Multi-Sensor Multi-Object Tracking With the Generalized Labeled Multi-Bernoulli Filter

Ba-Ngu Vo, Ba-Tuong Vo, and Michael Beard

***Abstract*—This paper proposes an efficient implementation of the multi-sensor generalized labeled multi-Bernoulli (GLMB) filter. Like its single-sensor counterpart, such implementation requires truncating the GLMB sum. However the single-sensor case requires solving 2-D ranked assignment problems whereas the multi-sensor case require solving multi-dimensional ranked assignment problems, which are NP-hard. The proposed implementation exploits the GLMB joint prediction and update together with a new technique for truncating the GLMB filtering density based on Gibbs sampling. The resulting algorithm has a quadratic complexity in the number of hypothesized objects and linear in the total number of measurements from all sensors.**

***Index Terms*—State estimation, Filtering, Random finite sets, Multi-dimensional assignment, Gibbs sampling.**

I. Introduction

THE objective of multi-object tracking is to jointly estimate the number of objects and their trajectories from sensor data [1]–[4]. Amongst a host of algorithms, Joint Probabilistic Data Association (JPDA) [1], Multiple Hypotheses Tracking (MHT) [2], and Random Finite Set (RFS) [3], [4] are regarded as the three main paradigms for multi-object tracking. Using data from multiple sensors, in principle, reduces uncertainty on the number of objects and their states, yielding improved multi-object tracking performance [1]–[4]. For a comprehensive overview of multi-sensor multi-object tracking techniques, we refer the reader to [5] and references therein. In this work, we are interested in multi-object filters that compute estimates on-line as data arrives, since these are well-suited for time-critical applications.

Many of the recent multi-sensor multi-object filters use the random finite set (RFS) framework [3], [4]. One of the few exceptions is a JPDA type filter that is quadratic in the number of targets, linear in the number of sensors, and linear in the number of measurements per sensor [6] . Multi-sensor versions of RFS-based filters such as the Probability Hypothesis Density (PHD) [7], Cardinalized PHD (CPHD) [8], multi-Bernoulli [3], [9], have been developed in [4], [10], and recently in [11] for hybrid multi-Bernoulli. These solutions have numerical complexities that are combinatorial in the number of measurements, and (with the exception of the hybrid multi-Bernoulli filter) linear in the number of objects. The earliest and conceptually simplest approximate multi-sensor PHD, CPHD (and multi-Bernoulli) filters are the heuristic "iterated corrector" versions that apply single-sensor updates, once for each sensor in turn [12], [13]. This approach yields final solutions that depend on the order in which the sensors are processed. Principled approximations of the multi-sensor PHD and CPHD filters that are computationally tractable, and independent of sensor order have been proposed in [4] (Section 10.6). However, this approach and the heuristic "iterated corrector" involve two levels of approximations since the exact multi-sensor PHD, CPHD and multi-Bernoulli filters are in fact approximations of the Bayes multi-sensor multi-object filter. Note that the multi-object filters discussed thus far are not trackers in the sense that only the current states are estimated, not their trajectories.

An analytic Bayes multi-object filter that estimates multi-object trajectories is the Generalized Labeled Multi-Bernoulli (GLMB) filter [14], [15]. The major hurdle in the multi-sensor GLMB filter implementation is the NP-hard multi-dimensional assignment problem. In principle, the "iterated corrector" strategy would yield the exact solution if all GLMB components are kept. However in practice truncation is performed at each single-sensor update, and an extremely large number of GLMB components at each single-sensor update would be needed, even if the final GLMB filtering density only contains a small number of significant components. Worse, insignificant components after one single-sensor update, which could become significant in the final GLMB filtering density, are discarded and cannot be recovered. An implementation of the two-sensor GLMB filter was developed in [16] using Murty's algorithm with a complexity of $\mathcal{O}((M^{(1)}M^{(2)})^4)$, where $M^{(s)}$ is the number of measurements from sensor s. A multi-sensor version of an approximate GLMB filter, known as the marginalized GLMB filter, was proposed in [17]. This solution has a complexity of $\mathcal{O}(\prod_{s=1}^{V}(M^{(s)})^4)$, where V is the number of sensors. While this approach is scalable in the number of sensors [17], it still involves two levels of approximations: the truncation of the GLMB density; and the functional approximation of the truncated GLMB density.

The multi-sensor multi-object filters discussed above were developed for a centralized fusion architecture, where measurements are sent to a central node for processing. Alternatively, in a decentralized setting, estimates and/or statistics computed

Manuscript received March 14, 2019; revised August 3, 2019 and September 15, 2019; accepted September 16, 2019. Date of publication October 7, 2019; date of current version November 19, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. David Ramirez. This work was supported by the Australian Research Council under Discovery Projects DP160104662 and DP170104854. *(Corresponding author: Ba Tuong Vo.)*

The authors are with the Department of Electrical and Computer Engineering, Curtin University, Bentley, WA 6102, Australia (e-mail: ba-ngu.vo@curtin.edu.au; ba-tuong.vo@curtin.edu.au; michael.beard@curtin.edu.au).

Digital Object Identifier 10.1109/TSP.2019.2946023

from individual sensors are fused together [18], [19]. Several decentralized multi-sensor fusion algorithms based on Generalized Covariance Intersection (GCI) [20] and its variants have been proposed for the PHD/CPHD filters [21]–[24] and multi-Bernoulli filter [25]–[28]. For multi-object tracking filters, GCI fusion rules for labeled multi-Bernoulli (LMB) and marginalized GLMB were derived and applied to distributed multi-sensor multi-object tracking via a sensor network in [29]. In [30] a new fusion rule for LMB was proposed by modifying the GCI fusion rule in [29] based on a Cauchy-Schwarz divergence criterion. However, tracking performance of GCI fusion can be sensitive to label inconsistencies between nodes, quantified by the so-called label inconsistency indicator [31]. A remedy was developed in [32] by seeking the best matching labels (that minimize the label inconsistency indicator) and performing label-wise GCI fusion with the matched multi-object densities. While fusion of multi-object densities alleviates the computational complexity associated with the centralized multi-sensor update and facilitates distributed multi-object tracking, the relationship between the fused density and the multi-sensor updated density is difficult to establish.

In this paper we present an efficient implementation of the (centralized) multi-sensor GLMB filter with the same quadratic complexity in the number of hypothesized objects as [16], [17], but linear in the total number of measurements from all sensors. The key lies in efficient solutions to multi-dimensional assignment problems. Unlike the multi-scan GLMB [33], in multi-sensor GLMB the multi-dimensional assignment problems can be solved by exploiting certain structural properties and suitable adaptation of the 2-D assignment Gibbs sampler of [34]. This approach generalizes our preliminary result in [35], and is generally quadratic in the number of hypothesized objects and linear in the product of the number of measurements. More importantly, we further develop a very practical and scalable solution that drastically reduces the complexity to being linear in the total number of measurements across the sensors.

The remainder of this article is organized as follows. Section II presents the background on GLMB filtering and its multi-sensor extension. In Section III we present our proposed Gibbs sampling based approach to the truncation of the multi-sensor GLMB filtering density, and the resulting implementation of the multi-sensor GLMB filter. Numerical studies are presented in Section IV, followed by some concluding remarks in Section V.

II. Background

This section summarizes the multi-object state space models and the GLMB filter. Adhering to the single-sensor GLMB filter implementation in [34] we adopt the following notation; the inner product $\langle f,g\rangle \triangleq \int f(x)g(x)dx$, the list of variables $X_{m:n} \triangleq X_m, X_{m+1}, \ldots, X_n$, and the generalized Kroneker delta that takes arbitrary arguments

$$\delta_Y[X] \triangleq \begin{cases} 1, & \text{if } X = Y \\ 0, & \text{otherwise} \end{cases}.$$

For a given set A, we denote its indicator function by $1_A(\cdot)$, and the class of finite subsets of A by $\mathcal{F}(A)$. For a finite set X, we denote its cardinality (or number of elements) by $|X|$, and the product $\prod_{x\in X} f(x)$ by f^X, with $f^{\emptyset}=1$.

A. Multi-Object State

An existing object at time k is represented by a vector x_k in some state space $\mathbb{X}$, and a unique label ℓ_k consisting of an ordered pair (t,α), where t is the time of birth and α is the index of individual objects born at the same time [14]. The label space for all objects up to time k (including those born prior to k) is the disjoint union $\mathbb{L}_k = \biguplus_{t=0}^{k} \mathbb{B}_t$, where $\mathbb{B}_t$ denotes the label space for objects born at time t, (note that $\mathbb{L}_k = \mathbb{L}_{k-1} \uplus \mathbb{B}_k$). Formally, the *labeled state* of an object at time k is a vector $\boldsymbol{x}_k = (x_k, \ell_k) \in \mathbb{X}\times\mathbb{L}_k$, and the *trajectory* of an object is a sequence of consecutive labeled states with a common label [14].

Suppose that there are N_k objects at time k, with states $\boldsymbol{x}_{k,1},\ldots,\boldsymbol{x}_{k,N_k}$, in the context of multi-object tracking, the collection of states, referred to as the *multi-object state*, is naturally represented as a finite set

$$\boldsymbol{X}_k = \{\boldsymbol{x}_{k,1},\ldots,\boldsymbol{x}_{k,N_k}\} \in \mathcal{F}(\mathbb{X}\times\mathbb{L}_k).$$

We denote the (set of) labels of $\boldsymbol{X}$, i.e. $\{\ell : (x,\ell)\in\boldsymbol{X}\}$, by $\mathcal{L}(\boldsymbol{X})$. Note that since no two objects in a multi-object state have the same label, $\delta_{|\boldsymbol{X}|}(|\mathcal{L}(\boldsymbol{X})|)=1$. Hence, we define the *distinct label indicator* as

$$\Delta(\boldsymbol{X}) \triangleq \delta_{|\boldsymbol{X}|}[|\mathcal{L}(\boldsymbol{X})|].$$

In what follows, we use the convention that single-object states are represented by lower-case letters (e.g. x, $\boldsymbol{x}$), while multi-object states are represented by upper-case letters (e.g. X, $\boldsymbol{X}$), symbols for labeled states and their distributions are bold-faced to distinguish them from unlabeled ones (e.g. $\boldsymbol{x}$, $\boldsymbol{X}$, $\boldsymbol{\pi}$, etc.), spaces are represented by blackboard bold (e.g. $\mathbb{X}$, $\mathbb{Z}$, $\mathbb{L}$, $\mathbb{N}$, etc.). Also, for notational compactness, we drop the subscript k for the current time, the next time is indicated by the subscript '$+$'.

B. Standard Multi-Object Dynamic Model

Given the multi-object state $\boldsymbol{X}$ (at time k), each $(x,\ell)\in\boldsymbol{X}$ either survives with probability $P_S(x,\ell)$ and evolves to a new state (x_+,ℓ_+) (at time $k+1$) with probability density $f_+(x_+|x,\ell)\delta_\ell[\ell_+]$ or dies with probability $1-P_S(x,\ell)$. The set $\boldsymbol{B}_+$ of new objects (born at time $k+1$) is distributed according to the labeled multi-Bernoulli (LMB) density[1]

$$\boldsymbol{f}_{B,+}(\boldsymbol{B}_+) = \Delta(\boldsymbol{B}_+)\left[1_{\mathbb{B}_+} r_{B,+}\right]^{\mathcal{L}(\boldsymbol{B}_+)}\left[1-r_{B,+}\right]^{\mathbb{B}_+-\mathcal{L}(\boldsymbol{B}_+)} p_{B,+}^{\boldsymbol{B}_+},$$

where $r_{B,+}(\ell)$ is the probability that a new object with label ℓ is born, and $p_{B,+}(\cdot,\ell)$ is the distribution of its kinematic state [14]. The multi-object state $\boldsymbol{X}_+$ (at time $k+1$) is the superposition

[1] Note that in this work we use Mahler's set derivatives for multi-object densities [3], [8]. While these are not actual probability densities, they are equivalent to probability densities relative to a certain reference measure [36].

of surviving objects and new born objects. Using the standard assumption that, conditional on $\boldsymbol{X}$, objects move, appear and die independently of each other, the expression for the multi-object transition density $\boldsymbol{f}_+$ is given by [14], [15]

$$\begin{aligned}&\boldsymbol{f}_+\left(\boldsymbol{X}_+|\boldsymbol{X}\right) =\\ &\boldsymbol{f}_{S,+}\left(\boldsymbol{X}_+\cap\left(\mathbb{X}\times\mathbb{L}\right)|\boldsymbol{X}\right)\boldsymbol{f}_{B,+}\left(\boldsymbol{X}_+-\left(\mathbb{X}\times\mathbb{L}\right)\right)\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{f}_{S,+}\left(\boldsymbol{W}|\boldsymbol{X}\right) &= \Delta\left(\boldsymbol{W}\right)\Delta\left(\boldsymbol{X}\right)1_{\mathcal{L}(\boldsymbol{X})}\left(\mathcal{L}\left(\boldsymbol{W}\right)\right)\left[\Phi\left(\boldsymbol{W};\cdot\right)\right]^{\boldsymbol{X}}\\ \Phi\left(\boldsymbol{W};x,\ell\right) &= \left(1-1_{\mathcal{L}(\boldsymbol{W})}\left(\ell\right)\right)\left(1-P_S\left(x,\ell\right)\right)\\ &\quad+\sum_{(x_+,\ell_+)\in\boldsymbol{W}}\delta_\ell\left[\ell_+\right]P_S\left(x,\ell\right)f_+\left(x_+|x,\ell\right).\end{aligned}$$

C. Standard Multi-Object Observation Model

Suppose that there are V sensors, numbered from 1 to V. For a given multi-object state $\boldsymbol{X}$, each $\boldsymbol{x}\in\boldsymbol{X}$ is either detected by sensor $s\in\{1:V\}$, with probability $P_D^{(s)}(\boldsymbol{x})$ and generates a detection $z^{(s)}\in Z^{(s)}$ with likelihood $g_D^{(s)}(z^{(s)}|\boldsymbol{x})$ or missed with probability $1-P_D^{(s)}(\boldsymbol{x})$. The *multi-object observation* from sensor s is the superposition of the observations from detected objects and Poisson clutter with intensity $\kappa^{(s)}$. The standard multi-object likelihood function for sensor s is given by [14], [15]

$$g^{(s)}(Z^{(s)}|\boldsymbol{X})\propto\sum_{\theta^{(s)}\in\Theta^{(s)}}1_{\Theta^{(s)}(\mathcal{L}(\boldsymbol{X}))}(\theta^{(s)})\left[\psi_{Z^{(s)}}^{(s,\theta^{(s)}\circ\mathcal{L}(\cdot))}(\cdot)\right]^{\boldsymbol{X}} \tag{1}$$

where: $\Theta^{(s)}$ is the set of positive 1–1 maps $\theta^{(s)}:\mathbb{L}\to\{0:|Z^{(s)}|\}$, i.e. maps such that *no two distinct arguments are mapped to the same positive value*; $\Theta^{(s)}(I)$ is the subset of $\Theta^{(s)}$ with domain I; $\theta^{(s)}\circ\mathcal{L}(\boldsymbol{x})=\theta^{(s)}(\mathcal{L}(\boldsymbol{x}))$; and

$$\psi_{\{z_{1:M^{(s)}}\}}^{(s,j)}(\boldsymbol{x})=\begin{cases}\frac{P_D^{(s)}(\boldsymbol{x})g^{(s)}(z_j|\boldsymbol{x})}{\kappa^{(s)}(z_j)}, & j=1{:}M^{(s)}\\ 1-P_D^{(s)}(\boldsymbol{x}), & j=0\end{cases}. \tag{2}$$

The map $\theta^{(s)}$ specifies that object ℓ generates detection $z_{\theta(\ell)}\in Z^{(s)}$, with undetected objects assigned to 0. The positive 1–1 property means that $\theta^{(s)}$ is 1–1 on the set of detected labels, i.e. $\{\ell:\theta^{(s)}(\ell)>0\}$, and ensures that any detection in $Z^{(s)}$ is assigned to at most one object.

Using the following abbreviations

$$Z\triangleq(Z^{(1)},\ldots,Z^{(V)}), \tag{3}$$

$$\Theta\triangleq\Theta^{(1)}\times\cdots\times\Theta^{(V)}, \tag{4}$$

$$\Theta(I)\triangleq\Theta^{(1)}(I)\times\cdots\times\Theta^{(V)}(I), \tag{5}$$

$$\theta\triangleq(\theta^{(1)},\ldots,\theta^{(V)}), \tag{6}$$

$$1_{\Theta(I)}\left(\theta\right)\triangleq\prod_{s=1}^{V}1_{\Theta^{(s)}(I)}(\theta^{(s)}), \tag{7}$$

$$\psi_Z^{(j^{(1)},\ldots,j^{(V)})}\left(x,\ell\right)\triangleq\prod_{s=1}^{V}\psi_{Z^{(s)}}^{(s,j^{(s)})}\left(x,\ell\right), \tag{8}$$

and the standard assumption that the sensors are conditionally independent,[2] the multi-sensor likelihood is given by

$$\begin{aligned}g\left(Z|\boldsymbol{X}\right) &= \prod_{s=1}^{V}g^{(s)}(Z^{(s)}|\boldsymbol{X})\\ &\propto\sum_{\theta\in\Theta}1_{\Theta(\mathcal{L}(\boldsymbol{X}))}\left(\theta\right)\left[\psi_Z^{(\theta\circ\mathcal{L}(\cdot))}\left(\cdot\right)\right]^{\boldsymbol{X}},\end{aligned} \tag{9}$$

which has the same form as its single-sensor counterpart. The multi-sensor association map θ is said to be positive 1–1, since all constituent $\theta^{(1)},\ldots,\theta^{(V)}$ are positive 1–1.

D. Generalized Label Multi-Bernoulli (GLMB)

A GLMB density, or simply GLMB, is a labeled multi-object density of the form[3]

$$\boldsymbol{\pi}\left(\boldsymbol{X}\right)=\Delta\left(\boldsymbol{X}\right)\sum_{I,\xi}\omega^{(I,\xi)}\delta_I\left[\mathcal{L}\left(\boldsymbol{X}\right)\right]\left[p^{(\xi)}\right]^{\boldsymbol{X}}, \tag{10}$$

where: $I\in\mathcal{F}(\mathbb{L})$; each $\xi\in\Xi$ represents a history of (multi-sensor) association maps, i.e. $\xi=(\theta_{1:k})$; each $p^{(\xi)}(\cdot,\ell)$ is a probability density on $\mathbb{X}$; and each $\omega^{(I,\xi)}$ is non-negative with

$$\sum_{I,\xi}\omega^{(I,\xi)}=1.$$

The cardinality distribution of a GLMB is given by

$$\Pr\left(|\boldsymbol{X}|=n\right)=\sum_{I,\xi}\delta_n\left[|I|\right]\omega^{(I,\xi)},$$

while the existence probability and probability density for the track with label $\ell\in\mathbb{L}$ are respectively given by

$$\begin{aligned}r\left(\ell\right) &= \sum_{I,\xi}1_I\left(\ell\right)\omega^{(I,\xi)},\\ p\left(x,\ell\right) &= \frac{1}{r\left(\ell\right)}\sum_{I,\xi}1_I\left(\ell\right)\omega^{(I,\xi)}p^{(\xi)}\left(x,\ell\right).\end{aligned}$$

Various multi-object estimators for GLMBs are discussed in [14], [15]. The most popular is a suboptimal version of the marginal multi-object estimator [3], which: first, determines the pair (L,ξ) with the highest weight $\omega^{(L,\xi)}$ such that $|L|$ coincides with the mode (most probable) cardinality; and second, compute the state estimate for each object (with label) $\ell\in L$ from $p^{(\xi)}(\cdot,\ell)$, e.g. the mode or the mean.

Remark: For GLMBs with $\xi=(\theta_{1:k})$, this estimator encompasses the entire trajectory of each object, because for each $\ell=(t,\alpha)\in L$ the initial state distribution is given by $p_{B,t}(\cdot,\ell)$ in the LMB birth model, and the sequence of associated measurements is given by $\theta_{1:k}(\ell)$. Such information is sufficient to

[2] More concisely, given the (multi-object) state, the uncertainty (due to measurement noise, misdetections, and clutter) from each sensor is independent from the others. This condition is valid when the sensors do not interfere nor influence each other in the process of obtaining the measurements or detections.

[3] In fact this is the δ-form of the GLMB, known as the δ-GLMB. In this paper we only use this form and hence, the prefix δ is omitted.

determine the joint density of the states along the object's trajectory. Moreover, all marginals of this joint density can be computed recursively from $p^{(\xi)}(\cdot,\ell)$ using the forward-backward algorithm.

E. Multi-Sensor GLMB Recursion

All information on the multi-object state is captured in the multi-object filtering density, which can be propagated forward recursively by the (multi-object) Bayes filter [3], [4]. Usually this recursion is decomposed into two separate steps, namely prediction and update [3], [4]. This work employs the joint prediction and update form that combines these two steps into a single expression

$$\boldsymbol{\pi}_+(\boldsymbol{X}_+) \propto g(Z_+|\boldsymbol{X}_+)\int \boldsymbol{f}_+(\boldsymbol{X}_+|\boldsymbol{X})\,\boldsymbol{\pi}(\boldsymbol{X})\,\delta\boldsymbol{X}, \quad (11)$$

where: $\boldsymbol{\pi}$ and $\boldsymbol{\pi}_+$ denote the multi-object filtering densities at times k and $k+1$, respectively; while the integral is the set integral given in [14]. Note that for notational compactness we have omitted the measurement histories from these densities.

The GLMB filter is an analytic solution to the Bayes multi-object filter (11) under the standard multi-object dynamic and observation models [15]. Since the multi-sensor likelihood function has the same form as the single-sensor case, it follows from [34] that given the filtering density (10) at time k, the filtering density at time $k+1$ is given by

$$\boldsymbol{\pi}_+(\boldsymbol{X}) \propto$$
$$\Delta(\boldsymbol{X})\sum_{I,\xi,I_+,\theta_+}\omega^{(I,\xi)}\omega_{Z_+}^{(I,\xi,I_+,\theta_+)}\delta_{I_+}[\mathcal{L}(\boldsymbol{X})]\left[p_{Z_+}^{(\xi,\theta_+)}\right]^{\boldsymbol{X}} \quad (12)$$

where $I\in\mathcal{F}(\mathbb{L})$, $\xi\in\Xi$, $I_+\in\mathcal{F}(\mathbb{L}_+)$, $\theta_+\in\Theta_+(I_+)$, and

$$\omega_{Z_+}^{(I,\xi,I_+,\theta_+)} = 1_{\Theta_+(I_+)}(\theta_+)\left[1-\bar{P}_S^{(\xi)}\right]^{I-I_+}\left[\bar{P}_S^{(\xi)}\right]^{I\cap I_+}$$
$$\times\,[1-r_{B,+}]^{\mathbb{B}_+-I_+}\,r_{B,+}^{\mathbb{B}_+\cap I_+}\left[\bar{\psi}_{Z_+}^{(\xi,\theta_+))}\right]^{I_+} \quad (13)$$

$$\bar{P}_S^{(\xi)}(\ell) = \left\langle p^{(\xi)}(\cdot,\ell),P_S(\cdot,\ell)\right\rangle \quad (14)$$

$$\bar{\psi}_{Z_+}^{(\xi,\theta_+)}(\ell_+) = \left\langle \bar{p}_+^{(\xi)}(\cdot,\ell_+),\psi_{Z_+}^{(\theta_+(\ell_+))}(\cdot,\ell_+)\right\rangle \quad (15)$$

$$\bar{p}_+^{(\xi)}(x_+,\ell_+)$$
$$= 1_{\mathbb{L}}(\ell_+)\frac{\left\langle P_S(\cdot,\ell_+)f_+(x_+|\cdot,\ell_+),p^{(\xi)}(\cdot,\ell_+)\right\rangle}{\bar{P}_S^{(\xi)}(\ell_+)}$$
$$+\,1_{\mathbb{B}_+}(\ell_+)\,p_{B,+}(x_+,\ell_+) \quad (16)$$

$$p_{Z_+}^{(\xi,\theta_+)}(x_+,\ell_+) = \frac{\bar{p}_+^{(\xi)}(x_+,\ell_+)\,\psi_{Z_+}^{(\theta_+(\ell_+))}(x_+,\ell_+)}{\bar{\psi}_{Z_+}^{(\xi,\theta_+)}(\ell_+)}. \quad (17)$$

The number of components in the GLMB filtering density grows exponentially with time, and needs to be truncated at every time step. Truncation by retaining components with the largest weights minimizes the L_1 approximation error [15], and can be formulated as a multi-dimensional assignment problem [37]. This problem is NP-hard for more than two dimensions. A multi-dimensional assignment problem with 5 dimensions and 20 measurements per dimension is equivalent to an integer linear programming problem with 3.2 million variables, see for example [38] and the references therein.

III. Multi-Sensor GLMB Filter Implementation

This section presents efficient implementations of the multi-sensor GLMB filter based on truncation of the filtering density (12). Following [34], we consider truncation by sampling the GLMB components (I,ξ,I_+,θ_+) from a discrete probability distribution π. Specifically, we consider

$$\pi(I,\xi,I_+,\theta_+) \propto \omega^{(I,\xi)}\pi(I_+,\theta_+|I,\xi), \quad (18)$$

where, for a given (I,ξ), $\pi(I_+,\theta_+|I,\xi)$ is approximately proportional[4] to $\omega_{Z_+}^{(I,\xi,I_+,\theta_+)}$. The rationale is that $\pi(I,\xi,I_+,\theta_+)$ would then be approximately proportional to the weight $\omega^{(I,\xi)}\omega_{Z_+}^{(I,\xi,I_+,\theta_+)}$ of component (I,ξ,I_+,θ_+) in the GLMB filtering density (12), thereby ensuring that sampling from $\pi(I,\xi,I_+,\theta_+)$ would generate high-weight components.

To draw a sample from (18), we first sample (I,ξ) from $\pi(I,\xi)\propto\omega^{(I,\xi)}$, and second sample (I_+,θ_+) from $\pi(I_+,\theta_+|I,\xi)$. The first operation is straight forward, the challenge lies in the second, which we address in subsection III-B and III-C via Gibbs sampling. Details of the multi-sensor GLMB filter implementation are outlined in subsection III-D.

To facilitate the Gibbs sampling formulation, we first start with a convenient representation of association maps in the following subsection.

A. Extended Association Map

Recall the single-sensor case [34], where there are M measurements. Given a component (I,ξ), the pair (I_+,θ_+) is represented by the extended association map

$$\gamma: I\uplus\mathbb{B}_+\to\{-1:M\},$$

defined by: $\gamma(\ell)=\theta_+(\ell)$, if $\ell\in I_+$, i.e. a live label; and $\gamma(\ell)=-1$, if ℓ is a dead/unborn label. An obvious generalization of extended association map to the multi-sensor case (where each sensor has $M^{(s)}$ measurements) is

$$\gamma: I\uplus\mathbb{B}_+\to\{-1:M^{(1)}\}\times\cdots\times\{-1:M^{(V)}\}.$$

defined by $\gamma(\ell)=(\gamma^{(1)}(\ell),\ldots,\gamma^{(V)}(\ell))$, where: $\gamma^{(s)}(\ell)=\theta_+^{(s)}(\ell)$, if $\ell\in I_+$; and $\gamma^{(s)}(\ell)=-1$, otherwise. However, this representation allows pathological cases, e.g. extended association maps with $\gamma(\ell)=(-1,1)$ in a two-sensor scenario, meaning that label ℓ is dead/unborn but generated measurement 1 in sensor 2. Hence, additional constructs are needed to design Gibbs samplers that do not generate such pathological extended association maps.

This problem can be avoided altogether by representing each pair $(I_+,\theta_+)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_+(I_+)$ of a given component (I,ξ),

[4] We say that two (unnormalized) distributions are *approximately proportional* when their normalized versions are approximately equal.

as the *extended association map*

$$\gamma \triangleq (\gamma^{(1)}, \ldots, \gamma^{(V)}) : I \uplus \mathbb{B}_+ \to \{-1\}^V \uplus \Lambda^{(1:V)},$$

where

$$\gamma(\ell) \triangleq \begin{cases} (\theta_+^{(1)}(\ell), \ldots, \theta_+^{(V)}(\ell)), & \text{if } \ell \in I_+ \\ (-1, \ldots, -1) & \text{otherwise} \end{cases},$$

$$\Lambda^{(s:t)} \triangleq \{0 : M^{(s)}\} \times \cdots \times \{0 : M^{(t)}\}.$$

Note that $\gamma(\ell)$ is a V-tuple that either belongs to $\Lambda^{(1:V)}$ when ℓ is a live label, or $\{-1\}^V$ when ℓ is a dead/unborn label, in which case $\gamma^{(s)}(\ell) = -1$ for all $s \in \{1 : V\}$. Given any γ in the set Γ of all positive 1–1 extended association maps, we can recover I_+ and $\theta_+ : I_+ \to \Lambda^{(1:V)}$, respectively, by

$$I_+ = \{\ell \in I \uplus \mathbb{B}_+ : \gamma(\ell) \in \Lambda^{(1:V)}\} \text{ and } \theta_+(\ell) = \gamma(\ell). \quad (19)$$

Thus, there is a 1–1 correspondence between the spaces $\Theta_+(I_+)$ and Γ, moreover

$$1_\Gamma(\gamma) = 1_{\Theta_+(I_+)}(\theta_+).$$

Hereon, we enumerate $I \uplus \mathbb{B}_+ = \{\ell_{1:P}\}$, and abbreviate

$$\gamma_n^{(s)} \triangleq \gamma^{(s)}(\ell_n) \in \{-1 : M^{(s)}\}$$

$$\gamma_n \triangleq (\gamma_n^{(1)}, \ldots, \gamma_n^{(V)}) \in \{-1\}^V \uplus \Lambda^{(1:V)},$$

for $n \in \{1 : P\}$, so that an extended association map γ can be represented as a $P \times V$ array in $(\{-1\}^V \uplus \Lambda^{(1:V)})^P$,

$$\gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_P \end{bmatrix} = \begin{bmatrix} \gamma_1^{(1)} & \gamma_1^{(2)} & \cdots & \gamma_1^{(V)} \\ \gamma_2^{(1)} & \gamma_2^{(2)} & \cdots & \gamma_2^{(V)} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_P^{(1)} & \gamma_P^{(2)} & \cdots & \gamma_P^{(V)} \end{bmatrix}. \quad (20)$$

The positive 1–1 property means that for each $s \in \{1 : V\}$ there are no distinct i, j with $\gamma_i^{(s)} = \gamma_j^{(s)} > 0$.

Similar to the single-sensor case, assuming that for each ℓ in $I \uplus \mathbb{B}_+$, the expected probabilities $\bar{P}_S^{(\xi)}(\ell) \in (0,1)$ and $\bar{P}_D^{(s,\xi)}(\ell) \triangleq \langle P_D^{(s)}(\cdot, \ell), \bar{p}_+^{(\xi)}(\cdot, \ell)\rangle \in (0,1)$, then it follows from (13) and (19) that

$$\omega_{Z_+}^{(I,\xi,I_+,\theta_+)} = 1_\Gamma(\gamma) \prod_{n=1}^{P} \eta_n(\gamma_n) \quad (21)$$

where

$$\eta_n(j^{(1:V)}) \triangleq \quad (22)$$

$$\begin{cases} 1 - \bar{P}_S^{(\xi)}(\ell_n), & \ell_n \in I, \quad j^{(1:V)} \in \{-1\}^V \\ \bar{P}_S^{(\xi)}(\ell_n)\, \bar{\psi}_{Z_+}^{(\xi, j^{(1:V)})}(\ell_n), & \ell_n \in I, \quad j^{(1:V)} \in \Lambda^{(1:V)} \\ 1 - r_{B,+}(\ell_n), & \ell_n \in \mathbb{B}_+, j^{(1:V)} \in \{-1\}^V \\ r_{B,+}(\ell_n)\, \bar{\psi}_{Z_+}^{(\xi, j^{(1:V)})}(\ell_n), & \ell_n \in \mathbb{B}_+, j^{(1:V)} \in \Lambda^{(1:V)} \end{cases}$$

$$\bar{\psi}_{Z_+}^{(\xi, j^{(1:V)})}(\ell_n) = \langle \bar{p}_+^{(\xi)}(\cdot, \ell_n), \psi_{Z_+}^{(j^{(1:V)})}(\cdot, \ell_n)\rangle \quad (23)$$

The assumptions on the expected survival and detection probabilities, $\bar{P}_S^{(\xi)}(\ell)$ and $\bar{P}_D^{(s,\xi)}(\ell)$, eliminate trivial and ideal sensing scenarios, as well as ensuring $\eta_n(j^{(1:V)}) > 0$. Note that $\eta_n(j^{(1:V)})$ depends on the given (I, ξ) and Z_+, which have been omitted for compactness.

Remark: In the single-sensor case, γ is a *P*-tuple, and GLMB truncation requires solving 2-D assignment problems [34]. In the multi-sensor case, γ is a $P \times V$ array, and hence GLMB truncation would require solving multi-dimensional assignment problems. Nonetheless, these problems can be solved by adapting the 2-D assignment Gibbs sampler of [34], and treating the rows of γ (which are elements of $\{-1\}^V \uplus \Lambda^{(1:V)}$) in the same way as the entries of γ for the single-sensor case (which are elements of $\{-1 : M\}$). In general, a multi-dimensional assignment problem may not necessarily be solved by the 2-D assignment Gibbs sampler. However, this is possible for multi-sensor assignment because for any live label there is no constraint between the columns of γ (representing the sensors). In contrast, the multi-dimensional assignment problems for multi-scan GLMB truncation have constraints on the columns (representing time) [33], and the 2-D assignment Gibbs sampler is no longer applicable.

B. GLMB Truncation via Gibbs Sampling

Recall that to generate high-weight components via sampling from (18), we first sample (I, ξ) from $\pi(I, \xi) \propto \omega^{(I,\xi)}$, and second sample (I_+, θ_+) from a distribution approximately proportional to $\omega_{Z_+}^{(I,\xi,I_+,\theta_+)}$. This subsection presents an approach for the second sampling step.

The extended association map representation (20) treats each(I_+, θ_+) as a $P \times V$ array in the space $(\{-1\}^V \uplus \Lambda^{(1:V)})^P$. Hence, sampling (I_+, θ_+) from a distribution approximately proportional to $\omega_{Z_+}^{(I,\xi,I_+,\theta_+)}$ amounts to sampling from a distribution on $(\{-1\}^V \uplus \Lambda^{(1:V)})^P$ that is approximately proportional to (21). Additionally, keeping in mind that all generated samples must be positive 1–1, we restrict ourselves to distributions of the same form as (21). Specifically, we consider distributions of the form

$$\pi(\gamma) \propto 1_\Gamma(\gamma) \prod_{n=1}^{P} \vartheta_n(\gamma_n), \quad (24)$$

where each $\vartheta_n : \{-1\}^V \uplus \Lambda^{(1:V)} \to [0, \infty)$ is chosen to approximate η_n, including the special case $\vartheta_n = \eta_n$.

Gibbs sampling for a stationary distribution π requires constructing a Markov chain with transition kernel [39], [40]

$$\pi(\gamma'|\gamma) = \prod_{n=1}^{P} \pi_n(\gamma_n'|\gamma_{1:n-1}', \gamma_{n+1:P}),$$

where $\pi_n(\gamma_n'|\gamma_{1:n-1}', \gamma_{n+1:P}) \propto \pi(\gamma_{1:n}', \gamma_{n+1:P})$. In other words, given the current state γ of the chain, the components $\gamma_1', \ldots, \gamma_P'$ of the next state γ' are distributed according to the sequence of conditionals

$$\pi_1\left(\gamma_1'|\gamma_{2:P}\right) \propto \pi\left(\gamma_1', \gamma_{2:P}\right)$$

$$\vdots$$

$$\pi_n\left(\gamma_n'|\gamma_{1:n-1}', \gamma_{n+1:P}\right) \propto \pi\left(\gamma_{1:n}', \gamma_{n+1:P}\right)$$

$$\vdots$$

$$\pi_P\left(\gamma_P'|\gamma_{1:P-1}'\right) \propto \pi\left(\gamma_{1:P}'\right).$$

Proposition 1: Let $\bar{n} = \{1:P\} - \{n\}$, and $\gamma_{\bar{n}} = (\gamma_{1:n-1}, \gamma_{n+1:P}) \in (\{-1\}^V \uplus \Lambda^{(1:V)})^{P-1}$ be positive 1–1. Then, the n-th conditional of the stationary distribution (24) is given by

$$\pi_n(j^{(1:V)}|\gamma_{\bar{n}}) \propto \vartheta_n(j^{(1:V)}) \prod_{s=1}^{V} \beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)}) \tag{25}$$

for $j^{(1:V)} \in \{-1\}^V \uplus \Lambda^{(1:V)}$, where

$$\beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)}) = 1 - 1_{\{1:M^{(s)}\} \bigcap \{\gamma_{1:n-1}^{(s)}, \gamma_{n+1:P}^{(s)}\}}(j^{(s)}). \tag{26}$$

The proof is given in Appendix A.

The above result shows that sampling from the conditional $\pi_n(\cdot|\gamma_{\bar{n}})$ amounts to sampling from a categorical distribution on $\{-1\}^V \uplus \Lambda^{(1:V)}$. Moreover, given a positive 1–1 $\gamma_{\bar{n}}$, the conditional $\pi_n(\cdot|\gamma_{\bar{n}})$ only generates γ_n such that γ is positive 1–1, otherwise the product in (25) vanishes and $\pi_n(\gamma_n|\gamma_{\bar{n}}) = 0$. Hence, starting with a positive 1–1 γ, all iterates of the Gibbs sampler, summarized in Algorithm 1, are also positive 1–1. Note from (25), (26) that $\pi_n(j^{(1:V)}|\gamma_{\bar{n}}) \propto \vartheta_n(j^{(1:V)})$ for $j^{(1:V)} \in \{-1\}^V$, and hence in Algorithm 1 we only need the for-loop over $j^{(1:V)} \in \Lambda^{(1:V)}$.

Further, as the number of iterates, T, tends to infinity the samples generated by Algorithm 1 are distributed according to the stationary distribution.

Proposition 2: If $\vartheta_n > 0$ on $\{-1\}^V \uplus \Lambda^{(1:V)}$ for each n, then starting from any positive 1–1 state, the Gibbs sampler defined by the family of conditionals (25) converges to the stationary distribution (24) at an exponential rate. More concisely, let π^j denote the j-th power of the transition kernel, then

$$\max_{\gamma,\gamma' \in \Gamma}\left(\left|\pi^j\left(\gamma'|\gamma\right) - \pi\left(\gamma'\right)\right|\right) \leq (1-2\beta)^{\left\lfloor \frac{j}{2} \right\rfloor},$$

where $\beta \triangleq \min_{\gamma,\gamma' \in \Gamma} \pi^2(\gamma'|\gamma) > 0$ is the least likely 2-step transition probability.

The proof follows along the same lines of arguments as Proposition 4 in [34].

Similar to the single-sensor case, increased efficiency can be achieved by using annealing or tempering techniques to modify the stationary distribution so as to induce the Gibbs sampler to seek more diverse samples [41], [42].

1) Optimal Stationary Distribution: Since each ϑ_n in the stationary distribution (24) should be chosen to approximate η_n, the *optimal stationary distribution* is obtained by setting

$$\vartheta_n\left(\gamma_n\right) = \eta_n\left(\gamma_n\right). \tag{27}$$

Algorithm 1: Gibbs (Optimal).

```
Inputs: $T, V, \gamma^{(1)} = [\gamma_n^{(1,s)}], \vartheta = [\vartheta_n(j^{(1:V)})]$
Outputs: $\gamma^{(1)}, \dots, \gamma^{(T)}$

$P = \mathsf{size}(\vartheta, 1),\ c = -1\mathsf{*ones}(1, V),$
for $s = 1:V$
    $M^{(s)} = \mathsf{size}(\vartheta, 1+s) - 2$
end for
for $j^{(1:V)} = \mathsf{zeros}(1, V) : [M^{(1:V)}]$
    $c = [c; [j^{(1:V)}]]$
end for
$[p_n(j^{(1:V)})] = [\vartheta_n(j^{(1:V)})]$
for $t = 2:T$
    $\phi^{(t)} = [\ ]$
    for $n = 1:P$
        for $j^{(1:V)} = \mathsf{zeros}(1, V) : [M^{(1:V)}]$
            $p_n(j^{(1:V)}) = p_n(j^{(1:V)}) \prod_{s=1}^{V} \beta_n^{(s)}$
            $(j^{(s)}|\phi_{1:n-1}^{(t,s)}, \gamma_{n+1:P}^{(t-1,s)})$
        end for
        $\phi_n^{(t)} \sim \mathsf{Categorical}(c, p_n),\quad \phi^{(t)} = [\phi^{(t)}; \phi_n^{(t)}]$
    end for
    $\gamma^{(t)} = \phi^{(t)}$
end for
```

In this case, the conditional $\pi_n(\cdot|\gamma_{\bar{n}})$ is a categorical distribution with $1 + \prod_{s=1}^{V}(M^{(s)} + 1)$ categories. Hence, sampling $\pi_n(\cdot|\gamma_{\bar{n}})$ in Algorithm 1 requires at least $1 + \prod_{s=1}^{V}(M^{(s)} + 1)$ memory locations. For example, 5 sensors with 15 measurements per sensor requires over $16^5 \simeq 1$ million memory locations. Moreover, Algorithm 1 incurs a complexity of $\mathcal{O}(TP\prod_{s=1}^{V} M^{(s)})$, since sampling from a categorical distribution is linear in the number of categories [43]. While this is orders of magnitude cheaper than Murty-based solutions [16], [17] such computational load is still prohibitive.

Sampling from the categorical distribution in Algorithm 1 can be replaced by a single iteration of the Metropolis-Hastings algorithm on the same stationary distribution. Other alternatives include adaptive rejection sampling [44], [45], [46]. However, the resultant samplers take longer to carry out each iteration, and much longer to converge because the conditionals have been replaced by their approximations.

C. Markovian Stationary Distribution

This subsection introduces suboptimal stationary distributions that drastically reduces memory requirement/complexity.

Sampling $\gamma_n = j^{(1:V)}$ directly from a distribution on $\{-1\}^V \uplus \Lambda^{(1:V)}$, as per the optimal stationary distribution, incurs large memory and computational costs. One strategy to circumvent such problems is to sample $j^{(1)}, \dots, j^{(V)}$ individually from respective distributions on $\{-1:M^{(1)}\}$,...,$\{-1:M^{(V)}\}$, provided that these distributions are inexpensive to compute. This can be accomplished by imposing the Markov property on

ϑ_n in (24), i.e.

$$\vartheta_n(j^{(1:V)}) = \prod_{s=2}^{V} \vartheta_n^{(s)}(j^{(s)}|j^{(s-1)})\vartheta_n^{(1)}(j^{(1)}) \tag{28}$$

where $\vartheta_n^{(1)}:\{-1:M^{(1)}\} \to [0,\infty)$, and for $s \in \{2:V\}$, $j^{(s-1)} \in \{-1:M^{(s-1)}\}$, $\vartheta_n^{(s)}(\cdot|j^{(s-1)}) : \{-1:M^{(s)}\} \to [0,\infty)$.

Effectually, we are considering a so-called *Markovian stationary distribution*

$$\pi(\gamma) \propto 1_\Gamma(\gamma) \prod_{n=1}^{P} \left(\prod_{s=2}^{V} \vartheta_n^{(s)}(\gamma_n^{(s)}|\gamma_n^{(s-1)}) \right) \vartheta_n^{(1)}(\gamma_n^{(1)}), \tag{29}$$

whose conditionals have the Markov property as shown in the following Proposition (see Appendix B for proof). Note that when each $\vartheta_n^{(s)}(\cdot|j^{(s-1)})$ is independent of $j^{(s-1)}$, the resultant ϑ_n is not a distribution on $\{-1\}^V \uplus \Lambda^{(1:V)}$ (because there is no mechanism to ensure that if $j^{(1)}$ is negative/non-negative then $j^{(2:V)}$ are also negative/non-negative), and hence (29) would not be a distribution on $(\{-1\}^V \uplus \Lambda^{(1:V)})^P$.

Proposition 3: If $\vartheta_n^{(1)}$, $\vartheta_n^{(s)}(\cdot|\cdot)$, $s \in \{2:V\}$ are such that (29) is a distribution on $(\{-1\}^V \uplus \Lambda^{(1:V)})^P$, then its n-th conditional, given a positive 1–1 $\gamma_{\bar{n}} = (\gamma_{1:n-1}, \gamma_{n+1:P})$ in $(\{-1\}^V \uplus \Lambda^{(1:V)})^{P-1}$, is

$$\pi_n(j^{(1:V)}|\gamma_{\bar{n}}) = \prod_{s=2}^{V} \pi_n^{(s)}(j^{(s)}|j^{(s-1)}, \gamma_{\bar{n}})\pi_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}) \tag{30}$$

where

$$\pi_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}) = \frac{K_n^{(2)}(j^{(1)})\beta_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j^{(1)})}{\sum_{j=-1}^{M^{(1)}} K_n^{(2)}(j)\beta_n^{(1)}(j|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j)} \tag{31}$$

$$\pi_n^{(s)}(j^{(s)}|j^{(s-1)}, \gamma_{\bar{n}}) = \begin{cases} 1, & j^{(s)}, j^{(s-1)} = -1 \\ \frac{K_n^{(s+1)}(j^{(s)})\beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})\vartheta_n^{(s)}(j^{(s)}|j^{(s-1)})}{K_n^{(s)}(j^{(s-1)})}, & j^{(s)}, j^{(s-1)} > -1 \end{cases} \tag{32}$$

$$K^{(s)}(j^{(s-1)}) = \sum_{j=0}^{M^{(s)}} K_n^{(s+1)}(j)\beta_n^{(s)}(j|\gamma_{\bar{n}}^{(s)})\vartheta_n^{(s)}(j|j^{(s-1)}) \tag{33}$$

for $s \in \{2:V\}$, with $K_n^{(V+1)}(j^{(V)}) = 1$.

Thus, sampling $j^{(1:V)}$ from the n-th conditional, i.e. (30), of a Markovian stationary distribution, can be achieved by

$$\begin{aligned} j^{(1)} &\sim \pi_n^{(1)}(\cdot|\gamma_{\bar{n}}), \\ j^{(2)} &\sim \pi_n^{(2)}(\cdot|j^{(1)}, \gamma_{\bar{n}}), \\ &\vdots \\ j^{(V)} &\sim \pi_n^{(V)}(\cdot|j^{(V-1)}, \gamma_{\bar{n}}). \end{aligned}$$

This strategy only requires $2 + \max_s M^{(s)}$ memory locations to store the categories, instead of $1 + \prod_{s=1}^{V}(1 + M^{(s)})$ as per the optimal stationary distribution. This means, for 5 sensors with 15 measurements per sensor, we only need 17 memory locations instead of over a million.

While the memory requirement has been addressed, sampling the n-th conditional is not necessarily scalable. For each $s \in \{2:V\}$ we need to compute $2 + M^{(s-1)}$ normalizing constants $K_n^{(s)}(j^{(s-1)})$, $j^{(s-1)} \in \{-1:M^{(s-1)}\}$ with $\mathcal{O}(M^{(s)})$ complexity each, which incurs a net complexity of $\mathcal{O}(M^{(s-1)}M^{(s)})$. Hence, sampling the n-th conditional generally incurs $\mathcal{O}(\sum_{s=2}^{V} M^{(s-1)}M^{(s)})$ complexity, since sampling a categorical distribution is linear in the number of categories.

Nonetheless, the following special case requires only one normalizing constant for each s (see Appendix C for proof), thereby achieving a complexity of $\mathcal{O}(\sum_{s=1}^{V} M^{(s)})$.

Corollary 4: In addition to the premises of Proposition 3, if the stationary distribution (29) is *minimally-Markovian*, i.e.

$$\vartheta_n^{(s)}(j^{(s)}|j^{(s-1)}) = \vartheta_n^{(s)}(j^{(s)})1_{\{-1\}^2 \uplus \Lambda^{(s-1:s)}}(j^{(s-1)}, j^{(s)}), \tag{34}$$

then (31) and (32) reduce to

$$\pi_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}) = \begin{cases} 1 - P_n(\Lambda^{(1:V)}), & j^{(1)} = -1 \\ \frac{P_n(\Lambda^{(1:V)})\beta_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j^{(1)})}{\Upsilon_n^{(1)}}, & j^{(1)} > -1 \end{cases} \tag{35}$$

$$\pi_n^{(s)}(j^{(s)}|j^{(s-1)}, \gamma_{\bar{n}}) = \begin{cases} 1, & j^{(s)}, j^{(s-1)} = -1 \\ \frac{\beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})\vartheta_n^{(s)}(j^{(s)})}{\Upsilon_n^{(s)}}, & j^{(s)}, j^{(s-1)} > -1 \end{cases} \tag{36}$$

for $s \in \{2:V\}$, where

$$\Upsilon_n^{(s)} \triangleq \sum_{j^{(s)}=0}^{M^{(s)}} \beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})\vartheta_n^{(s)}(j^{(s)}) \tag{37}$$

$$P_n(\Lambda^{(1:V)}) \triangleq \frac{\prod_{s=1}^{V} \Upsilon_n^{(s)}}{\prod_{s=1}^{V} \vartheta_n^{(s)}(-1) + \prod_{s=1}^{V} \Upsilon_n^{(s)}}. \tag{38}$$

The pseudocode for Gibbs sampling based on minimally-Markovian stationary distributions is given in Algorithm 2, MM-Gibbs, which has a complexity of $\mathcal{O}(TP\sum_{s=1}^{V} M^{(s)})$. In general, the conditionals, and hence performance depend on the sensor ordering, except in the following special case.

1) Suboptimal Stationary Distribution: Again, recall that each ϑ_n should be chosen to approximate η_n. This can be achieved with a minimally-Markovian stationary distribution by setting $\vartheta_n^{(s)}(j^{(s)})$ to

$$\eta_n^{(s)}(j^{(s)}) \triangleq \begin{cases} \left(1 - \bar{P}_S^{(\xi)}(\ell_n)\right)^{\delta_1[s]}, & \ell_n \in I,\ j^{(s)} = -1 \\ \left(\bar{P}_S^{(\xi)}(\ell_n)\right)^{\delta_1[s]} \bar{\psi}_{Z_+}^{(\xi,s,j^{(s)})}(\ell_n), & \ell_n \in I,\ j^{(s)} \in \Lambda^{(s)} \\ (1 - r_{B,+}(\ell_n))^{\delta_1[s]}, & \ell_n \in \mathbb{B}_+, j^{(s)} = -1 \\ (r_{B,+}(\ell_n))^{\delta_1[s]} \bar{\psi}_{Z_+}^{(\xi,s,j^{(s)})}(\ell_n), & \ell_n \in \mathbb{B}_+, j^{(s)} \in \Lambda^{(s)} \end{cases} \tag{39}$$

Algorithm 2: MM-Gibbs (Suboptimal).

```
Inputs: $T, V, \gamma^{(1)} = [\gamma_n^{(1,s)}], \vartheta = \{[\vartheta_n^{(s)}(j^{(s)})]\}_{s=1}^V$
Outputs: $\gamma^{(1)}, \dots, \gamma^{(T)}$

$P = \mathsf{size}(\vartheta, 1)$
for $s = 1 : V$
    $M^{(s)} = \mathsf{size}(\vartheta^{(s)}, 2) - 2, \quad c^{(s)} = [0 : M^{(s)}]$
end for
for $n = 1 : P$
    Compute $P_n(\Lambda^{(1:V)})$ via (38), $Q_n(\Lambda^{(1:V)}) =$
        $1 - P_n(\Lambda^{(1:V)})$
end for
for $t = 2 : T$
    $\phi^{(t)} = [\,]$
    for $n = 1 : P$
        $i_n \sim \mathsf{Categorical}([``+,"``-"], [P_n(\Lambda^{(1:V)}),$
            $Q_n(\Lambda^{(1:V)})])$
        if $i_n = ``+"$
            for $s = 1 : V$
                for $j^{(s)} = 0 : M^{(s)}$
                    $p_n^{(s)}(j^{(s)}) = \vartheta_n^{(s)}(j^{(s)})\beta_n^{(s)}(j^{(s)}|\phi_{1:n-1}^{(t,s)}, \gamma_{n+1:P}^{(t-1,s)})$
                end for
                $\phi_n^{(t,s)} \sim \mathsf{Categorical}(c^{(s)}, p_n^{(s)})$
            end for
            $\phi_n^{(t)} = [\phi_n^{(t,s)}]_{s=1}^V$
        else if
            $\phi_n^{(t)} = -1{*}\mathsf{ones}(1, V)$
        end if
        $\phi^{(t)} = [\phi^{(t)}; \phi_n^{(t)}]$
    end for
    $\gamma^{(t)} = \phi^{(t)}$
end for
```

where

$$\bar{\psi}_{Z_+}^{(\xi,s,j^{(s)})}(\ell_n) \triangleq \langle \bar{p}_+^{(\xi)}(\cdot, \ell_n), \psi_{Z_+}^{(s,j^{(s)})}(\cdot, \ell_n) \rangle, \tag{40}$$

in which case,

$$\vartheta_n(j^{(1:V)}) = \begin{cases} \eta_n^{(1)}(-1), & j^{(1:V)} \in \{-1\}^V \\ \prod_{s=1}^V \eta_n^{(s)}(j^{(s)}), & j^{(1:V)} \in \Lambda^{(1:V)} \end{cases}. \tag{41}$$

Equation (41) can be verified by substituting (39) for $\vartheta_n^{(s)}(j^{(s)})$ into (34), and the resulting $\vartheta_n^{(s)}(j^{(s)}|j^{(s-1)})$ into (28).

To gain an intuition into how ϑ_n in (41) approximates η_n, note firstly that, for $j^{(1:V)} \in \{-1\}^V$, it is straightforward to verify $\vartheta_n(j^{(1:V)}) = \eta_n^{(1)}(-1) = \eta_n(j^{(1:V)})$, by inspecting (22) and (39). Secondly, for $j^{(1:V)} \in \Lambda^{(1:V)}$, note from (22), (41), and (39) that the approximation of $\eta_n(j^{(1:V)})$ boils down to

$$\bar{\psi}_{Z_+}^{(\xi,j^{(1:V)})}(\ell_n) \simeq \prod_{s=1}^V \bar{\psi}_{Z_+}^{(\xi,s,j^{(s)})}(\ell_n).$$

Conditional on the history ξ and measurement Z_+: the left hand side, given by (23), can be interpreted as the probability that label ℓ_n jointly generates measurements $z_{j^{(1)}}^{(1)}, \dots, z_{j^{(V)}}^{(V)}$, i.e. $\Pr(z_{j^{(1)}}^{(1)}, \dots, z_{j^{(V)}}^{(V)} \sim \ell_n)$; the s-th term of the product, given by (40), can be interpreted as the probability that ℓ_n generates measurement $z_{j^{(s)}}^{(s)}$, i.e. $\Pr(z_{j^{(s)}}^{(s)} \sim \ell_n)$ (with $z_0^{(s)}$ representing a misdetection by sensor s). In essence, the suboptimal strategy approximates $\Pr(z_{j^{(1)}}^{(1)}, \dots, z_{j^{(V)}}^{(V)} \sim \ell_n)$ by $\Pr(z_{j^{(1)}}^{(1)} \sim \ell_n) \times \dots \times \Pr(z_{j^{(V)}}^{(V)} \sim \ell_n)$, which is reasonable, because intuitively the events "ℓ_n generates measurement $z_{j^{(s)}}^{(s)}$" and "ℓ_n generates measurement $z_{j^{(t)}}^{(t)}$" are almost independent of each other when $s \neq t$. Note also that for a single sensor $\vartheta_n = \eta_n = \eta_n^{(1)}$.

Remark: In (41), both $\prod_{s=1}^V \eta_n^{(s)}(j^{(s)})$ and $\eta_n^{(1)}(-1)$ (the latter only depends on $\bar{P}_S^{(\xi)}(\ell_n)$ and $r_{B,+}(\ell_n)$), are independent of the sensor ordering. Hence (41), and consequently Algorithm 2 with the suboptimal distribution defined by (41), are independent of the order of the sensors. Additionally, since each $\eta_n^{(s)}(j^{(s)}) > 0$, it follows that $\vartheta_n(j^{(1:V)}) > 0$, hence the convergence result of Proposition 2 holds.

The support of the (minimally-Markov) suboptimal stationary distribution contains the support of the optimal stationary distribution. To verify this, suppose that there exists an s such that $\bar{\psi}_{Z_+}^{(\xi,s,j^{(s)})}(\ell_n) = 0$. Then it follows from (40) that

$$\bar{p}_+^{(\xi)}(\cdot, \ell_n) = 0, \text{ or } \psi_{Z_+}^{(s,j^{(s)})}(\cdot, \ell_n) = 0.$$

Each of the above conditions implies $\bar{\psi}_{Z_+}^{(\xi,j^{(1:V)})}(\ell_n) = 0$. The implication of the first condition follows from (15), while that of the second follows from (8) and (15). Therefore, it follows from (39) and (22) that $\prod_{s=1}^V \eta_n^{(s)}(j^{(s)}) = 0$, implies $\eta_n(j^{(1:V)}) = 0$, i.e. the suboptimal stationary distribution is zero implies the optimal is also zero. Thus, the suboptimal stationary distribution is positive whenever the optimal is positive. This means the support of suboptimal stationary distribution contains that of the optimal.

The suboptimal sampling strategy can be viewed as importance sampling [47], with the suboptimal stationary distribution as the "proposal" or "importance function" since its support contains that of the optimal. As such, a larger number of iterations *T* would be needed to achieve the same effective number of samples as per the optimal stationary distribution [47]. The number of additional samples depends on how well the suboptimal stationary distribution approximates the optimal. The better the approximation the smaller the additional number of samples. Nonetheless, for GLMB truncation, the total number of distinct samples with significant weights is more relevant than the effective number of samples. As such, the optimal stationary distribution is not necessarily the most desirable in terms of sample diversity [34]. Indeed, the rationale behind tempering is to modify the stationary distribution to generate more diverse samples [34].

D. Multi-Sensor GLMB Filtering

A GLMB is completely characterized by parameters $(\omega^{(I,\xi)}, p^{(\xi)})$, $(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi$, which can be enumerated as

Algorithm 3: Multi-Sensor GLMB Filter.

```
 Inputs: ${(I^{(h)}, \omega^{(h)}, p^{(h)})}_{h=1}^{H}, Z_+, H_+^{\max}$
 Inputs: ${(r_{B,+}^{(\ell)}, p_{B,+}^{(\ell)})}, P_S, f_+(\cdot|\cdot)$
 Inputs: ${(\kappa_+^{(s)}, P_{D,+}^{(s)}, g_+^{(s)}(\cdot|\cdot))}_{s=1}^{V}$
Outputs: ${(I_+^{(h_+)}, \omega_+^{(h_+)}, p_+^{(h_+)})}_{h_+=1}^{H_+}$

Sample counts $[T_+^{(h)}]_{h=1}^{H}$ from multinomial distribution
 with $H_+^{\max}$ trials and weights $[\omega^{(h)}]_{h=1}^{H}$
for $h = 1 : H$
    Initialize $\gamma^{(h,1)}$
    Compute $\vartheta^{(h)} = \eta^{(h)}$ using(22) AND
    $\{\gamma^{(h,t)}\}_{t=1}^{\tilde{T}^{(h)}} = \mathsf{Unique}(\mathsf{Gibbs}(T_+^{(h)}, V, \gamma^{(h,1)}, \vartheta^{(h)}))$
                 OR
    Compute $\vartheta^{(h)} = \{\eta^{(h,s)}\}_{s=1}^{V}$using(39), AND
    $\{\gamma^{(h,t)}\}_{t=1}^{\tilde{T}^{(h)}} =$
    $\mathsf{Unique}(\mathsf{MM\text{-}Gibbs}(T_+^{(h)}, V, \gamma^{(h,1)}, \vartheta^{(h)}))$
    for $t = 1 : \tilde{T}_+^{(h)}$
      Compute
        $I_+^{(h,t)} = \{\ell_n \in I^{(h)} \cup \mathbb{B}_+ : \gamma_n^{(h,t)} \geq 0\}$
        $\omega_+^{(h,t)} \propto \omega^{(h)} \prod_{n=1}^{|I^{(h)} \cup \mathbb{B}_+|} \eta_n^{(h)}(\gamma_n^{(h,t)})$
        $p_+^{(h,t)}(\cdot, \ell_n) \propto \bar{p}_+^{(h)}(\cdot, \ell_n) \psi_{Z_+}^{(\gamma_n^{(h,t)})}(\cdot, \ell_n)$
    end for
end for
 $(\{(I_+^{(h_+)}, p_+^{(h_+)})\}_{h_+=1}^{H_+}, \sim, [U_{h,t}])$
        $= \mathsf{Unique}(\{(I_+^{(h,t)}, p_+^{(h,t)})\}_{(h,t)=(1,1)}^{(H, \tilde{T}_+^{(h)})})$
for  $h_+ = 1 : H_+$
    $\omega_+^{(h_+)} = \sum_{h,t: U_{h,t} = h_+} \omega_+^{(h,t)}$
end for
Normalize weights $\{\omega_+^{(h_+)}\}_{h_+=1}^{H_+}$
```

$\{(I^{(h)}, \xi^{(h)}, \omega^{(h)}, p^{(h)})\}_{h=1}^{H}$, where

$$\omega^{(h)} \triangleq \omega^{\left(I^{(h)}, \xi^{(h)}\right)}, \quad p^{(h)} \triangleq p^{\left(\xi^{(h)}\right)}.$$

Implementing the GLMB filter amounts to propagating forward the parameter set $\{(I^{(h)}, \omega^{(h)}, p^{(h)})\}_{h=1}^{H}$.

The multi-sensor GLMB filter implementation is the same as single-sensor case in [34], with the single-sensor Gibbs sampler and update replaced by their multi-sensor versions. For completeness the proposed multi-sensor GLMB filter is summarized in Algorithm 3. Note that to be consistent with the indexing by h instead of (I, ξ), we abbreviate

$$\eta_n^{(h)}(j^{(1:V)}) \triangleq \eta_n(j^{(1:V)}), \text{with } (I, \xi) = (I^{(h)}, \xi^{(h)})$$

$$\eta^{(h)}(j^{(1:V)}) \triangleq \left[\eta_1^{(h)}(j^{(1:V)}), \ldots, \eta_P^{(h)}(j^{(1:V)})\right]$$

$$\bar{P}_S^{(h)}(\ell_i) \triangleq \bar{P}_S^{\left(\xi^{(h)}\right)}(\ell_i)$$

$$\bar{p}_+^{(h)}(x, \ell_i) \triangleq \bar{p}_+^{\left(\xi^{(h)}\right)}(x, \ell_i)$$

$$\bar{\psi}_{Z_+}^{\left(h, j^{(1:V)}\right)}(\ell_i) \triangleq \bar{\psi}_{Z_+}^{\left(\xi^{(h)}, j^{(1:V)}\right)}(\ell_i)$$

$$\eta_n^{(h,s)}(j^{(s)}) \triangleq \eta_n^{(h,s)}(j^{(s)}), \text{with } (I, \xi) = (I^{(h)}, \xi^{(h)})$$

$$\bar{\psi}_{Z_+}^{\left(h, s, j^{(s)}\right)}(\ell_i) \triangleq \bar{\psi}_{Z_+}^{\left(\xi^{(h)}, s, j^{(s)}\right)}(\ell_i)$$

Computing $\bar{p}_+^{(h)}(\cdot, \ell_i)$, $\bar{P}_S^{(h)}$, $\bar{\psi}_{Z_+}^{(h, j^{(1:V)})}(\ell_i)$, $\bar{\psi}_{Z_+}^{(h, s, j^{(s)})}(\ell_i)$, $p_+^{(h,t)}(\cdot, \ell_i)$, can be done as in subsection IV.B of [15].

IV. Numerical Experiments

In this section, we present simulation results that demonstrate the performance of the multi-sensor GLMB filters discussed in Section III. The results are presented in two parts. In the first part, we use a scenario with a small number of sensors, and compare the following three algorithms:

- suboptimal multi-sensor GLMB filter (Gibbs sampling with suboptimal distribution, Subsection III-C1),
- optimal multi-sensor GLMB filter (Gibbs sampling with optimal distribution, Subsection III-B1), and
- iterated-corrector multi-sensor GLMB filter (standard prediction and iterated update implementation).

The aim is to show that the suboptimal multi-sensor GLMB filter can achieve near-optimal results, but at a significantly smaller computational cost. The scenario used for this comparison is limited to four sensors, because the optimal version does not scale well to a large number of sensors (see Subsection III-B1). In the second part, we compare the suboptimal multi-sensor GLMB filter with the iterated-corrector GLMB filter, on a scenario with a larger number of sensors with different types. This is designed to demonstrate the scalability and versatility of the suboptimal version.

Throughout this section a common ground truth is used with various scenarios of different sensor combinations. The ground truth involves a maximum of 10 objects simultaneously within a 2-D surveillance region over a period of 100 seconds. The objects move according to a discrete white noise acceleration model, and the number of objects in the surveillance region varies over time, as new objects can appear and existing objects can disappear. The state of an object at time k is represented by its 2-D position and velocity vectors, i.e. $x_k = [p_{x,k}, p_{y,k}, \dot{p}_{x,k}, \dot{p}_{y,k}]$, and the single-object transition density is given by

$$f(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; F_k x_k, Q_k),$$

where

$$F_k = \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix} \otimes I_2, \qquad Q_k = \sigma_a^2 \begin{bmatrix} \frac{\Delta^4}{4} & \frac{\Delta^3}{3} \\ \frac{\Delta^3}{3} & \Delta^2 \end{bmatrix} \otimes I_2,$$

I_2 is the 2×2 identity matrix, $\Delta = 1$s is the sampling period, and $\sigma_a = 0.15$ m/s^2 is the standard deviation of the process noise. The survival probability is $P_S = 0.98$. Object births are modeled by an LMB with parameters $\{(r_{B,k}, p_{B,k}^{(i)})\}_{i=1}^{6}$ where $r_{B,k} = 0.05$, $p_{B,k}^{(i)}(x) = \mathcal{N}(x; m_{B,k}^{(i)}, Q_{B,k})$,

$$m_{B,k}^{(1)} = (100, 100, 0, 0), \quad m_{B,k}^{(2)} = (100, 500, 0, 0),$$

$$m_{B,k}^{(3)} = (100, 900, 0, 0), \quad m_{B,k}^{(4)} = (900, 100, 0, 0),$$

$$m_{B,k}^{(5)} = (900, 500, 0, 0), \quad m_{B,k}^{(6)} = (900, 900, 0, 0),$$

and $Q_{B,k} = diag([25,\ 25,\ 5,\ 5]^2)$. Ten objects appear near these locations at various times during the first 40 steps. Seven of these disappear at various times over the last 40 steps.

Each sensor has a fixed position. If sensor i at position $s^{(i)} = (s_x^{(i)}, s_y^{(i)})$ with type $t^{(i)}$generates a detection $z_k^{(i)}$ for an object with state x_k, then $z_k^{(i)}$ is distributed according to

$$g^{(i)}(z_k^{(i)}|x_k, s^{(i)}, t^{(i)}) = \mathcal{N}(z_k^{(i)}; h_{t^{(i)}}(x_k, s^{(i)}), \sigma_{t^{(i)}}^2).$$

Based on the following measurement functions (bearing, range, range rate and position)

$$h_\theta(x_k, s^{(i)}) = \arctan\left(\frac{p_{x,k} - s_x^{(i)}}{p_{y,k} - s_y^{(i)}}\right),$$

$$h_r(x_k, s^{(i)}) = \sqrt{(p_{x,k} - s_x^{(i)})^2 + (p_{y,k} - s_y^{(i)})^2},$$

$$h_{rr}(x_k, s^{(i)}) = \frac{(p_{x,k} - s_x^{(i)})\dot{p}_{x,k} + (p_{y,k} - s_y^{(i)})\dot{p}_{y,k}}{h_r(x_k, s^{(i)})},$$

$$h_p(x_k) = \begin{bmatrix} p_{x,k} \\ p_{y,k} \end{bmatrix},$$

we simulate four different types of sensors (i.e. $t^{(i)} \in \{1, 2, 3, 4\}$) according to the following models:

- Type 1: Bearings only (e.g. passive radar)

$$h_1(x_k, s^{(i)}) = h_\theta(x_k, s^{(i)})$$

- Type 2: Bearing and range (e.g. Doppler insensitive active radar)

$$h_2(x_k, s^{(i)}) = \begin{bmatrix} h_\theta(x_k, s^{(i)}) \\ h_r(x_k, s^{(i)}) \end{bmatrix}$$

- Type 3: Bearing, range and range-rate (e.g. Doppler sensitive active radar)

$$h_3(x_k, s^{(i)}) = \begin{bmatrix} h_\theta(x_k, s^{(i)}) \\ h_r(x_k, s^{(i)}) \\ h_{rr}(x_k, s^{(i)}) \end{bmatrix}$$

- Type 4: Position (e.g. drone-mounted camera)

$$h_4(x_k, s^{(i)}) = h_p(x_k)$$

All three algorithms are run with approximately 3000 components during the update, which are pruned to approximately 300 best post-update. To examine the tracking performance, we consider both the OSPA [48] and OSPA$^{(2)}$ [49], [50] distances between the set of estimated tracks and the set of ground truth tracks. The OSPA distance is an instantaneous per-object error, accounting for estimation errors in both localization and cardinality, but does not capture track labelling errors, as can occur when objects are close together or cross each other in the measurement space. The OSPA$^{(2)}$ metric addresses this by using the OSPA distance with a suitable base-distance between two tracks (rather than vectors) [50]. Note that the OSPA$^{(2)}$ distance assesses multi-object tracking error over a time window (for window of one instant, OSPA$^{(2)}$ reduces to OSPA) [50]. Hence, an OSPA$^{(2)}$ versus time curve would show at each time k, the OSPA$^{(2)}$ error over a window ending at k.

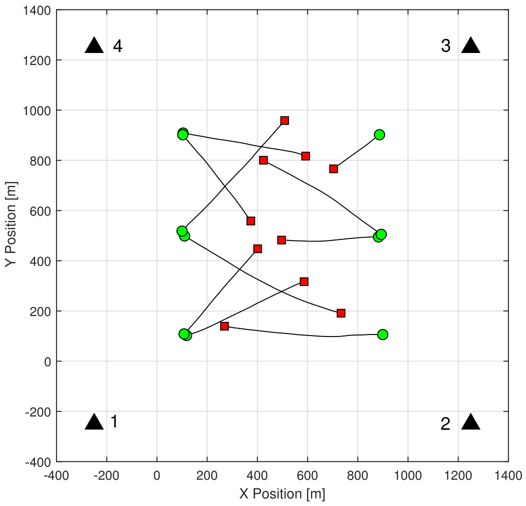

Fig. 1. Scenario 1: ground truth and sensor layout. The triangles (▲) are the locations of bearing-only sensors, the green circles (●) indicate the object starting positions, and the red squares (■) are the final positions.

A. Scenario 1

For the first experiment, we compare all three tracking algorithms on a scenario with up to four Type-1 (bearing-only) sensors. Each sensor has measurement noise standard deviation $\sigma_1 = \pi/180$ rad, and a fixed detection probability of $P_D = 0.67$. Each sensor also generates a set of false alarms at each time step, all with a Poisson cardinality distribution of mean 7. The false alarms are uniformly distributed on the interval $[0, 2\pi]$ rad. The sensor positions and ground truth are shown in Figure 1.

Each algorithm was tested on the following three configurations; 2 sensors (1 and 2), 3 sensors (1, 2 and 3), and 4 sensors (1, 2, 3 and 4). We ran 100 Monte Carlo (MC) trials, each with the same ground truth trajectories, but a different realization of sensor noise. The MC average OSPA (with cutoff $c = 100$ m, order $p = 1$) and OSPA$^{(2)}$ (with the same c, p, and window length $w = 20$) distances are plotted against time in Figure 2. The relative execution times (wrt. the suboptimal filter on sensor 1) of the algorithms are shown in Table I.

As expected, both the OSPA and OSPA$^{(2)}$ results (Figure 2) demonstrate that the optimal multi-sensor GLMB filter outperforms the suboptimal version. However, a comparison of the execution times (Table I) shows that such improved performance comes at a very large computational cost. The computational cost of the optimal version scales exponentially with the number of sensors, whereas the suboptimal version scales linearly. Despite being allocated an equal number of components, the iterated-corrector GLMB filter performs poorly compared to the other two algorithms. Note that the reduction in execution time of the iterated-corrector at the 4th sensor is due to component depletion in the previous updates.

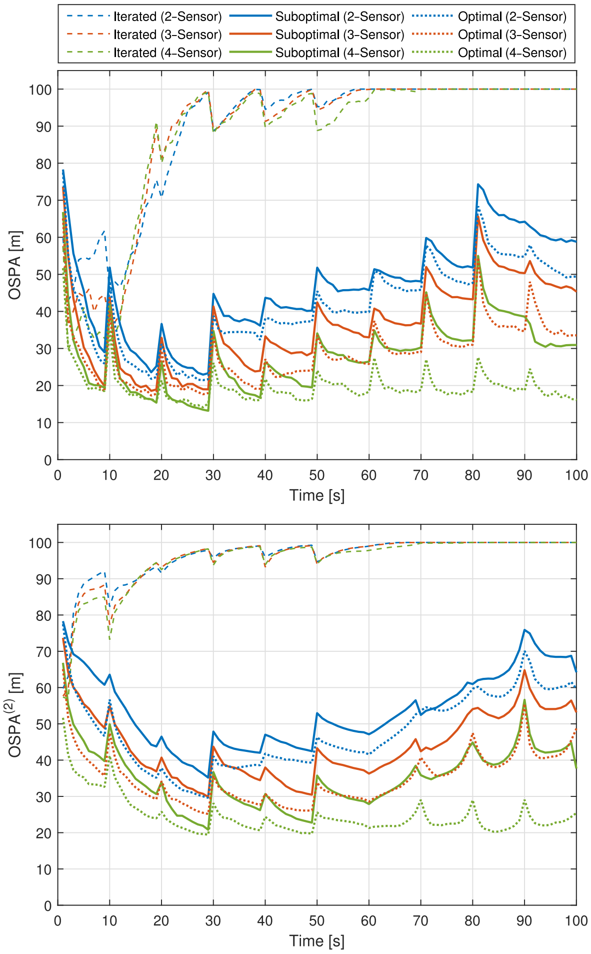

Fig. 2. Scenario 1: OSPA, OSPA$^{(2)}$ plots for different number of sensors.

TABLE I
SCENARIO 1: ALGORITHM EXECUTION TIMES

	2-Sensor	3-Sensor	4-Sensor
Optimal	225	439.4	1321.7
Suboptimal	1.5	2.3	3.1
Iterated-Corrector	6.5	6.6	5.9

TABLE II
SCENARIO 2: SENSOR MODEL PARAMETERS

Type (i)	Meas Noise (σ_i)	Clutter Dist
1	$(2\pi/180)$ rad	$[0, 2\pi]$ rad
2	$\begin{bmatrix} (2\pi/180)\text{ rad} & 0 \\ 0 & 50\text{ m} \end{bmatrix}$	$[0, 2\pi]$ rad $\times$ $[0, 2000]$ m
3	$\begin{bmatrix} (2\pi/180)\text{ rad} & 0 & 0 \\ 0 & 50\text{ m} & 0 \\ 0 & 0 & 2\text{ m/s} \end{bmatrix}$	$[0, 2\pi]$ rad $\times$ $[0, 2000]$ m $\times$ $[-10, 10]$ m/s
4	$\begin{bmatrix} 150\text{ m} & 0 \\ 0 & 150\text{ m} \end{bmatrix}$	$[0, 1000]$ m $\times$ $[0, 1000]$ m

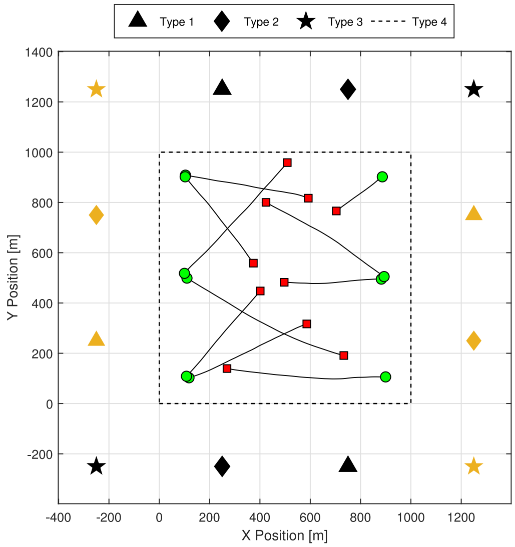

Fig. 3. Scenario 2: ground truth and sensor layout, 7-sensor configuration (black sensors), 13-sensor configuration (all sensors).

B. Scenario 2

To demonstrate the scalability and versatility of the suboptimal multi-sensor GLMB filter, we now study its tracking performance on a far more challenging scenario with unreliable sensors (considerably lower detection probability than Scenario 1), in larger number and more diverse types. All sensors have a detection probability of $P_D = 0.5$, and generate Poisson false alarms with a mean of 5 per scan. This is quite an adverse signal environment since there is only a 50% chance of detecting an object. The measurement noise standard deviation and the false alarm spatial distribution for each sensor type is given in Table II.

We test two different cases: the first with a total of 7 sensors (two each of Type-1, Type-2, Type-3, and one Type-4); and the second with a total of 13 sensors (four each of Type-1, Type-2, Type-3, and one Type-4). Ground truths and sensor configurations for the 7-sensor and 13-sensor cases are shown in Figure 3. In this experiment, we can only compare with the iterated-corrector multi-sensor GLMB filter, since the optimal version becomes computationally infeasible for more than four sensors. Note that even the latest scalable solution for multi-sensor multi-object filtering, see for example [11], can only cope with 3 sensors on a less challenging scenario (than this experiment) with a P_D of 0.6 (rather than 0.5), and up to 5 objects (rather than 10).

Again, we carried out 100 MC trials for each case with the same ground truth and different realizations of sensor noise. The MC average OSPA and OSPA$^{(2)}$ distances (with the same parameters as Scenario 1) are plotted against time in Figure 4. The mean (and one-sigma bounds) of the cardinality for the 7-sensor and 13-sensor cases are plotted in Figures 5 and 6, respectively. The relative execution times (wrt. the iterated-corrector strategy on the 7-sensor case) of the algorithms are shown in Table III.

The OSPA and OSPA$^{(2)}$ plots (Figure 4) indicate that the suboptimal multi-sensor GLMB filter significantly outperforms the iterated-corrector implementation in both the 7-sensor and 13-sensor cases. This concurs with the cardinality statistics (Figures 5 and 6), which show the iterated-corrector producing significantly less accurate estimates and with higher variability.

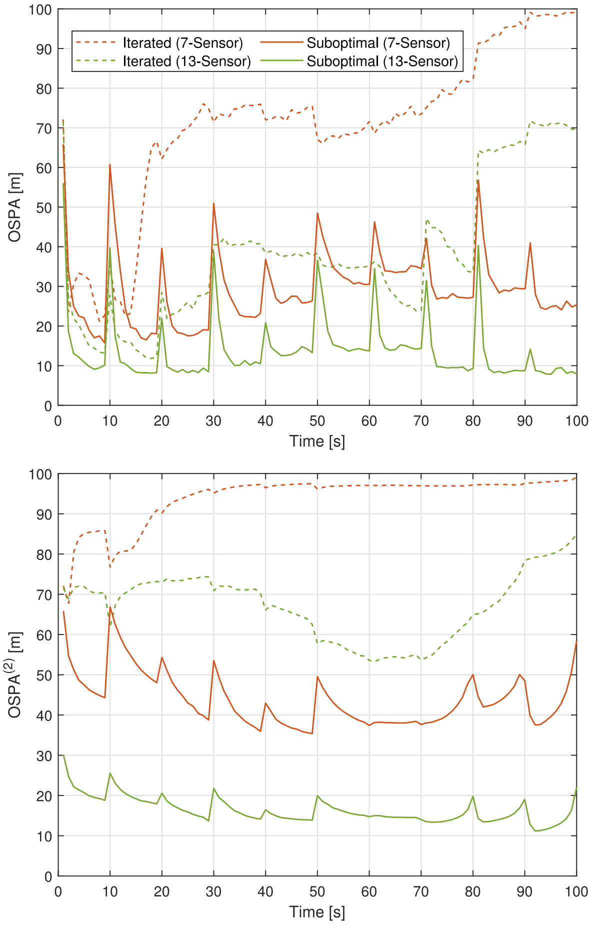

Fig. 4. Scenario 2: OSPA, OSPA$^{(2)}$ plots for 7-sensor and 13-sensor cases.

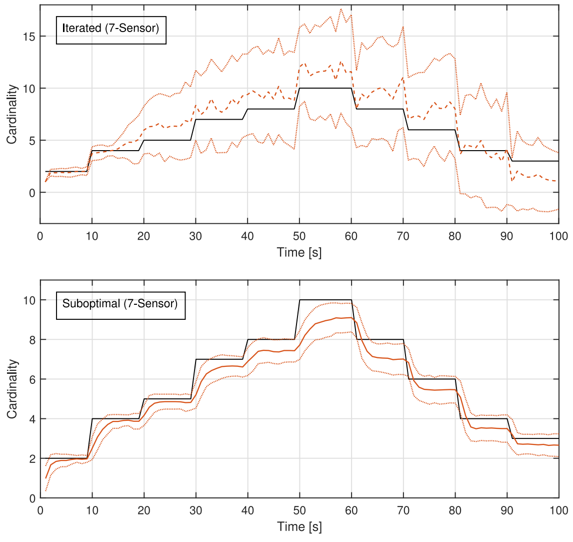

Fig. 5. 7-sensor case: true and mean cardinality (with 1-sigma bounds).

Note that the tracking performance of the iterated-corrector implementation depends on the order in which the sensors are processed. In our experiments the iterated-corrector processes the bottom left sensor first, and progresses anti-clockwise to the last sensor. The first sensor is a Type-3 sensor (bearing, range and range rate), which is the most informative amongst the sensors. The performance could be far worse by starting with a less informative sensor. Finding the best the performance over all sensor permutations is infeasible for large number of sensors, not to mention such optimal sensor permutation is data dependent.

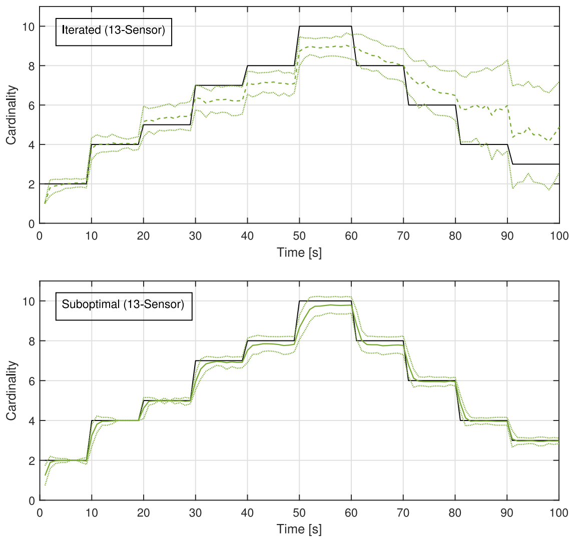

Fig. 6. 13-sensor case: true and mean cardinality (with 1-sigma bounds).

TABLE III
SCENARIO 2: ALGORITHM EXECUTION TIMES

	7-Sensor	13-Sensor
Iterated-Corrector	1.00	1.27
Suboptimal	1.11	2.16

The suboptimal (and optimal) multi-sensor GLMB filter uses the joint prediction and update, which simultaneously integrates information from the survival model as well as measurements from all sensors to generate significant components. In contrast, the iterated-corrector uses information from the survival model first, followed by information from the measurements of each sensor, one at a time. Due to such lack of information prior to each update, components that would be significant after subsequent updates are likely to be discarded, leading to component depletion. This explains the iterated-corrector's poorer tracking performance (compared to the suboptimal) while having a slower increase in execution times as the number of sensors increases.

V. CONCLUSIONS

This paper proposed an efficient implementation of the multi-sensor GLMB filter by integrating the prediction and update into one step along with an efficient algorithm for truncating the GLMB filtering density based on Gibbs sampling. The resulting algorithm is an on-line multi-sensor multi-object tracker with linear complexity in the total number of measurements across the sensors, and quadratic in the number of hypothesized tracks. Numerical studies verify the scalability of the proposed solution with respect to the total number of measurements.

The proposed multi-sensor GLMB implementation is also applicable to approximations such as the labeled multi-Bernoulli (LMB) and marginalized GLMB filters, since these filters require full GLMB updates to be performed [17], [51]. Conceptually, the proposed the multi-sensor solution can also be extended to the multi-scan case [33]. However, the multi-sensor multi-scan GLMB filtering problem is far more computationally intensive, and the key challenge lies in achieving real-time speed for practical applications.

Appendix

A. Proof of Proposition 1

Let $\bar{n} = \{1:P\} - \{n\}$ and $\Gamma(\bar{n})$ be the set of all $\gamma_{\bar{n}} = (\gamma_{1:n-1}, \gamma_{n+1:P}) \in (\{-1\}^V \uplus \Lambda^{(1:V)})^{P-1}$ that are positive 1–1 (i.e. $\gamma_{\bar{n}}$ such that for each $s \in \{1:V\}$ there are no distinct $i, j \in \bar{n}$ with $\gamma_i^{(s)} = \gamma_j^{(s)} > 0$). We are interested in the functional dependence of $\pi_n(\gamma_n|\gamma_{\bar{n}})$ on γ_n, while its dependence on all other variables is aggregated into the normalizing constant:

$$\begin{aligned}\pi_n\left(\gamma_n|\gamma_{\bar{n}}\right) &\triangleq \frac{\pi\left(\gamma\right)}{\pi\left(\gamma_{\bar{n}}\right)} \propto \pi\left(\gamma\right)\\ &\propto 1_\Gamma\left(\gamma\right)\prod_{j=1}^{P}\vartheta_j\left(\gamma_j\right)\\ &= \vartheta_n\left(\gamma_n\right)1_\Gamma\left(\gamma\right)\prod_{j\in\bar{n}}\vartheta_j\left(\gamma_j\right).\end{aligned}$$

Factorizing $1_\Gamma(\gamma)$ using Lemma A, gives

$$\begin{aligned}&\pi_n\left(\gamma_n|\gamma_{\bar{n}}\right)\\ &\quad\propto \vartheta_n\left(\gamma_n\right)\prod_{s=1}^{V}\prod_{i\in\bar{n}}\left(1-1_{\{1:M^{(s)}\}}(\gamma_n^{(s)})\delta_{\gamma_n^{(s)}}[\gamma_i^{(s)}]\right)\\ &\quad\times 1_{\Gamma(\bar{n})}\left(\gamma_{\bar{n}}\right)\prod_{j\in\bar{n}}\vartheta_j\left(\gamma_j\right)\\ &\quad\propto \vartheta_n\left(\gamma_n\right)\prod_{s=1}^{V}\prod_{i\in\bar{n}}\left(1-1_{\{1:M^{(s)}\}}(\gamma_n^{(s)})\delta_{\gamma_n^{(s)}}[\gamma_i^{(s)}]\right). \quad (42)\end{aligned}$$

If $j^{(1:V)} \in \Lambda^{(1:V)}$, then it follows from (42) that $\pi_n(j^{(1:V)}|\gamma_{\bar{n}}) \propto \vartheta_n(j^{(1:V)})$, unless there exist $i \in \bar{n}$ and s with $\gamma_i^{(s)} = j^{(s)} > 0$, in which case $\pi_n(j^{(1:V)}|\gamma_{\bar{n}}) = 0$ (because $1_{\{1:M^{(s)}\}}(j^{(s)})\delta_{j^{(s)}}[\gamma_i^{(s)}] = 1$). Thus, for $j^{(1:V)} \in \Lambda^{(1:V)}$

$$\begin{aligned}&\pi_n(j^{(1:V)}|\gamma_{\bar{n}})\\ &\quad\propto \vartheta_n(j^{(1:V)})\prod_{s=1}^{V}\left(1-1_{\{1:M^{(s)}\}\bigcap\{\gamma_{1:n-1}^{(s)},\gamma_{n+1:P}^{(s)}\}}(j^{(s)})\right).\end{aligned}$$

On the other hand, if $j^{(1:V)} \in \{-1\}^V$, then $1_{\{1:M^{(s)}\}}(j^{(s)}) = 0$ for all s, and (42) implies $\pi_n(j^{(1:V)}|\gamma_{\bar{n}}) \propto \vartheta_n(j^{(1:V)})$. Hence the above equation holds on $\{-1\}^V \uplus \Lambda^{(1:V)}$.

Lemma A:

$$1_\Gamma\left(\gamma\right) = 1_{\Gamma(\bar{n})}\left(\gamma_{\bar{n}}\right)\prod_{s=1}^{V}\prod_{i\in\bar{n}}\left(1-1_{\{1:M^{(s)}\}}(\gamma_n^{(s)})\delta_{\gamma_n^{(s)}}[\gamma_i^{(s)}]\right). \quad (43)$$

Proof: Note that the condition $\gamma_i^{(s)} = \gamma_j^{(s)} > 0$ is equivalent to $\delta_{\gamma_i^{(s)}}[\gamma_j^{(s)}]1_{\{1:M^{(s)}\}}(\gamma_i^{(s)}) = 1$. Hence, $\gamma^{(s)}$ is positive 1–1 iff for any distinct i, j, $\delta_{\gamma_i^{(s)}}[\gamma_j^{(s)}]1_{\{1:M^{(s)}\}}(\gamma_i^{(s)}) = 0$. Also, $\gamma^{(s)}$ is not positive 1–1 iff there exists distinct i, j such that $\delta_{\gamma_i^{(s)}}[\gamma_j^{(s)}]1_{\{1:M^{(s)}\}}(\gamma_i^{(s)}) = 1$. Similarly, $\gamma_{\bar{n}}^{(s)}$ is positive 1–1 iff for any distinct $i, j \in \bar{n}$, $\delta_{\gamma_i^{(s)}}[\gamma_j^{(s)}]1_{\{1:M^{(s)}\}}(\gamma_i^{(s)}) = 0$.

We will show that; (a) if γ is positive 1–1 then the right hand side (RHS) of (43) equates to 1, and (b) if γ is not positive 1–1, then the RHS of (43) equates to 0.

To establish (a), assume that γ is positive 1–1, then $\gamma_{\bar{n}}$ is also positive 1–1, i.e., $1_{\Gamma(\bar{n})}(\gamma_{\bar{n}}) = 1$, and for any $i \neq n$, $\delta_{\gamma_n^{(s)}}[\gamma_i^{(s)}]1_{\{1:M^{(s)}\}}(\gamma_n^{(s)}) = 0$ for all s. Hence the RHS of (43) equates to 1.

To establish (b), assume that γ is not positive 1–1. If $\gamma_{\bar{n}}$ is also not positive 1–1, i.e., $1_{\Gamma(\bar{n})}(\gamma_{\bar{n}}) = 0$, then the RHS of (43) trivially equates to 0. It remains to show that even if $\gamma_{\bar{n}}$ is positive 1–1, the RHS of (43) still equates to 0. Since γ is not positive 1–1, there exist an s and distinct i, j such that $\delta_{\gamma_i^{(s)}}[\gamma_j^{(s)}]1_{\{1:M^{(s)}\}}(\gamma_i^{(s)}) = 1$. Further, either i or j has to equal n, because the positive 1–1 property of $\gamma_{\bar{n}}$ implies that if such (distinct) i, j, are in $\bar{n}$, then $\delta_{\gamma_i^{(s)}}[\gamma_j^{(s)}]1_{\{1:M^{(s)}\}}(\gamma_i^{(s)}) = 0$ and we have a contradiction. Hence, there exist an s and $i \neq n$ such that $\delta_{\gamma_n^{(s)}}[\gamma_i^{(s)}]1_{\{1:M^{(s)}\}}(\gamma_n^{(s)}) = 1$, and thus the RHS of (43) equates to 0. ■

B. Proof of Proposition 3

Let us make the following abbreviations,

$$\begin{aligned}\tilde{\vartheta}_n^{(1)}(j^{(1)}) &\triangleq \beta_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j^{(1)}),\\ K_n^{(1)} &\triangleq \sum_{j^{(1)}=-1}^{M^{(1)}}\tilde{\vartheta}_n^{(1)}(j^{(1)})K_n^{(2)}(j^{(1)})\\ \tilde{\vartheta}_n^{(s)}(j^{(s)}|j^{(s-1)}) &\triangleq \beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})\vartheta_n^{(s)}(j^{(s)}|j^{(s-1)}),\\ K_n^{(s)}(j^{(s-1)}) &\triangleq \sum_{j^{(s)}=-1}^{M^{(s)}}\tilde{\vartheta}_n^{(s)}(j^{(s)}|j^{(s-1)})K_n^{(s+1)}(j^{(s)}).\end{aligned}$$

for $s \in \{2:V\}$, with $K_n^{(V+1)}(j^{(V)}) = 1$.

Substituting $K_n^{(2)}(j^{(1)})$ into $K_n^{(1)}$ gives

$$K_n^{(1)} = \sum_{j^{(1:2)}}\tilde{\vartheta}_n^{(1)}(j^{(1)})\tilde{\vartheta}_n^{(2)}(j^{(2)}|j^{(1)})K_n^{(3)}(j^{(2)}).$$

Further, repeating this substitution with $K_n^{(3)}(j^{(2)})$,..., and $K_n^{(V-1)}(j^{(V-2)})$ gives

$$K_n^{(1)} = \sum_{j^{(1:V)}}\tilde{\vartheta}_n^{(1)}(j^{(1)})\tilde{\vartheta}_n^{(2)}(j^{(2)}|j^{(1)})\ldots\tilde{\vartheta}_n^{(V)}(j^{(V)}|j^{(V-1)})$$

Using (25), the n-th conditional is

$$\begin{aligned}
\pi_n(j^{(1:V)}|\gamma_{\bar{n}}) &= \frac{\vartheta_n(j^{(1:V)})\prod_{s=1}^{V}\beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})}{\sum_{j^{(1:V)}}\vartheta_n(j^{(1:V)})\prod_{s=1}^{V}\beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})}\\
&= \frac{\prod_{s=2}^{V}\tilde{\vartheta}_n^{(s)}(j^{(s)}|j^{(s-1)})\tilde{\vartheta}_n^{(1)}(j^{(1)})}{\sum_{j^{(1:V)}}\prod_{s=2}^{V}\tilde{\vartheta}_n^{(s)}(j^{(s)}|j^{(s-1)})\tilde{\vartheta}_n^{(1)}(j^{(1)})}\\
&= \frac{1}{K_n^{(1)}}\prod_{s=2}^{V}\tilde{\vartheta}_n^{(s)}(j^{(s)}|j^{(s-1)})\tilde{\vartheta}_n^{(1)}(j^{(1)})\\
&= \frac{\tilde{\vartheta}_n^{(V)}(j^{(V)}|j^{(V-1)})}{K_n^{(V)}(j^{(V-1)})}\\
&\quad\times\frac{K_n^{(V)}(j^{(V-1)})\tilde{\vartheta}_n^{(V-1)}(j^{(V-1)}|j^{(V-2)})}{K_n^{(V-1)}(j^{(V-2)})}\times\dots\\
&\quad\times\frac{K_n^{(3)}(j^{(2)})\tilde{\vartheta}_n^{(2)}(j^{(2)}|j^{(1)})}{K_n^{(2)}(j^{(1)})}\\
&\quad\times\frac{K_n^{(2)}(j^{(1)})\tilde{\vartheta}_n^{(1)}(j^{(1)})}{K_n^{(1)}}\\
&= \prod_{s=2}^{V}\pi_n^{(s)}(j^{(s)}|j^{(s-1)},\gamma_{\bar{n}})\pi_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}),
\end{aligned}$$

where $\pi_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}) = K_n^{(2)}(j^{(1)})\tilde{\vartheta}_n^{(1)}(j^{(1)})/K_n^{(1)}$, which is indeed (31), and

$$\pi_n^{(s)}(j^{(s)}|j^{(s-1)},\gamma_{\bar{n}}) = \frac{K_n^{(s+1)}(j^{(s)})\tilde{\vartheta}_n^{(s)}(j^{(s)}|j^{(s-1)})}{K_n^{(s)}(j^{(s-1)})}.$$

Note that the normalizing constants $K_n^{(1)}$, $K_n^{(s)}(j^{(s-1)})$, $s\in\{2:V\}$, are all positive, otherwise (29) is not a probability distribution on $(\{-1\}^V\uplus\Lambda^{(1:V)})^P$.

Additionally, for (29) to be a probability distribution on $(\{-1\}^V\uplus\Lambda^{(1:V)})^P$, each conditional $\pi_n(j^{(1:V)}|\gamma_{\bar{n}})$, must be a probability distribution on $\{-1\}^V\uplus\Lambda^{(1:V)}$, which in turn implies that for each $s\in\{2:V\}$,

$$\begin{aligned}
\pi_n^{(s)}(-1|-1,\gamma_{\bar{n}}) &= 1\\
\pi_n^{(s)}(j^{(s)}|-1,\gamma_{\bar{n}}) &= 0, j^{(s)} > -1\\
\pi_n^{(s)}(-1|j^{(s-1)},\gamma_{\bar{n}}) &= 0, j^{(s-1)} > -1
\end{aligned}$$

Otherwise, we would have realizations from $\pi_n(j^{(1:V)}|\gamma_{\bar{n}})$ that are outside of $\{-1\}^V\uplus\Lambda^{(1:V)}$, the very space on which it is defined. Further, the last condition means that for $j^{(s-1)} > -1$, $K_n^{(s+1)}(-1)\tilde{\vartheta}_n^{(s)}(-1|j^{(s-1)}) = 0$, and hence

$$\begin{aligned}
K_n^{(s)}(j^{(s-1)}) &= \sum_{j=-1}^{M^{(s)}}K_n^{(s+1)}(j)\tilde{\vartheta}_n^{(s)}(j|j^{(s-1)})\\
&= \sum_{j=0}^{M^{(s)}}K_n^{(s+1)}(j)\tilde{\vartheta}_n^{(s)}(j|j^{(s-1)}).
\end{aligned}$$

Therefore $\pi_n^{(s)}(j^{(s)}|j^{(s-1)},\gamma_{\bar{n}})$ is given by (32).

C. Proof of Corollary to Proposition 3

Using (34), and noting that $\beta_n^{(s)}(-1|\gamma_{\bar{n}}^{(s)}) = 1$, the normalizing constants (33) can be written as

$$\begin{aligned}
&K_n^{(V)}(j^{(V-1)})\\
&= \begin{cases}\vartheta_n^{(V)}(-1), & j^{(V-1)} = -1\\ \sum_j\beta_n^{(V)}(j|\gamma_{\bar{n}}^{(V)})\vartheta_n^{(V)}(j), & j^{(V-1)} > -1\end{cases}\\
&= \begin{cases}\vartheta_n^{(V)}(-1), & j^{(V-1)} = -1\\ \Upsilon_n^{(V)}, & j^{(V-1)} > -1\end{cases}\\
&K_n^{(V-1)}(j^{(V-2)})\\
&= \begin{cases}\vartheta_n^{(V)}(-1)\,\vartheta_n^{(V-1)}(-1), & j^{(V-2)} = -1\\ \Upsilon_n^{(V)}\sum_j\beta_n^{(V-1)}(j|\gamma_{\bar{n}}^{(V-1)})\vartheta_n^{(V-1)}(j), & j^{(V-2)} > -1\end{cases}\\
&= \begin{cases}\vartheta_n^{(V)}(-1)\,\vartheta_n^{(V-1)}(-1), & j^{(V-2)} = -1\\ \Upsilon_n^{(V)}\Upsilon_n^{(V-1)}, & j^{(V-2)} > -1\end{cases}\\
&\quad\vdots\\
&K_n^{(s)}(j^{(s-1)})\\
&= \begin{cases}\vartheta_n^{(V)}(-1)\,\vartheta_n^{(V-1)}(-1)\dots\vartheta_n^{(s)}(-1), & j^{(s-1)} = -1\\ \Upsilon_n^{(V)}\Upsilon_n^{(V-1)}\dots\Upsilon_n^{(s)}, & j^{(s-1)} > -1\end{cases}.
\end{aligned}$$

Hence, (31) becomes

$$\begin{aligned}
&\pi_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}^{(1)})\\
&= \begin{cases}\dfrac{\vartheta_n^{(1)}(j^{(1)})\prod_{s=2}^{V}\vartheta_n^{(s)}(-1)}{\vartheta_n^{(1)}(-1)\prod_{s=2}^{V}\vartheta_n^{(s)}(-1)+\prod_{s=2}^{V}\Upsilon_n^{(s)}\sum_{j=0}^{M^{(1)}}\beta_n^{(1)}(j|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j)}, & j^{(1)} = -1\\ \dfrac{\beta_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j^{(1)})\prod_{s=2}^{V}\Upsilon_n^{(s)}}{\vartheta_n^{(1)}(-1)\prod_{s=2}^{V}\vartheta_n^{(s)}(-1)+\prod_{s=2}^{V}\Upsilon_n^{(s)}\sum_{j=0}^{M^{(1)}}\beta_n^{(1)}(j|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j)}, & j^{(1)} > -1\end{cases}\\
&= \begin{cases}\dfrac{\prod_{s=1}^{V}\vartheta_n^{(s)}(-1)}{\prod_{s=1}^{V}\vartheta_n^{(s)}(-1)+\prod_{s=1}^{V}\Upsilon_n^{(s)}} & j^{(1)} = -1\\ \dfrac{\prod_{s=2}^{V}\Upsilon_n^{(s)}\beta_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j^{(1)})}{\prod_{s=1}^{V}\vartheta_n^{(s)}(-1)+\prod_{s=1}^{V}\Upsilon_n^{(s)}} & j^{(1)} > -1\end{cases}\\
&= \begin{cases}1-P_n(\Lambda^{(1:V)}), & j^{(1)} = -1\\ \dfrac{P_n(\Lambda^{(1:V)})\beta_n^{(1)}(j^{(1)}|\gamma_{\bar{n}}^{(1)})\vartheta_n^{(1)}(j^{(1)})}{\Upsilon_n^{(1)}}, & j^{(1)} > -1\end{cases}
\end{aligned}$$

and (32) becomes

$$
\begin{aligned}
&\pi_n^{(s)}(j^{(s)}|j^{(s-1)},\gamma_{\bar{n}}^{(s)})\\
&=\begin{cases}
1, & j^{(s)},j^{(s-1)}=-1\\
\dfrac{\prod_{t=s+1}^{V}\Upsilon_n^{(t)}\beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})\vartheta_n^{(s)}(j^{(s)})}{\prod_{t=s+1}^{V}\Upsilon_n^{(t)}\sum_{j^{(s)}=0}^{M^{(s)}}\beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})\vartheta_n^{(s)}(j^{(s)})}, & \\
 & j^{(s)},j^{(s-1)}>-1
\end{cases}\\
&=\begin{cases}
1, & j^{(s)},j^{(s-1)}=-1\\
\dfrac{\beta_n^{(s)}(j^{(s)}|\gamma_{\bar{n}}^{(s)})\vartheta_n^{(s)}(j^{(s)})}{\Upsilon_n^{(s)}}, & j^{(s)},j^{(s-1)}>-1
\end{cases}
\end{aligned}
$$

for $s \in \{2:V\}$.

References

[1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. San Diego, CA, USA: Academic, 1988.

[2] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Norwood, MA, USA: Artech House, 1999.

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

[4] R. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, 2014.

[5] X. Chen, R. Tharmarasa, and T. Kirubarajan, "Multitarget multisensor tracking," in *Academic Press Library in Signal Processing*, vol. 2. New York, NY, USA: Elsevier, 2014, pp. 759–812.

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

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

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

[9] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409–423, Feb. 2009.

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

[11] F. Meyer *et al.*, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[12] B.-N. Vo, S. S. Singh, and W. K. Ma, "Tracking multiple speakers using random sets," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, 2004, pp. 357–360.

[13] N. T. Pham, W. Huang, and S. H. Ong, "Multiple sensor multiple object tracking with GMPHD filter," in *Proc. IEEE 10th Int. Conf. Inf. Fusion*, Jul. 2007, pp. 1–7.

[14] 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.

[15] B.-N. Vo, B.-T. Vo, and D. Phung, "Labeled random finite sets and the Bayes multi-target tracking filter," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6554–6567, Dec. 2014.

[16] B. S. Wei, B. Nener, W. F. Liu, and M. Liang, "Centralized multi-sensor multi-target tracking with labeled random finite sets," in *Proc. Int. Conf. Control, Autom., Inf. Sci.*, 2016, pp. 82–87.

[17] C. Fantacci and F. Papi, "Scalable multisensor multitarget tracking using the marginalized-GLMB density," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 863–867, Jun. 2016.

[18] M. Z. Win, Y. Shen, and W. H. Dai, "A theoretical foundation of network localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1136–1165, Jul. 2018.

[19] M. Z. Win, W. H. Dai, Y. Shen, G. Chrisikos, and H. V. Poor, "Network operation strategies for efficient localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1224–1254, Jul. 2018.

[20] R. Mahler, "Optimal/robust distributed data fusion: A unified approach," *Proc. SPIE*, vol. 4052, 2000, pp. 128–138.

[21] D. E. Clark, S. J. Julier, R. Mahler, and B. Ristic, "Robust multi-object sensor fusion with unknown correlations," in *Proc. Senors Signal Process. Defence*, Sep. 2010, pp. 1–5.

[22] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 508–520, Jun. 2013.

[23] M. Uney, D. E. Clark, and S. J. Julier, "Distributed fusion of PHD filters via exponential mixture densities," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 521–531, Jun. 2013.

[24] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and R. Mahler, "Distributed fusion of multitarget densities and consensus PHD/CPHD filters," *Proc. SPIE*, vol. 9474, 2015, Art. no. 94740E.

[25] M. B. Guldogan, "Consensus Bernoulli filter for distributed detection and tracking using multi-static Doppler shifts," *IEEE Signal Process. Lett.*, vol. 24, no. 6, pp. 672–676, Jun. 2014.

[26] W. Yi, M. Jiang, R. Hoseinnezhad, and B. Wang, "Distributed multisensor fusion using generalised multi-Bernoulli densities," *IET Radar, Sonar, Navigat.*, vol. 11, no. 3, pp. 434–443, Mar. 2016.

[27] B. L. Wang, W. Yi, R. Hoseinnezhad, S. Q. Li, L. J. Kong, and X. B. Yang, "Distributed fusion with multi-Bernoulli filter based on generalized covariance intersection," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 242–255, Jan. 2017.

[28] T. C. Li, J. M. Corchado, and S. D. Sun, "On generalized covariance intersection for distributed PHD filtering and a simple but better alternative," in *Proc. IEEE Int. Fusion Conf.*, 2017, pp. 1–8.

[29] C. Fantacci, B.-N. Vo, B.-T. Vo, G. Battistelli, and L. Chisci, "Robust fusion for multisensor multiobject tracking," *IEEE Signal Process. Lett.*, vol. 25, no. 5, pp. 640–644, May 2018.

[30] X. Wang, A. K. Gostar, T. Rathnayake, B. Xu, and A. Bab-Hadiashar, "Centralized multiple-view sensor fusion using labeled multi-Bernoulli filters," *Signal Process.*, vol. 150, pp. 75–84, 2018.

[31] S. Li, W. Yi, R. Hoseinnezhad, G. Battistelli, B. Wang, and L. Kong, "Robust distributed fusion with labeled random finite sets," *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 278–293, Jan. 2018.

[32] S. Li, G. Battistelli, L. Chisci, W. Yi, B. Wang, and L. Kong, "Computationally efficient multi-agent multi-object tracking with labeled random finite sets," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 260–275, Jan. 2019.

[33] B.-N. Vo and B.-T. Vo, "A multi-scan labeled random finite set model for multi-object state estimation," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4948–4963, Oct. 2019.

[34] B.-N. Vo, B.-T. Vo, and H. Hoang, "An efficient implementation of the generalized labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1975–1987, Apr. 2017.

[35] B.-N. Vo and B.-T. Vo, "An implementation of the multi-sensor generalized labeled multi-Bernoulli filter via Gibbs sampling," in *Proc. 20th Annu. Conf. Inf. Fusion*, Xian, China, Jul. 2017, pp. 1–8.

[36] 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.

[37] W. P. Pierskalla, "Letter to the editor - The multidimensional assignment problem," *Oper. Res.*, vol. 16, no. 2, pp. 422–431, 1968.

[38] D. M. Nguyen, H. A. Le Thi, and D. T. Pham, "Solving the multidimensional assignment problem by a cross-entropy method," *J. Combinatorial Optim.*, vol. 27, no. 4, pp. 808–823, 2014.

[39] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PMAI-6, no. 6, pp. 721–741, Nov. 1984.

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

[41] C. Geyer and E. Thompson, "Annealing Markov Chain Monte Carlo with applications to ancestral inference," *J. Amer. Statistical Assoc.*, vol. 90, no. 431, pp. 909–920, Sep. 1995.

[42] R. Neal, "Annealed importance sampling,"*Statist. Comput.*, vol. 11, pp. 125–139, 2000.

[43] L. Devroye, *Non-Uniform Random Variate Generation*. Berlin, Germany: Springer-Verlag, 1986.

[44] W. R. Gilks and P. Wild, "Adaptive rejection sampling for Gibbs sampling," *J. Roy. Statistical Soc. Ser. C*, vol. 41, no. 2, pp. 337–348, 1992.

[45] W. R. Gilks, N. G. Best, and K. K. C. Tan, "Adaptive rejection metropolis sampling within Gibbs sampling," *J. Roy. Statistical Soc. Ser. C*, vol. 44, no. 4, pp. 455–472, 1995.

[46] C. Ritter and M. A. Tanner, "Facilitating the Gibbs sampler: The Gibbs stopper and the Griddy-Gibbs sampler," *J. Amer. Statistical Assoc.*, vol. 87, no. 419, pp. 861–868, 1992.

[47] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook Nonlinear Filtering*, vol. 12, no. 3, pp. 656–704, 2009.

[48] D. Schumacher, 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.

[49] M. Beard, B.-T. Vo, and B.-N. Vo, "Performance evaluation for large-scale multi-target tracking algorithms," in *Proc. 21st Int. Conf. Inf. Fusion*, Cambridge, U.K., Jul. 2018, pp. 1–5.

[50] M. Beard, B.-T. Vo, and B.-N. Vo, "A solution for large-scale multi-object tracking," Apr. 2018, *arXiv:1804.06622*.

[51] 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.

Ba-Ngu Vo received the bachelor's degree in mathematics and electrical engineering (with first-class honors) in 1994, and the Ph.D. degree in 1997. He is currently a Professor and the Chair of Signals and Systems, Curtin University, Bentley, WA, Australia. He is best known as a pioneer in the stochastic geometric approach to multi-object system. His research interests are signal processing, systems theory, and stochastic geometry.

Ba-Tuong Vo received the B.Sc. degree in applied mathematics and the B.E. degree in electrical and electronic engineering (with first-class honors) in 2004 and the Ph.D. degree in engineering (with Distinction) in 2008 from the University of Western Australia, Perth, WA, Australia. He is currently a Professor of Signal Processing, Curtin University, Bentley, WA, Australia. His primary research interests in random sets, filtering and estimation, multiple object systems.

Michael Beard was born in Sydney, Australia, in 1981. He received the B.E. degree in computer systems engineering, in 2003, the B.Ma. and Computer Science degree from the University of Adelaide, Adelaide, SA, Australia, and the Ph.D. degree in engineering (with Chancellor's commendation) from Curtin University, Bentley, WA, Australia, in 2016. Since 2004, he has been employed by the Defence Science and Technology Group (DST Group, formerly the Defence Science and Technology Organization), Australia, where he is currently a tracking and data fusion Research Scientist. From 2017 to 2018, he undertook a secondment to Curtin University as a Postdoctoral Research Fellow with the School of Electrical Engineering, Computing and Mathematical Sciences. His research interests include multi-sensor multi-target tracking, nonlinear filtering, and sensor management.