Information Exchange track-before-detect Multi-Bernoulli filter for superpositional sensors

Elinor S. Davies, Ángel F. García-Fernández

***Abstract*—In this paper we derive the Information Exchange track-before-detect Multi-Bernoulli (IEMB) filter for multi-target filtering with superpositional sensors. The IEMB filter propagates a multi-Bernoulli density through the filtering recursion and each Bernoulli is propagated with its own prediction and update step. At each update step, each Bernoulli filter exchanges the predicted mean and covariance matrix of its measurement contribution with the other Bernoulli filters. The exchanged information is then used by the filters to perform the update step. Additionally, we propose the Iterated Posterior Linearisation Filter (IPLF) implementation of the IEMB filter (IEMB-IPLF). We compare the IEMB-IPLF filter to a number of other non-linear filtering methods showing the benefits of the proposed filter.**

***Index Terms*—Multi-target tracking, track-before-detect, multi-Bernoulli filtering, iterated posterior linearisation filter.**

I. Introduction

There are many areas of research in which multiple target tracking [1] is an important component. These include autonomous vehicles [2], space situational awareness [3], positioning and mapping in communication networks [4] and defence [1]. Most target tracking methods are detection based, i.e. measurements which exceed a certain threshold result in a detection being generated, which is then fed into the multiple target tracking algorithm [1], [5], [6].

In this paper we focus on the track-before-detect approach to multiple target tracking [7]–[11]. In track-before-detect, we do not threshold the received signal, and we directly feed the raw signal to the multiple target tracking algorithm. This is especially beneficial in situations where there is a low signal-to-noise ratio, as a detection-based tracker is more likely to miss the targets.

In a Bayesian setting, the information of interest about the current targets is encapsulated in the posterior density, which is the density of the current targets given all past measurements [12]. Computing the posterior in a track-before-detect scenario is challenging due to the non-linear track-before-detect measurement models. There are three situations of interest in which this challenge has been addressed: 1) single-target tracking, 2) multiple target tracking with fixed and known number of targets, and 3) multiple target tracking with an unknown and time-varying number of targets. We proceed to review previous work in these cases.

E. S. Davies (e.s.davies@liverpool.ac.uk) and Á. F. García-Fernández (angel.garcia-fernandez@liverpool.ac.uk) are with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, United Kingdom. Á. F. García-Fernández is also with the ARIES Research Center, Universidad Antonio de Nebrija, Madrid, Spain. This work was supported by the EPSRC Centre for Doctoral Training in Distributed Algorithms EP/S023445/1 and Leonardo UK.

Single-target tracking with non-linear and non-Gaussian measurement models can be addressed using particle filtering [13]–[15], which is an asymptotitally optimal method to approximate the posterior, or Gaussian filtering, which is suitable for unimodal posteriors [12]. Standard Gaussian filtering approaches to deal with non-linear measurements are the Extended Kalman filter (EKF) [12], and the sigma-point Kalman filters, such as the Unscented Kalman Filter (UKF) [16], the Cubature Kalman Filter [17], [18] and the Quadrature Kalman Filter [19]. All these non-linear Kalman filters perform a linearisation of the non-linear measurement in the update step, either by analytical linearisation, as in the EKF or by statistical linear regression (SLR) [19], as in the sigma-point Kalman filters. These methods generally work well unless the non-linearities are significant enough in relation to the covariance matrix of the measurement noise [20].

The optimal linearisation of the measurement function in a mean square error sense is given by its SLR with respect to the posterior [21]. This insight leads to a filter in which we refine the selection of the linearisation via iterations such that we can achieve a more accurate posterior approximation than non-iterated Kalman filters: the iterated posterior linearization filter (IPLF) [21], [22]. Specifically, the IPLF performs iterated SLRs, approximated via sigma points, based on the current approximation of the posterior, which enables an iterated improvement of the posterior approximation.

To deal with track-before-detect with a fixed and known number of targets, a challenge is to deal with the high-dimensional state space. In the context of particle filtering, a common approach is to use independent target approximations, which have been used in a variety of filters [7], [8], [23]–[26]. Of special relevance to this paper is the multiple particle filtering approach [23], [27] in which a particle filter is applied to each target independently, whilst an exchange of information between different targets is performed at each time step, thus providing, for example, the predicted mean, [23], or the predicted mean and its covariance matrix [27], [28].

The concept of multiple particle filtering can be extended to non-linear Gaussian filters. In multiple filtering, the state is partitioned, such that each partition corresponds to a target, and a separate filter is run for each partition. Again, the filters exchange information; the predicted mean in [29], and the predicted mean and its covariance matrix in [28], [30].

To deal with an unknown and variable number of targets, a conventional approach is the use of random finite sets (RFS) [5], where the multi-object state is a set of targets. Of particular relevance in the design of track-before-detect filters for an unknown and variable number of targets is the

type of measurement model. There are three main types of measurement models: a general measurement model [7], [8], [31], [32], a superpositional sensor model, in which each target contribution is additive [33]–[36], and a measurement model with non-overlapping target contributions to the measurement [37], [38]. This last option implies that targets are assumed to be sufficiently far away from each other at all times, which is less challenging.

For these measurement models, there are different track-before-detect RFS filters in the literature. For a superpositional measurement model and a Poisson point process (PPP) birth model, a probability hypothesis density (PHD) filter with a particle filter implementation was proposed in [33]. For a superpositional measurement model and an independent identically distributed cluster process birth model, cardinality probability hypothesis density (CPHD) filters with particle filtering implementations were proposed in [33], [35]. Also, for superpositional sensors and a PPP birth model, a filter based on marked PPPs and a particle filter implementation was proposed in [34].

A track-before-detect Bernoulli filter, which considers at most one alive target, with a general measurement model and a particle filter implementation was proposed in [39]. A multi-Bernoulli filter, with a multi-Bernoulli birth model, non-overlapping target measurement contributions and a particle filter implementation was proposed in [37]. An improved multi-Bernoulli particle filter with a general measurement model and iterated Kullback-Leibler divergence (KLD) minimisations to improve the posterior density approximation was proposed in [32]. A Poisson multi-Bernoulli filter, with Poisson birth model, and non-overlapping target contributions to the measurement is proposed in [38]. A multi-target particle filter with a general measurement model and PPP birth, implemented with a two-layer particle filter was proposed in [31]. Two multi-target particle filters with target births sampled from an existence grid were proposed in [7], [8]. In addition, in parallel to this paper, a multi-Bernoulli filter with a more general superpositional sensor model that includes a state-dependent covariance matrix, and an implementation based on belief propagation and particle filtering was proposed in [40].

In this paper, we propose a track-before-detect multi-Bernoulli filter for superpositional sensors, which we refer to as the Information Exchange Multi-Bernoulli (IEMB) filter. The IEMB filter can be considered an extension of the multiple filtering approach to deal with an unknown and time-varying number of targets. The IEMB filter is derived by applying KLD minimisation, augmenting the target states with auxiliary variables [41], after each update step, to optimally approximate the posterior as multi-Bernoulli, see Figure 1. In the IEMB filter, every potential Bernoulli target has an independent prediction step. In the update step, each Bernoulli target shares its predicted measurement information consisting of its predicted mean and covariance matrix. Each potential target then performs its own update using the means and covariance matrices shared by the other potential targets.

The second contribution of the paper is a Gaussian implementation of the IEMB filtering recursion via the IPLF that aims to make an optimal use of the information provided by the measurements in each update. This implementation of the IEMB is referred to as the IEMB-IPLF. The IEMB-IPLF filter is compared to other track-before-detect filters based on independent multi-Bernoulli filtering [37], and other Gaussian filters. A preliminary conference version of this paper is [42]. Compared to the conference version, the journal extension contains a more detailed presentation, the derivation of the IEMB via KLD minimisation with auxiliary variables, the IPLF implementation, and more thorough simulation results.

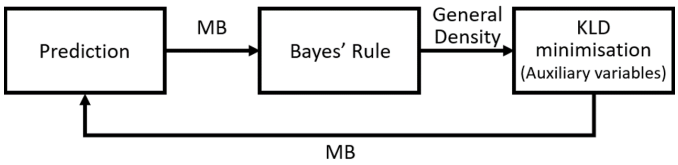


Figure 1: Diagram of the IEMB filter for track-before-detect. The filter propagates a multi-Bernoulli (MB) density on the set of potential targets for the current time step. After the update step, the IEMB filter performs a KLD minimisation on the posterior by introducing auxiliary variables to obtain the marginal multi-Bernoulli distributions for each potential target.

The rest of the paper is organised as follows. Section II presents the problem formulation. Section III derives the IEMB filter. The IPLF implementation is described in Section IV. Simulation results are analysed in Section V. Finally, conclusions are drawn in Section VI.

II. Problem formulation

We aim to approximate the posterior density of the set of targets at the current time step given the sequence of measurements up to the current time step. In particular, we propose a new method to perform the update step for multi-Bernoulli densities. We present the measurement model in Section II-A, the dynamic model in Section II-B, the prediction step in Section II-C and the update step in Section II-D.

A. Measurement model with superpositional sensors

A set of targets X_k at time step k is observed by way of a vector of measurements $z_k = \left[z_k^1, ..., z_k^M\right]^T$, where $z_k^j \in \mathbb{R}^{n_z}$ is the measurement from the j^{th} cell, n_z is the dimensionality of z_k^j, and M is the number of cells. The vector of measurements z_k has a likelihood

$$p\left(z_k|X_k\right) = \prod_{j=1}^{M} p\left(z_k^j|X_k\right) \tag{1}$$

which is the product of the likelihood from each cell, given by $p\left(z_k^j|X_k\right)$. This means that, given the set of targets, the measurement in each cell is independent of the rest.

We are interested in the case where there are superpositional sensors for example, radar sensors [5, Chapter 19]. Given that the measurements are superpositional, the measurement in the j^{th} cell is the sum of each target contribution to the measurement and is given by

$$z_k^j = \sum_{x_k \in X_k} h^j\left(x_k\right) + \eta^j \tag{2}$$

where x_k is a target with state $x_k \in \mathbb{R}^{n_x}$, where n_x is the dimensionality of x_k, and $h^j(\cdot)$ is the measurement function for the j th cell. The set of targets X_k therefore belongs to $\mathcal{F}(\mathbb{R}^{n_x})$, which denotes the set of finite subsets of $\mathbb{R}^{n_x}$. Also, η^j is an independent zero-mean Gaussian noise with a covariance matrix R^j.

From (2), we can see that the set of targets X_k only affects the measurements through the sum of terms in (2). Therefore, the likelihood of the j^{th} cell can be written as

$$p\left(z_k^j|X_k\right) = l\left(z_k^j|\sum_{x_k\in X_k} h^j(x_k)\right) \tag{3}$$

where

$$l\left(z_k^j|\sum_{x_k\in X_k} h^j(x_k)\right) = \mathcal{N}\left(z_k^j;\sum_{x_k\in X_k} h^j(x_k), R^j\right) \tag{4}$$

where $\mathcal{N}\left(z_k^j;\hat{z}^j,R^j\right)$ is a Gaussian density with mean $\hat{z}^j$, covariance matrix R^j, evaluated at z_k^j. It should be noted that (4) is a function of the sum of each target contribution to the measurement. The mean and the covariance for the j^{th} cell measurement given X_k are

$$\mathrm{E}\left[z_k^j|X_k\right] = \sum_{x_k\in X_k} h^j(x_k) \tag{5}$$

$$C\left[z_k^j|X_k\right] = R^j. \tag{6}$$

B. Dynamic model

Each target is considered to move independently of all other targets present. New targets can be born, independently of other targets, at each time step. The set X_{k+1} is the union of any surviving targets and any potential targets that are born independently.

1) Multi-Bernoulli birth: To model target birth, a multi-Bernoulli birth model with n_b Bernoulli components is used [5] [43, Example 2]. Each Bernoulli component represents a potential target that may be born. When considering the case where the l^{th} Bernoulli component is empty at time step $k-1$ for a target to exist in time step k it must be born into that time step. The density of the l^{th} Bernoulli component at birth is given by [39]

$$p_{k|k-1}^l(X_k) = \begin{cases} 1-p_b^l & X_k = \emptyset \\ p_b^l b_k^l(x) & X_k = \{x\} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where, p_b^l is the probability of birth and $b_k^l(\cdot)$ its single-target density at birth.

It should be noted that, in detection-based multi-target tracking, it is more convenient to model target birth as a PPP, instead of multi-Bernoulli due to these reasons: there is a more compact representation of the information on undetected targets via a PPP intensity (lowering the number of global hypotheses), the birth model does not cap the maximum number of new born targets, and the initiation of each Bernoulli potential target is driven by the measurements [44], [45]. However, in track-before-detect with PPP birth model, it is not clear how to initiate each Bernoulli potential target taking into account the measurements. Thus, in this work, we adopt a multi-Bernoulli birth model.

2) Bernoulli thinning: When considering the case where the l^{th} Bernoulli component exists at time step $k-1$, for the target to exist in time step k, it must survive to that time step. The density of the surviving l^{th} Bernoulli component given that it existed at the previous time step is [39]

$$p_{k|k-1}^l(X_k|\{x'\}) = \begin{cases} 1-p_s(x') & X_k = \emptyset \\ p_s(x')\pi_{k|k-1}(x|x') & X_k = \{x\} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $p_s(x')$ is the probability that the target survives to the next time step and transitions to a new state with transition density $\pi_{k|k-1}(x|x')$.

C. Multi-Bernoulli prediction

The predicted density $f_{k|k-1}(\cdot)$ is assumed to be multi-Bernoulli with $n_{k|k-1}$ potential targets. This multi-Bernoulli density can then be expressed using the convolution formula for independent RFSs [5]

$$f_{k|k-1}(X_k) = \sum_{\uplus_{l=1}^{n_{k|k-1}} X^l = X_k} \prod_{i=1}^{n_{k|k-1}} f_{k|k-1}^i\left(X^i\right) \tag{9}$$

where $\uplus$ represents the disjoint union, and each Bernoulli component is summed over $(X^1,\ldots,X^{n_{k|k-1}})$ where each l^{th} Bernoulli set X^l is a mutually disjoint (and possibly empty) subset of X_k, and the predicted Bernoulli density of the l^{th} Bernoulli component is [39]

$$f_{k|k-1}^l(X_k) = \begin{cases} 1-r_{k|k-1}^l & X_k = \emptyset \\ r_{k|k-1}^l p_{k|k-1}^l(x) & X_k = \{x\} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $r_{k|k-1}^l$represents the predicted probability of existence for the l^{th} Bernoulli component and $p_{k|k-1}^l(\cdot)$ is its single-target predicted density. The Bernoulli density has a non-zero value for sets with no more than one target.

It should also be noted that the multi-Bernoulli density integrates to one by using the set integral [5]

$$\begin{aligned} &\int f_{k|k-1}(X)\,\delta X \\ &= \sum_{n=0}^{\infty}\frac{1}{n!}\int f_{k|k-1}(\{x_1,...,x_n\})\,dx_{1:n} \qquad (11)\\ &= 1 \qquad (12)\end{aligned}$$

where $x_{1:n} = (x_1,...,x_n)$.

For a potentially new born target, the l^{th} Bernoulli predicted density is given by (7). For the surviving l^{th} potential target, the predicted probability of existence $r_{k|k-1}^l$ and single-target density $p_{k|k-1}^l(\cdot)$ [45]

$$r_{k|k-1}^l = r_{k-1|k-1}^l\left\langle p_s, p_{k-1|k-1}^l\right\rangle \tag{13}$$

$$p_{k|k-1}^l(x) = \frac{\int \pi_{k|k-1}^l(x|x')\,p_s(x')\,p_{k-1|k-1}^l(x')\,dx'}{\left\langle p_s, p_{k-1|k-1}^l\right\rangle} \tag{14}$$

where $\langle h, g \rangle$ is the notation for the inner product $\langle h, g \rangle = \int h(x)\, g(x)\, dx$.

D. Multi-Bernoulli update

Through the application of Bayes' law on the likelihood (3) and the predicted density (9), the posterior density is

$$f_{k|k}(X_k) = \frac{\left(\prod_{j=1}^{M} p\left(z_k^j | X_k\right)\right) f_{k|k-1}(X_k)}{\int \left[\left(\prod_{j=1}^{M} p\left(z_k^j | X_k\right)\right) f_{k|k-1}(X_k)\right] \delta X_k}. \quad (15)$$

Equation (15) is not a multi-Bernoulli density. Therefore, in order to develop a multi-Bernoulli filter for track-before-detect, we need to approximate this density as multi-Bernoulli, which is done in the next section.

III. Information exchange multi-Bernoulli filter

In this section we present a multi-Bernoulli approximation to the posterior density (15) and hence we provide a track-before-detect multi-Bernoulli update. In Section III-A we introduce auxiliary variables in the predicted and posterior densities. With these auxiliary variables, in Section III-B, we derive an optimal multi-Bernoulli approximation to the posterior via KLD minimisation.

A. Auxiliary variables

In this section, we introduce auxiliary variables in the predicted multi-Bernoulli density (9) to mark each Bernoulli component with an index [41]. We use these auxiliary variables to remove the convolution sum in (9) and then obtain the updated marginal distributions for each potential target minimising the KLD (see Section III-B). It should be noted that the use of auxiliary variables is common in Bayesian inference to remove sums over latent/hidden variables by making them explicit in the density under consideration, see for example, auxiliary particle filtering [46], expectation maximisation [47] and the concept of opening in factor graphs [48].

For a multi-Bernoulli prior, the use of auxiliary variables is equivalent to target labelling [31], [49] but without adding these variables to the dynamic/measurement models. In addition, in the case where the prior is a Poisson multi-Bernoulli mixture [44], [50], the use of auxiliary variables has the benefit of being able to handle multiple targets with the same auxiliary variable to represent undetected targets [41]. We proceed to explain how auxiliary variables are introduced in this setting.

Given (9), we augment the single-target space with an auxiliary variable u such that the single-target state is (u, x). The space of auxiliary variables is $\mathbb{U}_k = \{1, ..., n_{k|k-1}\}$, and the augmented single-target state space is $\mathbb{U}_k \times \mathbb{R}^{n_x}$ [41]. The set of targets with the corresponding auxiliary variables is denoted as $\widetilde{X}_k$. Then, the predicted multi-Bernoulli density with auxiliary variables is defined as follows.

Definition 1. Given the predicted multi-Bernoulli density $f_{k|k-1}(\cdot)$ in (9), we define the multi-Bernoulli density $\widetilde{f}_{k|k-1}(\cdot)$ with auxiliary variables as [41]

$$\widetilde{f}_{k|k-1}\left(\widetilde{X}_k\right) = \prod_{i=1}^{n_{k|k-1}} \widetilde{f}^i_{k|k-1}\left(\widetilde{X}^i_k\right) \quad (16)$$

where

$$\widetilde{X}^i_k = \left\{(u, x) \in \widetilde{X}_k : u = i\right\} \quad (17)$$

and the predicted density of the i^{th} Bernoulli component (10) with auxiliary variable is

$$\widetilde{f}^i_{k|k-1}\left(\widetilde{X}_k\right) = \begin{cases} 1 - r^i_{k|k-1} & \widetilde{X}_k = \emptyset \\ r^i_{k|k-1} p^i_{k|k-1}(x)\, \delta_i[u] & \widetilde{X}_k = \{(u, x)\} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

where $\delta_i[u]$ is the Kronecker delta, which meets $\delta_i[u] = 1$ if $u = i$, and $\delta_i[u] = 0$ if $u \neq i$.

It should be noted that the use of the Kronecker delta term $\delta_i[u]$ ensures that the i^{th} Bernoulli component $\widetilde{f}^i_{k|k-1}(\cdot)$ is non-zero only if evaluated on a set with auxiliary variable i. This implies that the convolution sum, which appears in (9), disappears when auxiliary variables are introduced, see (16), as there is only one term in the convolution sum that provides a non-zero density, resulting in (18). Note that the Bernoulli RFSs with auxiliary variable in (18) are independent, as the Bernoulli RFSs without auxiliary variables in (9). In addition, it holds that $\widetilde{X}_k = \widetilde{X}^1_k \uplus ... \uplus \widetilde{X}^{n_{k|k-1}}_k$.

As shown in [41, App. A] in a more general setting, if we integrate the auxiliary variables in (16), we recover the density without auxiliary variables (9). That is,

$$f_{k|k-1}(\{x_1, ..., x_n\}) = \sum_{u_{1:n} \in \mathbb{U}^n_k} \widetilde{f}_{k|k-1}(\{(u_1, x_1), ..., (u_n, x_n)\}) \quad (19)$$

where $\mathbb{U}^n_k = \mathbb{U}_k \times ... \times \mathbb{U}_k$ (with n factors).

In addition, the density $\widetilde{f}_{k|k-1}(\cdot)$ in (16) defines a multi-target density on $\mathcal{F}(\mathbb{U}_k \times \mathbb{R}^{n_x})$ that integrates to one such that

$$\int \widetilde{f}_{k|k-1}\left(\widetilde{X}\right) \delta \widetilde{X} = \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{u_{1:n} \in \mathbb{U}^n_k} \int \widetilde{f}_{k|k-1}(\{(u_1, x_1), ..., (u_n, x_n)\})\, dx_{1:n} \quad (20)$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!} \int f_{k|k-1}(\{x_1, ..., x_n\})\, dx_{1:n} \quad (21)$$

$$= 1 \quad (22)$$

where we have used (19) and the fact that the set integral of $f_{k|k-1}(\cdot)$ is one.

B. Update step via KLD minimisation

We obtain the update step by minimising the KLD using the auxiliary variables. Introducing auxiliary variables, the posterior (15) becomes

$$\widetilde{f}_{k|k}\left(\widetilde{X}_k\right) \propto \prod_{j=1}^{M} l\left(z_k^j|\sum_{(u,x)\in\widetilde{X}_k} h^j\left(x\right)\right)\prod_{i=1}^{n_{k|k-1}}\widetilde{f}_{k|k-1}^{i}\left(\widetilde{X}_k^i\right). \tag{23}$$

It should be noted that, if we integrate the auxiliary variables in the updated density (23), we obtain the updated density without auxiliary variables (15). That is,

$$f_{k|k}\left(\{x_1,...,x_n\}\right) = \sum_{u_{1:n}\in\mathbb{U}_k^n}\widetilde{f}_{k|k}\left(\{(u_1,x_1),...,(u_n,x_n)\}\right). \tag{24}$$

This property is a direct consequence of (19) and the fact that the likelihood does not depend on the auxiliary variables.

We aim to approximate $\widetilde{f}_{k|k}\left(\cdot\right)$ as a multi-Bernoulli density $\widetilde{q}\left(\cdot\right)$ (with auxiliary variables) of the form

$$\widetilde{q}\left(\widetilde{X}_k\right) = \prod_{i=1}^{n_{k|k-1}}\widetilde{q}^{i}\left(\widetilde{X}_k^i\right) \tag{25}$$

where $\widetilde{q}^{i}\left(\cdot\right)$ is a Bernoulli density of the form (18). The sequence of sets for all potential targets excluding the u^{th} potential target is denoted by

$$\widetilde{X}_k^{(-u)} = \left(\widetilde{X}_k^1,\ldots,\widetilde{X}_k^{u-1},\widetilde{X}_k^{u+1},\ldots,\widetilde{X}_k^{n_{k|k-1}}\right). \tag{26}$$

Lemma 2. *Given* $\widetilde{f}_{k|k}(\cdot)$ *in* (23)*, the multi-Bernoulli density* $\widetilde{q}\left(\cdot\right)$ *of the form* (25) *that minimises the KLD*

$$\mathrm{D}\left(\widetilde{f}_{k|k}||\widetilde{q}\right) = \int \widetilde{f}_{k|k}\left(\widetilde{X}_k\right)\log\frac{\widetilde{f}_{k|k}\left(\widetilde{X}_k\right)}{\widetilde{q}\left(\widetilde{X}_k\right)}\delta\widetilde{X}_k \tag{27}$$

has the Bernoulli density $\widetilde{q}^{u}\left(\cdot\right)$ *given by*

$$\widetilde{q}^{u}\left(\widetilde{X}_k^u\right) = \widetilde{f}_{k|k}^{u}\left(\widetilde{X}_k^u\right) \tag{28}$$

$$= \int \widetilde{f}_{k|k}\left(\widetilde{X}_k^1\uplus...\uplus\widetilde{X}_k^{n_{k|k-1}}\right)\delta\widetilde{X}_k^{(-u)} \tag{29}$$

$$\triangleq \int...\int\int...\int \widetilde{f}_{k|k}\left(\widetilde{X}_k^1\uplus...\uplus\widetilde{X}_k^{n_{k|k-1}}\right) \times\delta\widetilde{X}_k^1...\delta\widetilde{X}_k^{u-1}\delta\widetilde{X}_k^{u+1}...\delta\widetilde{X}_k^{n_{k|k-1}} \tag{30}$$

where $\widetilde{f}_{k|k}^{u}\left(\cdot\right)$ *is the marginal posterior of the* u^{th} *potential target and* (29) *integrates out all potential targets except the one with auxiliary variable* u*. In terms of the likelihood and predicted density,* $\widetilde{f}_{k|k}^{u}\left(\cdot\right)$ *has the expression*

$$\widetilde{f}_{k|k}^{u}\left(\widetilde{X}_k^u\right) = \frac{p^u\left(z_k|\widetilde{X}_k^u\right)\widetilde{f}_{k|k-1}^{u}\left(\widetilde{X}_k^u\right)}{\int p^u\left(z_k|\widetilde{X}_k^u\right)\widetilde{f}_{k|k-1}^{u}\left(\widetilde{X}_k^u\right)\delta\widetilde{X}_k^u} \tag{31}$$

where

$$p^u\left(z_k|\widetilde{X}_k^u\right) = \int\prod_{j=1}^{M} l\left(z_k^j|\sum_{(u,x)\in\widetilde{X}_k^1\uplus...\uplus\widetilde{X}_k^{n_{k|k-1}}} h^j\left(x\right)\right)$$

$$\prod_{i=1:i\neq u}^{n_{k|k-1}}\widetilde{f}_{k|k-1}^{i}\left(\widetilde{X}_k^i\right)\delta\widetilde{X}_k^{(-u)}. \tag{32}$$

The proof is provided in Appendix A.

This lemma indicates that, in order to minimise the KLD, $q^u\left(\cdot\right)$ is given by the marginal distributions for the u^{th} potential target. As indicated by (31), this can be achieved by performing an update using a modified likelihood for each potential target. The modified likelihood for the u^{th} potential target is (32) which is found by integrating out the contribution of the other potential targets.

Equation (31) can be used to perform a Bernoulli update for the u^{th} potential target separately from the rest of the targets. The updated probability of existence and single-target density of the u-th potential target, from (31), are given by [39]

$$r_{k|k}^{u} = \frac{r_{k|k-1}^{u}p^u\left(z_k|\left|\widetilde{X}_k^u\right|=1\right)}{l_{k|k}^{u}} \tag{33}$$

$$l_{k|k}^{u} = \left(1-r_{k|k-1}^{u}\right)p^u\left(z_k|\emptyset\right) + r_{k|k-1}^{u}p^u\left(z_k|\left|\widetilde{X}_k^u\right|=1\right)$$

$$p_{k|k}^{u}\left(x\right) = \frac{p_{k|k-1}^{u}\left(x\right)p^u\left(z_k|\left\{x\right\}\right)}{\int p_{k|k-1}^{u}\left(x\right)p^u\left(z_k|\left\{x\right\}\right)dx} \tag{34}$$

$$p^u\left(z_k|\left|\widetilde{X}_k^u\right|=1\right) = \int p_{k|k-1}^{u}\left(x\right)p^u\left(z_k|\left\{x\right\}\right)dx \tag{35}$$

where $p^u\left(z_k|\left\{x\right\}\right)$ is the likelihood given that the u^{th} potential target has state $\{x\}$, $p^u\left(z_k|\left|\widetilde{X}_k^u\right|=1\right)$ is the likelihood given that the u^{th} potential target exists, and $p^u\left(z_k|\emptyset\right)$ is the likelihood given that the u^{th} potential target does not exist.

We would like to remark that, as proved in [41], the KLD with auxiliary variables is an upper bound of the KLD without auxiliary variables, which is the one we would be mainly interested in. That is,

$$\mathrm{D}\left(f_{k|k}\left\|q\right.\right) \leq \mathrm{D}\left(\widetilde{f}_{k|k}\left\|\widetilde{q}\right.\right) \tag{36}$$

where density $q(\cdot)$ is obtained by integrating out the auxiliary variables in $\widetilde{q}(\cdot)$, see (19).

IV. Gaussian IPLF implementation

In this section, we propose a Gaussian implementation of the IEMB filter based on the IPLF. In Section IV-A, we present the prediction step. In Section IV-B, we calculate the conditional moments of the measurements given the set of targets. In Section IV-C, we provide the updated Bernoulli when we perform a Gaussian update. Section IV-D presents the concept of statistical linear regression, Section IV-E, the IPLF update, and Section IV-F, the sigma-point implementation.

A. Prediction

The IEMB filter described in Section III is implemented by using Gaussian approximations. We assume target birth is a multi-Bernoulli where the single-target density for the i^{th} target is Gaussian with mean $\bar{x}_b^i$ and covariance matrix P_b^i.

Therefore, for new born targets, $r^i_{k|k-1}$, $\bar{x}^i_{k|k-1}$ and $P^i_{k|k-1}$ are

$$r^i_{k|k-1} = p^i_b, \qquad \bar{x}^i_{k|k-1} = \bar{x}^i_b, \qquad P^i_{k|k-1} = P^i_b.$$

The transition density for a single target is linear and Gaussian

$$\pi^i_{k|k-1}\left(x|x^i_{k-1}\right) = \mathcal{N}\left(x; Fx^i_{k-1}, Q\right) \tag{37}$$

where F is the transition matrix and Q the covariance matrix of the measurement noise. The predicted mean and covariance are found by using Kalman filter equations [12]

$$\bar{x}^u_{k|k-1} = F\bar{x}^u_{k-1|k-1} \tag{38}$$

$$P^u_{k|k-1} = FP^u_{k-1|k-1}F^T + Q. \tag{39}$$

Assuming that the probability of survival is a constant p_s, the probability of existence is predicted using [5]

$$r^u_{k|k-1} = p_s r^u_{k-1|k-1}. \tag{40}$$

B. Conditional moments

To perform the update of the IEMB filter using the IPLF, as explained in the following two Sections (Subsection IV-C and Section IV-D), we require the calculation of the conditional mean and covariance of the measurement [22], which we provide in this section.

The conditional mean and covariance of (32) given $\widetilde{X}^u_k = \{(u, x^u)\}$ for the measurement model in (1) are provided by the following proposition.

Proposition 3. *Assuming the predicted density is multi-Bernoulli of the form $\widetilde{f}_{k|k-1}(\cdot)$ in* (16)*, the conditional mean and covariance matrix of z^j_k given $\widetilde{X}^u_k = \{(u, x^u)\}$ are*

$$\mathrm{E}\left[z^j_k|\widetilde{X}^u_k = \{(u,x^u)\}\right] = h^j\left(x^u\right) + \sum_{i=1:i\neq u}^{n_{k|k-1}} r^i \mathrm{E}_i\left[h^j(x)\right] \tag{41}$$

$$\mathrm{C}\left[z^j_k|\widetilde{X}^u_k = \{(u,x^u)\}\right] = R^j + \sum_{i=1:i\neq u}^{n_{k|k-1}} \widetilde{\mathrm{C}}_i\left[h^j\left(x^i\right)\right], \tag{42}$$

where

$$\mathrm{E}_i\left[h^j(x)\right] = \int h^j(x)\, p^i_{k|k-1}(x)\, dx \tag{43}$$

$$\widetilde{\mathrm{C}}_i\left[h^j\left(x^i\right)\right] = r^i \mathrm{E}_i\left[h^j\left(x^i\right)\left(h^j\left(x^i\right)\right)^T\right] - \left(r^i\right)^2 \mathrm{E}_i\left[h^j\left(x^i\right)\right]\left(\mathrm{E}_i\left[h^j\left(x^i\right)\right]\right)^T \tag{44}$$

$$\mathrm{E}_i\left[h^j(x)\left(h^j(x)\right)^T\right] = \int h^j(x)\left(h^j(x)\right)^T p^i_{k|k-1}(x)\, dx. \tag{45}$$

and the sums in (41) *and* (42) *go through all $i \in \{1, ..., n_{k|k-1}\}$ except $i = u$.*

In addition, the conditional mean and and covariance matrix of z^j_k given $\widetilde{X}^u_k = \emptyset$ are

$$\mathrm{E}\left[z^j_k|\widetilde{X}^u_k = \emptyset\right] = \sum_{i=1:i\neq u}^{n_{k|k-1}} r^i \mathrm{E}_i\left[h^j(x)\right] \tag{46}$$

$$\mathrm{C}\left[z^j_k|\widetilde{X}^u_k = \emptyset\right] = R^j + \sum_{i=1:i\neq u}^{n_{k|k-1}} \widetilde{\mathrm{C}}_i\left[h^j\left(x^i\right)\right]. \tag{47}$$

The proof of Proposition 3 is provided in Appendix B. It should be noted that the conditional covariance of z^j_k is the same for the cases $\widetilde{X}^u_k = \{(u, x^u)\}$ and $\widetilde{X}^u_k = \emptyset$. For the conditional mean of z^j_k, the case $\widetilde{X}^u_k = \{(u, x^u)\}$ has the additional term $h^j(x^u)$.

C. Gaussian update

We now proceed to explain how to perform a Gaussian update based on the conditional moments (41) and (42). The measurement function and noise covariance for all sensors is written as

$$h(x) = \left[h^1(x), ..., h^M(x)\right]^T \tag{48}$$

$$R = \mathrm{diag}\left(R^1, ..., R^M\right) \tag{49}$$

where $\mathrm{diag}(\cdot)$ is used to represent a block diagonal matrix with the indicated matrices in the block diagonal. To perform a Gaussian update in closed form, we make the approximation [21]

$$h\left(x^u\right) \approx Ax^u + b + e \tag{50}$$

where $A \in \mathbb{R}^{Mn_z \times n_x}$, $b \in \mathbb{R}^{Mn_z}$ and $e \in \mathbb{R}^{Mn_z}$ is a zero-mean Gaussian distributed random variable with covariance matrix Ω. The variable e is uncorrelated with x and η. Then, the density of the measurement given the state can be approximated as Gaussian

$$p\left(z_k|\left\{\widetilde{X}^u_k = \{(u,x^u)\}\right\}\right) \approx \mathcal{N}\left(z_k; Ax^u + b + \hat{z}^u_{\mathrm{corr}}, R + S^u_{\mathrm{corr}} + \Omega\right), \tag{51}$$

where

$$\hat{z}^u_{\mathrm{corr}} = \sum_{i=1:i\neq u}^{n_{k|k-1}} r^i \mathrm{E}_i\left[h(x)\right] \tag{52}$$

$$S^u_{\mathrm{corr}} = \sum_{i=1:i\neq u}^{n_{k|k-1}} \left[r^i \mathrm{E}_i\left[h(x)\left(h(x)\right)^T\right] - \left(r^i\right)^2 \mathrm{E}_i\left[h(x)\right]\left(\mathrm{E}_i\left[h(x)\right]\right)^T\right] \tag{53}$$

as required by (41) and (42).

Given (51), the single-target updated mean and covariance matrix are

$$\overline{x}^u_{k|k} = \overline{x}^u_{k|k-1} + K^u\left(z_k - \hat{z}^u_{k|k-1}\right) \tag{54}$$

$$P^u_{k|k} = P^u_{k|k-1} - K^u S^u \left(K^u\right)^T \tag{55}$$

$$\hat{z}^u_{k|k-1} = A\overline{x}^u_{k|k-1} + b + \hat{z}^u_{\mathrm{corr}} \tag{56}$$

$$K^u = P^u_{k|k-1} A^T \left(S^u\right)^{-1} \tag{57}$$

$$S^u = AP^u_{k|k-1}A^T + R + S^u_{\mathrm{corr}} + \Omega. \tag{58}$$

The likelihoods for the existence and non-existence of each potential target, which are required to obtain the updated probability of existence, see (33), are

$$p^u\left(z_k|\left|\widetilde{X}^u_k\right| = 1\right) = \mathcal{N}\left(z_k; \hat{z}^u_{k|k-1}, S^u\right) \tag{59}$$

$$p^u(z_k|\emptyset) = \mathcal{N}(z_k; \hat{z}^u_{\text{corr}}, R + S^u_{\text{corr}}). \quad (60)$$

Therefore, once we make approximation (51), the accuracy of the posterior moments $\overline{x}^u_{k|k}$ and $P^u_{k|k}$, and the updated probability of existence $r^u_{k|k}$ only depend on how we choose the parameters (A, b, Ω). In the following sections, we explain the proposed approach to the selection of these parameters.

D. Statistical linear regression

In this section, we explain the key concept of statistical linear regression to optimally select the parameters (A, b, Ω) [19]. With SLR, we obtain an optimal linearisation of a function $h(\cdot)$ that minimises the mean square error with respect to a probability density $p(\cdot)$, with mean $\overline{x}$ and covariance matrix P. That is, with SLR, we obtain the linearisation

$$\left(A^+, b^+\right) = \underset{(A,b)}{\arg\min}\, \mathrm{E}\left[\|h(x) - Ax - b\|^2\right] \quad (61)$$

where the expected value is taken w.r.t. density $p(\cdot)$. That is, the best (affine) approximation of the function $h(\cdot)$ in the sense of minimising the mean square error w.r.t. density $p(\cdot)$ is $A^+x + b^+$.

We can solve the optimisation problem (61) by setting

$$A^+ = \Psi^T P^{-1} \quad (62)$$

$$b^+ = \overline{z} - A^+\overline{x} \quad (63)$$

where

$$\overline{z} = \mathrm{E}[h(x)]$$
$$\Psi = \mathrm{E}\left[(x - \overline{x})(h(x) - \overline{z})^T\right].$$

The resulting mean square error matrix gives us the noise covariance parameter

$$\begin{aligned}\Omega^+ &= \mathrm{E}\left[\left(h(x) - A^+x - b^+\right)\left(h(x) - A^+x - b^+\right)^T\right] \\ &= \mathrm{C}[h(x)] - A^+P\left(A^+\right)^T \end{aligned} \quad (64)$$

where $\mathrm{C}[h(x)]$ is the covariance of $h(x)$ w.r.t. density $p(\cdot)$.

The moments $\mathrm{E}[h(x)]$, $\mathrm{E}\left[(x - \overline{x})(h(x) - \overline{z})^T\right]$ and $\mathrm{C}[h(x)]$ required to compute the SLR can be approximated using first order Taylor series [22], sigma-point methods (such as unscented and cubature transform) [16], [17] and quadrature methods [19]. It should be noted that, if the measurement function is linear $h(x) = Hx$, then, $A^+ = H$, $b^+ = 0$ and $\Omega^+ = 0$, which indicates that there are no linearisation errors.

E. IPLF

A standard approach to the use of SLR in Gaussian nonlinear filtering is to obtain (A, b, Ω) w.r.t. the predicted density and then to apply the resulting (affine) Kalman filter update [12]. This is the approach followed by non-iterated sigma-point Kalman filtering methods such as the UKF and the CKF [15], [16], [18], [30]. This approach works well for mild non-linearities or sufficiently high measurement noise, as can be proved by a KLD analysis [20]. One reason for its lower performance in high nonlinearities is that this selection of (A, b, Ω) is sub-optimal, as it is not using all available information. In particular, it does not use the value of the current measurement z.

Once we know z, the SLR, should be done w.r.t. the updated density so that we take the value of the measurement into account. We cannot perform the SLR w.r.t. the posterior as it is the density we would like to approximate. Nevertheless, this gives rise to the IPLF that performs iterated SLRs w.r.t. our best approximation of the posterior. In the IEMB setting, this implies that we will carry out iterated SLR for each potential target, also taking into account $\hat{z}^u_{\text{corr}}$ and S^u_{corr}, which are computed beforehand. That is, we start with the predicted mean and covariance $\overline{x}^{u,0}_{k|k} = \overline{x}^u_{k|k-1}$ and $P^{u,0}_{k|k} = P^u_{k|k-1}$. Then, we perform SLR of $h(\cdot)$ w.r.t. $\overline{x}^{u,0}_{k|k}$ and $P^{u,0}_{k|k}$ using (62)-(64) to obtain $\left(A^1, b^1, \Omega^1\right)$. We plug these parameters into (54) and (55) to obtain the updated mean and covariance at the first iteration $\overline{x}^{u,1}_{k|k}$ and $P^{u,1}_{k|k}$. Now, we calculate the SLR of of $h(\cdot)$ w.r.t. $\overline{x}^{u,1}_{k|k}$ and $P^{u,1}_{k|k}$ to obtain $\left(A^2, b^2, \Omega^2\right)$ and we continue the iteration for a fixed number of steps, or until a certain stopping criterion is met.

It should be noted that, when we apply (54) and (55), the only parameters that change with the iterations are (A, b, Ω), the predicted mean $\overline{x}^u_{k|k-1}$, covariance $P^u_{k|k-1}$, and the exchanged parameters $\hat{z}^u_{\text{corr}}$ and S^u_{corr} remain unaltered with iterations. That is, with each iteration we are refining the choice of the parameters (A, b, Ω), but it is still the same update.

F. Sigma point implementation

To implement the IEMB update, we use sigma points to calculate z^u_{corr} and S^u_{corr} and also to select the SLR parameters A, b, Ω. The sigma points are selected using the unscented transform [16].

To approximate an integral w.r.t. a density with mean $\bar{x}$ and covariance matrix Σ_x, we select a total $2n_x + 1$ sigma points. Given the weight ω^0 of the sigma point located at the origin, the weighting of each sigma point s, for $s > 0$, is given by [16]

$$\omega^s = \frac{1 - \omega^0}{2n_x} \quad (65)$$

where ω^0 is a parameter that must be equal or greater than zero for numerical stability. Using $\bar{x}$ and the corresponding covariance Σ_x, the sigma points are determined using [16]

$$\mathcal{X}^0 = \bar{x} \quad (66)$$

$$\mathcal{X}^i = \bar{x} + \left(\sqrt{\frac{n_x}{1 - \omega^0}\Sigma_x}\right)_i \quad (67)$$

$$\mathcal{X}^{i+n_x} = \bar{x} - \left(\sqrt{\frac{n_x}{1 - \omega^0}\Sigma_x}\right)_i \quad (68)$$

where $\left(\sqrt{\Sigma_x}\right)_i$ is the i^{th} column of the matrix square root of Σ_x (see [16, page 6]) and the matrix square root is calculated using the Cholesky decomposition. We can use these sigma-points to approximate the following moments

$$\hat{h} = E[h(x)] \approx \sum_{s=0}^{2n_x} \omega^s h(\mathcal{X}^s) \quad (69)$$

$$E\left[(x-\overline{x})\left(h(x)-\hat{h}\right)^{T}\right] \approx \sum_{s=0}^{2n_x}\omega^{s}\left(\mathcal{X}^{s}-\bar{x}\right)\left(h\left(\mathcal{X}^{s}\right)-\hat{h}\right)^{T} \tag{70}$$

$$\widehat{h\,h^{T}} = E\left[h(x)\,h(x)^{T}\right] \approx \sum_{s=0}^{2n_x}\omega^{s}h\left(\mathcal{X}^{s}\right)\left(h\left(\mathcal{X}^{s}\right)\right)^{T} \tag{71}$$

$$\mathrm{C}\left[h(x)\right] \approx \sum_{s=0}^{2n_x}\omega^{s}\left(\widehat{h\,h^{T}}-\hat{h}^{i}\left(\hat{h}\right)^{T}\right) \tag{72}$$

1) Sigma-points to compute $\hat{z}^u_{\text{corr}}$ and S^u_{corr}: We use equations (52) and (53) to compute $\hat{z}^u_{\text{corr}}$ and S^u_{corr}. This requires the computation of the moments (69) and (70) w.r.t. the predicted density. Therefore, we select sigma-points using (65)-(68) that match the mean $\overline{x}^u_{k|k-1}$ and covariance matrix $P^u_{k|k-1}$.

2) Sigma-points to compute (A, b, Ω): Once we have approximated $\hat{z}^u_{\text{corr}}$ and S^u_{corr}, we use the sigma points to compute (A, b, Ω) in an iterated fashion using the IPLF. For the i^{th} iteration, we select the sigma-points using (65)-(68) that match the mean $\overline{x}^{u,i}_{k|k}$ and covariance matrix $P^{u,i}_{k|k}$. We then compute the moments (69)-(72). A pseudocode of the IEMB for a general choice of (A, b, Ω) is given in Algorithm 1. The choice of these parameters using the IPLF is provided in Algorithm 2.

Algorithm 1 IEMB update for the u^{th} potential target

Input: The predicted means $\overline{x}^{1:n_{k|k-1}}_{k|k-1}$, covariance $P^{1:n_{k|k-1}}_{k|k-1}$ and probabilities $r^{1:n_{k|k-1}}_{k|k-1}$.
Output: The updated mean $\overline{x}^u_{k|k}$, covariance $P^u_{k|k}$ and probability $r^u_{k|k}$.
For $i = 1 : n_{k|k-1}$

- Calculate $\mathrm{E}_i\left[h(x)\right]$ using (69) with $\overline{x}^i_{k|k-1}$ and $P^i_{k|k-1}$.
- Calculate $\mathrm{E}_i\left[h(x)\left(h(x)\right)^{T}\right]$ using (71) with $\overline{x}^i_{k|k-1}$ and $P^i_{k|k-1}$.

End
For $u = 1 : n_{k|k-1}$

- Calculate $\hat{z}^u_{\text{corr}}$ and S^u_{corr} using (52) and (53).

End
For $u = 1 : n_{k|k-1}$

- Calculate the updated mean $\overline{x}^u_{k|k}$ and covariance $P^u_{k|k}$, and the values of A, b and Ω using Algorithm 2.
- Find $p^u\left(z_k\middle|\left|\widetilde{X}^u_k\right|=1\right)$ and $p^u\left(z_k|\emptyset\right)$ using (59) and (60).
- Calculate $r^u_{k|k}$ using (33).

End

Algorithm 2 IPLF update for the u^{th} potential target

Input: $\hat{z}^u_{\text{corr}}$, S^u_{corr}, and the predicted mean $\overline{x}^u_{k|k-1}$ and covariance $P^u_{k|k-1}$.
Output: Updated mean $\overline{x}^u_{k|k}$ and covariance $P^u_{k|k}$, and linearisation parameters (A, b, Ω).
Set $\bar{x} = \overline{x}^u_{k|k-1}$ and $\Sigma_x = P^u_{k|k-1}$.
Repeat:

- Select sigma-points $\widetilde{\mathcal{X}}_s$ matching mean $\bar{x}$ and covariance Σ_x.
- Calculate $\hat{h}$, $\mathrm{E}\left[(x-\overline{x})\left(h(x)-\hat{h}\right)^{T}\right]$ and $\mathrm{C}\left[h(x)\right]$ using (69) to (72).
- Find A, b and Ω using (62) to (64) using Σ_x for the covariance and $\bar{x}$ for the mean.
- Calculate $\overline{x}^u_{k|k}$ and $P^u_{k|k}$ using using (54) to (58) using predicted mean $\bar{x}$ and covariance Σ_x.

Until: Convergence [21] or after a given number of iterations.

V. Simulation results

In this section we test the performance of the proposed IEMB-IPLF filter[1] in two track-before-detect multi-target scenarios. In Section V-A we describe the other algorithms the IEMB-IPLF was compared to and the two scenarios considered. In Section V-B we describe the measurement and dynamic models, In Section V-C and V-D we describe and provide the results for each scenario respectively.

[1]Matlab code will be shared at https://github.com/ESDavies/IPLF-IEMB-filter.

A. Algorithms

In our implementation of the IEMB filter we used various linearisation methods to compute the required single-target updates. Specifically, two sigma-point methods, the IPLF and a UKF (as in [42]), and two analytical linearisation methods, the EKF and the iterated EKF (IEKF) [12]. We have compared these results with the independent multi-Bernoulli (IMB) filter [37] using an IPLF and a UKF for the single-target updates. The unscented transforms for both the UKF and IPLF have been implemented with $\omega^0 = \frac{1}{3}$ [16].

We tested the convergence of the IPLF by calculating the KLD between the current iteration and the iteration that immediately preceded it with a threshold of 0.1 [21]. Here, we set the maximum number of iterations to be 20. This is a high limit which could be reduced in order to improve the computational speed.

In addition, we have also implemented the generalised parallel partition multi-Bernoulli (GPP-MB) particle filter proposed in [32], which was shown to outperform other particle filters for track-before-detect. The GPP-MB particle filter has been implemented with 10^4 particles. The considered implementation does not make use of the Markov Chain Monte Carlo steps to improve the multi-Bernoulli approximation in [32].

We examine the performance of the filters in two different situations:

1) An uninformative birth scenario: In this scenario, at most one target may be born anywhere in the surveillance area.
2) An informative birth scenario: In this scenario, targets may be born at certain known locations with low positional uncertainty.

B. Measurement and dynamic model

The surveillance region used was a 120m $\times$ 120m area. This surveillance region is divided into 12 $\times$ 12 pixels with a sensor at each midpoint. Each sensor generates intensity measurements from a single target by using the measurement function [28]

$$h^j\left(x_k\right) = \frac{\phi}{\left(d_j\left(x_k\right)\right)^\beta + \epsilon} \tag{73}$$

where $d_j\left(x_k\right)$ is the distance from a target with state x_k to the j^{th} sensor and $\phi = 500$, $\epsilon = 25$ and $\beta = 2$. Each sensors' intensity measurement is found by summing the single-target measurements (73) from each target, see (5) [28]. The vector of measurements consists of 144 elements with each element corresponding to a multi-target measurement from each sensor. Each j^{th} measurement also has additional measurement noise η^j, see (2), which is a zero mean Gaussian noise covariance with covariance $R^j = 1$. This type of measurement function represents a received signal strength indicator [28].

Each target has a state vector consisting of a 2D position and velocity vector $[p_x, v_x, p_y, v_y]^T$. The matrices F and Q which characterise the dynamics of the targets (37) are given by the nearly constant velocity model [51] such that

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Q = \sigma_q^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} \tag{74}$$

where $\sigma_q = 0.5\,\text{m/s}^{3/2}$ and the length between each time step is $T = 1\,\text{s}$. The probability of each target's birth was set to be $p_b = 10^{-4}$ and the probability of survival to be $p_s = 0.99$. This means that it is considered to be unlikely for new targets to be born, and, once born, targets survive across many time steps.

C. Scenario 1: Uninformative Bernoulli birth model

In this section, we describe the target trajectories in Section V-C1, the birth model in Section V-C2 and the results in Section V-C3.

1) The Trajectory: To test the performance of the proposed filter we simulated the trajectories of 5 different targets which come into close proximity and then separate. This setting is more complex because of the interactions between the different targets in the received measurements.

Each target's trajectory was created using a similar approach to the one seen in [52]. The simulation ran for 81 time steps. At the midpoint of the simulation all targets were found to be at, approximately, the center of the surveillance region. From here, the trajectories are found by using forwards and backwards dynamics, as in [44]. The resulting set of trajectories is shown in Figure 2.

2) The birth model: The birth model is designed to allow for a maximum of one target to be born in each time step. The single-target density for new born targets $b_k^l(\cdot)$ (see (7)) is Gaussian with mean m_b and covariance matrix P_b. The

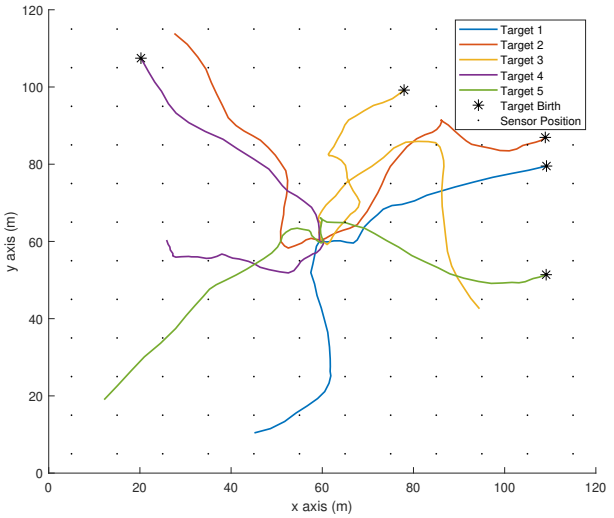


Figure 2: A figure showing the target trajectories. The target births for each target were $t^1_{\text{birth}} = 18$, $t^2_{\text{birth}} = 2$, $t^3_{\text{birth}} = 3$, $t^4_{\text{birth}} = 16$ and $t^5_{\text{birth}} = 9$. The target death for each target was $t^1_{\text{death}} = 64$, $t^2_{\text{death}} = 79$, $t^3_{\text{death}} = 78$, $t^4_{\text{death}} = 77$ and $t^5_{\text{death}} = 71$. All targets are roughly at the same location at around time step 40.

vector m_b and the matrix P_b contain prior information about the target's birth. The covariance matrix is

$$P_b = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & \sigma_y^2 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \tag{75}$$

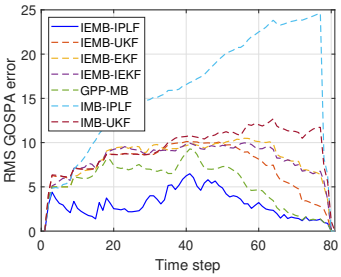
where σ_x^2 is the x-position variance and σ_y^2 is the y-position variance.

In this scenario, m_b is set to $[c_x, 0, c_y, 0]^T$, where $(c_x, c_y) = (60\,\text{m}, 60\,\text{m})$ are the coordinates for the centre of the surveillance region. The variances of the position along the x-axis and y-axis are set to $\sigma_x^2 = 1000\,\text{m}^2$ and $\sigma_y^2 = 1000\,\text{m}^2$. This implies that the birth model covers a large area.

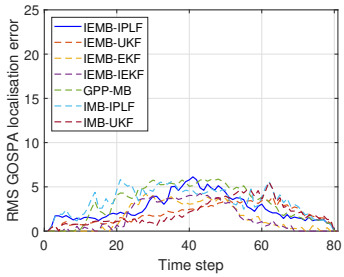
3) Results: The performance of the filter was assessed using the generalised optimal sub-pattern assignment (GOSPA) metric [53]. The GOSPA metric has the parameters: c as the cutoff distance, α as the cardinality factor and p as the exponent. The values assigned were as follows, $c = \frac{W}{2}$, where W is the cell width, in meters, $\alpha = 2$ and $p = 2$.

The root mean square GOSPA (RMS GOSPA) error per time step was calculated using 100 Monte Carlo runs. Also found, were the decomposition of the RMS GOSPA error into its three components: localisation error, missed target error and false target error. The RMS GOSPA per time step, for the uninformative birth scenario, is seen in Figure 3a. We omit showing the results of the IMB with the EKF and IEKF as these methods have a worse performance.

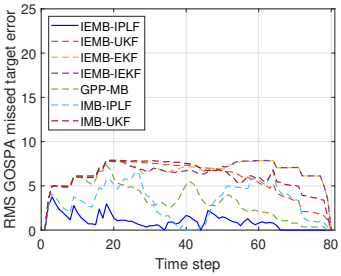
In Figure 3a the IEMB-IPLF is shown to have the best performance across all time steps with the two IMB methods having the worst performance. The GPP-MB is the second best performing filter, performing better than the IEMB-UKF but worse than the IEMB-IPLF. In Figure 3b we see that all the filters have a comparable localisation error, which, for the majority of filters, peaks when the targets are in close proximity, towards the midpoint of the simulation. However,



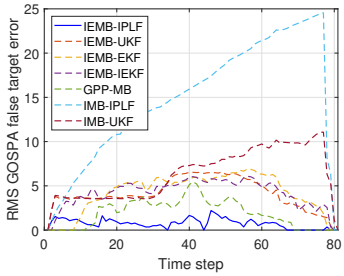
(a) RMS total GOSPA Error



(b) RMS localisation GOSPA Error



(c) RMS missed target GOSPA error



(d) RMS false target GOSPA error

Figure 3: RMS GOSPA errors and their decompositions across time for the different filters (Scenario 1). The IEMB-IPLF is the best performing filter.

Table I: Average RMS GOSPA errors (Scenario 1)

Filter	Error tot.	Error Loc.	Error Fal.	Error Mis.
IEMB-IPLF	3.39	3.00	0.909	1.27
IEMB-UKF	7.81	2.21	4.50	6.00
IEMB-IEKF	8.31	2.25	4.46	6.64
IEMB-EKF	8.47	2.18	4.74	6.68
GPP-MB	5.91	3.78	2.43	3.83
IMB-IPLF	17.0	3.71	16.2	3.77
IMB-UKF	9.64	2.46	6.93	6.24

the IMB methods have a reduced localisation error at the midpoint.

In Figure 3c we see the missed target GOSPA error where the IEMB-IPLF filter performs better than all the other filtering methods. The IMB-IPLF has no missed target error when the targets are in close proximity, towards the midpoint of the simulation. This is because the point of crossing is the mean of the birth location. The IMB-IPLF estimates many targets in this area, most of them false, as can be seen in Figure 3d. The GPP-MB has the third lowest missed target error. The missed target error increases with each additional target born into the surveillance region, reaching a peak at approximately time-step 20, before decreasing. There is a smaller peak around time-step 40 after which the missed target error continues to decrease.

In Figure 3d the IEMB-IPLF has the lowest false target error for the majority of time steps and the GPP-MB has the second lowest. The IMB-IPLF has the greatest false target error with false targets being generated at each time step.

The RMS GOSPA errors averaged across all time steps are given in Table I. It can be seen that the IEMB-IPLF has the lowest total RMS GOSPA error. It also has the lowest false target and missed target errors. While it does have a higher localisation error than other filters with only the IMB-IPLF and the GPP-MB having a higher localisation error, this, we can assume, to be a consequence of having more targets accurately estimated within the cut-off region.

The GPP-MB has the second lowest total RMS GOSPA error performing better than the IEMB-UKF. The IEMB-UKF has the third lowest total RMS GOSPA error, lower than the two IEMB filters which use the analytical approach to linearisation. This suggests that the sigma point approach performs better than the analytical linearisation approach.

The IMB-IPLF has the second lowest missed target error. This suggests that the IPLF method is the best at reducing the number of targets missed. However, the low number of missed targets could also be explained by the high number of false targets. Considering the number of targets being tracked, it is likely that some would fall within the cut-off region.

The GPP-MB's missed and false target GOSPA errors fall approximately midway between the IEMB filters which use a sigma point approach and the IEMB filters which use an analytical approach to linearisation.

As found in [42] the IMB approach results in high numbers of false targets. Apart from the IEMB-IPLF the IEMB filters show similar performances overall. The IEMB-IEKF shows little improvement on the IEMB-EKF only having a 0.28 reduction in the false target error, a 0.04 reduction in missed target error and a small increase in the localisation error. Again, this could possibly be because of reduced missed target error.

The higher GOSPA error for the IEMB-UKF, as opposed to the IEMB-IPLF, is mostly because of higher missed and false target errors. This is probably due to a lower accuracy, resulting in targets being predicted outside the cut-off region as well as more targets being missed entirely. Excluding the two IMB filters, the IEMB-EKF has the highest missed target and false target errors and performed worse than the other IEMB filters overall.

It can be seen in Figure 3 and in Table I that the IEMB-IPLF performs better than all the other methods for the uninformative birth scenario. The main advantage of the IEMB-IPLF is that it generates a lower number of missed and false targets.

Table II shows the average runtime of each filter for the uninformative scenario. Here, the fastest performing filter is the IEMB-EKF, followed by the IEMB-UKF and then the IMB-UKF. The three filters do no perform iterated SLRs resulting in them having faster runtimes than those which do. The IEMB-EKF which uses an analytical linearisation method is shown to be faster than the IEMB-UKF which uses the sigma-point linearisation method. This is because the sigma-

Table II: Average runtime (Scenario 1)

Filter	Mean Runtime (s)
IEMB-IPLF	1.9
IEMB-UKF	0.5
IEMB-IEKF	2.3
IEMB-EKF	0.3
GPP-MB	24.9
IMB-IPLF	24.5
IMB-UKF	0.6

Table III: Average RMS GOSPA errors (Scenario 2)

Filter	Error tot.	Error Loc.	Error Fal.	Error Mis.
IEMB-IPLF	3.14	2.98	0.695	0.697
IEMB-UKF	3.46	3.19	0.958	0.956
IEMB-IEKF	8.42	2.34	4.70	6.59
IEMB-EKF	8.14	2.31	4.15	6.61
GPP-MB	4.62	3.48	1.99	2.30
IMB-IPLF	32.1	3.07	31.8	2.93
IMB-UKF	35.1	3.48	34.9	1.91

Table IV: Average runtime (Scenario 2)

Filter	Mean Runtime (s)
IEMB-IPLF	3.7
IEMB-UKF	1.2
IEMB-IEKF	4.2
IEMB-EKF	0.6
GPP-MB	11.7
IMB-IPLF	92.4
IMB-UKF	11.6

point linearisation method iterates over the sigma-points when approximating the sensor measurements. The IEMB-IPLF is the fastest of all the filters which perform iterated SLRs this is likely because of the smaller number of false targets. The GPP-MB filter has the slowest runtime in Scenario 1.

D. Scenario 2: Informative birth model

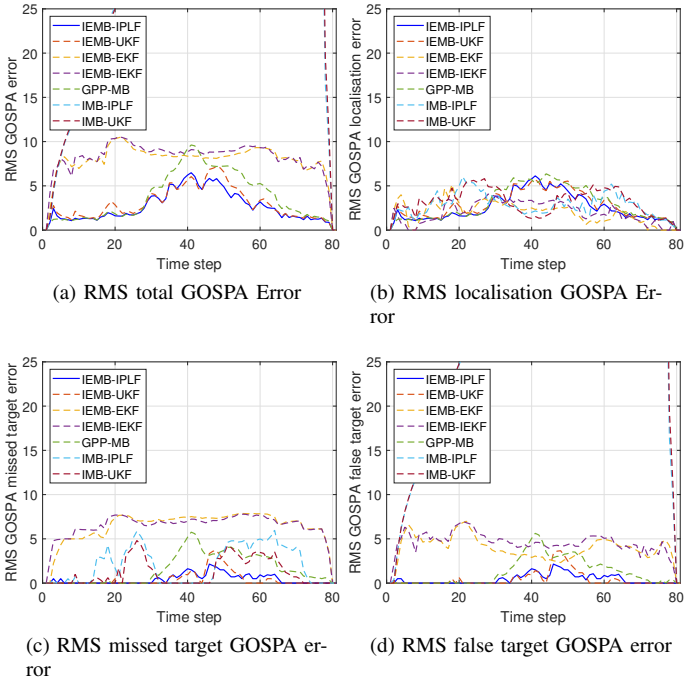
In this section, we describe the birth model in Section V-D1 and the results in Section V-D2.

1) The birth model: In this scenario, we have a multi-Bernoulli birth where each Bernoulli has a probability of existence $r^i = p_b$ at birth and Gaussian single-target density with means $m_b^1 = \begin{bmatrix} 75\,\mathrm{m} & 0 & 100\,\mathrm{m} & 0 \end{bmatrix}^T$, $m_b^2 = \begin{bmatrix} 20\,\mathrm{m} & 0 & 105\,\mathrm{m} & 0 \end{bmatrix}^T$, $m_b^3 = \begin{bmatrix} 110\,\mathrm{m} & 0 & 50\,\mathrm{m} & 0 \end{bmatrix}^T$ and $m_b^4 = \begin{bmatrix} 110\,\mathrm{m} & 0 & 85\,\mathrm{m} & 0 \end{bmatrix}^T$, and covariances P_b^i are in the form of (75) with $\sigma_x^2 = 10\,\mathrm{m}^2$ and $\sigma_y^2 = 10\,\mathrm{m}^2$.

The single-target densities contain prior knowledge of the approximate birth locations for the targets and the low covariance implies there is little uncertainty. In this scenario, we had a multi-Bernoulli birth which allows for a maximum of four targets to potentially be born in a single time step. These four Bernoulli densities represent the birth locations of the five targets, see Figure 2.

2) Results: The RMS GOSPA error per time step was calculated in the same way as it was in Section V-C3. The RMS GOSPA per time step and its decomposition into the localisation error, missed target error and false target error is seen in Figure 4. We omit showing the results of the IMB with the EKF and IEKF as these methods have a worse performance.

In Figure 4a the IEMB-IPLF and the IEMB-UKF are the best performing filters, with a comparable performance. This is because there is little uncertainty in our prior and therefore less of an improvement to be made through iterative SLRs [20]. The IMB filters have the worst performance. The GPP-MB filter is the third best performing filter with its performance closely following that of the IEMB-IPLF and the IEMB-UKF up until around time-step 25. After this time step, as the scenario increases in complexity, the RMS GOSPA error for



(a) RMS total GOSPA Error

(b) RMS localisation GOSPA Error

(c) RMS missed target GOSPA error

(d) RMS false target GOSPA error

Figure 4: RMS GOSPA errors and their decompositions across time for the different filters (Scenario 2). The IEMB-IPLF and IEMB-UKF are the best performing filters. The IMB errors are considerably higher due to false targets.

the GPP-MB filter becomes significantly larger than the RMS GOSPA error of the IEMB-UKF and the IEMB-IPLF.

In Figure 4b, all the examined filters had a similar localisation error. For the GPP-MB filter and for all the IEMB filters the localisation error increases as the targets come into close proximity, whereas, the IMB methods have a reduced localisation error when the targets are in close proximity. This is because many false targets, once generated, are not removed by the IMB filters. These false targets continue to be tracked across many time steps and tend to follow the approximate trajectories of the true targets. This results in a situation, when the targets are in close proximity, where the IMB filters estimate a high number of targets concentrated around the point of crossing. This is the cause of the IMB filters having a lower localisation error around the midpoint of the simulation.

In Figure 4c it can be seen that the analytical linearisation approach misses a fairly consistent number of targets over the course of the simulation. This results in the IEMB-EKF and the IEMB-IEKF having the highest missed target error while the IEMB-IPLF and IEMB-UKF have the lowest missed target error. The GPP-MB filter, which uses particle filtering, results in a greater number of missed targets than those using sigma points but fewer than those using an analytical linearisation approach.

The IEMB-UKF and IEMB-IPLF have the lowest false target errors in Figure 4d. The IEMB-IEKF and the IEMB-EKF have a fairly consistent false target error over the course of the simulation. The IEMB-IPLF and IEMB-UKF false target errors increase slightly at midpoint. The two IMB

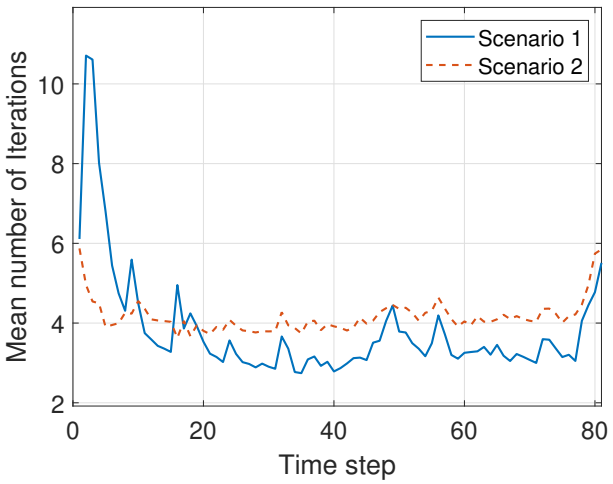


Figure 5: A figure showing, for the IEMB-IPLF, the average number of IPLF iterations per potential target at each time-step. The figure presents the results obtained from both the uninformative birth model (Scenario 1) and the informative birth model (Scenario 2).

methods have the worst false target error with increasing numbers of false targets being generated at each time step. The GPP-MB produces fewer false targets than the IEMB-IEKF and the IEMB-EKF which both use an analytical approach to linearisation. However, it produces a greater number of false targets than the IEMB-UKF and the IEMB-IPLF, which use sigma points.

The RMS GOSPA errors averaged across all time steps are given in Table III. The IEMB-IPLF has the lowest total RMS GOSPA error, the lowest false target error and the lowest missed target error. In the informative birth scenario, the difference in the error between the IEMB-IPLF and the IEMB-UKF is small. The GPP-MB filter resulted in a total GOSPA error which was, on average, lower than the IEMB-IEKF and the IEMB-EKF but not as low as the IEMB-UKF or the IEMB-IPLF. The GPP-MB filter GOSPA error fell approximately halfway between the IEMB filters which use sigma points and those which use an analytical approach to linearisation.

The IEMB-IEKF and the IEMB-EKF have a comparable performance. Both of these filters use analytical linearisation methods which results in increased numbers of false targets and missed targets. They also have a lower localisation error than the two IEMB filters which use sigma points. The higher false and missed target errors suggest that more targets are being predicted to be outside the cut-off region. The GPP-MB had significantly higher numbers of false and missed targets than both the IEMB-UKF and the IEMB-IPLF and approximately half the number produced by the IEMB-EKF and the IEMB-IEKF. The GPP-MB also had the greatest localisation error. However, this is only significant when compared to the IEMB filters.

The two IMB filters have high numbers of false targets. Additionally, they also have high localisation errors. The IMB filters have a lower missed target than the IEMB-IEKF and the IEMB-UKF.

In the informative birth scenario (Scenario 2) more targets could potentially be born in a single time step than in the uninformative birth scenario (Scenario 1). This is a consequence of the multi-Bernoulli birth model used in Scenario 2, see Section V-D. Because of this, both IMB filters show an increased false target error in the informative birth scenario.

Table IV shows the average runtime of each filter for the informative scenario. Here, the fastest performing filter is the IEMB-EKF, followed by the IEMB-UKF. Both the IEMB-EKF and IEMB-UKF do no perform iterated SLRs resulting in them having faster runtimes but higher GOSPA errors than the IEMB-IPLF and the IEMB-IEKF which do perform iterated SLRs. The IEMB-EKF which uses an analytical linearisation method is shown to be faster than the IEMB-UKF, which uses the sigma-point linearisation method. This is because the sigma-point linearisation method iterates over the sigma-points when approximating the sensor measurements. The IEMB-IPLF is the fastest of all the filters which perform iterated SLRs. This is likely because of the smaller number of false targets while the IMB-IPLF has the slowest runtime likely because of the combination of iterated SLRs and the high numbers of false targets produced by the IMB method in Scenario 2. The GPP-MB filter has a similar runtime to the IMB-UKF and they are both the slowest filters after the IMB-IPLF. The slow runtime of the IMB-UKF is likely because of high number of false targets whereas the GPP-MB's slow runtime is probably because of the computational demand of the particle linearisation method.

Finally, to conclude the analysis of experimental results, Figure 5 shows the average number of IPLF iterations in the IEMB-IPLF at each time-step, for Scenario 1 and Scenario 2. For the majority of time-steps, the average number of IPLF iterations is higher for Scenario 2. This is because Scenario 2 considers a higher number of potential new born targets that do not really exist, which increases the number of IPLF iterations.

VI. Conclusions

In this paper, we have proposed the information exchange multi-Bernoulli filter for track-before-detect of multiple targets using superpositional sensors. The filter update has been derived by minimising the KLD with the use of auxiliary variables. We have also proposed an IPLF implementation of the filter to be able to accommodate for a broad birth model with a non-linear measurement function.

In conclusion the IEMB-IPLF performed better than the other filters in both of the two scenarios we tested. The results show that the filters which used sigma points (IEMB-IPLF and IEMB-UKF) performed better than non sigma point methods (IEMB-EKF and IEMB-IEKF). It was also seen that the exchange of information significantly improved results, via a thorough numerical comparison with the IMB filters and a state-of-the-art track-before-detect particle filter.

Future work includes the implementation of the IEMB using particle filtering, its implementation using parallel computing and the application of the filter to real data. It is also relevant to carry out future research in exploiting the flexibility of being able to change the auxiliary variables to improve the multi-Bernoulli approximation, as is done in the set joint

probabilistic data association filter [54], the track-before-detect filter in [32] and in the variational Poisson multi-Bernoulli filters in [55]. Another line of future work is to extend the information exchange multi-Bernoulli filter to sets of trajectories to keep full target trajectory information and estimate the set of trajectories directly from the posterior [56].

APPENDIX A

In this appendix, we prove Lemma 2. We separate the proof into two parts. We first prove (29) and then we prove (31). Prior to presenting the proof, we review a preliminary result on set integrals on joint spaces.

A. Set integrals on joint spaces

Let us consider a multi-target function $\widetilde{f}(\cdot)$ on the single-target space with auxiliary variables $\mathbb{U}_k \times \mathbb{R}^{n_x}$. The single-target state space augmented with auxiliary variables can also be written as the disjoint union of the spaces for each auxiliary variables. That is, $\mathbb{U}_k \times \mathbb{R}^{n_x} = \uplus_{u=1}^{n_{k|k-1}} (\{u\} \times \mathbb{R}^{n_x})$.

For a given $\widetilde{X}_k \subset \mathbb{U}_k \times \mathbb{R}^{n_x}$, we can then write $\widetilde{X}_k = \widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}$ with $\widetilde{X}_k^u \subset \{u\} \times \mathbb{R}^{n_x}$. We can now apply the formula for the set integral on joint spaces [5, Eq. (3.53)] such that the set integral becomes

$$\int \widetilde{f}\left(\widetilde{X}_k\right) \delta\widetilde{X}_k = \int ... \int \widetilde{f}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right) \delta\widetilde{X}_k^1 ... \delta\widetilde{X}_k^{n_{k|k-1}} \quad (76)$$

which is a multiple set integral, with each of the set integrals over a single-target space $\{u\} \times \mathbb{R}^{n_x}$. That is, for a given multi-target function $\widetilde{g}(\cdot)$, the set integral with single-target space $\{u\} \times \mathbb{R}^{n_x}$ is

$$\int \widetilde{g}\left(\widetilde{X}_k^u\right) \delta\widetilde{X}_k^u = \sum_{n=0}^{\infty} \frac{1}{n!} \int \widetilde{g}\left(\{(u, x_1), ..., (u, x_n)\}\right) dx_{1:n}. \quad (77)$$

B. Proof of (29)

The KLD $\mathrm{D}\left(\widetilde{f}_{k|k} || \widetilde{q}\right)$ is given by (27), which is a set integral with a single-target space $\mathbb{U}_k \times \mathbb{R}^{n_x}$. We can now apply the formula for the set integral on joint spaces (76), such that (27) becomes

$$\mathrm{D}\left(\widetilde{f}_{k|k} || \widetilde{q}\right) = \int ... \int \widetilde{f}_{k|k}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right) \times \log \frac{\widetilde{f}_{k|k}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right)}{\widetilde{q}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right)} \delta\widetilde{X}_k^1 ... \delta\widetilde{X}_k^{n_{k|k-1}}. \quad (78)$$

Substituting (25) into (78), we obtain

$$\mathrm{D}\left(\widetilde{f}_{k|k} || \widetilde{q}\right) = \int ... \int \widetilde{f}_{k|k}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right) \times \log \frac{\widetilde{f}_{k|k}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right)}{\prod_{i=1}^{n_{k|k-1}} \widetilde{q}^i\left(\widetilde{X}_k^i\right)} \delta\widetilde{X}_k^1 ... \delta\widetilde{X}_k^{n_{k|k-1}}. \quad (79)$$

We seek to minimise $\mathrm{D}\left(\widetilde{f}_{k|k} || \widetilde{q}\right)$ with respect to $\widetilde{q}(\cdot)$. We use z to denote arbitrary constants that do not affect the minimisation. Then, we can write

$$\begin{aligned}
\mathrm{D}\left(\widetilde{f}_{k|k} || \widetilde{q}\right) =& \mathrm{z} - \int ... \int \widetilde{f}_{k|k}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right) \\
& \times \log \prod_{i=1}^{n_{k|k-1}} \widetilde{q}^i\left(\widetilde{X}_k^i\right) \delta\widetilde{X}_k^1 ... \delta\widetilde{X}_k^{n_{k|k-1}} \\
=& \mathrm{z} - \sum_{u=1}^{n_{k|k-1}} \int ... \int \widetilde{f}_{k|k}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right) \\
& \log \widetilde{q}^u\left(\widetilde{X}_k^u\right) \delta\widetilde{X}_k^1 ... \delta\widetilde{X}_k^{n_{k|k-1}} \\
=& \mathrm{z} - \sum_{u=1}^{n_{k|k-1}} \int \log \widetilde{q}^u\left(\widetilde{X}_k^i\right) \\
& \left[\int \widetilde{f}_{k|k}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right) \delta\widetilde{X}_k^{(-u)}\right] \delta\widetilde{X}_k^u.
\end{aligned}$$

where the set integral over $\widetilde{X}_k^{(-u)}$ is defined in Lemma 2. By standard KLD minimisation applied to each term in the above sum, we can see that the $\widetilde{q}^u(\cdot)$ that minimises $\mathrm{D}\left(\widetilde{f}_{k|k} || \widetilde{q}\right)$ are given by its marginal distribution

$$\widetilde{q}^u\left(\widetilde{X}_k^u\right) = \int \widetilde{f}_{k|k}\left(\widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}\right) \delta\widetilde{X}_k^{(-u)}. \quad (80)$$

This result proves (29).

C. Proof of (31)

We now substitute (23) into (80) to obtain

$$\begin{aligned}
\widetilde{q}^u\left(\widetilde{X}_k^u\right) \propto& \int \prod_{j=1}^{M} l\left(z_k^j | \sum_{(u,x)\in\widetilde{X}_k} h^j(x)\right) \\
& \prod_{i=1}^{n_{k|k-1}} \widetilde{f}_{k|k-1}^i\left(\widetilde{X}_k^i\right) \delta\widetilde{X}_k^{(-u)} \\
=& \widetilde{f}_{k|k-1}^u\left(\widetilde{X}_k^u\right) \int \prod_{j=1}^{M} l\left(z_k^j | \sum_{(u,x)\in\widetilde{X}_k} h^j(x)\right) \\
& \prod_{i=1:i\neq u}^{n_{k|k-1}} \widetilde{f}_{k|k-1}^i\left(\widetilde{X}_k^i\right) \delta\widetilde{X}_k^{(-u)}.
\end{aligned}$$

Substituting (32) into the above equation yields

$$\widetilde{q}^u\left(\widetilde{X}_k^u\right) \propto p^u\left(z_k | \widetilde{X}_k^u\right) \widetilde{f}_{k|k-1}^u\left(\widetilde{X}_k^u\right). \quad (81)$$

To finish the proof of (31), we just need to normalise the density such that it integrates to one, and substitute $\widetilde{X}_k = \widetilde{X}_k^1 \uplus ... \uplus \widetilde{X}_k^{n_{k|k-1}}$.

APPENDIX B

In this appendix, the proof for Proposition 3 is provided. Firstly, the result for the mean is proved considering $\widetilde{X}_k^u = \{(u, x^u)\}$.

The conditional mean is given by

$$\begin{aligned}\mathrm{E}\left[z_k^j|\widetilde{X}_k^u = \{(u,x^u)\}\right] &= \int \mathrm{E}\left[z_k^j|\left\{(u,x^u)\right\} \cup \widetilde{X}_k^{(-u)}\right] \\ &\quad \prod_{i=1:i\neq u}^{n_{k|k-1}} \widetilde{f}_{k|k-1}^i\left(\widetilde{X}_k^i\right) \delta\widetilde{X}_k^{(-u)}. \quad (82)\end{aligned}$$

Using (5), Equation (82) can be simplified to

$$\mathrm{E}\left[z_k^j|\widetilde{X}_k^u = \{(u,x^u)\}\right] \quad (83)$$

$$= h^j\left(x^u\right) + \sum_{i=1:i\neq u}^{n_{k|k-1}} r^i \mathrm{E}_i\left[h^j\left(x\right)\right], \quad (84)$$

which completes the proof of the conditional mean.

Now, the proof for the conditional covariance is shown. Using the law of total covariance, it follows that

$$\begin{aligned}&\mathrm{C}\left[z_k^j|\widetilde{X}_k^u = \{(u,x^u)\}\right] \\ &= \int \mathrm{C}\left[z_k^j|\left\{(u,x^u)\right\} \cup \widetilde{X}_k^{(-u)}\right] \\ &\prod_{i=1:i\neq u}^{n_{k|k-1}} \widetilde{f}_{k|k-1}^i\left(\widetilde{X}_k^i\right) \delta\widetilde{X}_k^{(-u)} \\ &+ \mathrm{C}\left[\mathrm{E}\left[z_k^j|\left\{(u,x^u)\right\} \cup \widetilde{X}_k^{(-u)}\right]|\widetilde{X}_k^u = \{(u,x^u)\}\right].\end{aligned}$$

Then, using (6), we obtain

$$\mathrm{C}\left[z_k^j|\left\{(u,x^u)\right\} \cup \widetilde{X}_k^{(-u)}\right] = R^j$$

Using (5), we obtain

$$\begin{aligned}&\mathrm{C}\left[\mathrm{E}\left[z_k^j|\left\{(u,x^u)\right\} \cup \widetilde{X}_k^{(-u)}\right]|\widetilde{X}_k^u = \{(u,x^u)\}\right] \\ &= \mathrm{C}\left[h^j\left(x^u\right) + \sum_{x\in\widetilde{X}_k\setminus\widetilde{X}_k^u} h^j(x)|\widetilde{X}_k^u = \{(u,x^u)\}\right] \\ &= \mathrm{C}\left[\sum_{x\in\widetilde{X}_k\setminus\widetilde{X}_k^u} h^j(x)\right] \\ &= \sum_{i=1:i\neq u}^{n_{k|k-1}} \widetilde{\mathrm{C}}_\mathrm{i}\left[h^j\left(x^i\right)\right].\end{aligned}$$

In the above expression $h^j\left(x^u\right)$ is a constant, given $\widetilde{X}_k^u = \{(u,x^u)\}$, so it can be removed when computing the conditional covariance. Then, as the distribution of each target in $\widetilde{X}_k$ is independent, the covariance of z_k^j can be obtained from the sum of the covariances of the potential targets.

The proof for the case $\widetilde{X}_k^u = \emptyset$ is analogous, with the only difference that the term $h^j\left(x^u\right)$ is removed from the conditional mean, and the rest of the results remain unchanged.

REFERENCES

[1] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, Jan. 2004.

[2] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer, "Multi-target tracking using a 3D-lidar sensor for autonomous vehicles," in *16th International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 881–886.

[3] E. Delande, J. Houssineau, J. Franco, C. Frueh, D. Clark, and M. Jah, "A new multi-target tracking algorithm for a large number of orbiting objects," *Advances in Space Research*, vol. 64, pp. 645–667, 2019.

[4] H. Kim, K. Granström, L. Gao, G. Battistelli, S. Kim, and H. Wymeersch, "5G mmWave Cooperative Positioning and Mapping Using Multi-Model PHD Filter and Map Fusion," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3782–3795, 2020.

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

[6] S. Challa, M. R. Morelande, D. Musicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge University Press, 2011.

[7] C. Kreucher, K. Kastella, and A. O. Hero III, "Multitarget tracking using the joint multitarget probability density," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.

[8] M. R. Morelande, C. M. Kreucher, and K. Kastella, "A Bayesian approach to multiple target detection and tracking," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1589–1604, May. 2007.

[9] S. Davey, M. Wieneke, and H. Vu, "Histogram-PMHT unfettered," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 435–447, June 2013.

[10] A.-A. Saucan, C. Sintes, T. Chonavel, and J.-M. Le Caillec, "Robust, track before detect particle filter for bathymetric sonar application," in *17th International Conference on Information Fusion*, 2014.

[11] W. Yi, Z. Fang, W. Li, R. Hoseinnezhad, and L. Kong, "Multi-frame track-before-detect algorithm for maneuvering target tracking," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4104–4118, 2020.

[12] S. Särkkä and L. Svensson, *Bayesian Filtering and Smoothing*, 2nd ed. Cambridge University Press, 2023.

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

[14] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[15] F. Daum, "Nonlinear filters: beyond the Kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, Aug. 2005.

[16] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[17] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.

[18] I. Arasaratnam, S. Haykin, and T. Hurd, "Cubature Kalman Filtering for Continuous-Discrete Systems: Theory and Simulations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, Oct. 2010.

[19] I. Arasaratnam, S. Haykin, and R. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, May 2007.

[20] M. R. Morelande and A. F. García-Fernández, "Analysis of Kalman filter approximations for nonlinear measurements," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5477–5484, Nov. 2013.

[21] Á. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior Linearization Filter: Principles and Implementation Using Sigma Points," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5561–5573, Oct. 2015.

[22] F. Tronarp, Á. F. García-Fernández, and S. Särkkä, "Iterative Filtering and Smoothing In Non-Linear and Non-Gaussian Systems Using Conditional Moments," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 408–412, Mar. 2018.

[23] P. M. Djuric, T. Lu, and M. F. Bugallo, "Multiple Particle Filtering," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. IEEE, Apr. 2007.

[24] W. Yi, M. R. Morelande, L. Kong, and J. Yang, "A computationally efficient particle filter for multitarget tracking using an independence approximation," *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 843–856, Feb. 2013.

[25] L. Úbeda-Medina, Á. F. García-Fernández, and J. Grajal, "Adaptive Auxiliary Particle Filter for Track-Before-Detect With Multiple Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2317–2330, Oct. 2017.

[26] R. Min, C. Garnier, F. Septier, and J. Klein, "State space partitioning based on constrained spectral clustering for block particle filtering," *Signal Processing*, vol. 201, p. 108727, 2022.

[27] L. Úbeda-Medina, Á. F. García-Fernández, and J. Grajal, "Sigma-point multiple particle filtering," *Signal Processing*, vol. 160, pp. 271–283, Jul. 2019.

[28] J. P. Beaudeau, M. F. Bugallo, and P. M. Djuric, "RSSI-Based Multi-Target Tracking by Cooperative Agents Using Fusion of Cross-Target Information," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5033–5044, Oct. 2015.

[29] P. Closas, C. Fernández-Prades, and J. Vilà-Valls, "Multiple Quadrature Kalman Filtering," *IEEE Transactions on Signal Processing*, vol. 12, pp. 6125–6137, Dec. 2012.

[30] J. Vila-Valls, P. Closas, and Á. F. García-Fernández, "Uncertainty Exchange Through Multiple Quadrature Kalman Filtering," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1825–1829, Dec. 2016.

[31] Á. F. García-Fernández, J. Grajal, and M. R. Morelande, "Two-layer particle filter for multiple target detection and tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1569–1588, Jul. 2013.

[32] Á. F. García-Fernández, "A track-before-detect labelled multi-Bernoulli particle filter with label switching," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2123–2138, Oct. 2016.

[33] S. Nannuru, M. Coates, and R. Mahler, "Computationally-tractable approximate PHD and CPHD filters for superpositional sensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 410–420, June 2013.

[34] A.-A. Saucan, T. Chonavel, C. Sintes, and J.-M. Le Caillec, "Track before detect DOA tracking of extended targets with marked Poisson point processes," in *18th International Conference on Information Fusion*, 2015, pp. 754–760.

[35] A. Masnadi-Shirazi and B. D. Rao, "A Covariance-Based Superpositional CPHD Filter for Multisource DOA Tracking," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 309–323, Jan. 2018.

[36] I. A. Bol'shakov and V. G. Latysh, "Separating an unknown number of fluctuating signals from noise on the basis of the theory of random points," *Radio Engineering and Electronic Physics*, vol. 3, pp. 326–334, 1964.

[37] B.-N. Vo, B.-T. Vo, N.-T. Pham, and D. Suter, "Joint Detection and Estimation of Multiple Objects From Image Observations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5129–5141, Oct. 2010.

[38] T. Kropfreiter, J. L. Williams, and F. Meyer, "A Scalable Track-Before-Detect Method With Poisson/Multi-Bernoulli Model," *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, Sep. 2021.

[39] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina, "A Tutorial on Bernoulli Filters: Theory, Implementation and Applications," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3406–3430, Jul. 2013.

[40] M. Liang, T. Kropfreiter, and F. Meyer, "A BP method for track-before-detect," *IEEE Signal Processing Letters*, vol. 30, pp. 1137–1141, 2023.

[41] Á. F. García-Fernández, L. Svensson, J. L. Williams, Y. Xia, and K. Granström, "Trajectory Poisson multi-Bernoulli filters," *in IEEE Transactions on Signal Processing, vol. 68, pp. 4933-4945*, Mar. 2020.

[42] E. S. Davies and Á. F. García-Fernández, "A multi-Bernoulli Gaussian filter for track-before-detect with superpositional sensors," in *2022 25th International Conference on Information Fusion (FUSION)*. IEEE, jul 2022.

[43] P. A. Bakut and N. A. Ivanchuk, "Calculation of the a posteriori characteristics of flow of resolved objects," *Engineering Cybernetics*, vol. 14, no. 6, pp. 148–156, 1976. [Online]. Available: www.stochasticflows.com

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

[45] Á. F. García-Fernández, Y. Xia, K. Granström, L. Svensson, and J. L. Williams, "Gaussian implementation of the multi-Bernoulli mixture filter," *Proceedings of the 22nd International Conference on Information Fusion, 2019*, Aug. 2019.

[46] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, Jun. 1999.

[47] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[48] H. Wymeersch, *Iterative receiver design*. Cambridge University Press, 2007.

[49] B. T. Vo and B. N. Vo, "Labeled Random Finite Sets and Multi-Object Conjugate Priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, Jul. 2013.

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

[51] Y. Bar-Shalom, T. Kirubarajan, and X. R. Li, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[52] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems, vol. 54, no. 4, pp. 1883-1901, Aug. 2018*, Mar. 2017.

[53] A. S. Rahmathullah, A. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *20th International Conference on Information Fusion*, 2017, pp. 1–8.

[54] L. Svensson, D. Svensson, M. Guerriero, and P. Willett, "Set JPDA Filter for Multitarget Tracking," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4677–4691, Oct. 2011.

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

[56] A. F. García-Fernández, L. Svensson, and M. R. Morelande, "Multiple target tracking based on sets of trajectories," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1685–1707, Jun. 2020.