})^{m_{k,s}}}{m_{k,s}!} e^{-\mu_{\mathrm{c}}^{(s)}} \prod_{j=1}^{j_{k,s}} \big(1-\underline{r}_{k,s}^{(j)} p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}\big)\big),$$

where $e^{-\mu_{\mathrm{n}}^{(s)}} (\mu_{\mathrm{n}}^{(s)})^{m_{k,s}}/m_{k,s}!$ is the Poisson pmf of the number of newly detected targets evaluated at $m_{k,s}$. In the *no-new-detections, all-legacy-detections, no-clutter* case, i.e., $m_{k,s}=j_{k,s}^{\mathrm{d}}$, $\overline{\boldsymbol{r}}_{k,s}=\mathbf{0}$, and $\boldsymbol{a}_{k,s}=\boldsymbol{a}_{k,s}^{\mathrm{d}}$, where $j_{k,s}^{\mathrm{d}} \triangleq \sum_{j=1}^{j_{k,s}} \underline{r}_{k,s}^{(j)}$ is the number of existing legacy PTs and $\boldsymbol{a}_{k,s}^{\mathrm{d}}$ is any vector that assigns exactly one measurement to each existing legacy PT,

$$p\big(\boldsymbol{a}_{k,s}=\boldsymbol{a}_{k,s}^{\mathrm{d}}, \overline{\boldsymbol{r}}_{k,s}=\mathbf{0}, m_{k,s}=j_{k,s}^{\mathrm{d}}\big|\underline{\boldsymbol{y}}_{k,s}\big) = \frac{1}{j_{k,s}^{\mathrm{d}}!} e^{-\mu_{\mathrm{n}}^{(s)}} e^{-\mu_{\mathrm{c}}^{(s)}} \prod_{j\in\mathcal{D}_{\boldsymbol{a}_{k,s}^{\mathrm{d}}}} \underline{r}_{k,s}^{(j)} p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}\big).$$

Here, $e^{-\mu_{\mathrm{n}}^{(s)}}$ and $e^{-\mu_{\mathrm{c}}^{(s)}}$ are again the Poisson pmfs of the number of newly detected targets and the number of clutter measurements evaluated at 0, respectively; $\prod_{j\in\mathcal{D}_{\boldsymbol{a}_{k,s}^{\mathrm{d}}}} \underline{r}_{k,s}^{(j)} p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}\big)$ is the probability that all existing legacy PTs are detected by sensor s, and the factor $1/j_{k,s}^{\mathrm{d}}!$ arises because there are $j_{k,s}^{\mathrm{d}}!$ different measurement-target associations and, thus, $j_{k,s}^{\mathrm{d}}!$ different $\boldsymbol{a}_{k,s}^{\mathrm{d}}$.

We can express (56) more compactly as

$$p(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}|\underline{\boldsymbol{y}}_{k,s}) = C(m_{k,s})\, \psi(\boldsymbol{a}_{k,s}) \left(\prod_{j=1}^{j_{k,s}} q_1\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; m_{k,s}\big) \right) \times \prod_{m\in\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}} \frac{\mu_{\mathrm{n}}^{(s)}}{\mu_{\mathrm{c}}^{(s)}} \Gamma_{\boldsymbol{a}_{k,s}}^{(m)}, \quad (57)$$

where $C(m_{k,s})$ is a normalization factor that depends only on $m_{k,s}$ and $q_1\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; m_{k,s}\big)$ is defined as

$$q_1\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1, a_{k,s}^{(j)}; m_{k,s}\big) \triangleq \begin{cases} \dfrac{p_{\mathrm{d}}^{(s)}(\underline{\boldsymbol{x}}_{k,s}^{(j)})}{\mu_{\mathrm{c}}^{(s)}}, & a_{k,s}^{(j)} \in \{1,\ldots,m_{k,s}\} \\ 1-p_{\mathrm{d}}^{(s)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}\big), & a_{k,s}^{(j)}=0 \end{cases}$$

$$q_1\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 0, a_{k,s}^{(j)}; m_{k,s}\big) \triangleq 1\big(a_{k,s}^{(j)}\big), \quad (58)$$

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

Next, using Vu5 and the fact that new PT states with $\overline{\boldsymbol{r}}_{k,s}=1$ represent newly detected targets, the prior pdf of the states $\overline{\mathbf{x}}_{k,s}$ of new PTs conditioned on $\overline{\mathbf{r}}_{k,s}$ and $m_{k,s}$ is obtained as

$$f(\overline{\boldsymbol{x}}_{k,s}|\overline{\boldsymbol{r}}_{k,s}, m_{k,s}) = \left(\prod_{m'\in\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}} f_{\mathrm{n}}\big(\overline{\boldsymbol{x}}_{k,s}^{(m')}\big) \right) \prod_{m\notin\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}} f_{\mathrm{D}}\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}\big), \quad (59)$$

where $m\notin\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}$ is short for $m\in\{1,\ldots,m_{k,s}\}\backslash\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}$.

Finally, using Vu6, we obtain for the conditional pdf of $\mathbf{a}_{k,s}$, $\overline{\mathbf{y}}_{k,s}$, and $\mathsf{m}_{k,s}$

$$\begin{aligned} &f(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{y}}_{k,s}, m_{k,s}|\underline{\boldsymbol{y}}_{k,s}) \\ &\quad = f(\overline{\boldsymbol{x}}_{k,s}|\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}, \underline{\boldsymbol{y}}_{k,s})\, p(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}|\underline{\boldsymbol{y}}_{k,s}) \\ &\quad = f(\overline{\boldsymbol{x}}_{k,s}|\overline{\boldsymbol{r}}_{k,s}, m_{k,s})\, p(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{r}}_{k,s}, m_{k,s}|\underline{\boldsymbol{y}}_{k,s}). \end{aligned}$$

Inserting (57) and (59), we obtain further

$$\begin{aligned} &f(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{y}}_{k,s}, m_{k,s}|\underline{\boldsymbol{y}}_{k,s}) \\ &\quad = C(m_{k,s})\, \psi(\boldsymbol{a}_{k,s}) \left(\prod_{j=1}^{j_{k,s}} q_1\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; m_{k,s}\big) \right) \\ &\qquad \times \prod_{m=1}^{m_{k,s}} v_1\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}, \overline{r}_{k,s}^{(m)}, \boldsymbol{a}_{k,s}\big), \end{aligned} \quad (60)$$

where $v_1\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}, \overline{r}_{k,s}^{(m)}, \boldsymbol{a}_{k,s}\big)$ is defined as

$$v_1\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}, 1, \boldsymbol{a}_{k,s}\big) \triangleq \begin{cases} 0, & \exists j\in\{1,\ldots,j_{k,s}\} \\ & \text{such that } a_{k,s}^{(j)}=m \\ \frac{\mu_{\mathrm{n}}^{(s)}}{\mu_{\mathrm{c}}^{(s)}} f_{\mathrm{n}}\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}\big), & \text{otherwise} \end{cases} \quad (61)$$

and

$$v_1\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}, 0, \boldsymbol{a}_{k,s}\big) \triangleq f_{\mathrm{D}}\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}\big). \quad (62)$$

E. Conditional pdf of the Measurements

Using A4, A7, A8, and A9, the dependence of the measurement vector $\mathbf{z}_{k,s}$ on $\underline{\mathbf{y}}_{k,s}$, $\overline{\mathbf{y}}_{k,s}$, $\mathbf{a}_{k,s}$, and $\mathsf{m}_{k,s}$ is described by the conditional pdf (cf. (12))

$$\begin{aligned} &f(\boldsymbol{z}_{k,s}|\underline{\boldsymbol{y}}_{k,s}, \overline{\boldsymbol{y}}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}) \\ &\quad = \left(\prod_{m=1}^{m_{k,s}} f_{\mathrm{c}}^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big) \right) \left(\prod_{j\in\mathcal{D}_{\boldsymbol{a}_{k,s}}} f^{(s)}\Big(\boldsymbol{z}_{k,s}^{(a_{k,s}^{(j)})}\Big|\underline{\boldsymbol{x}}_{k,s}^{(j)}\Big) \right) \\ &\qquad \times \prod_{m'\in\mathcal{N}_{\overline{\boldsymbol{r}}_{k,s}}} f^{(s)}\big(\boldsymbol{z}_{k,s}^{(m')}|\overline{\boldsymbol{x}}_{k,s}^{(m')}\big). \end{aligned} \quad (63)$$

(Note that this expression presupposes that $\overline{\boldsymbol{r}}_{k,s}$ and $\boldsymbol{a}_{k,s}$ are consistent with A4.) We can write (63) as

$$\begin{aligned} f(\boldsymbol{z}_{k,s}|\underline{\boldsymbol{y}}_{k,s}, \overline{\boldsymbol{y}}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}) \\ = C(\boldsymbol{z}_{k,s}) \left(\prod_{j=1}^{j_{k,s}} q_2\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s}\big) \right) \\ \times \prod_{m=1}^{m_{k,s}} v_2\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}, \overline{r}_{k,s}^{(m)}; \boldsymbol{z}_{k,s}^{(m)}\big), \end{aligned} \tag{64}$$

where $C(\boldsymbol{z}_{k,s})$ is a normalization factor that depends only on $\boldsymbol{z}_{k,s}$ (and, thus, also on $m_{k,s}$), $q_2\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s}\big)$ is defined as

$$q_2\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s}\big) \triangleq \begin{cases} f^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}|\underline{\boldsymbol{x}}_{k,s}^{(j)}\big), & a_{k,s}^{(j)} = m \in \{1, \dots, m_{k,s}\} \\ 1, & a_{k,s}^{(j)} = 0 \end{cases}$$

$$q_2\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 0, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s}\big) \triangleq 1, \tag{65}$$

and $v_2\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}, \overline{r}_{k,s}^{(m)}; \boldsymbol{z}_{k,s}^{(m)}\big)$ is defined as

$$v_2\big(\overline{\boldsymbol{x}}_{k.s}^{(m)}, \overline{r}_{k,s}^{(m)}; \boldsymbol{z}_{k,s}^{(m)}\big) \triangleq \begin{cases} f^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}|\overline{\boldsymbol{x}}_{k,s}^{(m)}\big), & \overline{r}_{k,s}^{(m)} = 1 \\ 1, & \overline{r}_{k,s}^{(m)} = 0. \end{cases} \tag{66}$$

The vector of legacy PTs $\underline{\boldsymbol{y}}_{k,s}$ may be empty, i.e., $j_{k,s} = 0$. In that case, an expression of $f(\boldsymbol{z}_{k,s}|\underline{\boldsymbol{y}}_{k,s}, \overline{\boldsymbol{y}}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}) = f(\boldsymbol{z}_{k,s}|\overline{\boldsymbol{y}}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s})$ can be obtained by replacing in (64) the product involving $j_{k,s}$ by 1.

F. Conditional pdf of Measurements, Number of Measurements, DA Vector, and Augmented Target States

The pdf of $\mathsf{z}_{k,s}$, $\mathsf{a}_{k,s}$, $\mathsf{m}_{k,s}$, and $\overline{\mathsf{y}}_{k,s}$ conditioned on $\underline{\mathsf{y}}_{k,s}$ is obtained as

$$\begin{aligned} &f(\boldsymbol{z}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}, \overline{\boldsymbol{y}}_{k,s}|\underline{\boldsymbol{y}}_{k,s}) \\ &= f(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{y}}_{k,s}, m_{k,s}|\underline{\boldsymbol{y}}_{k,s}) f(\boldsymbol{z}_{k,s}|\underline{\boldsymbol{y}}_{k,s}, \overline{\boldsymbol{y}}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}), \end{aligned} \tag{67}$$

with $f(\boldsymbol{a}_{k,s}, \overline{\boldsymbol{y}}_{k,s}, m_{k,s}|\underline{\boldsymbol{y}}_{k,s})$ given by (60) and $f(\boldsymbol{z}_{k,s}|\underline{\boldsymbol{y}}_{k}, \overline{\boldsymbol{y}}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s})$ given by (64). Note that $f(\boldsymbol{z}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}, \overline{\boldsymbol{y}}_{k,s}|\underline{\boldsymbol{y}}_{k,s})$ in (67) is a valid hybrid pdf/pmf in the sense that $\sum_{m_{k,s}=0}^{\infty} \sum_{\boldsymbol{a}_{k,s}} \sum_{\overline{\boldsymbol{r}}_{k,s} \in \{0,1\}^{m_{k,s}}} \int \int f(\boldsymbol{z}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}, \overline{\boldsymbol{y}}_{k,s}|\underline{\boldsymbol{y}}_{k,s}) \mathrm{d}\boldsymbol{z}_{k,s} \mathrm{d}\overline{\boldsymbol{x}}_{k,s} = 1$ for arbitray $\underline{\boldsymbol{y}}_{k,s}$. Using A10 and the fact that the new PT states $\overline{\boldsymbol{y}}_{k,s'}$ related to sensors $s' = 1, \dots, s-1$ become legacy PT states at successive sensors $s, \dots, n_{\mathrm{s}}$, the conditional pdf of z_k, a_k, m_k, and $\overline{\mathsf{y}}_k$ given $\underline{\mathsf{y}}_k$ is obtained as (cf. (15))

$$f(\boldsymbol{z}_k, \boldsymbol{a}_k, \boldsymbol{m}_k, \overline{\boldsymbol{y}}_k|\underline{\boldsymbol{y}}_k) = \prod_{s=1}^{n_{\mathrm{s}}} f(\boldsymbol{z}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}, \overline{\boldsymbol{y}}_{k,s}|\underline{\boldsymbol{y}}_{k,s}), \tag{68}$$

where $f(\boldsymbol{z}_{k,s}, \boldsymbol{a}_{k,s}, m_{k,s}, \overline{\boldsymbol{y}}_{k,s}|\underline{\boldsymbol{y}}_{k,s})$ is given by (67).

Next, we develop the conditional pdf of z_k, a_k, m_k, and y_k given y_{k-1}. We obtain

$$\begin{aligned} &f(\boldsymbol{z}_k, \boldsymbol{a}_k, \boldsymbol{m}_k, \boldsymbol{y}_k|\boldsymbol{y}_{k-1}) \\ &\quad = f(\boldsymbol{z}_k, \boldsymbol{a}_k, \boldsymbol{m}_k, \underline{\boldsymbol{y}}_k, \overline{\boldsymbol{y}}_k|\boldsymbol{y}_{k-1}) \\ &\quad = f(\boldsymbol{z}_k, \boldsymbol{a}_k, \boldsymbol{m}_k, \overline{\boldsymbol{y}}_k|\underline{\boldsymbol{y}}_k, \boldsymbol{y}_{k-1}) f(\underline{\boldsymbol{y}}_k|\boldsymbol{y}_{k-1}) \\ &\quad = f(\boldsymbol{z}_k, \boldsymbol{a}_k, \boldsymbol{m}_k, \overline{\boldsymbol{y}}_k|\underline{\boldsymbol{y}}_k) f(\underline{\boldsymbol{y}}_k|\boldsymbol{y}_{k-1}), \end{aligned} \tag{69}$$

where A11 was used. (Note that A11 implies an analogous assumption in which the $\boldsymbol{x}_{k'}^{(i)}$, $k' \neq k$ in A11 are replaced by $\boldsymbol{y}_{k'}^{(j)}$, $k' \neq k$.) In case $\boldsymbol{y}_{k-1}$ is empty, $\underline{\boldsymbol{y}}_k$ is empty as well and (69) reduces to $f(\boldsymbol{z}_k, \boldsymbol{a}_k, \boldsymbol{m}_k, \boldsymbol{y}_k|\boldsymbol{y}_{k-1}) = f(\boldsymbol{z}_k, \boldsymbol{a}_k, \boldsymbol{m}_k, \overline{\boldsymbol{y}}_k)$. Using again A11 and the chain rule, the unconditional joint pdf for all times up to k is given by

$$f(\boldsymbol{z}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{m}_{1:k}, \boldsymbol{y}_{1:k}) = \prod_{k'=1}^{k} f(\boldsymbol{z}_{k'}, \boldsymbol{a}_{k'}, \boldsymbol{m}_{k'}, \boldsymbol{y}_{k'}|\boldsymbol{y}_{k'-1}), \tag{70}$$

where $f(\boldsymbol{z}_1, \boldsymbol{a}_1, \boldsymbol{m}_1, \boldsymbol{y}_1|\boldsymbol{y}_0) = f(\boldsymbol{z}_1, \boldsymbol{a}_1, \boldsymbol{m}_1, \boldsymbol{y}_1) = f(\boldsymbol{z}_1, \boldsymbol{a}_1, \boldsymbol{m}_1, \overline{\boldsymbol{y}}_1)$.

G. Joint Posterior pdf and Factor Graph

Next, similarly to Section V, we derive the joint posterior pdf $f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k}|\boldsymbol{z}_{1:k})$ and the corresponding factor graph. As in Section V-A, we start by performing the first stretching step, i.e., we replace $f(\boldsymbol{y}_{1:k}|\boldsymbol{z}_{1:k})$ by $f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k})$. With $\boldsymbol{z}_{1:k}$ observed, we obtain for the stretched joint posterior pdf

$$\begin{aligned} f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k}) &= f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k}, \boldsymbol{m}_{1:k}) \\ &\propto f(\boldsymbol{z}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{m}_{1:k}, \boldsymbol{y}_{1:k}) \\ &= \prod_{k'=1}^{k} f(\boldsymbol{z}_{k'}, \boldsymbol{a}_{k'}, \boldsymbol{m}_{k'}, \boldsymbol{y}_{k'}|\boldsymbol{y}_{k'-1}), \end{aligned}$$

where (70) was used. Next, using (69) and subsequently (68) and (67), we obtain further

$$\begin{aligned} &f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k}) \\ &\quad \propto \prod_{k'=1}^{k} f(\underline{\boldsymbol{y}}_{k'}|\boldsymbol{y}_{k'-1}) \prod_{s=1}^{n_{\mathrm{s}}} f(\boldsymbol{a}_{k',s}, \overline{\boldsymbol{y}}_{k',s}, m_{k',s}|\underline{\boldsymbol{y}}_{k',s}) \\ &\qquad \times f(\boldsymbol{z}_{k',s}|\underline{\boldsymbol{y}}_{k',s}, \overline{\boldsymbol{y}}_{k',s}, \boldsymbol{a}_{k',s}, m_{k',s}), \end{aligned}$$

where we recall that $f(\underline{\boldsymbol{y}}_1|\boldsymbol{y}_0) = 1$. Finally, inserting (53), (60), and (64) yields

$$\begin{aligned} &f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k}) \\ &\quad \propto \prod_{k'=1}^{k} \left(\prod_{j'=1}^{j_{k'-1}} f\big(\underline{\boldsymbol{y}}_{k'}^{(j')}|\boldsymbol{y}_{k'-1}^{(j')}\big) \right) \prod_{s=1}^{n_{\mathrm{s}}} \psi(\boldsymbol{a}_{k',s}) \\ &\qquad \times \left(\prod_{j=1}^{j_{k',s}} q\big(\underline{\boldsymbol{x}}_{k',s}^{(j)}, \underline{r}_{k',s}^{(j)}, a_{k',s}^{(j)}; \boldsymbol{z}_{k',s}\big) \right) \\ &\qquad \times \prod_{m=1}^{m_{k',s}} v_1\big(\overline{\boldsymbol{x}}_{k',s}^{(m)}, \overline{r}_{k',s}^{(m)}, \boldsymbol{a}_{k',s}\big)\, v_2\big(\overline{\boldsymbol{x}}_{k',s}^{(m)}, \overline{r}_{k',s}^{(m)}; \boldsymbol{z}_{k',s}^{(m)}\big), \end{aligned} \tag{71}$$

with $f(\underline{\boldsymbol{y}}_1^{(j)}|\boldsymbol{y}_0^{(j)}) = 1$ and

$$\begin{aligned} &q\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s}\big) \\ &\quad \triangleq q_1\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; m_{k,s}\big)\, q_2\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}, a_{k,s}^{(j)}; \boldsymbol{z}_{k,s}\big). \end{aligned} \tag{72}$$

Next, we perform the second stretching step, similarly to Section V-B, i.e., we replace $f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}|\boldsymbol{z}_{1:k})$ by $f(\boldsymbol{y}_{1:k}, \boldsymbol{a}_{1:k}, \boldsymbol{b}_{1:k}|\boldsymbol{z}_{1:k})$. This is done by replacing $\psi(\boldsymbol{a}_{k,s})$

in (71) by $\psi(\boldsymbol{a}_{k,s},\boldsymbol{b}_{k,s}) = \prod_{j=1}^{j_{k,s}}\prod_{m=1}^{m_{k,s}}\Psi_{j,m}\big(a_{k,s}^{(j)},b_{k,s}^{(m)}\big)$ (cf. (21)) with $\Psi_{j,m}\big(a_{k,s}^{(j)},b_{k,s}^{(m)}\big)$ given by (22). The factorization (71) can now be simplified by the following modification. Let $\boldsymbol{a}_{k,s}$ and $\boldsymbol{b}_{k,s}$ be a valid pair of DA vectors in the sense that $\psi(\boldsymbol{a}_{k,s},\boldsymbol{b}_{k,s}) = 1$. Then, the condition in (61), "$\exists j \in \{1,\ldots,j_{k,s}\}$ such that $a_{k,s}^{(j)} = m$", is equal to "$b_{k,s}^{(m)} \in \{1,\ldots,j_{k,s}\}$". By inspecting $v_1\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)},\boldsymbol{a}_{k,s}\big)$ defined in (61) and (62) as well as $v_2\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)};\boldsymbol{z}_{k,s}^{(m)}\big)$ defined in (66), one can conclude that the product $v_1\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)},\boldsymbol{a}_{k,s}\big)\,v_2\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)};\boldsymbol{z}_{k,s}^{(m)}\big)$ in (71) can be replaced by $v\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)},b_{k,s}^{(m)};\boldsymbol{z}_{k,s}^{(m)}\big)$ defined as

$$v\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},1,b_{k,s}^{(m)};\boldsymbol{z}_{k,s}^{(m)}\big) \triangleq \begin{cases} 0, & b_{k,s}^{(m)} \in \{1,\ldots,j_{k,s}\} \\ \frac{\mu_{\mathrm{n}}^{(s)}}{\mu_{\mathrm{c}}^{(s)}}\,f_{\mathrm{n}}\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}\big)f^{(s)}\big(\boldsymbol{z}_{k,s}^{(m)}\big|\overline{\boldsymbol{x}}_{k,s}^{(m)}\big), & b_{k,s}^{(m)}=0 \end{cases} \tag{73}$$

and

$$v\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},0,b_{k,s}^{(m)};\boldsymbol{z}_{k,s}^{(m)}\big) = f_{\mathrm{D}}\big(\overline{\boldsymbol{x}}_{k,s}^{(m)}\big). \tag{74}$$

With this simplification, the stretched marginal posterior pdf replacing (71) is obtained as

$$\begin{aligned} f(\boldsymbol{y}_{1:k},\boldsymbol{a}_{1:k},\boldsymbol{b}_{1:k}|\boldsymbol{z}_{1:k}) &\propto \prod_{k'=1}^{k}\left(\prod_{j'=1}^{j_{k'-1}} f\big(\underline{\boldsymbol{y}}_{k'}^{(j')}\big|\boldsymbol{y}_{k'-1}^{(j')}\big)\right) \\ &\quad\times \prod_{s=1}^{n_{\mathrm{s}}}\left(\prod_{j=1}^{j_{k',s}} q\big(\underline{\boldsymbol{x}}_{k',s}^{(j)},\underline{r}_{k',s}^{(j)},a_{k',s}^{(j)};\boldsymbol{z}_{k',s}\big)\right. \\ &\quad\left.\times \prod_{m'=1}^{m_{k',s}}\Psi_{j,m'}\big(a_{k',s}^{(j)},b_{k',s}^{(m')}\big)\right) \\ &\quad\times \prod_{m=1}^{m_{k',s}} v\big(\overline{\boldsymbol{x}}_{k',s}^{(m)},\overline{r}_{k',s}^{(m)},b_{k',s}^{(m)};\boldsymbol{z}_{k',s}^{(m)}\big). \end{aligned} \tag{75}$$

Here, we recall that $j_0 = 0$ (cf. Vu7). We note that $f(\boldsymbol{y}_{1:k},\boldsymbol{a}_{1:k},\boldsymbol{b}_{1:k}|\boldsymbol{z}_{1:k})$ is still consistent with the original joint posterior pdf $f(\boldsymbol{y}_{1:k}|\boldsymbol{z}_{1:k})$ in the sense that $\sum_{\boldsymbol{a}_{1:k}}\sum_{\boldsymbol{b}_{1:k}} f(\boldsymbol{y}_{1:k},\boldsymbol{a}_{1:k},\boldsymbol{b}_{1:k}|\boldsymbol{z}_{1:k}) = f(\boldsymbol{y}_{1:k}|\boldsymbol{z}_{1:k})$, which means that the marginal posterior pdfs $f\big(\boldsymbol{y}_k^{(j)}\big|\boldsymbol{z}_{1:k}\big)$ calculated from $f(\boldsymbol{y}_{1:k},\boldsymbol{a}_{1:k},\boldsymbol{b}_{1:k}|\boldsymbol{z}_{1:k})$ are equal to the ones calculated from $f(\boldsymbol{y}_{1:k}|\boldsymbol{z}_{1:k})$.

The factorization in (75) can now be represented by a factor graph. This factor graph is shown for the single-sensor case ($n_{\mathrm{s}}=1$) in Fig. 4. As a difference from the factor graph for a known, fixed number of targets shown in Fig. 2(c), for each measurement, an additional variable node "$\overline{\boldsymbol{y}}_{k,s}^{(m)}$", representing the state of a new PT is introduced.

IX. SPA-Based Vector-Type MTT Methods for an Unknown, Time-Varying Number of Targets

In this section, building on the vector-type system model presented in Section VIII and the factor graph shown in Fig. 4, we develop total-SPA MTT methods for tracking an unknown, time-varying number of targets. An important building block of these methods is the "bipartite" formulation of the probabilistic DA problem developed in Section VIII-G and represented graphically in Fig. 4. This formulation enables the application of SPADA, which leads to excellent scalability and, thus, suitability even for large tracking scenarios with closely spaced targets. The proposed SPA-based MTT methods are improved variants of the total-SPA method presented in [53] and can be seen as SPA-based versions of the JIPDA filter [2], [63]. Differently from the method in [53] and the JIPDA filter, they do not use a heuristic to model the generation of new PTs. A particle-based implementation is described in [53].

A. Single-Sensor MTT

We first consider the single-sensor case ($n_{\mathrm{s}}=1$) and thus drop the sensor index s. For MTT with an unknown number of targets, approximations $\tilde{f}\big(\underline{\boldsymbol{y}}_k^{(j)}\big)$ and $\tilde{f}\big(\overline{\boldsymbol{y}}_k^{(m)}\big)$ of the marginal posterior pdfs $f\big(\underline{\boldsymbol{y}}_k^{(j)}\big|\boldsymbol{z}_{1:k}\big)$ and $f\big(\overline{\boldsymbol{y}}_k^{(m)}\big|\boldsymbol{z}_{1:k}\big)$ can be obtained in an efficient way by running the iterative SPA on the loopy factor graph in Fig. 4. As before, messages are only sent forward in time. The generic SPA rules in Section III-B then yield the following operations at time k. (With an abuse of notation, we denote messages that are analogous to messages in previously presented methods in the same way, even if the functional forms are different.)

1) Prediction: First, the messages $\alpha_k\big(\underline{\boldsymbol{y}}_k^{(j)}\big) = \alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},\underline{r}_k^{(j)}\big)$ passed from the factor nodes "$f\big(\underline{\boldsymbol{y}}_k^{(j)}\big|\boldsymbol{y}_{k-1}^{(j)}\big)$" to the variable nodes "$\underline{\boldsymbol{y}}_k^{(j)}$" are calculated according to

$$\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},\underline{r}_k^{(j)}\big) = \sum_{r_{k-1}^{(j)}\in\{0,1\}}\int f\big(\underline{\boldsymbol{x}}_k^{(j)},\underline{r}_k^{(j)}\big|\boldsymbol{x}_{k-1}^{(j)},r_{k-1}^{(j)}\big) \times \tilde{f}\big(\boldsymbol{x}_{k-1}^{(j)},r_{k-1}^{(j)}\big)\,\mathrm{d}\boldsymbol{x}_{k-1}^{(j)}.$$

Here, $\tilde{f}\big(\boldsymbol{x}_{k-1}^{(j)},r_{k-1}^{(j)}\big)$ was calculated at the previous time k–1. Inserting (54) and (55), we obtain for $\underline{r}_k^{(j)}=1$

$$\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},1\big) = \int p_{\mathrm{s}}\big(\boldsymbol{x}_{k-1}^{(j)}\big)\,f\big(\underline{\boldsymbol{x}}_k^{(j)}\big|\boldsymbol{x}_{k-1}^{(j)}\big)\,\tilde{f}\big(\boldsymbol{x}_{k-1}^{(j)},1\big)\,\mathrm{d}\boldsymbol{x}_{k-1}^{(j)}, \tag{76}$$

and for $\underline{r}_k^{(j)}=0$ we have (cf. (51)) $\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},0\big) = \alpha_k^{(j)} f_{\mathrm{D}}\big(\underline{\boldsymbol{x}}_k^{(j)}\big)$ with

$$\alpha_k^{(j)} = \tilde{f}_{k-1}^{(j)} + \int\big(1-p_{\mathrm{s}}\big(\boldsymbol{x}_{k-1}^{(j)}\big)\big)\,\tilde{f}\big(\boldsymbol{x}_{k-1}^{(j)},1\big)\,\mathrm{d}\boldsymbol{x}_{k-1}^{(j)}, \tag{77}$$

where $\tilde{f}_{k-1}^{(j)} = \int\tilde{f}\big(\boldsymbol{x}_{k-1}^{(j)},0\big)\,\mathrm{d}\boldsymbol{x}_{k-1}^{(j)}$. (Note also that $\alpha_k^{(j)} = \int\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},0\big)\,\mathrm{d}\underline{\boldsymbol{x}}_k^{(j)}$; furthermore, $\tilde{f}\big(\boldsymbol{x}_{k-1}^{(j)},0\big) = \tilde{f}_{k-1}^{(j)} f_{\mathrm{D}}\big(\boldsymbol{x}_{k-1}^{(j)}\big)$ according to (51).) Because $\tilde{f}\big(\boldsymbol{x}_{k-1}^{(j)},r_{k-1}^{(j)}\big)$ is normalized, so is $\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},\underline{r}_k^{(j)}\big)$, i.e., $\sum_{\underline{r}_k^{(j)}\in\{0,1\}}\int\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},\underline{r}_k^{(j)}\big)\,\mathrm{d}\underline{\boldsymbol{x}}_k^{(j)} = 1$. Thus, we also have $\alpha_k^{(j)} = 1-\int\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},1\big)\,\mathrm{d}\underline{\boldsymbol{x}}_k^{(j)}$.

2) Measurement Evaluation: Once $\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)},1\big)$ and $\alpha_k^{(j)}$ have been calculated, a "measurement evaluation" step is performed for both the legacy PTs and the new PTs. For the legacy PTs, the messages $\beta_k^{(j)}\big(a_k^{(j)}\big)$ passed from the factor nodes "$q\big(\underline{\boldsymbol{x}}_k^{(j)},\underline{r}_k^{(j)},a_k^{(j)};\boldsymbol{z}_k\big)$" to the variable nodes "$a_k^{(j)}$" are calculated according to

$$\beta_k^{(j)}\big(a_k^{(j)}\big)$$

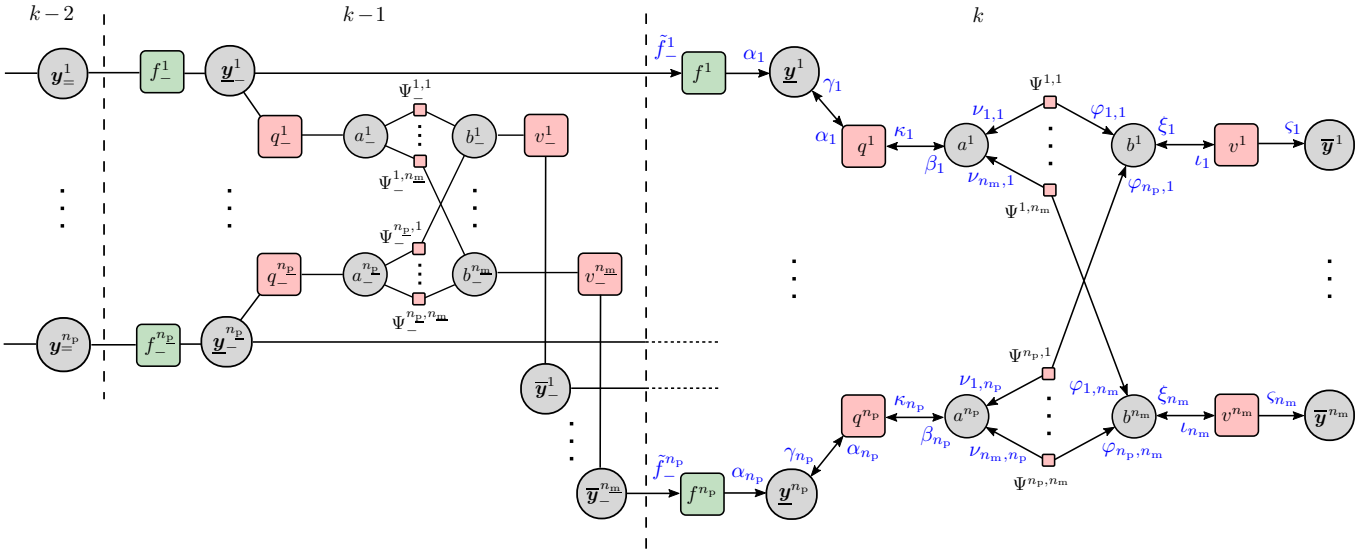


Fig. 4. Factor graph for single-sensor MTT with an unknown, time-varying number of targets, corresponding to the factorization (75) for $n_\mathrm{s}=1$. Two complete consecutive sections of the factor graph (for times $k-1$ and k) are shown; the section for time k also depicts the messages passed between adjacent nodes. Factor nodes in green represent factors related to the state-transition function, factor nodes in red represent the remaining factors, and messages are depicted in blue. The time index k and the sensor index s are omitted, and the following short notations are used: $\boldsymbol{y}_{=}^{j} \triangleq \boldsymbol{y}_{k-2}^{(j)}$, $n_{\underline{\mathrm{m}}} \triangleq m_{k-1}$, $n_{\underline{\mathrm{p}}} \triangleq j_{k-2}$, $\underline{\boldsymbol{y}}_{-}^{j} \triangleq \underline{\boldsymbol{y}}_{k-1}^{(j)}$, $\overline{\boldsymbol{y}}_{-}^{m} \triangleq \overline{\boldsymbol{y}}_{k-1}^{(m)}$, $\boldsymbol{a}_{-}^{j} \triangleq \boldsymbol{a}_{k-1}^{(j)}$, $\boldsymbol{b}_{-}^{m} \triangleq \boldsymbol{b}_{k-1}^{(m)}$, $n_\mathrm{m} \triangleq m_k$, $n_\mathrm{p} \triangleq j_{k-1}$, $\underline{\boldsymbol{y}}^{j} \triangleq \underline{\boldsymbol{y}}_{k}^{(j)}$, $\overline{\boldsymbol{y}}^{m} \triangleq \overline{\boldsymbol{y}}_{k}^{(m)}$, $\boldsymbol{a}^{j} \triangleq \boldsymbol{a}_{k}^{(j)}$, $\boldsymbol{b}^{m} \triangleq \boldsymbol{b}_{k}^{(m)}$, $f_{-}^{j} \triangleq f\big(\underline{\boldsymbol{y}}_{k-1}^{(j)} \big| \underline{\boldsymbol{y}}_{k-2}^{(j)}\big)$, $f^{j} \triangleq f\big(\underline{\boldsymbol{y}}_{k}^{(j)} \big| \underline{\boldsymbol{y}}_{k-1}^{(j)}\big)$, $q_{-}^{j} \triangleq q\big(\underline{\boldsymbol{x}}_{k-1}^{(j)}, \underline{r}_{k-1}^{(j)}, a_{k-1}^{(j)}; \boldsymbol{z}_{k-1}\big)$, $\Psi_{-}^{j,m} \triangleq \Psi_{j,m}\big(a_{k-1}^{(j)}, b_{k-1}^{(m)}\big)$, $\upsilon_{-}^{m} \triangleq \upsilon\big(\overline{\boldsymbol{x}}_{k-1}^{(m)}, \overline{r}_{k-1}^{(m)}, b_{k-1}^{(m)}; \boldsymbol{z}_{k-1}^{(m)}\big)$, $q^{j} \triangleq q\big(\underline{\boldsymbol{x}}_{k}^{(j)}, \underline{r}_{k}^{(j)}, a_{k}^{(j)}; \boldsymbol{z}_{k}\big)$, $\Psi^{j,m} \triangleq \Psi_{j,m}\big(a_{k}^{(j)}, b_{k}^{(m)}\big)$, $\upsilon^{m} \triangleq \upsilon\big(\overline{\boldsymbol{x}}_{k}^{(m)}, \overline{r}_{k}^{(m)}, b_{k}^{(m)}; \boldsymbol{z}_{k}^{(m)}\big)$, $\tilde{f}_{-}^{j} \triangleq \tilde{f}\big(\underline{\boldsymbol{y}}_{k-1}^{(j)}\big)$, $\alpha_j \triangleq \alpha_k\big(\underline{\boldsymbol{y}}_{k}^{(j)}\big)$, $\gamma_j \triangleq \gamma_k^{(j)}\big(\underline{\boldsymbol{y}}_{k}^{(j)}\big)$, $\kappa_j \triangleq \kappa_k^{(j)}\big(a_{k}^{(j)}\big)$, $\beta_j \triangleq \beta_k^{(j)}\big(a_{k}^{(j)}\big)$, $\nu_{m,j} \triangleq \nu^{[\ell]}_{\Psi_{j,m}\to a_k^{(j)}}\big(a_{k}^{(j)}\big)$, $\varphi_{j,m} \triangleq \varphi^{[\ell]}_{\Psi_{j,m}\to b_k^{(m)}}\big(b_{k}^{(m)}\big)$, $\xi_m \triangleq \xi_k^{(m)}\big(b_{k}^{(m)}\big)$, $\iota_m \triangleq \iota_k^{(m)}\big(b_{k}^{(m)}\big)$, and $\varsigma_m \triangleq \varsigma_k^{(m)}\big(\overline{\boldsymbol{y}}_{k}^{(m)}\big)$.

$$= \sum_{\underline{r}_k^{(j)} \in \{0,1\}} \int q\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}, a_k^{(j)}; \boldsymbol{z}_k\big)\, \alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big)\, \mathrm{d}\underline{\boldsymbol{x}}_k^{(j)}$$

$$= \int q\big(\underline{\boldsymbol{x}}_k^{(j)}, 1, a_k^{(j)}; \boldsymbol{z}_k\big)\, \alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)}, 1\big)\, \mathrm{d}\underline{\boldsymbol{x}}_k^{(j)} + 1\big(a_k^{(j)}\big)\, \alpha_k^{(j)}. \quad (78)$$

In the last expression, we used $q\big(\underline{\boldsymbol{x}}_k^{(j)}, 0, a_k^{(j)}; \boldsymbol{z}_k\big) = 1\big(a_k^{(j)}\big)$, which follows from (72) along with (65) and (58). For the new PTs, the messages $\xi_k^{(m)}\big(b_k^{(m)}\big)$ passed from the factor nodes "$\upsilon\big(\overline{\boldsymbol{x}}_k^{(m)}, \overline{r}_k^{(m)}, b_k^{(m)}; \boldsymbol{z}_k^{(m)}\big)$" to the variable nodes "$b_k^{(m)}$" are calculated according to

$$\xi_k^{(m)}\big(b_k^{(m)}\big) = \sum_{\overline{r}_k^{(m)} \in \{0,1\}} \int \upsilon\big(\overline{\boldsymbol{x}}_k^{(m)}, \overline{r}_k^{(m)}, b_k^{(m)}; \boldsymbol{z}_k^{(m)}\big)\, \mathrm{d}\overline{\boldsymbol{x}}_k^{(m)}$$

$$= \int \upsilon\big(\overline{\boldsymbol{x}}_k^{(m)}, 1, b_k^{(m)}; \boldsymbol{z}_k^{(m)}\big)\, \mathrm{d}\overline{\boldsymbol{x}}_k^{(m)} + 1, \quad (79)$$

where (74) was used. Inserting (73), we find that for $b_k^{(m)} = 0$, expression (79) simplifies to

$$\xi_k^{(m)}(0) = \frac{\mu_\mathrm{n}}{\mu_\mathrm{c}} \int f_\mathrm{n}\big(\overline{\boldsymbol{x}}_k^{(m)}\big) f\big(\boldsymbol{z}_k^{(m)} \big| \overline{\boldsymbol{x}}_k^{(m)}\big)\, \mathrm{d}\overline{\boldsymbol{x}}_k^{(m)} + 1,$$

and for $b_k^{(m)} \neq 0$, it simplifies to $\xi_k^{(m)}\big(b_k^{(m)}\big) = 1$.

For future reference, we note that the belief of the joint DA vector $[\mathbf{a}_k^\mathrm{T}\ \mathbf{b}_k^\mathrm{T}]^\mathrm{T}$ is obtained as

$$\tilde{p}(\boldsymbol{a}_k, \boldsymbol{b}_k) \propto \psi(\boldsymbol{a}_k, \boldsymbol{b}_k) \left(\prod_{j=1}^{j_{k-1}} \beta_k^{(j)}\big(a_k^{(j)}\big) \right) \prod_{m=1}^{m_k} \xi_k^{(m)}\big(b_k^{(m)}\big), \quad (80)$$

where $\psi(\boldsymbol{a}_k, \boldsymbol{b}_k)$ is given by $\psi(\boldsymbol{a}_k, \boldsymbol{b}_k) = \prod_{j=1}^{j_{k-1}} \prod_{m=1}^{m_k} \Psi_{j,m}\big(a_k^{(j)}, b_k^{(m)}\big)$ (cf. (21)).

3) Iterative probabilistic DA: Next, probabilistic DA is performed using the iterative SPA-based algorithm from Section VI-A, which yields at each iteration ℓ the messages $\varphi^{[\ell]}_{\Psi_{j,m}\to b_k^{(m)}}\big(b_k^{(m)}\big)$ and $\nu^{[\ell]}_{\Psi_{j,m}\to a_k^{(j)}}\big(a_k^{(j)}\big)$. Here, the messages $\beta_k^{(j)}\big(a_k^{(j)}\big)$ from (78) are used in (27) and (29), and the messages $\xi_k^{(m)}\big(b_k^{(m)}\big)$ from (79) are used in (28). After the last iteration $\ell = n_\mathrm{it}$, the messages passed from "$a_k^{(j)}$" to "$q\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}, a_k^{(j)}; \boldsymbol{z}_k\big)$" are obtained as

$$\kappa_k^{(j)}\big(a_k^{(j)}\big) = \prod_{m=1}^{m_k} \nu^{[n_\mathrm{it}]}_{\Psi_{j,m}\to a_k^{(j)}}\big(a_k^{(j)}\big), \quad (81)$$

and the messages passed from "$b_k^{(m)}$" to "$\upsilon\big(\overline{\boldsymbol{x}}_k^{(m)}, \overline{r}_k^{(m)}, b_k^{(m)}; \boldsymbol{z}_k^{(m)}\big)$" are obtained as

$$\iota_k^{(m)}\big(b_k^{(m)}\big) = \prod_{j=1}^{j_{k-1}} \varphi^{[n_\mathrm{it}]}_{\Psi_{j,m}\to b_k^{(m)}}\big(b_k^{(m)}\big). \quad (82)$$

For an efficient implementation, one can use SPADA. The messages $\nu_k^{[n_\mathrm{it}](m\to j)}$ in (31) and $\varphi_k^{[n_\mathrm{it}](j\to m)}$ in (30) then have to be converted into the messages $\nu^{[n_\mathrm{it}]}_{\Psi_{j,m}\to a_k^{(j)}}\big(a_k^{(j)}\big)$ used in (81) and $\varphi^{[n_\mathrm{it}]}_{\Psi_{j,m}\to b_k^{(m)}}\big(b_k^{(m)}\big)$ used in (82), respectively. For $\nu_k^{[n_\mathrm{it}](m\to j)}$, this is done as explained in Section VII-A2. For $\varphi_k^{[n_\mathrm{it}](j\to m)}$, similarly, $\varphi^{[n_\mathrm{it}]}_{\Psi_{j,m}\to b_k^{(m)}}\big(b_k^{(m)}\big) = \varphi_k^{[n_\mathrm{it}](j\to m)}$ for $b_k^{(m)} = j$ and $\varphi^{[n_\mathrm{it}]}_{\Psi_{j,m}\to b_k^{(m)}}\big(b_k^{(m)}\big) = 1$ for $b_k^{(m)} \neq j$.

4) Measurement Update: A "measurement update" step is now performed for both the legacy PTs and the new PTs. For the legacy PTs, the messages $\gamma_k^{(j)}\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big)$ passed from

"$q\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}, a_k^{(j)}; \boldsymbol{z}_k\big)$" to "$\underline{\boldsymbol{y}}_k^{(j)}$" are calculated according to

$$\gamma_k^{(j)}\big(\underline{\boldsymbol{x}}_k^{(j)}, 1\big) = \sum_{a_k^{(j)}=0}^{m_k} q\big(\underline{\boldsymbol{x}}_k^{(j)}, 1, a_k^{(j)}; \boldsymbol{z}_k\big)\, \kappa_k^{(j)}\big(a_k^{(j)}\big)$$

and $\gamma_k^{(j)}\big(\underline{\boldsymbol{x}}_k^{(j)}, 0\big) = \gamma_k^{(j)}$ with

$$\gamma_k^{(j)} = \kappa_k^{(j)}(0).$$

For the new PTs, the messages $\varsigma_k^{(m)}\big(\overline{\boldsymbol{x}}_k^{(m)}, \overline{r}_k^{(m)}\big)$ passed from "$v\big(\overline{\boldsymbol{x}}_k^{(m)}, \overline{r}_k^{(m)}, b_k^{(m)}; \boldsymbol{z}_k^{(m)}\big)$" to "$\overline{\boldsymbol{y}}_k^{(m)}$" are calculated as

$$\varsigma_k^{(m)}\big(\overline{\boldsymbol{x}}_k^{(m)}, 1\big) = \frac{\mu_{\mathrm{n}}}{\mu_{\mathrm{c}}}\, f_{\mathrm{n}}\big(\overline{\boldsymbol{x}}_k^{(m)}\big) f\big(\boldsymbol{z}_k^{(m)} \big| \overline{\boldsymbol{x}}_k^{(m)}\big)\, \iota_k^{(m)}(0)$$

and $\varsigma_k^{(m)}\big(\overline{\boldsymbol{x}}_k^{(m)}, 0\big) = \varsigma_k^{(m)} f_{\mathrm{D}}\big(\overline{\boldsymbol{x}}_k^{(m)}\big)$ with

$$\varsigma_k^{(m)} = \sum_{b_k^{(m)}=0}^{j_{k-1}} \iota_k^{(m)}\big(b_k^{(m)}\big).$$

5) Belief Calculation: Finally, for the legacy PTs, beliefs $\tilde{f}\big(\underline{\boldsymbol{y}}_k^{(j)}\big) = \tilde{f}\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big)$ approximating the marginal posterior pdfs $f\big(\underline{\boldsymbol{y}}_k^{(j)} \big| \boldsymbol{z}_{1:k}\big) = f\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)} \big| \boldsymbol{z}_{1:k}\big)$ are obtained as

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

and $\tilde{f}\big(\underline{\boldsymbol{x}}_k^{(j)}, 0\big) = \underline{f}_k^{(j)} f_{\mathrm{D}}\big(\underline{\boldsymbol{x}}_k^{(j)}\big)$ with

$$\underline{f}_k^{(j)} = \frac{1}{\underline{C}_k^{(j)}}\, \alpha_k^{(j)} \gamma_k^{(j)}, \tag{83}$$

where $\underline{C}_k^{(j)} \triangleq \int \alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)}, 1\big) \gamma_k^{(j)}\big(\underline{\boldsymbol{x}}_k^{(j)}, 1\big)\, \mathrm{d}\underline{\boldsymbol{x}}_k^{(j)} + \alpha_k^{(j)} \gamma_k^{(j)}$. Similarly, for the new PTs, beliefs $\tilde{f}\big(\overline{\boldsymbol{y}}_k^{(m)}\big) = \tilde{f}\big(\overline{\boldsymbol{x}}_k^{(m)}, \overline{r}_k^{(m)}\big)$ approximating the marginal posterior pdfs $f\big(\overline{\boldsymbol{y}}_k^{(m)} \big| \boldsymbol{z}_{1:k}\big) = f\big(\overline{\boldsymbol{x}}_k^{(m)}, \overline{r}_k^{(m)} \big| \boldsymbol{z}_{1:k}\big)$ are obtained as

$$\tilde{f}\big(\overline{\boldsymbol{x}}_k^{(m)}, 1\big) = \frac{1}{\overline{C}_k^{(m)}}\, \varsigma_k^{(m)}\big(\overline{\boldsymbol{x}}_k^{(m)}, 1\big) \tag{84}$$

and $\tilde{f}\big(\overline{\boldsymbol{x}}_k^{(m)}, 0\big) = \overline{f}_k^{(m)} f_{\mathrm{D}}\big(\overline{\boldsymbol{x}}_k^{(m)}\big)$ with

$$\overline{f}_k^{(m)} = \frac{1}{\overline{C}_k^{(m)}}\, \varsigma_k^{(m)}, \tag{85}$$

where $\overline{C}_k^{(m)} \triangleq \int \varsigma_k^{(m)}\big(\overline{\boldsymbol{x}}_k^{(m)}, 1\big) \mathrm{d}\overline{\boldsymbol{x}}_k^{(m)} + \varsigma_k^{(m)}$.

6) Target Declaration, State Estimation, and Pruning: For the legacy PTs, beliefs $\tilde{p}\big(\underline{r}_k^{(j)}\big)$ approximating the marginal posterior pmfs $p\big(\underline{r}_k^{(j)} \big| \boldsymbol{z}_{1:k}\big)$ of the existence indicators $\underline{r}_k^{(j)}$ are obtained from the beliefs $\tilde{f}\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big)$ as

$$\tilde{p}\big(\underline{r}_k^{(j)}\big) = \int \tilde{f}\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big)\, \mathrm{d}\underline{\boldsymbol{x}}_k^{(j)}.$$

Target detection—hereafter termed "target declaration" to avoid confusion with the detection performed by the sensors—is now performed by comparing $\tilde{p}\big(\underline{r}_k^{(j)} = 1\big)$ to a threshold P_{th}, i.e., legacy PT j is declared to exist at time k if $\tilde{p}\big(\underline{r}_k^{(j)} = 1\big) > P_{\mathrm{th}}$ [149, Ch. 2]. For these PTs j, state estimation is then performed, e.g., using (3) with $f(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ replaced by $\tilde{f}\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)} = 1\big)/\tilde{p}\big(\underline{r}_k^{(j)} = 1\big)$. For the new PTs, target declaration and state estimation are done as for the legacy PTs but with $\tilde{f}\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big)$ replaced by $\tilde{f}\big(\overline{\boldsymbol{x}}_k^{(m)}, \overline{r}_k^{(m)}\big)$.

Finally, a pruning step is typically performed. According to (52), the number of PTs would grow with time k. Therefore, legacy and new PTs whose existence beliefs $\tilde{p}\big(\underline{r}_k^{(j)} = 1\big)$ and $\tilde{p}\big(\overline{r}_k^{(m)} = 1\big)$, respectively are below a threshold P_{pr} are removed.

B. Multisensor MTT

For multiple sensors ($n_{\mathrm{s}} \geq 2$), the factor graph differs from that for a single sensor shown in Fig. 4 in that an additional section of the factor graph is introduced for each additional sensor, and for each time k, there is a factor node "$f\big(\underline{\boldsymbol{y}}_k^{(j)} \big| \underline{\boldsymbol{y}}_{k-1}^{(j)}\big)$" in the section corresponding to the first sensor. The total-SPA MTT method for an unknown, time-varying number of targets presented in the previous subsection can be extended to the case of multiple sensors by using a sensor-sequential or sensor-parallel processing. In either case, the first step is the prediction step, i.e., calculation of the messages $\alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big)$ according to (76) and (77).

1) Sequential Processing: As in Section VII-C1, message passing operations are performed sequentially for each sensor $s = 1, \ldots, n_{\mathrm{s}}$. (However, contrary to MTT methods for a known number of targets, the sequence of sensors was already defined in the derivation of the factor graph, see Section VIII.) Let us denote the beliefs—for both the legacy and new PTs—calculated at the $(s-1)$th sensor update step (i.e., at sensor $s-1$) as $\tilde{f}_{s-1}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}\big)$, $j = 1, \ldots, j_{k,s}$. (We recall from Section VIII-B that $\underline{\boldsymbol{y}}_{k,s} = [\underline{\boldsymbol{y}}_{k,s-1}^{\mathrm{T}}\, \overline{\boldsymbol{y}}_{k,s-1}^{\mathrm{T}}]^{\mathrm{T}}$ and $j_{k,s} = j_{k,s-1} + m_{k,s-1}$.) These beliefs serve as input to the sth sensor update step (i.e., at sensor s), in which the message passing and belief calculation operations described in Sections IX-A2 through IX-A5 are performed. This sensor-recursive processing is initialized by $\tilde{f}_0\big(\underline{\boldsymbol{x}}_{k,1}^{(j)}, \underline{r}_{k,1}^{(j)}\big) = \alpha_k\big(\underline{\boldsymbol{x}}_k^{(j)}, \underline{r}_k^{(j)}\big)$, $j = 1, \ldots, j_{k,1}$, which were calculated in the prediction step. (Recall that $\underline{\boldsymbol{y}}_{k,1} = \underline{\boldsymbol{y}}_k$ and $j_{k,1} = j_{k-1}$.) The beliefs $\tilde{f}_{n_{\mathrm{s}}}\big(\underline{\boldsymbol{x}}_{k,n_{\mathrm{s}}}^{(j)}, \underline{r}_{k,n_{\mathrm{s}}}^{(j)}\big)$, $j = 1, \ldots, j_{k,n_{\mathrm{s}}}$ and $\tilde{f}_{n_{\mathrm{s}}}\big(\overline{\boldsymbol{x}}_{k,n_{\mathrm{s}}}^{(m)}, \overline{r}_{k,n_{\mathrm{s}}}^{(m)}\big)$, $m = 1, \ldots, m_{k,n_{\mathrm{s}}}$ calculated at the last sensor $s = n_{\mathrm{s}}$ take into account the measurements of all sensors; they are used for target declaration and state estimation as described in Section IX-A6, and to perform prediction at the next time step $k+1$.

An advantage of this sensor-sequential method over the sensor-parallel method considered next is that it performs sensor fusion for new PTs and it executes SPADA only once per sensor. On the other hand, sequential processing is not well suited to a parallel or distributed implementation.

2) Parallel Processing: The SPA-based message passing method for sensor-parallel MTT can be summarized as follows. For all sensors $s \geq 2$, the messages $\alpha_k\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, \underline{r}_{k,s}^{(j)}\big)$, $j > j_{k-1}$ are not available and are thus initialized by $\alpha_k\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 0\big) = f_{\mathrm{D}}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}\big)$ and $\alpha_k\big(\underline{\boldsymbol{x}}_{k,s}^{(j)}, 1\big) = 0$. (These messages belong to legacy PTs at sensor s and to new PTs at sensors preceding sensor s with respect to the sensor ordering defined in Section VIII.) Then, the message passing operations described in Sections IX-A2 through IX-A4 are carried out separately for each sensor $s \in \{1, \ldots, n_{\mathrm{s}}\}$.

As a result, for each sensor $s \in \{1,\dots,n_\mathrm{s}\}$, messages[6] $\gamma_{k,s}^{(j)}\big(\underline{\boldsymbol{x}}_{k,s}^{(j)},\underline{r}_{k,s}^{(j)}\big) = \gamma_{k,s}^{(j)}\big(\underline{\boldsymbol{x}}_{k}^{(j)},\underline{r}_{k}^{(j)}\big)$ are available for each legacy PT $j \in \{1,\dots,j_{k-1}\}$, and messages $\varsigma_{k,s}^{(m)}\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)}\big)$ are available for each new PT $m \in \{1,\dots,m_{k,s}\}$. Finally, beliefs approximating the marginal posterior pdfs for the legacy PTs $j \in \{1,\dots,j_{k-1}\}$, $f\big(\underline{\boldsymbol{y}}_{k}^{(j)}\big|\boldsymbol{z}_{1:k}\big) = f\big(\underline{\boldsymbol{x}}_{k}^{(j)},\underline{r}_{k}^{(j)}\big|\boldsymbol{z}_{1:k}\big)$, are calculated as (cf. (83) and (83))

$$\tilde{f}\big(\underline{\boldsymbol{x}}_{k}^{(j)},\underline{r}_{k}^{(j)}\big) \propto \alpha_{k}^{(j)}\big(\underline{\boldsymbol{x}}_{k}^{(j)},\underline{r}_{k}^{(j)}\big)\prod_{s=1}^{n_\mathrm{s}}\gamma_{k,s}^{(j)}\big(\underline{\boldsymbol{x}}_{k}^{(j)},\underline{r}_{k}^{(j)}\big).$$

Furthermore, beliefs approximating the marginal posterior pdfs for the new PTs, $f\big(\overline{\boldsymbol{y}}_{k,s}^{(m)}\big|\boldsymbol{z}_{1:k}\big) = f\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)}\big|\boldsymbol{z}_{1:k}\big)$, for $s \in \{1,\dots,n_\mathrm{s}\}$ and $m \in \{1,\dots,m_{k,s}\}$, are directly obtained as (cf. (84) and (85))

$$\tilde{f}\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)}\big) \propto \varsigma_{k,s}^{(m)}\big(\overline{\boldsymbol{x}}_{k,s}^{(m)},\overline{r}_{k,s}^{(m)}\big).$$

This method facilitates a parallel or distributed implementation. On the other hand, its performance may be poorer than that of the sequential method described previously. This is because no sensor fusion is performed for new PTs, i.e., each sensor attempts to infer new targets individually. As demonstrated in [53], the computational complexity of sensor-parallel processing scales strictly linearly with the number of sensors n_s. We note that, similarly to [52], sensor-parallel processing can be extended to multiple SPA iterations over an "outer loop" that spans across the different sensors. In this way, sensor fusion gains can be leveraged also for the inference of new targets. However, this comes at the cost of an increased computational complexity since SPADA has to be executed multiple times for each sensor (once for each outer-loop iteration).

X. Introduction to RFSs

We now turn to the set-type MTT methods. We first give an introduction to RFSs, which underly the set-type system model and MTT methods to be presented in Sections XI–XIII.

A. Basic Description

An RFS $\mathsf{X} = \{\mathbf{x}^{(1)},\dots,\mathbf{x}^{(\mathsf{n})}\}$ (also known as a simple finite point process [150], [151]) is a set-valued random variable whose realizations $\mathcal{X}$ are finite sets $\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)}\}$ of vectors $\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)} \in \mathbb{R}^{d_x}$. Both the number of elements $\mathsf{n} = |\mathsf{X}| \in \mathbb{N}_0$—the cardinality of X—and the elements $\mathbf{x}^{(i)}$ are random, and the elements $\mathbf{x}^{(i)}$ are unordered. Adopting the framework of finite set statistics (FISST) [3], an RFS can be described by its pdf $f_\mathsf{X}(\mathcal{X})$, briefly denoted $f(\mathcal{X})$. For a realization $\mathcal{X} = \{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)}\}$ with cardinality $|\mathcal{X}| = n$, the pdf is given by

$$f(\mathcal{X}) = f(\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)}\}) = n!\,\rho(n)\,f_n(\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)}). \tag{86}$$

Here, $\rho(n) \triangleq \mathbb{P}\{|\mathsf{X}| = n\}$ is the cardinality distribution, i.e., the pmf of $\mathsf{n} = |\mathsf{X}|$, and $f_n(\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)})$ is a pdf of the random vectors $\mathbf{x}^{(1)},\dots,\mathbf{x}^{(n)}$ that is invariant to a permutation of its arguments $\boldsymbol{x}^{(i)}$. It will be convenient to define the *set integral* of a real-valued set function $g(\mathcal{X})$ as [3]

$$\int g(\mathcal{X})\,\delta\mathcal{X} \triangleq \sum_{n=0}^{\infty}\frac{1}{n!}\int_{\mathbb{R}^{n d_x}} g(\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)}\})\,\mathrm{d}\boldsymbol{x}^{(1)}\cdots\mathrm{d}\boldsymbol{x}^{(n)}. \tag{87}$$

For an RFS pdf $f(\mathcal{X})$, we have $\int f(\mathcal{X})\,\delta\mathcal{X} = 1$.

For an RFS X, the *probability hypothesis density* (PHD) $D(\boldsymbol{x}) : \mathbb{R}^{d_x} \to \mathbb{R}$ is defined by the property that its integral over a region $\mathcal{S} \subseteq \mathbb{R}^{d_x}$ equals the expected number of elements present in $\mathcal{S}$, i.e., $\int_\mathcal{S} D(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mathbb{E}\{|\mathsf{X}\cap\mathcal{S}|\}$ [3]. An expression of $D(\boldsymbol{x})$ in terms of $f(\mathcal{X})$ can be found in [3].

B. Special RFSs

Next, we review four types of RFSs that are especially relevant to RFS-based MTT methods. Given the cardinality $|\mathcal{X}| = n$, the elements of a *Poisson RFS* X are iid with some "spatial pdf" $f(\boldsymbol{x})$, i.e., $f_n(\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)}) = \prod_{i=1}^{n} f(\boldsymbol{x}^{(i)}) = \prod_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$. Furthermore, the cardinality is Poisson distributed with mean μ, i.e., $\rho(n) = e^{-\mu}\mu^n/n!$, $n \in \mathbb{N}_0$. Hence, (86) yields the pdf of X as

$$f(\mathcal{X}) = e^{-\mu}\prod_{\boldsymbol{x}\in\mathcal{X}}\mu f(\boldsymbol{x}).$$

The PHD of the Poisson RFS is given by $D(\boldsymbol{x}) = \mu f(\boldsymbol{x})$; it is also known as *intensity function*.

A *Bernoulli RFS* either contains one element $\mathbf{x}$ with probability of existence r or is empty with probability $1-r$. If it is nonempty, the element $\mathbf{x}$ is distributed according to the "spatial pdf" $s(\boldsymbol{x})$. Hence, the RFS pdf is given by

$$f(\mathcal{X}) = \begin{cases} 1-r, & \mathcal{X} = \emptyset, \\ r\,s(\boldsymbol{x}), & \mathcal{X} = \{\boldsymbol{x}\}, \\ 0, & \text{otherwise.} \end{cases} \tag{88}$$

We note that $\sum_j \gamma_j f^{(j)}(\mathcal{X})$ with normalized γ_j (i.e., $\sum_j \gamma_j = 1$) and Bernoulli pdfs $f^{(j)}(\mathcal{X})$ is again a Bernoulli pdf.

A *multi-Bernoulli (MB) RFS* is the union of n_c independent component RFSs $\mathsf{X}^{(j)}$, $j = 1,\dots,n_\mathrm{c}$, which are Bernoulli RFSs with existence probabilities $r^{(j)}$ and spatial pdfs $s^{(j)}(\boldsymbol{x})$. Let $f^{(j)}(\mathcal{X})$ denote the pdf of Bernoulli component $\mathsf{X}^{(j)}$. To express $f(\mathcal{X})$ for a realization $\mathcal{X} = \{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)}\}$ of cardinality $n \leqslant n_\mathrm{c}$, we introduce a mapping α of an index $j \in \{1,\dots,n_\mathrm{c}\}$ to an index $\alpha(j) \in \{0,\dots,n\}$. This mapping is such that the set of all $\alpha(j)$ includes $\{1,\dots,n\}$, and $j_1 \neq j_2$ with $\alpha(j_1),\alpha(j_2) \in \{1,\dots,n\}$ implies $\alpha(j_1) \neq \alpha(j_2)$. In our context, α conveys a mapping of n of the n_c Bernoulli component pdfs $f^{(j)}(\mathcal{X})$ to single-vector element sets $\{\boldsymbol{x}^{(\alpha(j))}\}$ and the other $n_\mathrm{c}-n$ Bernoulli component pdfs to empty sets. Let $\mathcal{P}_{n_\mathrm{c},n}$ denote the set of all such "components-to-elements" mappings α; note that $|\mathcal{P}_{n_\mathrm{c},n}| = n_\mathrm{c}!/(n_\mathrm{c}-n)!$. The pdf can now be expressed as [51]

$$f(\mathcal{X}) = \sum_{\alpha\in\mathcal{P}_{n_\mathrm{c},n}}\prod_{j=1}^{n_\mathrm{c}} f^{(j)}\big(\mathcal{X}^{(\alpha(j))}\big), \tag{89}$$

where $n = |\mathcal{X}|$ and $\mathcal{X}^{(\alpha(j))}$ is $\{\boldsymbol{x}^{(\alpha(j))}\}$ for $\alpha(j) \in \{1,\dots,n\}$ and $\emptyset$ for $\alpha(j) = 0$.

A *labeled MB (LMB) RFS* is an MB RFS where each state variable is augmented by a label [108]. More specifically, a

[6] We recall that for $j \in \{1,\dots,j_{k-1}\}$, $\underline{\boldsymbol{y}}_{k,s}^{(j)} = \underline{\boldsymbol{y}}_{k}^{(j)}$ for all $s \in \{1,\dots,n_\mathrm{s}\}$.

track-labeling function $\tau(\cdot)$ assigns to each Bernoulli component $\mathsf{X}^{(j)}$, $j \in \{1,\ldots,n_\mathrm{c}\}$ a distinct label $l \in \mathcal{L}$, where $\mathcal{L} = \{\tau(j) : j \in \{1,\ldots,n_\mathrm{c}\}\}$. The realization of an LMB RFS is a set $\mathcal{X} = \{(\boldsymbol{x}^{(1)}, l^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, l^{(n)})\}$ of tuples $(\boldsymbol{x}^{(i)}, l^{(i)}) \in \mathbb{R}^{d_x} \times \mathcal{L}$, $i \in \{1,\ldots,n\}$, with $l^{(i)} \neq l^{(i')}$ for $i \neq i'$. The pdf is given by [108]

$$f(\mathcal{X}) = \prod_{j=1}^{n_\mathrm{c}} f^{(j)}\big(\mathcal{X}^{(\tau(j))}\big), \tag{90}$$

where $f^{(j)}(\mathcal{X})$ is the pdf of Bernoulli component $\mathsf{X}^{(j)}$ and $\mathcal{X}^{(\tau(j))}$ is given by $\mathcal{X}^{(\tau(j))} = \{(\boldsymbol{x}^{(i)}, l^{(i)}) \in \mathcal{X} : \tau(j) = l^{(i)}\}$. For sets $\mathcal{X} = \{(\boldsymbol{x}^{(1)}, l^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, l^{(n)})\}$ that have elements with nondistinct labels, i.e., $\exists i, i'$ with $l^{(i)} = l^{(i')} = \tau(j)$, we have $|\mathcal{X}^{(\tau(j))}| > 1$ and thus $f(\mathcal{X}) = 0$ according to (88). Note that in the evaluation of $f(\mathcal{X})$, a sum over all possible components-to-elements mappings as in (89) is avoided since the mapping is fixed, i.e., given by the labels $l^{(1)}, \ldots, l^{(n)}$.

XI. Set-Type System Model

The system model for set-type MTT describes all the target states and all the measurements of a sensor as time-varying RFSs. Several set-type system models are available. In particular, the "unlabeled" model proposed in [51], [107] describes newborn targets by a Poisson RFS. The posterior pdf consists of an MB mixture component representing *detected targets*, which are targets that potentially exist and generated at least one measurement so far, and a Poisson component representing *undetected targets*, which are targets that potentially exist but did not generate any measurement yet. (The latter are termed *unknown targets* in [51], [152].) A new Bernoulli component is generated for each measurement, representing the hypothesis that this measurement is the first detection of a target that was previously undetected and thus was previously modeled by a Poisson component. This model is able to maintain track continuity implicitly based on information provided by metadata. As an alternative, the "labeled" model proposed in [108], [109] explicitly maintains track continuity through labels. Target birth is modeled by an LMB RFS, where each Bernoulli component represents a potential new target and has a distinct label. In cases of limited prior birth information, one typically uses a heuristic to generate new Bernoulli components based on measurements from the previous time step [109]. Such heuristics can be avoided with the MB–Poisson model [51], [107].

Here, we propose a hybrid "labeled-unlabeled" system model that combines the benefits of the unlabeled and labeled models, i.e., generation of new Bernoulli components based on the Poisson RFS modeling of undetected targets and indexing of already detected targets by a distinct label providing explicit track continuity. We represent targets that have been detected by an LMB RFS and targets that remain undetected by an unlabeled Poisson RFS in which each component is formally augmented by the same nonunique label $l_k = 0$. When a target is detected for the first time, this "dummy label" is replaced by a new unique label that identifies the underlying measurement. This is not the result of an update step but an incorporation of exogenous information; it can be seen as an additional external measurement of the label.[7]

A. State-Transition pdf and Prior Distribution

1) Assumptions: The entirety of the target states is modeled as a time-varying RFS $\mathsf{X}_k \triangleq \{(\mathsf{x}_k^{(1)}, \mathsf{l}_k^{(1)}), \ldots, (\mathsf{x}_k^{(\mathsf{i}_k)}, \mathsf{l}_k^{(\mathsf{i}_k)})\}$. Furthermore, we use the following assumptions in addition to the common assumptions A1–A11 [3], [51]:

- S1: The number of targets, i_k, is time-varying and unknown.
- S2: The target states (elements of X_k) are unordered.
- S3: A target i that exists at time $k-1$ survives (i.e., still exists at time k) with survival probability $p_\mathrm{s}(\boldsymbol{x}_{k-1}^{(i)})$ and disappears with probability $1 - p_\mathrm{s}(\boldsymbol{x}_{k-1}^{(i)})$. The survival probability $p_\mathrm{s}(\boldsymbol{x}_{k-1}^{(i)})$ does not depend on the target label. Targets that survive preserve their label.
- S4: Each target i is a survived target or a newborn target.
- S5: The states of the newborn targets at time k are independent of the states of the survived targets at time k.
- S6: The states of the newborn targets at time k are iid. Each of them has label $l_k^{(i)} = 0$ and is distributed according to the birth pdf $f_\mathrm{b}(\boldsymbol{x}_k^{(i)})$.
- S7: The number of newborn targets at time k is Poisson distributed with mean μ_b.
- S8: At time $k = 0$, all target states have label $l_0^{(i)} = 0$ (since no targets have been detected); they are iid and distributed according to a prior pdf $f_\mathrm{p}(\boldsymbol{x}_0^{(i)})$.
- S9: The number of targets at time $k = 0$ is Poisson distributed with mean μ_p.

The overall multitarget state RFS X_k is the union of the RFSs of the detected and undetected targets, i.e.,

$$\mathsf{X}_k = \mathsf{X}_k^\mathrm{d} \cup \mathsf{X}_k^\mathrm{u}. \tag{91}$$

Here, X_k^d is a labeled RFS whose elements have unique nonzero labels, and X_k^u is an unlabeled RFS whose elements have the nonunique dummy label $\mathsf{l}_k = 0$. The label of a detected target is a unique identifier of the first measurement generated by that target. Undetected targets become detected targets as soon as they produce a measurement; conversely, detected targets cannot become undetected targets.

2) Detected Targets: Let $\mathcal{X}_{k-1}^\mathrm{d} \triangleq \{(\boldsymbol{x}_{k-1}^{(1)}, l_{k-1}^{(1)}), \ldots, (\boldsymbol{x}_{k-1}^{(i_{k-1}^\mathrm{d})}, l_{k-1}^{(i_{k-1}^\mathrm{d})})\}$ comprise the states of the detected targets that existed at time $k-1$. Because of A1, A2, and S3, each detected target j that existed at time $k-1$ gives rise to a—possibly nonexistent—"survived target" at time k. The state of this survived target is modeled as a Bernoulli RFS $\mathsf{X}_{k,j}^\mathrm{d}$ (cf. (88)) with $r = p_\mathrm{s}(\boldsymbol{x}_{k-1}^{(j)})$ and $s(\boldsymbol{x}_k) = f(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(j)})$. The associated multitarget (RFS) state-transition pdf is thus

$$f\big(\mathcal{X}_{k,j}^\mathrm{d} \big| \{(\boldsymbol{x}_{k-1}^{(j)}, l_{k-1}^{(j)})\}\big) = \begin{cases} 1 - p_\mathrm{s}(\boldsymbol{x}_{k-1}^{(j)}), & \mathcal{X}_{k,j}^\mathrm{d} = \emptyset, \\ p_\mathrm{s}(\boldsymbol{x}_{k-1}^{(j)})\, f(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(j)}), & \mathcal{X}_{k,j}^\mathrm{d} = \{(\boldsymbol{x}_k, l_{k-1}^{(j)})\}, \\ 0, & \text{otherwise.} \end{cases} \tag{92}$$

[7] If the label is viewed as a proxy for the trajectory, the formulation in [153] provides a similar result directly via a Bayes update.

In view of A2, S1, and S2, the overall multitarget state RFS of the detected targets at time k is the LMB RFS $\mathsf{X}_k^{\mathrm{d}} = \{(\mathbf{x}_k^{(1)}, \mathsf{l}^{(1)}), \ldots, (\mathbf{x}_k^{(\mathsf{i}_k^{\mathrm{d}})}, \mathsf{l}^{(\mathsf{i}_k^{\mathrm{d}})})\}$ given by

$$\mathsf{X}_k^{\mathrm{d}} = \bigcup_{j=1}^{i_{k-1}^{\mathrm{d}}} \mathsf{X}_{k,j}^{\mathrm{d}}. \tag{93}$$

Here, because of A2, the Bernoulli components $\mathsf{X}_{k,j}^{\mathrm{d}}$ are conditionally independent given $\mathsf{X}_{k-1}^{\mathrm{d}}$.

The joint evolution of the detected target states is described by the multitarget state-transition pdf $f(\mathcal{X}_k^{\mathrm{d}}|\mathcal{X}_{k-1}^{\mathrm{d}})$, where $\mathcal{X}_k^{\mathrm{d}} = \{(\boldsymbol{x}_k^{(1)}, l^{(1)}), \ldots, (\boldsymbol{x}_k^{(i_k^{\mathrm{d}})}, l^{(i_k^{\mathrm{d}})})\}$ and $\mathcal{X}_{k-1}^{\mathrm{d}} = \{(\boldsymbol{x}_{k-1}^{(1)}, l^{(1)}), \ldots, (\boldsymbol{x}_{k-1}^{(i_{k-1}^{\mathrm{d}})}, l^{(i_{k-1}^{\mathrm{d}})})\}$. (Note that $i_k^{\mathrm{d}} \leqslant i_{k-1}^{\mathrm{d}}$, where $i_k^{\mathrm{d}} = i_{k-1}^{\mathrm{d}}$ if all survived targets exist at time k and $i_k^{\mathrm{d}} < i_{k-1}^{\mathrm{d}}$ otherwise.) Using (93), one obtains [108]

$$f(\mathcal{X}_k^{\mathrm{d}}|\mathcal{X}_{k-1}^{\mathrm{d}}) = \prod_{j=1}^{i_{k-1}^{\mathrm{d}}} f\big(\mathcal{X}_k^{(\tau(j))}\big|\{(\boldsymbol{x}_{k-1}^{(j)}, l_{k-1}^{(j)})\}\big), \tag{94}$$

where $f\big(\mathcal{X}_k^{(\tau(j))}\big|\{(\boldsymbol{x}_{k-1}^{(j)}, l_{k-1}^{(j)})\}\big)$ is $f\big(\mathcal{X}_{k,j}^{\mathrm{d}}\big|\{(\boldsymbol{x}_{k-1}^{(j)}, l_{k-1}^{(j)})\}\big)$ in (92) evaluated at $\mathcal{X}_{k,j}^{\mathrm{d}} = \mathcal{X}_k^{(\tau(j))} = \{(\boldsymbol{x}_k^{(i)}, l^{(i)}) \in \mathcal{X}_k^{\mathrm{d}} : \tau(j) = l^{(i)}\}$. (We recall that $\tau(\cdot)$ is the track-labeling function of the LMB RFS introduced in Section X-B.) As discussed after (90), for sets $\mathcal{X}_k^{\mathrm{d}}$ with nondistinct labels, $f(\mathcal{X}_k^{\mathrm{d}}|\mathcal{X}_{k-1}^{\mathrm{d}}) = 0$. At $k=0$, we have $\mathcal{X}_0^{\mathrm{d}} = \emptyset$.

3) Undetected Targets: Let $\mathcal{X}_{k-1}^{\mathrm{u}} \triangleq \{(\boldsymbol{x}_{k-1}^{(1)}, 0), \ldots, (\boldsymbol{x}_{k-1}^{(i_{k-1}^{\mathrm{u}})}, 0)\}$ comprise the states of the undetected targets that existed at time $k-1$. Because of A1 and S3, each undetected target j that existed at time $k-1$ gives rise to a—possibly nonexistent—survived target at time k. Again, the state of this (undetected) legacy target is modeled as a Bernoulli RFS $\mathsf{X}_{k,j}^{\mathrm{u}}$ with pdf $f\big(\mathcal{X}_{k,j}^{\mathrm{u}}\big|\{(\boldsymbol{x}_{k-1}^{(j)}, 0)\}\big)$ given by (92) (with obvious modifications). Furthermore, there may be targets newly appearing at time k. The entirety of the states of these newborn targets will be described by an RFS $\mathsf{\Gamma}_k$. Because of S6 and S7, $\mathsf{\Gamma}_k$ is a Poisson RFS with mean μ_{b} and spatial pdf $f_{\mathrm{b}}(\boldsymbol{x}_k)$, and all elements have label $l_k = 0$. Thus, the intensity function of $\mathsf{\Gamma}_k$ is $\lambda_{\mathrm{b}}(\boldsymbol{x}_k, l_k) = \mu_{\mathrm{b}} f_{\mathrm{b}}(\boldsymbol{x}_k) 1(l_k)$. With S4 and S5, the overall multitarget state RFS of the undetected targets at time k, $\mathsf{X}_k^{\mathrm{u}} = \{(\mathbf{x}_k^{(1)}, 0), \ldots, (\mathbf{x}_k^{(\mathsf{i}_k^{\mathrm{u}})}, 0)\}$, is then obtained as

$$\mathsf{X}_k^{\mathrm{u}} = \left(\bigcup_{j=1}^{i_{k-1}^{\mathrm{u}}} \mathsf{X}_{k,j}^{\mathrm{u}}\right) \cup \mathsf{\Gamma}_k. \tag{95}$$

Here, because of A2 and S5, the components $\mathsf{X}_{k,j}^{\mathrm{u}}$ and $\mathsf{\Gamma}_k$ are all conditionally independent given $\mathsf{X}_{k-1}^{\mathrm{u}}$.

The joint evolution of the undetected target states is described by the multitarget state-transition pdf $f(\mathcal{X}_k^{\mathrm{u}}|\mathcal{X}_{k-1}^{\mathrm{u}})$, where $\mathcal{X}_k^{\mathrm{u}} = \{(\boldsymbol{x}_k^{(1)}, 0), \ldots, (\boldsymbol{x}_k^{(i_k^{\mathrm{u}})}, 0)\}$. Let α be a mapping of an index $j \in \{1, \ldots, i_{k-1}^{\mathrm{u}}\}$ to an index $\alpha(j) \in \{0, \ldots, i_k^{\mathrm{u}}\}$. This mapping is such that $j_1 \neq j_2$ with $\alpha(j_1), \alpha(j_2) \in \{1, \ldots, m\}$ implies $\alpha(j_1) \neq \alpha(j_2)$. Furthermore, let $\mathcal{M}_{i_{k-1}^{\mathrm{u}}, i_k^{\mathrm{u}}}$ be the set of all possible mappings α. Using (95), (92), and (89) together with S6 and S7, one obtains [3, Ch. 13]

$$\begin{aligned} &f(\mathcal{X}_k^{\mathrm{u}}|\mathcal{X}_{k-1}^{\mathrm{u}}) \\ &= e^{-\mu_{\mathrm{b}}} \left(\prod_{i=1}^{i_k^{\mathrm{u}}} \mu_{\mathrm{b}} f_{\mathrm{b}}(\boldsymbol{x}_k^{(i)})\right) \left(\prod_{j=1}^{i_{k-1}^{\mathrm{u}}} \big(1 - p_{\mathrm{s}}(\boldsymbol{x}_{k-1}^{(j)})\big)\right) \\ &\quad \times \sum_{\alpha \in \mathcal{M}_{i_{k-1}^{\mathrm{u}}, i_k^{\mathrm{u}}}} \prod_{j': \alpha(j') > 0} \frac{p_{\mathrm{s}}(\boldsymbol{x}_{k-1}^{(j')}) f(\boldsymbol{x}_k^{(\alpha(j'))}|\boldsymbol{x}_{k-1}^{(j')})}{\mu_{\mathrm{b}} \big(1 - p_{\mathrm{s}}(\boldsymbol{x}_{k-1}^{(j')})\big) f_{\mathrm{b}}(\boldsymbol{x}_k^{(\alpha(j'))})}. \end{aligned} \tag{96}$$

An analysis of this expression shows that for $\alpha(j') > 0$, target j' survives and its state $\boldsymbol{x}_{k-1}^{(j')}$ is mapped to $\boldsymbol{x}_k^{(\alpha(j'))}$, whereas for $\alpha(j') = 0$, target j does not survive. At $k=0$, due to S8 and S9, $\mathsf{X}_0^{\mathrm{u}}$ is a Poisson RFS with mean μ_{p} and spatial pdf $f_{\mathrm{p}}(\boldsymbol{x}_0)$, and all elements have label $l_0 = 0$.

B. Likelihood Function

The entirety of the measurements of sensor s at time k is modeled as a time-varying RFS $\mathsf{Z}_{k,s} \triangleq \{\mathbf{z}_{k,s}^{(1)}, \ldots, \mathbf{z}_{k,s}^{(\mathsf{m}_{k,s})}\}$. Let $\mathcal{X}_k = \{(\boldsymbol{x}_k^{(1)}, l_k^{(1)}), \ldots, (\boldsymbol{x}_k^{(i_k)}, l_k^{(i_k)})\}$ be the set of the states of the targets at time k. Based on A5 and A9, a—possibly nonexistent—measurement $\mathbf{z}_{k,s}$ originating from target j is modeled as a Bernoulli RFS $\mathsf{Z}_{k,s,j}$ (cf. (88)) with $r = p_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k^{(j)})$ and $s(\boldsymbol{z}_{k,s}) = f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k^{(j)})$. Both r and $s(\boldsymbol{z}_{k,s})$ do not dependent on the target label. The associated RFS likelihood function is thus given by

$$f\big(\mathcal{Z}_{k,s,j}\big|\{\boldsymbol{x}_k^{(j)}\}\big) = \begin{cases} 1 - p_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k^{(j)}), & \mathcal{Z}_{k,s,j} = \emptyset, \\ p_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k^{(j)}) f(\boldsymbol{z}_{k,s}|\boldsymbol{x}_k^{(j)}), & \mathcal{Z}_{k,s,j} = \{\boldsymbol{z}_{k,s}\}, \\ 0, & \text{otherwise.} \end{cases} \tag{97}$$

Because of A4 and A9, the entirety of the target-originated measurements then forms the MB RFS $\bigcup_{j=1}^{i_k} \mathsf{Z}_{k,s,j}$. Furthermore, the entirety of the clutter-originated measurements is described by an RFS $\mathsf{\Lambda}_{k,s}$. Because of A6 and A8, $\mathsf{\Lambda}_{k,s}$ is a Poisson RFS with mean $\mu_{\mathrm{c}}^{(s)}$ and spatial pdf $f_{\mathrm{c}}^{(s)}(\boldsymbol{z}_{k,s})$ and, thus, intensity function $\lambda_{\mathrm{c}}^{(s)}(\boldsymbol{z}_{k,s}) = \mu_{\mathrm{c}}^{(s)} f_{\mathrm{c}}^{(s)}(\boldsymbol{z}_{k,s})$. Using A3, A4, A8, and A11, the overall measurement RFS at sensor s and time k, $\mathsf{Z}_{k,s} = \{\mathbf{z}_{k,s}^{(1)}, \ldots, \mathbf{z}_{k,s}^{(\mathsf{m}_{k,s})}\}$, is then obtained as

$$\mathsf{Z}_{k,s} = \left(\bigcup_{j=1}^{i_k} \mathsf{Z}_{k,s,j}\right) \cup \mathsf{\Lambda}_{k,s}. \tag{98}$$

Here, because of A7, A8, and A9, the components $\mathsf{Z}_{k,s,j}$ and $\mathsf{\Lambda}_{k,s}$ are all conditionally independent given X_k.

The dependence of $\mathsf{Z}_{k,s}$ on the multitarget state RFS $\mathsf{X}_k = \{(\mathbf{x}_k^{(1)}, \mathsf{l}_k^{(1)}), \ldots, (\mathbf{x}_k^{(\mathsf{i}_k)}, \mathsf{l}_k^{(\mathsf{i}_k)})\}$ is described by the single-sensor multitarget likelihood function $f(\mathcal{Z}_{k,s}|\mathcal{X}_k)$. Using (98) and (97) together with A6–A9, one obtains [3, Ch. 12]

$$\begin{aligned} f(\mathcal{Z}_{k,s}|\mathcal{X}_k) &= e^{-\mu_{\mathrm{c}}^{(s)}} \left(\prod_{m=1}^{m_{k,s}} \mu_{\mathrm{c}}^{(s)} f_{\mathrm{c}}^{(s)}(\boldsymbol{z}_{k,s}^{(m)})\right) \\ &\quad \times \sum_{\alpha \in \mathcal{M}_{i_k^{\mathrm{u}}, m_{k,s}}} \prod_{j=1}^{i_k} g\big(\boldsymbol{x}_k^{(j)}, \alpha(j); \boldsymbol{z}_{k,s}\big), \end{aligned} \tag{99}$$

where $g\big(\boldsymbol{x}_k^{(j)}, \alpha(j); \boldsymbol{z}_{k,s}\big)$ is as in (14) with $a_{k,s}^{(i)}$ replaced by $\alpha(j)$. An analysis of this expression shows that for $\alpha(j) > 0$, target j (with state $\boldsymbol{x}_k^{(j)}$) generates measurement

$\boldsymbol{z}_{k,s}^{(\alpha(j))}$, whereas for $\alpha(j)=0$, target j does not generate a measurement at sensor s. Note that due to missed detections and clutter measurements, the number of measurements $m_{k,s}$ may also be smaller or larger than the number of targets i_k. According to (99), the measurements do not provide information about the target labels. (If the labels could be measured as well, there would not be a DA uncertainty.) Comparing (99) with (17), it can be shown that for i_k fixed,[8] $f(\mathcal{Z}_{k,s}|\mathcal{X}_k) = m_{k,s}!\, f(\boldsymbol{z}_{k,s}, m_{k,s}|\boldsymbol{x}_k)$. This is not surprising as the assumptions underlying the vector-type and set-type measurement models (i.e., A3–A9) are equal.

Finally, using A10, the multisensor multitarget likelihood function is obtained as

$$f(\mathcal{Z}_k|\mathcal{X}_k) = \prod_{s=1}^{n_s} f(\mathcal{Z}_{k,s}|\mathcal{X}_k), \tag{100}$$

where $\mathcal{Z}_k \triangleq (\mathcal{Z}_{k,1}, \ldots, \mathcal{Z}_{k,n_s})$ is an ordered list of the sets $\mathcal{Z}_{k,s}$. For later use, we also introduce the ordered list $\mathcal{Z}_{1:k} \triangleq (\mathcal{Z}_1, \ldots, \mathcal{Z}_k)$.

XII. Set-Type Multitarget State Propagation

In this section, extending [113] and [51], we develop prediction and update steps for the set-type system model from Section XI. The prediction step and an approximation of the update step will be used in Section XIII to devise a hybrid labeled/unlabeled variant of the TOMB/P filter proposed in [51], [113]. We consider the single-sensor case ($n_s=1$) and thus drop the sensor index s. We note that two alternative derivations of our results are provided in [51] and [152].

According to (91), the multitarget state RFS X_k is the union of the detected target RFS $\mathsf{X}_k^{\mathrm{d}}$ and the undetected target RFS $\mathsf{X}_k^{\mathrm{u}}$. It can be shown [51] that the posterior pdfs of $\mathsf{X}_k^{\mathrm{d}}$ and $\mathsf{X}_k^{\mathrm{u}}$ can be propagated in parallel, in both cases by carrying out a prediction step and an update step. While the two prediction steps are performed completely separately, the update step for $\mathsf{X}_k^{\mathrm{d}}$ involves the prediction results for both $\mathsf{X}_k^{\mathrm{d}}$ and $\mathsf{X}_k^{\mathrm{u}}$. We note that explicit propagation of the pdf of the undetected targets $\mathsf{X}_k^{\mathrm{u}}$ enables the consideration of targets that were born at an earlier time but so far have not been detected by any sensor. This ability is beneficial especially if the detection probabilities $p_{\mathrm{d}}(\boldsymbol{x}_k)$ are low, time-varying, or nonuniform (i.e., different for different values of $\boldsymbol{x}_k$).

In the following development of the prediction and update steps, we use the fact that the posterior pdf of the overall multitarget state RFS X_{k-1} at time $k-1$ factorizes as

$$f(\mathcal{X}_{k-1}|\mathcal{Z}_{1:k-1}) = f(\mathcal{X}_{k-1}^{\mathrm{d}}|\mathcal{Z}_{1:k-1})\, f(\mathcal{X}_{k-1}^{\mathrm{u}}|\mathcal{Z}_{1:k-1}). \tag{101}$$

Here, according to Section X-B, $\mathsf{X}_{k-1}^{\mathrm{d}}$ is an LMB RFS consisting of $|\mathcal{L}_{k-1}|$ Bernoulli components, where $\mathcal{L}_{k-1}$ is a set of nonzero labels. Furthermore, $\mathsf{X}_{k-1}^{\mathrm{u}}$ is a Poisson RFS with mean parameter μ_{k-1}^{u}, spatial pdf $f_{\mathrm{u}}(\boldsymbol{x}_{k-1})$, dummy label l_{k-1} (which is 0) and, thus, intensity function $\lambda_{k-1}^{\mathrm{u}}(\boldsymbol{x}_{k-1}, l_{k-1}) = \mu_{k-1}^{\mathrm{u}} f_{\mathrm{u}}(\boldsymbol{x}_{k-1}) 1(l_{k-1})$. The sets $\mathcal{X}_{k-1}^{\mathrm{d}}$ and $\mathcal{X}_{k-1}^{\mathrm{u}}$ in (101) are given by $\mathcal{X}_{k-1}^{\mathrm{d}} = \{(\boldsymbol{x}_{k-1}, l_{k-1}) \in \mathcal{X}_{k-1} : l_{k-1} \neq 0\}$ and $\mathcal{X}_{k-1}^{\mathrm{u}} = \{(\boldsymbol{x}_{k-1}, l_{k-1}) \in \mathcal{X}_{k-1} : l_{k-1} = 0\}$. Indeed, for $k-1=0$, the form in (101) is a consequence of the initial condition defined by S8 and S9, which implies that at $k=0$, $\mathsf{X}_0 = \mathsf{X}_0^{\mathrm{u}}$ is a Poisson RFS with intensity function $\lambda_0^{\mathrm{u}}(\boldsymbol{x}_0, l_0) = \mu_{\mathrm{p}} f_{\mathrm{p}}(\boldsymbol{x}_0) 1(l_0)$. For $k-1 \geqslant 1$, it can be shown that the form in (101) is preserved by the prediction and update steps presented in what follows, and therefore it is valid for all values of $k-1$. (We note that a partial proof of this fact is provided in [51], [152].)

For simplicity of notation, we will index the existence probabilities and spatial pdfs of the Bernoulli components directly by their labels (assigned upon first detection), i.e., we write them as $r_{k-1}^{(l)}$ and $s_{k-1}^{(l)}(\boldsymbol{x}_{k-1})$ rather than $r_{k-1}^{(\tau^{-1}(l))}$ and $s_{k-1}^{(\tau^{-1}(l))}(\boldsymbol{x}_{k-1})$, respectively, with $l \in \mathcal{L}_{k-1}$.

A. Prediction Step

In the prediction step, the preceding posterior pdf $f(\mathcal{X}_{k-1}|\mathcal{Z}_{1:k-1})$ is converted into a predicted posterior pdf $f(\mathcal{X}_k|\mathcal{Z}_{1:k-1})$ via the RFS version of the Chapman-Kolmogorov equation (cf. (1))

$$f(\mathcal{X}_k|\mathcal{Z}_{1:k-1}) = \int f(\mathcal{X}_k|\mathcal{X}_{k-1})\, f(\mathcal{X}_{k-1}|\mathcal{Z}_{1:k-1})\, \delta\mathcal{X}_{k-1}.$$

The posterior pdf of the detected targets, $f(\mathcal{X}_k^{\mathrm{d}}|\mathcal{Z}_{1:k})$, and that of the undetected targets, $f(\mathcal{X}_k^{\mathrm{u}}|\mathcal{Z}_{1:k}) = f(\mathcal{X}_k^{\mathrm{u}})$, can be predicted separately. (Note that $\mathsf{X}_k^{\mathrm{u}}$ is independent of $\mathsf{Z}_{1:k}$, but impacted by the characteristics of the detection process—in particular, by $p_{\mathrm{d}}(\boldsymbol{x}_k)$.) These predictions involve the state-transition pdfs (94) and (96), respectively.

More specifically, $f(\mathcal{X}_k^{\mathrm{d}}|\mathcal{Z}_{1:k-1})$ is again an LMB pdf, with the Bernoulli parameters $r_{k|k-1}^{(l)}$ and $s_{k|k-1}^{(l)}(\boldsymbol{x}_k)$ given by

$$r_{k|k-1}^{(l)} = r_{k-1}^{(l)} \int p_{\mathrm{s}}(\boldsymbol{x}_{k-1})\, s_{k-1}^{(l)}(\boldsymbol{x}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1} \tag{102}$$

$$s_{k|k-1}^{(l)}(\boldsymbol{x}_k) = \frac{\int f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\, p_{\mathrm{s}}(\boldsymbol{x}_{k-1})\, s_{k-1}^{(l)}(\boldsymbol{x}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}}{\int p_{\mathrm{s}}(\boldsymbol{x}_{k-1})\, s_{k-1}^{(l)}(\boldsymbol{x}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}}, \tag{103}$$

for $l \in \mathcal{L}_{k-1}$. If at time k–1 the existence of a target is perfectly known, i.e., $r_{k-1}^{(l)} = 1$, and if also $p_{\mathrm{s}}(\boldsymbol{x}_{k-1}) = 1$, and thus $r_{k|k-1}^{(l)} = 1$ according to (102), then the spatial-pdf prediction in (103) simplifies to the conventional prediction for a single target performed by the JPDA filter (see (38)). In the general case, (102) and (103) correspond to the vector prediction (76) (here, $r_{k|k-1}^{(l)} s_{k|k-1}^{(l)}(\boldsymbol{x}_k)$ corresponds to $\alpha_k(\boldsymbol{x}_k, 1)$).

Furthermore, $f(\mathcal{X}_k^{\mathrm{u}}|\mathcal{Z}_{1:k-1})$ is again a Poisson pdf, whose state intensity function is given by

$$\lambda_{k|k-1}^{\mathrm{u}}(\boldsymbol{x}_k, l_k) = 1(l_k) \int f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\, p_{\mathrm{s}}(\boldsymbol{x}_{k-1}) \times \lambda_{k-1}^{\mathrm{u}}(\boldsymbol{x}_{k-1}, l_{k-1}=0)\, \mathrm{d}\boldsymbol{x}_{k-1} + \lambda_{\mathrm{b}}(\boldsymbol{x}_k, l_k).$$

This prediction step converting $\lambda_{k-1}^{\mathrm{u}}(\boldsymbol{x}_{k-1}, l_{k-1})$ into $\lambda_{k|k-1}^{\mathrm{u}}(\boldsymbol{x}_k, l_k)$ is equivalent to the prediction step of the PHD filter [111]. The labels of the elements of $f(\mathcal{X}_k^{\mathrm{u}}|\mathcal{Z}_{1:k-1})$ are again $l_k = 0$. It is important to note that the entire prediction step preserves the LMB–Poisson model assumed for X_{k-1}.

[8] Here, the factor $m_{k,s}!$ is related to the fact that $f(\mathcal{Z}_{k,s}|\mathcal{X}_k)$ integrates to one using the set integral (87) whereas $f(\boldsymbol{z}_{k,s}, m_{k,s}|\boldsymbol{x}_k)$ integrates to one using conventional integration.

B. Update Step

In the update step, the predicted posterior pdf $f(\mathcal{X}_k|\mathcal{Z}_{1:k-1})$ is converted into the posterior pdf $f(\mathcal{X}_k|\mathcal{Z}_{1:k})$ via the RFS version of Bayes' rule (cf. (2))

$$f(\mathcal{X}_k|\mathcal{Z}_{1:k}) \propto f(\mathcal{Z}_k|\mathcal{X}_k)\, f(\mathcal{X}_k|\mathcal{Z}_{1:k-1}).$$

Here, the single-sensor multitarget likelihood function $f(\mathcal{Z}_k|\mathcal{X}_k)$ (see (99)) incorporates the current measurement set $\mathcal{Z}_k$.

1) Update for the Detected Targets: The posterior pdf describing the detected targets is given by [51]

$$f(\mathcal{X}_k^{\mathrm{d}}|\mathcal{Z}_{1:k}) = \sum_{\boldsymbol{c}_k} p(\boldsymbol{c}_k)\, f_{\boldsymbol{c}_k}^{\mathrm{LMB}}(\mathcal{X}_k^{\mathrm{d}}) \tag{104}$$

$$= \sum_{\boldsymbol{c}_k} p(\boldsymbol{c}_k) \prod_{l\in\mathcal{L}_k} f^{(l,c_k^{(l)})}\big(\mathcal{X}_k^{(l)}\big), \tag{105}$$

where $\mathcal{X}_k^{\mathrm{d}} = \big\{\big(\boldsymbol{x}_k^{(1)}, l^{(1)}\big), \ldots, \big(\boldsymbol{x}_k^{(i_k^{\mathrm{d}})}, l^{(i_k^{\mathrm{d}})}\big)\big\}$, $\boldsymbol{c}_k$ is a DA vector with entries $c_k^{(l)}$, $l \in \mathcal{L}_k$ and pmf $p(\boldsymbol{c}_k)$ (see below), $f^{(l,c_k^{(l)})}\big(\mathcal{X}_k^{(l)}\big)$ is a Bernoulli pdf, and $\mathcal{X}_k^{(l)} = \big\{(\boldsymbol{x}_k^{(i)}, l^{(i)}) \in \mathcal{X}_k^{\mathrm{d}} : l^{(i)} = l\big\}$. According to (90), $f_{\boldsymbol{c}_k}^{\mathrm{LMB}}(\mathcal{X}_k^{\mathrm{d}}) = \prod_{l\in\mathcal{L}_k} f^{(l,c_k^{(l)})}\big(\mathcal{X}_k^{(l)}\big)$ is an LMB pdf. Thus, (104) means that $f(\mathcal{X}_k^{\mathrm{d}}|\mathcal{Z}_{1:k})$ is a mixture of LMB pdfs, where each LMB pdf is indexed by $\boldsymbol{c}_k$, weighted by $p(\boldsymbol{c}_k)$, and consists of $|\mathcal{L}_k|$ Bernoulli components. The label set is updated as

$$\mathcal{L}_k = \mathcal{L}_{k-1} \cup \mathcal{L}_k^{\mathrm{new}}, \tag{106}$$

where $\mathcal{L}_{k-1}$ corresponds to legacy components that are taken over from time $k-1$ and $\mathcal{L}_k^{\mathrm{new}}$ corresponds to new components that were undetected so far and generate a measurement for the first time. Note that $\mathcal{L}_{k-1} \cap \mathcal{L}_k^{\mathrm{new}} = \emptyset$ and thus $|\mathcal{L}_k| = |\mathcal{L}_{k-1}| + |\mathcal{L}_k^{\mathrm{new}}|$. For each measurement $m \in \{1, \ldots, m_k\}$, a new Bernoulli component is created, to which the unique label $l \triangleq (k,m) \in \mathcal{L}_k^{\mathrm{new}}$ is assigned. Thus, in particular, $|\mathcal{L}_k^{\mathrm{new}}| = m_k$.

The legacy and new Bernoulli components are the set-type counterparts of, respectively, the legacy and new PTs used in the vector-type system model described in Section VIII. In the vector-type system model, the number of PTs increased roughly linearly with time k (see (52)). As evidenced by (106), the same increase is now exhibited by the number of Bernoulli components $f^{(l,c_k^{(l)})}\big(\mathcal{X}_k^{(l)}\big)$ in each LMB pdf $f_{\boldsymbol{c}_k}^{\mathrm{LMB}}(\mathcal{X}_k^{\mathrm{d}})$. In addition, the number of possible association vectors $\boldsymbol{c}_k$ and, in turn, the number of different LMB pdfs $f_{\boldsymbol{c}_k}^{\mathrm{LMB}}(\mathcal{X}_k^{\mathrm{d}})$ increases exponentially. These issues will be addressed in Section XIII-A.

It will be convenient to partition the DA vector $\mathbf{c}_k = \big[\mathsf{c}_k^{(l)}\big]_{l\in\mathcal{L}_k}$ in (104), (105) as $\mathbf{c}_k \triangleq \big[\mathbf{a}_k^{\mathsf{T}}\ \mathbf{b}_k^{\mathsf{T}}\big]^{\mathsf{T}}$, with the "legacy component" DA vector $\mathbf{a}_k = \big[\mathsf{a}_k^{(l)}\big]_{l\in\mathcal{L}_{k-1}}$ and the "new component" DA vector $\mathbf{b}_k = \big[\mathsf{b}_k^{(l)}\big]_{l\in\mathcal{L}_k^{\mathrm{new}}}$. The entries of these vectors are defined as follows. For $l \in \mathcal{L}_{k-1}$, $\mathsf{c}_k^{(l)} = \mathsf{a}_k^{(l)}$ with

$$\mathsf{a}_k^{(l)} \triangleq \begin{cases} m \in \{1,\ldots,m_k\}, & \text{if at time } k, \text{ legacy component } l \text{ generates measurement } m \\ 0, & \text{if at time } k, \text{ legacy component } l \text{ does not generate a measurement.} \end{cases}$$

For $l \in \mathcal{L}_k^{\mathrm{new}}$, $\mathsf{c}_k^{(l)} = \mathsf{b}_k^{(l)}$ with

$$\mathsf{b}_k^{(l)} \triangleq \begin{cases} l' \in \mathcal{L}_{k-1}, & \text{if at time } k, \text{ measurement } m \text{ is generated by legacy component } l' \\ 0, & \text{if at time } k, \text{ measurement } m \text{ is not generated by a legacy component.} \end{cases}$$

(We recall that $l = (k,m)$ for $l \in \mathcal{L}_k^{\mathrm{new}}$.) These definitions of $\mathsf{a}_k^{(l)}$ and $\mathsf{b}_k^{(l)}$ are similar to (10) and (20), respectively. Note that $\boldsymbol{c}_k \in \{0,\ldots,m_k\}^{|\mathcal{L}_{k-1}|} \times (\{0\} \cup \mathcal{L}_{k-1})^{m_k}$. To guarantee that $p(\boldsymbol{c}_k) = 0$ for all $\boldsymbol{c}_k = \big[\boldsymbol{a}_k^{\mathsf{T}}\ \boldsymbol{b}_k^{\mathsf{T}}\big]^{\mathsf{T}}$ that violate A4, similarly as in (21) and (22), we introduce the indicator function

$$\psi(\boldsymbol{a}_k, \boldsymbol{b}_k) = \prod_{l\in\mathcal{L}_{k-1}} \prod_{m=1}^{m_k} \Psi_{l,m}\big(a_k^{(l)}, b_k^{(k,m)}\big),$$

with

$$\Psi_{l,m}\big(a_k^{(l)}, b_k^{(k,m)}\big) \triangleq \begin{cases} 0, & a_k^{(l)} = m,\ b_k^{(k,m)} \neq l \\ & \text{or } b_k^{(k,m)} = l,\ a_k^{(l)} \neq m \\ 1, & \text{otherwise.} \end{cases}$$

The DA pmf $p(\boldsymbol{c}_k)$ is now given by

$$p(\boldsymbol{c}_k) = p(\boldsymbol{a}_k, \boldsymbol{b}_k) \propto \psi(\boldsymbol{a}_k, \boldsymbol{b}_k) \left(\prod_{l\in\mathcal{L}_{k-1}} \beta_k^{(l)}\big(a_k^{(l)}\big) \right) \prod_{l'\in\mathcal{L}_k^{\mathrm{new}}} \xi_k^{(l')}\big(b_k^{(l')}\big), \tag{107}$$

where the association weights $\beta_k^{(l)}(\cdot)$ and $\xi_k^{(l)}(\cdot)$ are calculated as follows. For $l \in \mathcal{L}_{k-1}$, we have for $a_k^{(l)} = m \in \{1,\ldots,m_k\}$

$$\beta_k^{(l)}(m) = r_{k|k-1}^{(l)} C_k^{(l)}\big(\boldsymbol{z}_k^{(m)}\big),$$

with $C_k^{(l)}\big(\boldsymbol{z}_k^{(m)}\big) \triangleq \int f\big(\boldsymbol{z}_k^{(m)}\big|\boldsymbol{x}_k\big)\, p_{\mathrm{d}}(\boldsymbol{x}_k)\, s_{k|k-1}^{(l)}(\boldsymbol{x}_k)\, \mathrm{d}\boldsymbol{x}_k$, and for $a_k^{(l)} = 0$

$$\beta_k^{(l)}(0) = 1 - r_{k|k-1}^{(l)} D_k^{(l)},$$

with $D_k^{(l)} \triangleq \int p_{\mathrm{d}}(\boldsymbol{x}_k)\, s_{k|k-1}^{(l)}(\boldsymbol{x}_k)\, \mathrm{d}\boldsymbol{x}_k$. For $l = (k,m) \in \mathcal{L}_k^{\mathrm{new}}$, we have $\xi_k^{(l)}\big(b_k^{(l)}\big) = 1$ if $b_k^{(l)} \neq 0$, and

$$\xi_k^{(l=(k,m))}(0) = C_k^{\mathrm{u}}\big(\boldsymbol{z}_k^{(m)}\big) + \lambda_{\mathrm{c}}\big(\boldsymbol{z}_k^{(m)}\big)$$

if $b_k^{(l)} = 0$, with $C_k^{\mathrm{u}}\big(\boldsymbol{z}_k^{(m)}\big) \triangleq \int f\big(\boldsymbol{z}_k^{(m)}\big|\boldsymbol{x}_k\big) p_{\mathrm{d}}(\boldsymbol{x}_k) \lambda_{k|k-1}^{\mathrm{u}}(\boldsymbol{x}_k, l_k = 0)\, \mathrm{d}\boldsymbol{x}_k$ and $\lambda_{\mathrm{c}}\big(\boldsymbol{z}_k^{(m)}\big) = \mu_{\mathrm{c}} f_{\mathrm{c}}\big(\boldsymbol{z}_k^{(m)}\big)$. It can be shown that for $p_{\mathrm{d}}(\boldsymbol{x}_k)\lambda_{k|k-1}^{u}(\boldsymbol{x}_k, l_k = 0) = \mu_{\mathrm{n}} f_{\mathrm{n}}(\boldsymbol{x}_k)$, the joint marginal DA pmf $p(\boldsymbol{c}_k) = p(\boldsymbol{a}_k, \boldsymbol{b}_k)$ in (107) equals the belief $\tilde{p}(\boldsymbol{a}_k, \boldsymbol{b}_k)$ in (80), i.e., the belief calculated by the total-SPA method for an unknown number of targets.

Each Bernoulli component $f^{(l,c_k^{(l)})}\big(\mathcal{X}_k^{(l)}\big)$ in (105) is parameterized by an existence probability $r_k^{(l,c_k^{(l)})}$ and a spatial pdf $s_k^{(l,c_k^{(l)})}(\boldsymbol{x}_k)$. These parameters are calculated as follows. Let us first consider the legacy components, i.e., $l \in \mathcal{L}_{k-1}$, where $c_k^{(l)} = a_k^{(l)}$. Here, the existence probability is given for $a_k^{(l)} \in \{1,\ldots,m_k\}$ by $r_k^{(l,a_k^{(l)})} = 1$ (this means that legacy component l corresponds to an existing target that generated

measurement $m = a_k^{(l)}$) and for $a_k^{(l)} = 0$ by

$$r_k^{(l,0)} = \frac{r_{k|k-1}^{(l)}\big(1 - D_k^{(l)}\big)}{1 - r_{k|k-1}^{(l)} D_k^{(l)}}. \tag{108}$$

The spatial pdf is given for all $a_k^{(l)} \in \{0, \ldots, m_k\}$ by

$$s_k^{(l,a_k^{(l)})}(\boldsymbol{x}_k) = \frac{g\big(\boldsymbol{x}_k, a_k^{(l)}; \boldsymbol{z}_k\big)\, s_{k|k-1}^{(l)}(\boldsymbol{x}_k)}{\int g\big(\boldsymbol{x}'_k, a_k^{(l)}; \boldsymbol{z}_k\big)\, s_{k|k-1}^{(l)}(\boldsymbol{x}'_k)\, \mathrm{d}\boldsymbol{x}'_k}, \tag{109}$$

where $g\big(\boldsymbol{x}_k, a_k^{(l)}; \boldsymbol{z}_k\big)$ was defined in (14). Next, we consider the new components, i.e., $l = (k,m) \in \mathcal{L}_k^{\text{new}}$, where $c_k^{(l)} = b_k^{(l)}$. Here, the existence probability is given for $b_k^{(l)} = 0$ by

$$r_k^{(l=(k,m),0)} = \frac{C_k^{\text{u}}\big(\boldsymbol{z}_k^{(m)}\big)}{C_k^{\text{u}}\big(\boldsymbol{z}_k^{(m)}\big) + \lambda_{\text{c}}\big(\boldsymbol{z}_k^{(m)}\big)} \tag{110}$$

and for $b_k^{(l)} \neq 0$ by $r_k^{(l,b_k^{(l)})} = 0$ (this means that measurement m was generated by a legacy component). The spatial pdf is given for $b_k^{(l)} = 0$ by

$$s_k^{(l=(k,m),0)}(\boldsymbol{x}_k) = \frac{f\big(\boldsymbol{z}_k^{(m)} \big| \boldsymbol{x}_k\big)\, p_{\text{d}}(\boldsymbol{x}_k)\, \lambda_{k|k-1}^{\text{u}}(\boldsymbol{x}_k, l_k = 0)}{C_k^{\text{u}}\big(\boldsymbol{z}_k^{(m)}\big)}, \tag{111}$$

whereas for $b_k^{(l)} \neq 0$, $s_k^{(l,b_k^{(l)})}(\boldsymbol{x}_k)$ does not need to be defined (since $r_k^{(l,b_k^{(l)})} = 0$).

2) Update for the Undetected Targets: We recall that for the targets that remain undetected, the posterior pdf $f(\mathcal{X}_k^{\text{u}}|\mathcal{Z}_{1:k}) = f(\mathcal{X}_k^{\text{u}})$ does not involve $\mathcal{Z}_{1:k}$ (although it involves $p_{\text{d}}(\boldsymbol{x}_k)$). The update step for these targets yields again a Poisson pdf $f(\mathcal{X}_k^{\text{u}})$, whose intensity function is calculated as

$$\lambda_k^{\text{u}}(\boldsymbol{x}_k, l_k) = \big(1 - p_{\text{d}}(\boldsymbol{x}_k)\big)\, \lambda_{k|k-1}^{\text{u}}(\boldsymbol{x}_k, l_k).$$

This is equivalent to the update step of the PHD filter [111] for the case where the sensor does not produce any measurements. The labels of all elements of $\mathcal{X}_k^{\text{u}}$ are again zero.

XIII. SPA-Based Set-Type MTT Methods

Building on the set-type system model from Section XI and the set-type multitarget state propagation rules from Section XII, we will now develop a hybrid labeled/unlabeled variant of the TOMB/P filter proposed in [51], [113] (cf. Section II-B2). This labeled/unlabeled TOMB/P filter represents the joint state of the detected targets by a labeled RFS—more specifically, an LMB RFS—and is thus able to maintain track continuity, i.e., to estimate entire target trajectories.

Whereas the prediction step presented in Section XII-A and the update step for the undetected targets presented in Section XII-B2 preserve the LMB–Poisson model used for X_k in Section XII, the update step for the detected targets presented in Section XII-B1 converts the LMB RFS describing the detected targets into a mixture of LMB RFSs. In order to reobtain an LMB pdf also for the detected targets, we will use an approximation that is similar to the one used in the derivation of the JPDA filter. The marginal posterior DA pmfs are then calculated by SPADA. As in the vector-type MTT methods discussed in Section VII, this is key to obtaining scalability of the resulting filter. We first consider the single-sensor case ($n_{\text{s}} = 1$) and later extend to multiple sensors. A detailed derivation of our results and a closed-form implementation for linear-Gaussian models can be found in [51], [152], and a particle-based implementation for nonlinear and non-Gaussian models is presented in [107].

A. Approximate Update Step for the Detected Targets

According to (105) and (106), the update step for the detected targets produces a mixture of LMB pdfs, and its computational complexity scales exponentially with the number $|\mathcal{L}_k|$ of Bernoulli components and the number m_k of measurements. To address these issues, we approximate the joint DA pmf $p(\boldsymbol{c}_k)$ by the product of its marginals, i.e.,

$$p(\boldsymbol{c}_k) \approx \prod_{l \in \mathcal{L}_k} p\big(c_k^{(l)}\big), \quad \text{with } p\big(c_k^{(l)}\big) = \sum_{\sim c_k^{(l)}} p(\boldsymbol{c}_k). \tag{112}$$

Note that a similar approximation was used in the derivation of the JPDA filter, cf. (42). Inserting (112) in (105) yields

$$\begin{aligned} f(\mathcal{X}_k^{\text{d}}|\mathcal{Z}_{1:k}) &\approx \prod_{l \in \mathcal{L}_k} \sum_{c_k^{(l)}} p\big(c_k^{(l)}\big)\, f^{(l,c_k^{(l)})}\big(\mathcal{X}_k^{(l)}\big) \\ &= \prod_{l \in \mathcal{L}_k} f^{(l)}\big(\mathcal{X}_k^{(l)}\big). \end{aligned} \tag{113}$$

Here, $f^{(l)}\big(\mathcal{X}_k^{(l)}\big) \triangleq \sum_{c_k^{(l)}} p\big(c_k^{(l)}\big)\, f^{(l,c_k^{(l)})}\big(\mathcal{X}_k^{(l)}\big)$ is a weighted sum of Bernoulli pdfs, where the weights $p\big(c_k^{(l)}\big)$ are normalized. Thus (cf. Section X-B), $f^{(l)}\big(\mathcal{X}_k^{(l)}\big)$ is again a Bernoulli pdf. This means that $f(\mathcal{X}_k^{\text{d}}|\mathcal{Z}_{1:k})$ is approximated in (113) by the product of Bernoulli pdfs $f^{(l)}\big(\mathcal{X}_k^{(l)}\big)$, each associated with one of the components $l \in \mathcal{L}_k$. We conclude that the obtained approximation of $f(\mathcal{X}_k^{\text{d}}|\mathcal{Z}_{1:k})$ is again an LMB pdf, i.e., the approximate update step preserves the LMB structure.

The existence probability $r_k^{(l)}$ and spatial pdf $s_k^{(l)}(\boldsymbol{x}_k)$ parameterizing the Bernoulli pdf $f^{(l)}\big(\mathcal{X}_k^{(l)}\big)$ are obtained as follows. For the legacy components, i.e., $l \in \mathcal{L}_{k-1}$, where $c_k^{(l)} = a_k^{(l)}$,

$$r_k^{(l)} = \sum_{a_k^{(l)}=0}^{m_k} p\big(a_k^{(l)}\big)\, r_k^{(l,a_k^{(l)})} \tag{114}$$

$$s_k^{(l)}(\boldsymbol{x}_k) = \frac{1}{r_k^{(l)}} \sum_{a_k^{(l)}=0}^{m_k} p\big(a_k^{(l)}\big)\, r_k^{(l,a_k^{(l)})}\, s_k^{(l,a_k^{(l)})}(\boldsymbol{x}_k), \tag{115}$$

and for new components, i.e., $l \in \mathcal{L}_k^{\text{new}}$, where $c_k^{(l)} = b_k^{(l)}$,

$$r_k^{(l)} = p\big(b_k^{(l)} = 0\big)\, r_k^{(l,0)}, \quad s_k^{(l)}(\boldsymbol{x}_k) = s_k^{(l,0)}(\boldsymbol{x}_k). \tag{116}$$

Here, expressions of $r_k^{(l,c_k^{(l)})}$ and $s_k^{(l,c_k^{(l)})}(\boldsymbol{x}_k)$ were provided in Section XII-B1. In analogy to the JPDA filter, where the approximation (42) ensured that the approximate posterior pdf $\tilde{f}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ of the multitarget state factorizes into the approximate posterior pdfs $\tilde{f}\big(\boldsymbol{x}_k^{(i)}\big|\boldsymbol{z}_{1:k}\big)$ of the single-target states, the similar approximation (112) ensures that $f(\mathcal{X}_k^{\text{d}}|\mathcal{Z}_{1:k})$ is an LMB pdf.

It remains to calculate the marginal DA pmfs $p\big(a_k^{(l)}\big)$, $l \in \mathcal{L}_{k-1}$ and $p\big(b_k^{(l)}\big)$, $l \in \mathcal{L}_k^{\text{new}}$ occurring in (114)–(116).

The factorization in (107) is analogous to that in (26) with $\{1,\dots,n_\mathrm{t}\}$ and $\{1,\dots,m_k\}$ replaced by $\mathcal{L}_{k-1}$ and $\mathcal{L}_k^{\text{new}}$, respectively. Hence, we can use SPADA with association weights $\beta_k^{(l)}(a_k^{(l)})$, $l \in \mathcal{L}_{k-1}$ and $\xi_k^{(l)}(b_k^{(l)})$, $l \in \mathcal{L}_k^{\text{new}}$—to calculate approximations of $p(a_k^{(l)})$ and $p(b_k^{(l)})$.

After (114)–(116) have been evaluated, target declaration and state estimation can be performed. LMB component $l \in \mathcal{L}_k$ is declared to be an existing target at time k if $r_k^{(l)}$ is larger than a threshold P_th. Furthermore, for a target l that is declared to exist, an estimate $\hat{\boldsymbol{x}}_k^{(l)}$ of the state $\boldsymbol{x}_k^{(l)}$ is calculated as in (3) with $f(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ replaced by $s_k^{(l)}(\boldsymbol{x}_k)$.

Whereas the approximate update step discussed above avoids the exponential scaling of complexity with $|\mathcal{L}_k|$ and m_k, it does not avoid the roughly linear increase of $|\mathcal{L}_k|$ with time k (cf. (106)). Therefore, a pruning step is typically performed, which removes Bernoulli components l whose existence probability $r_k^{(l)}$ is below a predefined threshold P_pr.

It is interesting to note that the expression (109) of the spatial pdf $s_k^{(l,a_k^{(l)})}(\boldsymbol{x}_k)$ is analogous to the expression (41) of the approximate posterior pdf $\tilde{f}(\boldsymbol{x}_k^{(i)}|a_k^{(i)},\boldsymbol{z}_{1:k})$ in the JPDA filter. Indeed, if the predicted existence probability of all legacy components $l \in \mathcal{L}_{k-1}$ is $r_{k|k-1}^{(l)} = 1$, and thus (cf. (108) and the discussion preceding it) $r_k^{(l,a_k^{(l)})} = 1$ for all $l \in \mathcal{L}_{k-1}$ and all $a_k^{(l)} \in \{0,\dots,m_k\}$, and if there are no predicted undetected targets, i.e., $\lambda_{k|k-1}^{\mathrm{u}}(\boldsymbol{x}_k,l_k) = 0$, then $r_k^{(l=(k,m),0)} = 0$ in (110). This means that the existence probability of all new components is zero. As a consequence, (105) is equivalent to (39), and thus the update step of the TOMB/P filter reduces to the update step of the JPDA filter. Furthermore, it can be shown that if the intensity function $\lambda_\mathrm{n}(\boldsymbol{x}_k) = \mu_\mathrm{n} f_\mathrm{n}(\boldsymbol{x}_k)$ for newly detected targets in the vector-type total-SPA method from Section IX-A is set to $\lambda_\mathrm{n}(\boldsymbol{x}_k) = p_\mathrm{d}(\boldsymbol{x}_k)\lambda_{k|k-1}^{\mathrm{u}}(\boldsymbol{x}_k, l_k = 0)$, then the beliefs resulting from that method are equivalent to the Bernoulli component pdfs $f^{(l)}(\mathcal{X}_k^{(l)})$. More specifically, let $\tilde{f}(\boldsymbol{x}_k^{(j)}, r_k^{(j)})$ be the belief produced by the vector-type total-SPA method for the PT that was detected for the first time due to measurement m at time k, i.e., $j = j_{k-1} + m$. Furthermore, let $f^{(l)}(\mathcal{X}_k^{(l)})$ be the pdf of the Bernoulli component with label $l = (m,k)$ obtained by the hybrid labeled/unlabeled variant of the TOMB/P filter. Then the existence probability $r_k^{(l)}$ and spatial pdf $s_k^{(l)}(\boldsymbol{x}_k^{(l)})$ of that Bernoulli component are related to $\tilde{f}(\boldsymbol{x}_k^{(j)}, r_k^{(j)})$ according to $r_k^{(l)} s_k^{(l)}(\boldsymbol{x}_k^{(l)}) = \tilde{f}(\boldsymbol{x}_k^{(j)}, 1)$. While in the vector-type total-SPA method the time step k and the measurement index m related to the first measurement of a PT are implicitly encoded by the order of the subvectors $\boldsymbol{y}_k^{(j)}$, $j = 1,\dots j_k$ in the joint PT vector $\boldsymbol{y}_k$, in the hybrid labeled/unlabeled variant of the TOMB/P filter this information is explicitly given by the Bernoulli component label l.

B. Multisensor Extensions

In the case of multiple sensors ($n_\mathrm{s} \geq 2$), the prediction step discussed in Section XII-A is unchanged as it does not involve the sensor measurements. Regarding the update step, using Bayes' rule, A11, and (100), we obtain

$$\begin{aligned} f(\mathcal{X}_k|\mathcal{Z}_{1:k}) &\propto f(\mathcal{Z}_k|\mathcal{X}_k) f(\mathcal{X}_k|\mathcal{Z}_{1:k-1}) \\ &= \left(\prod_{s=1}^{n_\mathrm{s}} f(\mathcal{Z}_{k,s}|\mathcal{X}_k) \right) f(\mathcal{X}_k|\mathcal{Z}_{1:k-1}). \end{aligned}$$

Based on this expression, the update step can be performed sensor-sequentially within an iterated-corrector method. Iterated marginal posterior pdfs $f(\mathcal{X}_k|\mathcal{Z}_{1:k}^s)$ are calculated for each sensor $s = 1,\dots,n_\mathrm{s}$. Here, $f(\mathcal{X}_k|\mathcal{Z}_{1:k}^s)$ denotes the pdf of $\mathcal{X}_k$ conditioned on $\mathcal{Z}_{1:k-1}$ and on $\mathcal{Z}_{k,s'}$ for $s' = 1,\dots,s$. The sth update step converts $f(\mathcal{X}_k|\mathcal{Z}_{1:k}^{s-1})$ into $f(\mathcal{X}_k|\mathcal{Z}_{1:k}^s)$, thereby incorporating the measurement $\mathcal{Z}_{k,s}$ of sensor s. This recursion is initialized by $f(\mathcal{X}_k|\mathcal{Z}_{1:k}^0) = f(\mathcal{X}_k|\mathcal{Z}_{1:k-1})$. The sth update step is equal to the single-sensor update step discussed in Section XIII-A except for the following differences. The input $f(\mathcal{X}_k|\mathcal{Z}_{1:k-1})$ of the single-sensor update step is replaced by $f(\mathcal{X}_k|\mathcal{Z}_{1:k}^{s-1})$, and the measurements $\boldsymbol{z}_k^{(m)}$, $m \in \{1,\dots,m_k\}$ are replaced by $\boldsymbol{z}_{k,s}^{(m)}$, $m \in \{1,\dots,m_{k,s}\}$. Furthermore, for the labels of the newly detected targets to be unique, they are now given by the triple (k,s,m).

This sequential multisensor MTT method exhibits excellent performance in many scenarios; an example will be considered in Section XIV. Due to the approximation performed in each update step, the final result of the method depends on the order in which the sensors are updated. In certain set-type filters, such as the PHD, CPHD, and MeMBer filters, a sensor-sequential update can lead to a poorer performance than in the single-sensor case [112], [154]. In the method discussed above, because of the different approximation employed, the influence of the sensor order on performance is significantly less strong. We note that "parallel" set-type multisensor MTT methods that perform the update steps for all sensors simultaneously have been proposed recently [104], [105]. However, these methods do not exploit the independence of the measurements of different sensors (cf. A10) and hence do not scale well with the number of sensors.

XIV. Performance Evaluation

In this section, we present simulation and real-data experiments in order to demonstrate the performance of SPA-based MTT methods and compare it with that of other MTT methods.

A. Simulation Results

We consider a region of interest (ROI) given by $[-3000\,\mathrm{m}, 3000\,\mathrm{m}] \times [-3000\,\mathrm{m}, 3000\,\mathrm{m}]$, with up to five targets. The target states comprise 2-D position and velocity, i.e., $\boldsymbol{x}_k^{(i)} = \left[x_{1,k}^{(i)}\ x_{2,k}^{(i)}\ \dot{x}_{1,k}^{(i)}\ \dot{x}_{2,k}^{(i)}\right]^\mathrm{T}$, and they evolve according to the near constant-velocity motion model [155, Sec. 6.3.2] such that the target trajectories tend to intersect at the ROI center. The sensors are placed uniformly on a circle of radius 3000 m about the ROI center. The ROI with the sensor positions (for $n_\mathrm{s} = 3$ sensors) and an example realization of the target trajectories is shown in [53, Fig. 4]. The sensors perform range and bearing measurements within a measurement range of 6000 m. The mean number of clutter measurements is $\mu_\mathrm{c}^{(s)} = 2$. The clutter pdf $f_\mathrm{c}^{(s)}(\boldsymbol{z}_{n,m}^{(s)})$ is uniform on $[0\,\mathrm{m}, 6000\,\mathrm{m}]$ with

respect to the range component and uniform on $[0°, 360°)$ with respect to the angle component.

We present simulation results for the following SPA-based multisensor MTT methods: the sequential and parallel multisensor JPDA methods with SPA-based DA described in Section VII-C (abbreviated[9] "JPDA-SPA"), the sequential and parallel multisensor total-SPA methods described in Section IX-B (abbreviated "Total-SPA-S" and "Total-SPA-P," respectively), and the sequential multisensor TOMB/P filter with SPA-based DA described in Section XIII-B (abbreviated "TOMB/P-SPA"). These SPA-based methods are compared with the iterated-corrector PHD filter [3], [101], [112] ("IC-PHD"), the iterated-corrector CPHD filter [3], [102], [112] ("IC-CPHD"), the partition-based multisensor PHD filter [104] ("MS-PHD"), and the partition-based multisensor CPHD filter [104] ("MS-CPHD"). Particle implementations are used for all filters.

The simulation parameters are as follows. For methods that model the number of targets as unknown, the birth pdf $f_{\mathrm{b}}(\boldsymbol{x}_k)$ is uniform on the ROI, the number of newborn targets is Poisson distributed with mean $\mu_{\mathrm{b}} = 10^{-2}$, and the survival probability is $p_{\mathrm{s}}(\boldsymbol{x}_k) = p_{\mathrm{s}} = 0.999$. The threshold for target declaration is $P_{\mathrm{th}} = 0.5$. The SPA-based methods perform $n_{\mathrm{it}} = 20$ message passing iterations. In Total-SPA-S, we choose the intensity function for newly detected targets as $\lambda_{\mathrm{n}}^{(s)}(\boldsymbol{x}_k) = p_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k)\mu_{\mathrm{b}} f_{\mathrm{b}}(\boldsymbol{x}_k)$ for all sensors s. (Note that $\mu_{\mathrm{n}}^{(s)} = \int \lambda_{\mathrm{n}}^{(s)}(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k$ and $f_{\mathrm{n}}(\boldsymbol{x}_k) = \lambda_{\mathrm{n}}^{(s)}(\boldsymbol{x}_k)/\mu_{\mathrm{n}}^{(s)}$.) In Total-SPA-P, we set $\lambda_{\mathrm{n}}^{(1)}(\boldsymbol{x}_k) = p_{\mathrm{d}}^{(1)}(\boldsymbol{x}_k)\mu_{\mathrm{b}} f_{\mathrm{b}}(\boldsymbol{x}_k)$ and $\lambda_{\mathrm{n}}^{(s)}(\boldsymbol{x}_k) = 0$ for $s = 2, \ldots, n_{\mathrm{s}}$. In TOMB/P-SPA, the mean number of targets at time $k=0$ is $\mu_{\mathrm{p}}=0$ (cf. S9). In TOMB/P-SPA, Total-SPA-P, and Total-SPA-S, the pruning threshold is $P_{\mathrm{pr}} = 10^{-5}$. For track management in JPDA-SPA, we perform gating with a gate threshold of 18.4, we use the m-of-n heuristic [1] with $m = 4$ and $n = 6$ across time and sensors for track initialization, and we terminate a track if for six consecutive update steps no measurement falls into the gate of the respective target. The maximum numbers of subsets and partitions used by MS-PHD and MS-CPHD are 120 and 720, respectively, similarly to [104]. We performed 400 simulation runs, each with 150 time steps k. A new target trajectory was generated in each simulation run. Further details of the simulation setup and parameters (e.g., the number of particles used by the various filters) are provided in [53].

We measure the performance of the various MTT methods by the Euclidean distance based optimal sub-pattern assignment (OSPA) metric with cutoff parameter 200 [156]. (We recall that complementary performance results assessing DA accuracy were presented in Section VI-C.) For $n_{\mathrm{s}} = 3$ sensors and a detection probability of $p_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k^{(i)}) = p_{\mathrm{d}} = 0.8$, Fig. 5 shows the mean OSPA (MOSPA) error—averaged over 400 simulation runs—of all methods versus time k. One can see that for all methods, the error peaks at times $k = 5$, 10, 15, 20, and 25, i.e., when there are target births. However, very soon after each target birth, all methods except IC-PHD reliably estimate the number of targets. The performance of TOMB/P-SPA, Total-SPA-S, and Total-SPA-P is seen to be almost identical. The performance of JPDA-SPA is inferior to that of the other SPA-based methods immediately after target births; this can be explained by the fact that JPDA-SPA uses a heuristic to initialize new targets. Otherwise, JPDA-SPA performs like the other SPA-based methods. The SPA-based methods are seen to outperform all the other simulated methods. In particular, IC-CPHD, MS-PHD, and MS-CPHD perform worse than the SPA-based methods because particle implementations of (C)PHD filters involve a potentially unreliable clustering step. This clustering step is especially unreliable when targets are closely spaced, which occurs in our simulation around time $k = 100$, i.e., when the five target trajectories intersect in the ROI center. In fact, the MOSPA error of IC-CPHD, MS-PHD, and MS-CPHD is seen to be higher around that time. Finally, IC-PHD performs significantly worse than all the other filters because it is unable to reliably estimate the number of targets in the simulated scenario. We note that simulation results for a significantly larger scenario with closely spaced targets are presented in [53].

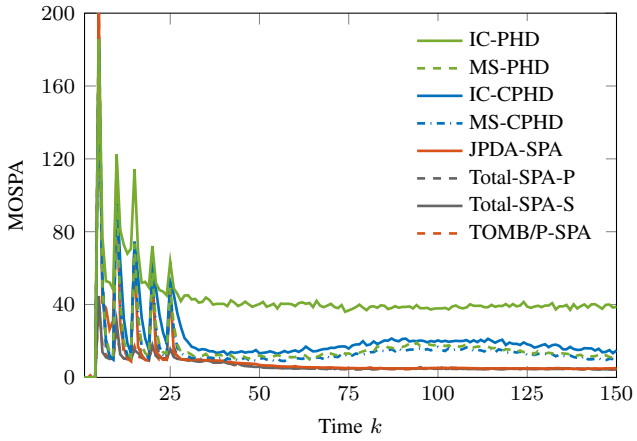


Fig. 5. MOSPA error versus time k for $n_{\mathrm{s}} = 3$ and $p_{\mathrm{d}} = 0.8$. (The curves for Total-SPA-S and TOMB/P-SPA coincide.)

The average computation time per time step k for a MATLAB implementation on a single core of an Intel Xeon E5-2640 v3 CPU was measured as 0.03 s for JPDA-SPA; 0.08 s for Total-SPA-S, Total-SPA-P, and TOMB/P-SPA; 0.11 s for IC-PHD; 0.14 s for IC-CPHD; 13.21 s for MS-PHD; and 13.82 s for MS-CPHD. We note that JPDA-SPA used gating whereas the other filters did not. The high computation times of the MS-PHD and MS-CPHD filters are due to the fact that the trellis algorithm used for partition extraction is tailored to a Gaussian mixture implementation and becomes computationally intensive in a particle-based implementation. Further results demonstrating the excellent scalability of the SPA-based methods are reported in [52], [53], [148].

Finally, for detection probability $p_{\mathrm{d}} = 0.6$, Fig. 6 shows the time-averaged MOSPA error—i.e., averaged over all 150 simulated time steps—versus the number of sensors n_{s}. JPDA-SPA performs worse than TOMB/P-SPA, Total-SPA-S, and Total-SPA-P, due to its heuristic track initialization. Total-SPA-P performs worse than Total-SPA-S and TOMB/P-SPA, because it does not implement sensor fusion for new PTs. The increase of the time-averaged MOSPA error of MS-PHD

[9]We do not distinguish between the sequential and parallel multisensor JPDA methods because they exhibited identical performance.

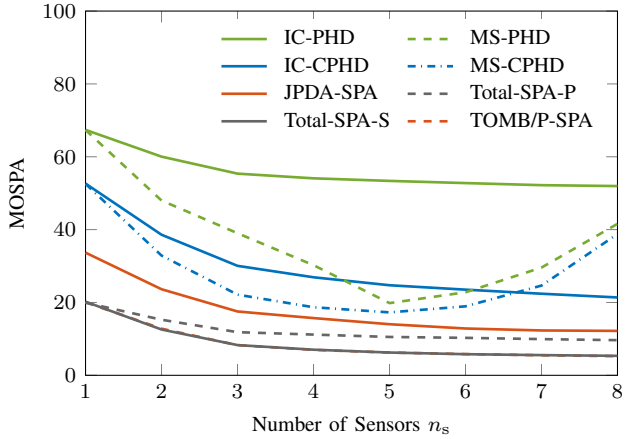


Fig. 6. Time-averaged MOSPA error versus number of sensors n_s for $p_d = 0.6$. (The curves for Total-SPA-S and TOMB/P-SPA coincide.)

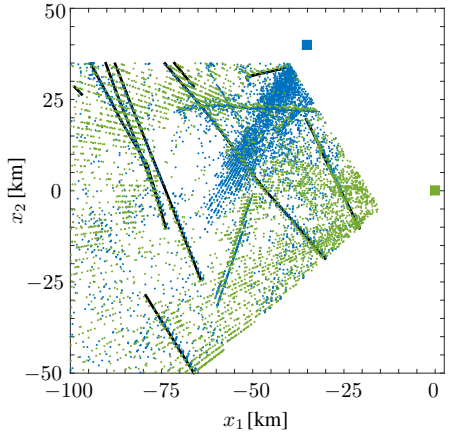


Fig. 7. Three hours of measurements acquired by two HFSW radar stations on the west coast of Italy. The positions of radar stations #1 (near Pisa) and #2 (near La Spezia) are indicated by a green and a blue square, respectively. The corresponding measurements are indicated by green and blue dots, respectively. AIS ground truth information is indicated by black lines.

and MS-CPHD for $n_s > 5$ is due to the fact that the chosen maximum numbers of subsets (120) and partitions (720) are too small for that case; however, larger maximum numbers would lead to excessive simulation times.

B. Results for a Radar Tracking Experiment

For further validation of the SPA-based MTT methods, we use real measurements that were acquired by two high-frequency surface-wave (HFSW) radars for maritime surveillance [157], [158]. HFSW radars feature over-the-horizon coverage and a continuous-time mode of operation. On the other hand, they suffer from poor range and azimuth resolution, high nonlinearity, and strong clutter; these effects make MTT a challenging task [157], [158]. The two radar stations are located close to the cities of Pisa and La Spezia in Italy. The ROI is the intersection of the fields-of-view of the two radar stations. All vessels in the ROI are considered as targets. Ground truth information about the target positions is reported by the Automatic Identification System (AIS). However, this information is incomplete since AIS reports do not include vessels below a certain gross tonnage and military vessels. The overall tracking scenario is shown in Fig. 7. At each of the two radar sensors, measurements are available every 33.28 s.

We processed the real measurements provided by the two sensors by the same MTT methods as in the previous subsection. Target motion is modeled by the near constant-velocity model with driving process variance $0.01\,\mathrm{m}^2/\mathrm{s}^4$. We use a range-bearing measurement model involving a Gaussian measurement noise vector with covariance matrix $\mathrm{diag}\{(150\mathrm{m})^2, (1.5^\circ)^2\}$. The probability of detection is $p_d^{(s)}(\boldsymbol{x}_k^{(i)}) = p_d = 0.65$ for both sensors. The clutter pdf $f_c^{(s)}(\boldsymbol{z}_{k,s}^{(m)})$ is chosen uniform on the ROI, and the mean number of clutter measurements is $\mu_c^{(s)} = 15$. For methods that model the number of targets as unknown, the birth pdf $f_b(\boldsymbol{x}_k)$ is uniform on the ROI, the number of newborn targets is Poisson distributed with mean $\mu_b = 10^{-1}$, and the survival probability is $p_s(\boldsymbol{x}_k) = p_s = 0.999$. The threshold for target declaration is $P_{th} = 0.5$. Track management in JPDA-SPA is as described earlier, except that the gate threshold is 9 and tracks are terminated if for six consecutive update steps less than two measurements fall into the gate of the respective target. The (C)PHD-type filters use 45000 particles to represent the PHD of the target states. JPDA-SPA, TOMB/P-SPA, Total-SPA-S, and Total-SPA-P use 3000 particles for each target. In TOMB/P-SPA, the mean number of targets at time $k = 0$ is $\mu_p = 5$, and the spatial pdf $f_p(\boldsymbol{x}_0)$ is uniform on the ROI. In TOMB/P-SPA, Total-SPA-S, and Total-SPA-P, after each single-sensor update step, targets with existence probability smaller than $P_{pr} = 10^{-3}$ are removed. All remaining parameters are as in Section XIV-A.

Table I shows the time-on-target (TOT) and the false alarm rate (FAR) [1] obtained with the various methods for 24 h of measurement data. TOT is the fraction of time that the target is successfully detected, which is considered to be true if the estimated target position is within 500 m of the true target position. FAR is the number of false trajectories (or false detections) generated in the surveillance region per unit of 2-D space and unit of time. Ideally, the TOT would be one and the FAR would be zero. It can be seen in Table I that the SPA-based methods achieve an attractive TOT-FAR compromise. In particular, the TOT of Total-SPA-S, Total-SPA-P, and TOMB/P-SPA is very similar, and larger than that of the other filters. The smaller TOT of JPDA-SPA-S and JPDA-SPA-P (the sequential and parallel multisensor extensions of JPDA-SPA discussed in Section VII-C) can be explained by the heuristic that is used to initialize new targets. The relatively high TOT of MS-PHD and MS-CPHD is seen to come at the cost of an increased FAR, while Total-SPA-S, Total-SPA-P, and TOMB/P-SPA achieve both a high TOT and a rather low FAR. We caution that the FAR reported in Table I is pessimistic in that the AIS ground truth information is incomplete and thus measurements that are generated by targets without ground truth information and detected by a tracking algorithm are considered as false alarms. In this light, the relatively low FAR of IC-PHD, JPDA-SPA-S, and JPDA-SPA-P can be partly attributed to the fact that these methods,

	TOT [%]	FAR $[\mathrm{s}^{-1}\mathrm{km}^{-2}]$
IC-PHD	60.8	$1.588 \cdot 10^{-5}$
MS-PHD	71.9	$2.041 \cdot 10^{-5}$
IC-CPHD	68.9	$1.933 \cdot 10^{-5}$
MS-CPHD	73.2	$2.120 \cdot 10^{-5}$
JPDA-SPA-S	67.7	$1.518 \cdot 10^{-5}$
JPDA-SPA-P	68.1	$1.572 \cdot 10^{-5}$
Total-SPA-S	75.5	$1.883 \cdot 10^{-5}$
Total-SPA-P	75.0	$1.663 \cdot 10^{-5}$
TOMB/P-SPA	75.6	$1.688 \cdot 10^{-5}$

TABLE I
TIME-ON-TARGET (TOT) AND FALSE ALARM RATE (FAR).

as evidenced by their low TOT, are unable to track some targets that do not provide AIS information (and that would thus contribute to the FAR if successfully tracked).

XV. CONCLUSION

Multitarget tracking (MTT) is an important contribution to acquiring and maintaining an awareness of the environment. Although measurement origin uncertainty makes MTT a complicated and challenging task, large-scale scenarios and real-time operation on resource-limited devices call for MTT methods whose complexity is moderate and scales well with the number of targets, sensors, and measurements. In this tutorial paper, we showed that the development of high-performance MTT methods with moderate complexity and excellent scalability can be based on the recently emerged paradigm of factor graphs and message passing using the sum-product algorithm (SPA). We presented SPA-based Bayesian MTT methods within both a random vector framework and a random finite set framework. A core component of these methods, and a major reason for their scalability, is a highly effective and efficient SPA-based algorithm for probabilistic data association. We discussed the integration of SPA-based probabilistic data association into existing MTT methods and showed that certain existing methods can be reformulated within the SPA framework. We also presented new vector-type and set-type MTT methods in which the SPA is used either for probabilistic data association or for the entire MTT problem.

The SPA approach to designing MTT methods has important advantages regarding scalability, accuracy, complexity, versatility, and intuitiveness. While rooted in the framework of optimum Bayesian inference, the SPA-based approach easily accomodates different system models and application-related aspects. In particular, it is suited to general nonlinear, non-Gaussian, and time-varying scenarios. The factor graph formulation of the MTT system model combined with the principle of stretching factor nodes introduces a beneficial intuitiveness and flexibility into the SPA-based design of MTT methods. Finally, SPA-based MTT methods are able to cope with unknown and time-varying hyperparameters, such as detection probabilities [54] and motion model parameters [159]. Using both simulated and real measurements, we demonstrated the excellent performance and low complexity of the presented SPA-based MTT methods.

The beliefs obtained by the loopy SPA can be overconfident in that their spread underestimates the uncertainty of the estimates [145]. This can be a limitation in certain applications [160]. A variational message passing approach that performs iterative data association across multiple sensors and/or time steps and avoids overconfident beliefs will be proposed in forthcoming work [145].

Possible directions of future research include an extension of the network localization and navigation (NLN) paradigm [161]–[165] to noncollaborative objects; a combination of MTT with resource allocation, sensor selection, and control [166]–[173]; network experimentation [174]–[176] with noncollaborative objects; network localization systems [177]–[179] for MTT; and distributed SPA-based MTT methods for decentralized wireless sensor networks [180]–[182].

REFERENCES

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

[2] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge, UK: Cambridge University Press, 2011.

[3] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2007.

[4] W. Koch, *Tracking and Sensor Data Fusion: Methodological Framework and Selected Applications*. Berlin, Germany: Springer, 2014.

[5] Y. Bar-Shalom, "Extension of the probabilistic data association filter in multi-target tracking," in *Proc. Symposium on Nonlinear Estimation Theory and its Applications*, San Diego, CA, Sep. 1974.

[6] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[7] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, no. 3, pp. 173–184, Jul. 1983.

[8] C. Rasmussen and G. D. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 560–576, Jun. 2001.

[9] Q. Lu, Y. Bar-Shalom, P. Willett, K. Granström, R. Ben-Dov, and B. Milgrom, "Tracking initially unresolved thrusting objects using an optical sensor," *IEEE Trans. Aerosp. Electron. Syst.*, vol. IEEE Xplore Early Access, 2018.

[10] R. J. Adrian, "Particle-imaging techniques for experimental fluid mechanics," *Annu. Rev. Fluid Mech.*, vol. 23, no. 1, pp. 261–304, 1991.

[11] A. Genovesio, T. Liedl, V. Emiliani, W. J. Parak, M. Coppey-Moisan, and J. C. Olivo-Marin, "Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1062–1070, May 2006.

[12] M. Maška and al., "A benchmark for comparison of cell tracking algorithms," *Bioinformatics*, vol. 30, no. 11, pp. 1609–1617, Jun. 2014.

[13] C. Urmson and al., "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, vol. 25, no. 8, pp. 425–466, Jun. 2008.

[14] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE IV 2011*, Baden-Baden, Germany, Jun. 2011, pp. 163–168.

[15] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.

[16] 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.

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

[18] J. Shen and A. F. Molisch, "Estimating multiple target locations in multi-path environments," *IEEE Trans. Wireless Commun.*, vol. 13, no. 8, pp. 4547–4559, Aug. 2014.

[19] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ: Princeton University Press, 2009.

[20] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: architecture, challenges and applications," *IEEE Network*, vol. 26, no. 3, pp. 21–28, May 2012.

[21] G. Ferri, A. Munafò, A. Tesei, P. Braca, F. Meyer, K. Pelekanakis, R. Petroccia, J. Alves, C. Strode, and K. LePage, "Cooperative robotic networks for underwater surveillance: an overview," *IET Radar Sonar Navig.*, vol. 11, no. 12, pp. 1740–1761, 2017.

[22] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE CVPR-08*, Anchorage, Alaska, Jun. 2008.

[23] S. Depatla, A. Muralidharan, and Y. Mostofi, "Occupancy estimation using only WiFi power measurements," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1381–1393, Jul. 2015.

[24] S. Bartoletti, A. Conti, and M. Z. Win, "Device-free counting via wideband signals," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1163–1174, May 2017.

[25] F. R. Kschischang, B. J. 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.

[26] J. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[27] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007.

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

[29] J. Pearl, "Reverend Bayes on inference engines: A distributed hierarchical approach," in *Proc. AAAI-82*, Pittsburgh, PA, Aug. 1982, pp. 133–136.

[30] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel, Eds., San Francisco, CA, 2003, ch. 8, pp. 239–269.

[31] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, Jan. 2008.

[32] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press, 2009.

[33] E. Riegler, G. E. Kirkelund, C. N. Manchon, M. A. Badiu, and B. H. Fleury, "Merging belief propagation and the mean field approximation: A free energy approach," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 588–602, Jan. 2013.

[34] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[35] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.

[36] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[37] T. Richardson and R. Urbanke, *Modern Coding Theory*. New York City, NY: Cambridge University Press, 2008.

[38] A. P. Worthen and W. E. Stark, "Unified design of iterative receivers using factor graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 843–849, Feb. 2001.

[39] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1772–1793, Jul. 2002.

[40] K. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Application*. Norwell, MA: Kluwer Academic Publishers, 2001.

[41] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[42] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[43] 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.

[44] L. Chen, M. J. Wainwright, M. Çetin, and A. S. Willsky, "Multitarget-multisensor data association using the tree-reweighted max-product algorithm," in *Proc. SPIE-03*, vol. 5096, Orlando, FL, Aug. 2003, pp. 127–138.

[45] L. Chen, M. Çetin, and A. S. Willsky, "Distributed data association for multi-target tracking in sensor networks," in *Proc. FUSION-08*, Cologne, Germany, Jul. 2005.

[46] L. Chen, M. J. Wainwright, M. Çetin, and A. S. Willsky, "Data association based on optimization in graphical models with application to sensor networks," *Math. Comput. Model.*, vol. 43, no. 9–10, pp. 1114–1135, 2006.

[47] M. Çetin, L. Chen, J. W. Fisher, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 42–55, Jul. 2006.

[48] M. Chertkov, L. Kroc, F. Krzakala, M. Vergassola, and L. Zdeborová, "Inference in particle tracking experiments by passing messages between images," *PNAS*, vol. 107, no. 17, pp. 7663–7668, Apr. 2010.

[49] J. L. Williams and R. A. Lau, "Data association by loopy belief propagation," in *Proc. FUSION-10*, Edinburgh, UK, Jul. 2010.

[50] J. L. 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.

[51] 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.

[52] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch, "Scalable multitarget tracking using multiple sensors: A belief propagation approach," in *Proc. FUSION-15*, Washington D.C., Jul. 2015, pp. 1778–1785.

[53] ——, "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.

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

[55] T. Li, H. Pareek, P. Ravikumar, D. Balwada, and K. Speer, "Tracking with ranked signals," in *Proc. UAI-15*, Amsterdam, The Netherlands, 2015, pp. 474–483.

[56] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

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

[58] B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[59] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[60] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Autom. Control*, vol. 17, no. 4, pp. 439–448, Aug. 1972.

[61] N. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to non-linear and non-Gaussian Bayesian state estimation," *IEE Proc. F: Radar Signal Process.*, vol. 140, pp. 107–113, 1993.

[62] S. Mori, C.-Y. Chong, E. Tse, and R. Wishner, "Tracking and classifying multiple targets without a priori identification," *IEEE Trans. Autom. Control*, vol. 31, no. 5, pp. 401–409, May 1986.

[63] D. Musicki, R. Evans, and S. Stankovic, "Integrated probabilistic data association," *IEEE Trans. Autom. Control*, vol. 39, no. 6, pp. 1237–1241, Jun. 1994.

[64] 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.

[65] J. Vermaak, S. Maskell, and M. Briers, "A unifying framework for multi-target tracking and existence," in *Proc. FUSION-05*, Philadelphia, PA, Jul. 2005.

[66] D. J. Salmond, "Mixture reduction algorithms for target tracking in clutter," in *Proc. SPIE*, vol. 1305, Orlando, FL, Apr. 1990, pp. 434–445.

[67] L. Y. Pao, "Multisensor multitarget mixture reduction algorithms for tracking," *J. Guid. Control Dyn.*, vol. 17, no. 6, pp. 1205–1211, 1994.

[68] R. Fitzgerald, "Development of practical PDA logic for multitarget tracking by microprocessor," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Norwood, MA: Artech-House, 1990.

[69] B. Bakhtiar, H. Alavi, and F. Amoozegar, "Efficient algorithm for computing data association probabilities for multitarget tracking," in *Automatic Object Recognition VI*, F. A. Sadjadi, Ed. Orlando, Fl: SPIE, 1996, vol. 2756.

[70] Musicki, D. and La Scala, B., "Multi-target tracking in clutter without measurement assignment," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 877–896, Jul. 2008.

[71] S. Maskell, M. Briers, and R. Wright, "Fast mutual exclusion," in *Proc. SPIE-04*, vol. 5428, Orlando, FL, Apr. 2004, pp. 526–536.

[72] P. Horridge and S. Maskell, "Real-time tracking of hundreds of targets with efficient exact JPDAF implementation," in *Proc. FUSION-06*, Florence, Italy, Jul. 2006.

[73] J. A. Roecker, "A class of near optimal JPDA algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 2, pp. 504–510, Apr. 1994.

[74] K. Pattipati, R. L. Popp, and T. Kirubarajan, "Survey of assignment techniques for multitarget tracking," in *Multitarget-Multisensor Tracking: Applications and Advances*, Y. Bar-Shalom and W. D. Blair, Eds. Norwood, MA: Artech-House, 2000, vol. 3, ch. 2, pp. 77–159.

[75] H. A. P. Blom and E. A. Bloem, "Probabilistic data association avoiding track coalescence," *IEEE Trans. Autom. Control*, vol. 45, no. 2, pp. 247–259, Feb. 2000.

[76] H. A. P. Blom, E. A. Bloem, and D. Musicki, "JIPDA: Automatic target tracking avoiding track coalescence," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 2, pp. 962–974, Apr. 2015.

[77] L. Svensson, D. Svensson, M. Guerriero, and P. Willett, "Set JPDA filter for multitarget tracking," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4677–4691, Oct. 2011.

[78] J. L. Williams, "An efficient, variational approximation of the best fitting multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 258–273, Jan. 2015.

[79] R. D. Turner, S. Bottone, and B. Avasarala, "A complete variational tracker," in *Proc. NIPS-14*, Montreal, Canada, Dec. 2014, pp. 496–504.

[80] R. A. Lau and J. L. Williams, "A structured mean field approach for existence-based multiple target tracking," in *Proc. FUSION-16*, Heidelberg, Germany, Jul. 2016, pp. 1111–1118.

[81] L. Y. Pao and C. W. Frei, "A comparison of parallel and sequential implementations of a multisensor multitarget tracking algorithm," in *Proc. ACC-95*, vol. 3, Seattle, WA, Jun. 1995, pp. 1683–1687.

[82] 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.

[83] R. Danchick and G. E. Newnam, "A fast method for finding the exact n-best hypotheses for multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 2, pp. 555–560, Apr 1993.

[84] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 2, pp. 138–150, Feb. 1996.

[85] R. L. Popp, K. R. Pattipati, and Y. Bar-Shalom, "Dynamically adaptable m-best 2-D assignment algorithm and multilevel parallelization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 4, pp. 1145–1160, Oct. 1999.

[86] R. Danchick and G. E. Newnam, "Reformulating Reid's MHT method with generalised Murty K-best ranked linear assignment algorithm," *IEE Radar Sonar Navig.*, vol. 153, no. 1, pp. 13–22, Feb. 2006.

[87] T. Kurien, "Issues in the design of practical multitarget tracking algorithms," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Norwood, MA: Artech-House, 1990, pp. 43–83.

[88] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 19, pp. 5–18, Jan. 2004.

[89] C. Morefield, "Application of 0-1 integer programming to multitarget tracking problems," *IEEE Trans. Autom. Control*, vol. 22, no. 3, pp. 302–312, Jun 1977.

[90] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. J. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Trans. Autom. Control*, vol. 37, no. 2, pp. 198–213, Feb. 1992.

[91] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A generalized S-D assignment algorithm for multisensor-multitarget state estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 2, pp. 523–538, Apr. 1997.

[92] A. P. Poore and N. Rijavec, "A Lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking," *SIAM J. Optimiz.*, vol. 3, no. 3, pp. 544–563, 1993.

[93] A. B. Poore and S. Gadaleta, "Some assignment problems arising from multiple target tracking," *Math. Comput. Model.*, vol. 43, no. 9–10, pp. 1074–1091, 2006.

[94] S. Coraluppi and C. A. Carthel, "If a tree falls in the woods, it does make a sound: Multiple-hypothesis tracking with undetected target births," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 3, pp. 2379–2388, July 2014.

[95] S. Mori, C.-Y. Chong, and K. C. Chang, "Three formalisms of multiple hypothesis tracking," in *Proc. FUSION-16*, Jul. 2016.

[96] S. Coraluppi and C. Carthel, "Multi-stage multiple-hypothesis tracking," *J. Adv. Inf. Fusion*, vol. 6, no. 1, pp. 57–67, Jun. 2011.

[97] G. Castañón and L. Finn, "Multi-target tracklet stitching through network flows," in *Proc. AeroConf-11*, Big Sky, MT, Mar. 2011.

[98] C. Y. Chong, "Graph approaches for data association," in *Proc. FUSION-12*, Singapore, Singapore, Jul. 2012, pp. 1578–1585.

[99] S. Coraluppi, C. Carthel, W. Kreamer, and A. Willsky, "Recent advances in multi-INT track fusion," in *Proc. AeroConf-16*, Big Sky, MT, Mar. 2016.

[100] ——, "New graph-based and MCMC approaches to multi-INT surveillance," in *Proc. FUSION-17*, Heidelberg, Germany, Jul. 2016, pp. 394–401.

[101] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, Oct. 2005.

[102] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3553–3567, Jul. 2007.

[103] ——, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409–423, Feb. 2009.

[104] S. Nannuru, S. Blouin, M. Coates, and M. Rabbat, "Multisensor CPHD filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1834–1854, Aug. 2016.

[105] A. A. Saucan, M. J. Coates, and M. Rabbat, "A multi-sensor multi-bernoulli filter," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5495–5509, Oct. 2017.

[106] R. P. S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[107] 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.

[108] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3460–3475, Jul. 2013.

[109] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3246–3260, Jun. 2014.

[110] P. Braca, S. Marano, V. Matta, and P. Willett, "Asymptotic efficiency of the PHD in multitarget/multisensor estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 553–564, Jun. 2013.

[111] R. P. S. Mahler, "PHD filters of higher order in target number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, Oct. 2007.

[112] S. Nagappa and D. Clark, "On the ordering of the sensors in the iterated-corrector probability hypothesis density (PHD) filter," in *Proc. SPIE-11*, Orlando, FL, Apr. 2011, pp. 26–28.

[113] J. L. Williams, "Experiments with graphical model implementations of multiple target multiple Bernoulli filters," in *Proc. ISSNIP-11*, Adelaide, Australia, Dec. 2011, pp. 532–537.

[114] F. Meyer, A. Tesei, and M. Z. Win, "Localization of multiple sources using time-difference arrival measurements," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, New Orleans, LA, Mar. 2017, pp. 3151–3155.

[115] A. Gning and L. Mihaylova, "Dynamic clustering and belief propagation for distributed inference in random sensor networks with deficient links," in *Proc. FUSION-09*, Seattle, WA, Jul. 2009, pp. 656–663.

[116] V. P. Panakkal and R. Velmurugan, "Effective data association scheme for tracking closely moving targets using factor graphs," in *Proc NCC-11)*, Jan. 2011.

[117] Z. Chen, L. Chen, M. Cetin, and A. S. Willsky, "An efficient message passing algorithm for multi-target tracking," in *Proc. FUSION-09*, Seattle, Washington, Jul. 2009, pp. 826–833.

[118] M. Bayati, D. Shah, and M. Sharma, "Max-product for maximum weight matching: Convergence, correctness, and LP duality," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1241–1251, Mar. 2008.

[119] R. A. Lau and J. L. Williams, "Multidimensional assignment by dual decomposition," in *Proc. ISSNIP-11*, Adelaide, Australia, Dec. 2011, pp. 437–442.

[120] Q. Li, J. Sun, and W. Sun, "An efficient multiple hypothesis tracker using max product belief propagation," in *Proc. FUSION-17*, Jul. 2017.

[121] H. Zhu, C. Han, and C. Li, "Graphical models-based track association algorithm," in *Proc. FUSION-07*, Québec, Canada, Jul. 2007.

[122] J. Johnson, D. Malioutov, and A. Willsky, "Lagrangian relaxation for MAP estimation in graphical models," in *Proc 45th Annual Allerton Conference on Communication, Control, and Computing*, 2007, pp. 64–73.

[123] N. Komodakis, N. Paragios, and G. Tziritas, "MRF energy minimization and beyond via dual decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 531–552, Mar. 2011.

[124] A. Frank, P. Smyth, and A. Ihler, "Beyond MAP estimation with the track-oriented multiple hypothesis tracker," *IEEE Trans. Signal Process.*, vol. 62, no. 9, pp. 2413–2423, May 2014.

[125] M. Chertkov, L. Kroc, and M. Vergassola, "Belief propagation and beyond for particle tracking," arXiv, e-print arXiv:0806.1199v1, June 2008. [Online]. Available: http://arxiv.org/abs/0806.1199

[126] B. Huang and T. Jebara, "Approximating the permanent with belief propagation," Aug. 2009, available online: http://arxiv.org/abs/0908.1769.

[127] P. O. Vontobel, "The Bethe permanent of a nonnegative matrix," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1866–1901, Mar. 2013.

[128] J. L. Williams and R. A. Lau, "Convergence of loopy belief propagation for data association," in *Proc. ISSNIP-10*, Brisbane, Australia, Dec. 2010, pp. 175–180.

[129] F. Meyer, A. Tesei, and M. Z. Win, "Localization of multiple sources using time-difference of arrival measurements," in *Proc. IEEE ICASSP-17*, New Orleans, LA, Mar. 2017, pp. 3151–3155.

[130] 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. IEEE ICL-GNSS-16*, Barcelona, Spain, Jun. 2016.

[131] 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, May 2017, pp. 652–658.

[132] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Z. Win, "A scalable belief propagation algorithm for radio signal based SLAM," 2018, in preparation.

[133] R. A. Lau and J. L. Williams, "Tracking a coordinated group using expectation maximisation," in *Proc. ISSNIP-13*, Melbourne, Australia, Apr. 2013.

[134] A. S. Rahmathullah, R. Selvan, and L. Svensson, "A batch algorithm for estimating trajectories of point targets using expectation maximization," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4792–4804, Sep. 2016.

[135] H. Lan, Q. Pan, F. Yang, S. Sun, and L. Li, "Variational Bayesian approach for joint multitarget tracking of multiple detection systems," in *Proc. FUSION-16*, Heidelberg, Germany, Jul. 2016, pp. 1260–1267.

[136] S. Sun, H. Lan, Z. Wang, Q. Pan, and H. Zhang, "The application of sum-product algorithm for data association," in *Proc. FUSION-16*, Heidelberg, Germany, Jul. 2016, pp. 416–423.

[137] B. Etzlinger, H. Wymeersch, and A. Springer, "Cooperative synchronization in wireless networks," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2837–2849, Jun. 2014.

[138] 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.

[139] R. Kindermann and J. L. Snell, *Markov Random Fields and their Applications*. Providence, RI: AMS, 1980.

[140] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*, 2nd ed. New York City, NY: Springer, 2007.

[141] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "A new class of upper bounds on the log partition function," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2313–2335, Jul. 2005.

[142] M. Chertkov and A. B. Yedidia, "Approximating the permanent with fractional belief propagation," *J. Mach. Learn. Res.*, vol. 14, pp. 2029–2066, 2013.

[143] F. Meyer, O. Hlinka, and F. Hlawatsch, "Sigma point belief propagation," *IEEE Signal Process. Lett.*, vol. 21, no. 2, pp. 145–149, Feb. 2014.

[144] F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Belief propagation methods for scalable multitarget tracking – Supplementary material," 2018, in preparation.

[145] J. L. Williams and R. A. Lau, "Multiple scan data association by convex variational inference," 2016, available online: http://arxiv.org/abs/1607.07942.

[146] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for multi-target tracking," *IEEE Trans. Autom. Control*, vol. 54, no. 3, pp. 481–497, Mar. 2009.

[147] J. M. Mooij, "libDAI: A free and open source C++ library for discrete approximate inference in graphical models," *J. Mach. Learn. Res.*, vol. 11, pp. 2169–2173, Aug. 2010.

[148] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch, "Tracking an unknown number of targets using multiple sensors: A belief propagation method," in *Proc. FUSION-16*, Heidelberg, Germany, Jul. 2016, pp. 719–726.

[149] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York City, NY: Springer, 1994.

[150] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes. Vol. I*, 2nd ed. New York City, NY: Springer, 2003.

[151] R. L. Streit, *Poisson Point Processes: Imaging, Tracking, and Sensing*. New York City, NY: Springer, 2010.

[152] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: Direct derivation and implementation," 2017, available online: http://arxiv.org/abs/1703.04264.

[153] Á. F. García-Fernández, L. Svensson, and M. R. Morelande, "Multiple target tracking based on sets of trajectories," 2016, available online: http://arxiv.org/abs/1605.08163.

[154] D. Franken, M. Schmidt, and M. Ulmke, ""Spooky action at a distance" in the cardinalized probability hypothesis density filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 4, pp. 1657–1664, Oct. 2009.

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

[156] 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.

[157] S. Maresca, P. Braca, J. Horstmann, and R. Grasso, "Maritime surveillance using multiple high-frequency surface-wave radars," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 5056–5071, Aug. 2014.

[158] P. Braca, S. Maresca, R. Grasso, K. Bryan, and J. Horstmann, "Maritime surveillance with multiple over-the-horizon HFSW radars: An overview of recent experimentation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 12, pp. 4–18, Dec. 2015.

[159] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.

[160] S. Davey, N. Gordon, I. Holland, M. Rutten, and J. Williams, *Bayesian Methods in the Search for MH370*. New York City, NY: Springer, 2016.

[161] M. Z. Win, A. Conti, S. Mazuelas, Y. Shen, W. M. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[162] Y. Shen and M. Z. Win, "Fundamental limits of wideband localization – Part I: A general framework," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4956–4980, Oct. 2010.

[163] Y. Shen, H. Wymeersch, and M. Z. Win, "Fundamental limits of wideband localization – Part II: Cooperative networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4981–5000, Oct. 2010.

[164] M. Z. Win, Y. Shen, and W. Dai, "Theoretical foundation of network localization and navigation," *Proc. IEEE*, 2018, special issue on *Foundations and Trends in Localization Technologies*.

[165] F. Meyer, B. Etzlinger, Z. Liu, F. Hlawatsch, and M. Z. Win, "Scalable cooperative synchronization and navigation for the Internet of Things," *IEEE Internet Things J.*, 2018, to appear.

[166] E. Masazade, R. Niu, and P. K. Varshney, "Dynamic bit allocation for object tracking in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5048–5063, Oct. 2012.

[167] T. Wang, Y. Shen, A. Conti, and M. Z. Win, "Network navigation with scheduling: Error evolution," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7509–7534, Nov. 2017.

[168] W. Dai, Y. Shen, and M. Z. Win, "A computational geometry framework for efficient network localization," *IEEE Trans. Inf. Theory*, vol. IEEE Xplore Early Access, 2018.

[169] H. Godrich, A. P. Petropulu, and H. V. Poor, "Power allocation strategies for target localization in distributed multiple-radar architectures," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3226–3240, Jul. 2011.

[170] M. Z. Win, W. Dai, and Y. Shen, "Network operation for localization and navigation," *Proc. IEEE*, 2018, special issue on *Foundations and Trends in Localization Technologies*.

[171] X. Shen and P. K. Varshney, "Sensor selection based on generalized information gain for target tracking in large sensor networks," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 363–375, Jan. 2014.

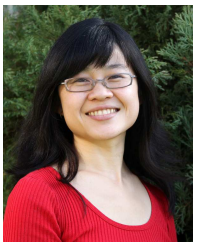
[172] F. Meyer, H. Wymeersch, M. Fröhle, and F. Hlawatsch, "Distributed estimation with information-seeking control in agent networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, Nov. 2015.
[173] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *Int. J. Robot. Res.*, vol. 31, no. 10, pp. 1134–1154, Sep. 2012.
[174] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experimentation for cooperative localization," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 467–475, Feb. 2012.
[175] S. Bartoletti, A. Giorgetti, M. Z. Win, and A. Conti, "Blind selection of representative observations for sensor radar networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1388–1400, Apr. 2015.
[176] A. Conti, D. Dardari, M. Guerra, L. Mucchi, and M. Z. Win, "Experimental characterization of diversity navigation," *IEEE Syst. J.*, vol. 8, no. 1, pp. 115–124, Mar. 2014.
[177] X. Wang, L. Gao, and S. Mao, "BiLoc: Bi-modal deep learning for indoor localization with commodity 5GHz WiFi," *IEEE Access*, vol. 5, pp. 4209–4220, 2017.
[178] B. Teague, Z. Liu, F. Meyer, and M. Z. Win, "Peregrine: 3-D network localization and navigation," in *Proc. IEEE Latin-American Conf. Commun.*, Guatemala City, Guatemala, Nov. 2017, pp. 1–6.
[179] Z. Liu, W. Dai, and M. Z. Win, "Mercury: An infrastructure-free system for network localization and navigation," *IEEE Trans. Mobile Comput.*, vol. IEEE Xplore Early Access, 2018.
[180] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 2000–2016, May 2009.
[181] ——, "DILAND: An algorithm for distributed sensor localization with noisy distance measurements," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1940–1947, Mar. 2010.
[182] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. Hoboken, NJ: Wiley, 2007.

Florian Meyer (S'12–M'15) received the Dipl.-Ing. (M.Sc.) and Ph.D. degrees in electrical engineering from TU Wien, Vienna, Austria in 2011 and 2015, respectively. He was a visiting researcher with the Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden in 2013 and with the NATO Centre of Maritime Research and Experimentation (CMRE), La Spezia, Italy in 2014 and 2015. In 2016 he joint CMRE as a research scientist. Currently, Dr. Meyer is an Erwin Schrödinger Fellow at the Massachusetts Institute of Technology (MIT). His research interests include signal processing for wireless sensor networks, localization and tracking, information-seeking control, message passing algorithms, and finite set statistics. Dr. Meyer served as reviewer for several IEEE journals and as a TPC member in IEEE conferences.

Thomas Kropfreiter received the B.Sc. degree in electrical engineering and the Dipl.-Ing degree (M.Sc. equivalent) in telecommunication engineering with distinction from TU Wien, Vienna, Austria, in 2012 and 2014, respectively. From 2013 to 2015, he was a member of the Mobile Communications Group, TU Wien, where he worked on the topic of beamforming in multi-cell scenarios. Since 2015, he has been a member of the Signal Processing Group, TU Wien, where he is currently pursuing the Ph.D. degree. His research interests include multi-object tracking, distributed signal processing in wireless sensor networks, message passing algorithms, and finite set statistics.

Jason L. Williams (S'01–M'07–SM'16) received degrees of BE(Electronics)/BInfTech from Queensland University of Technology in 1999, MSEE from the United States Air Force Institute of Technology in 2003, and PhD from Massachusetts Institute of Technology in 2007.

He worked for several years as an engineering officer in the Royal Australian Air Force, before joining Australia's Defence Science and Technology Group (formerly Defence Science and Technology Organisation) in 2007. He is also an adjunct associate professor at Queensland University of Technology. His research interests include target tracking, sensor resource management, Markov random fields and convex optimisation.



Roslyn A. Lau (S'14) received the degrees of BE (computer systems)/BMa&CS (statistics) in 2005, and MS (signal processing) in 2009, all from the University of Adelaide, Adelaide, Australia.

She is currently a PhD candidate at the Australian National University. She is also a scientist at the Defence Science and Technology Group, Australia. Her research interests include target tracking, probabilistic graphical models, and variational inference.

Franz Hlawatsch (S'85–M'88–SM'00–F'12) received the Diplom-Ingenieur, Dr. techn., and Univ.-Dozent (habilitation) degrees in electrical engineering/signal processing from TU Wien, Vienna, Austria in 1983, 1988, and 1996, respectively. Since 1983, he has been with the Institute of Telecommunications, TU Wien, where he is currently an Associate Professor. During 1991–1992, as a recipient of an Erwin Schrödinger Fellowship, he spent a sabbatical year with the Department of Electrical Engineering, University of Rhode Island, Kingston, RI, USA. In 1999, 2000, and 2001, he held one-month Visiting Professor positions with INP/ENSEEIHT, Toulouse, France and IRCCyN, Nantes, France. He (co)authored a book, three review papers that appeared in the IEEE SIGNAL PROCESSING MAGAZINE, about 200 refereed scientific papers and book chapters, and three patents. He coedited three books. He was Technical Program Co-Chair of EUSIPCO 2004 and served on the technical committees of numerous IEEE conferences. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2003 to 2007 and for the IEEE TRANSACTIONS ON INFORMATION THEORY from 2008 to 2011. From 2004 to 2009, he was a member of the IEEE SPCOM Technical Committee. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. He coauthored papers that won an IEEE Signal Processing Society Young Author Best Paper Award and a Best Student Paper Award at IEEE ICASSP 2011. His research interests include statistical and compressive signal processing methods and their application to sensor networks and wireless communications.

Paolo Braca (M'14–SM'17) received the Laurea degree (summa cum laude) in electronic engineering, and the Ph.D. degree (highest rank) in information engineering from the University of Salerno, Italy, in 2006 and 2010, respectively. In 2009, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA. In 2010?2011, he was a Postdoctoral Associate with the University of Salerno, Italy. In 2011, Dr. Braca joined the NATO Science & Technology Organization Centre for Maritime Research and Experimentation (CMRE), and currently is a Senior Scientist with the Research Department. He conducts research in the general area of statistical signal processing with emphasis on detection and estimation theory, wireless sensor network, multi-agent algorithms, target tracking and data fusion, adaptation and learning over graphs, radar (sonar) signal processing. He is coauthor of about one hundred publications in international scientific journals and conference proceedings. In 2017 Dr. Braca was awarded with National Scientific Qualification to function as associate professor in Italian Universities. Dr. Braca serves as an Associate Editor of IEEE Transactions on Signal Processing, IEEE Transactions on Aerospace and Electronic Systems, ISIF Journal of Advances in Information Fusion, EURASIP Journal on Advances in Signal Processing. In 2017 he was Lead Guest Editor of the special issue "Sonar Multi-Sensor Applications and Techniques" in IET Radar Sonar and Navigation. He served as Associate Editor of IEEE Signal Processing Magazine (E-Newsletter) from 2014 to 2016. He is in the Technical Committee of the major international conferences in the field of signal processing and data fusion. He was the recipient of the Best Student Paper Award (first runner-up) at the 12th International Conference on Information Fusion in 2009.

Moe Z. Win (S'85–M'87–SM'97–F'04) is a professor at the Massachusetts Institute of Technology (MIT) and the founding director of the Wireless Information and Network Sciences Laboratory. Prior to joining MIT, he was with AT&T Research Laboratories and NASA Jet Propulsion Laboratory.

His research encompasses fundamental theories, algorithm design, and network experimentation for a broad range of real-world problems. His current research topics include network localization and navigation, network interference exploitation, and quantum information science. He has served the IEEE Communications Society as an elected Member-at-Large on the Board of Governors, as elected Chair of the Radio Communications Committee, and as an IEEE Distinguished Lecturer. Over the last two decades, he held various Editorial posts for IEEE journals and organized numerous international conferences. Currently, he is serving on the SIAM Diversity Advisory Committee.

Dr. Win is an elected Fellow of the AAAS, the IEEE, and the IET. He was honored with two IEEE Technical Field Awards: the IEEE Kiyo Tomiyasu Award (2011) and the IEEE Eric E. Sumner Award (2006, jointly with R. A. Scholtz). Together with students and colleagues, his papers have received numerous awards. Other recognitions include the IEEE Communications Society Edwin H. Armstrong Achievement Award (2016), the International Prize for Communications Cristoforo Colombo (2013), the Copernicus Fellowship (2011) and the *Laurea Honoris Causa* (2008) from the University of Ferrara, and the U.S. Presidential Early Career Award for Scientists and Engineers (2004). He is an ISI Highly Cited Researcher.