Data-Driven Clustering and Bernoulli Merging for the Poisson Multi-Bernoulli Mixture Filter

MARCO FONTANA
University of Liverpool, Liverpool, U.K.

ÁNGEL F. GARCÍA-FERNÁNDEZ
University of Liverpool, Liverpool, U.K.
Universidad Antonio de Nebrija, Madrid, Spain

SIMON MASKELL, Member, IEEE
University of Liverpool, Liverpool, U.K.

This article proposes a clustering and merging approach for the Poisson multi-Bernoulli mixture (PMBM) filter to lower its computational complexity and make it suitable for multiple target tracking with a high number of targets. We define a measurement-driven clustering algorithm to reduce the data association problem into several subproblems, and we provide the derivation of the resulting clustered PMBM posterior density via Kullback–Leibler divergence minimization. Furthermore, we investigate different strategies to reduce the number of single target hypotheses by approximating the posterior via merging and intertrack swapping of Bernoulli components. We evaluate the performance of the proposed algorithm on simulated tracking scenarios with more than 1000 targets.

Manuscript received 26 May 2022; revised 13 November 2022; accepted 1 March 2023. Date of publication 7 March 2023; date of current version 11 October 2023.

DOI. No. 10.1109/TAES.2023.3253662

Refereeing of this contribution was handled by D. Clark.

This work was supported by Sintela Ltd.

Authors' addresses: Marco Fontana and Simon Maskell are with the Department of Electrical Engineering and Electronics, University of Liverpool, L69 3GJ Liverpool, U.K., E-mail: (marco.fontana@liverpool.ac.uk; s.maskell@liverpool.ac.uk). Ángel F. García-Fernández is with the Department of Electrical Engineering and Electronics, University of Liverpool, L69 3GJ Liverpool, U.K., and also with the ARIES Research Centre, Universidad Antonio de Nebrija, 28015 Madrid, Spain, E-mail: (angel.garcia-fernandez@liverpool.ac.uk). *(Corresponding author: Marco Fontana.)*

This article has supplementary downloadable material available at https://doi.org/10.1109/TAES.2023.3253662, provided by the authors.

I. INTRODUCTION

Multitarget tracking (MTT) is a well-known problem of interest in many application fields, including surveillance, traffic control, and autonomous driving [1], [2], [3]. The main goal is the estimation of the number of targets and their states based on the noisy measurements recorded by a sensor, which includes false alarms and missed detections. The targets move in a dynamic scenario, appearing and disappearing from the field of view of the sensor.

MTT has been studied for decades, and several solutions have been proposed to improve the tradeoff between performance and computational efficiency. Among the most widely used approaches, we mention multiple hypothesis tracking (MHT) [4], [5], [6], [7], [8], joint probabilistic data association (JPDA) [9], and random finite sets (RFS) [10].

In the last decade, several solutions to the MTT problem have been based on different birth models. With the Poisson point process (PPP) birth model and the standard measurement/dynamic models, the posterior is a Poisson multi-Bernoulli mixture (PMBM) [11], [12]. With multi-Bernoulli birth, the conjugate prior is a multi-Bernoulli mixture (MBM), which can be labeled and written in δ-generalized labeled multi-Bernoulli (δ-GLMB) form [12, Sec. IV] [13]. Approximate filters based on the PMBM and δ-GLMB filters are the Poisson multi-Bernoulli (PMB) filters [14], [15] and the labeled multi-Bernoulli (LMB) filter [16].

The PMBM filter can be considered a state-of-the-art fully Bayesian MHT filter, with an efficient representation of global hypotheses with probabilistic target existence (Bernoulli components) and information on undetected targets. The (track-oriented) PMB filter, which only propagates one component of the PMBM filter, can be seen as a fully Bayesian version of the integrated JPDA filter, with differences explained in [11, Sec. IV-A].

Of particular importance in real-world scenarios is to be able to track a large number of targets. The main challenge in large-scale MTT problems is the evaluation of all the possible measurement-target associations, which is known as data association problem. Several approaches were developed to efficiently manage the large number of hypotheses resulting from this combinatorial problem [17], [18], [19], [20], [21]. Among several methods, clustering and hypothesis merging are some of the most popular and most effective solutions. We briefly review the literature on these topics.

Clustering is considered one of the most effective strategies to increase the scalability of tracking algorithms. In a context of sufficiently sparse targets, clustering addresses large data association problems defining several independent subproblems. In the original MHT paper [4], the author describes a procedure to associate measurements with clusters of independent potential targets, merging the clusters associated to common measurements, and creating new individual clusters for the measurements not associated with any potential target in the prior. This method,

as the following ones derived from [4], can be referred to hypothesis clustering [22]. Similar procedures have been used in [23] to cluster targets based on the data association in multiple scans. In [24] the authors propose a spatial clustering of the tracks based on a minimum separation distance, addressing the assignment of new measurements to the most appropriate cluster using the gating procedure. In [25], a review of the split method presented in [5] is used to initialize new clusters based on the independent components in each cluster, detected by using rectangular areas in the measurement space. In [26], the authors describe an efficient cluster management approach based on a dynamic data structure, which implements the hypothesis tree. These methods do not have a straightforward application on multi-Bernoulli filters, as they are based on the concept of confirmed target, which is not defined for targets with probabilistic target existence.

For the δ-GLMB filter, a clustering algorithm for large-scale tracking based on predicted measurements gating regions is proposed in [27]. A drawback of this approach is that the δ-GLMB representation of a labeled MBM involves an exponential increase in the number of global hypothesis, which requires extra computational time [12]. In addition, this implementation neglects information on undetected targets, which is required in fully Bayesian MTT [28] and important in many applications, for example, autonomous vehicles and search-and-track [29].

Merging is another popular approach used to decrease the number of hypotheses in the filters and therefore computational time. One of the first contributions of this kind can be found in [30], where the authors suggest merging all tracks which share measurements for the past N times.

The first approaches to improve the computational efficiency of the PMBM filter were proposed in [12], where the authors suggested to cap the number of global hypotheses in the filter and pruning the Poisson and Bernoulli components whose weight is below a threshold. An alternative approach is to perform track-oriented N-scan pruning [31]. Two efficient PMB approximations of the PMBM posterior density were proposed in [32] for multiple extended object filtering, based on the original paper [11] and the variational approximation via Kullback–Leibler divergence (KLD) minimization.

In this work, we focus on developing novel clustering and merging algorithms to provide an efficient implementation of the PMBM filter. We proceed to explain the contributions. Our first contribution is a new data-driven clustering technique with low computational burden for the PMBM filter. To the best of our knowledge, this is the first attempt to cluster Bernoulli components in a PMBM filter implementation. We first define the concept of clustered PMBM density that is of general validity for any clustering algorithm. A clustered PMBM density is the union of an independent PPP and a number of independent MBMs, one for each cluster. We obtain the best fitting clustered PMBM by minimizing the KLD after introducing auxiliary variables over the track indexes in the target space [33], see diagram in Fig. 1. Then, the proposed clustering algorithm

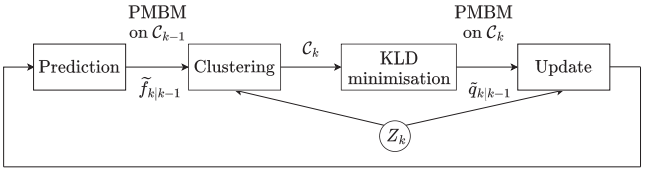

Fig. 1. Schematic of the clustered PMBM filter. The prediction step propagates a clustered PMBM density on the set of clusters $\mathcal{C}_{k-1}$ computed at the previous time instant. Once the filter receives the set of measurements Z_k for the current time instant k, we obtain the new set of clusters $\mathcal{C}_k$ and perform a KLD minimization (with auxiliary variables) to obtain the current clustered PMBM. The clustering algorithm can be defined by the user.

takes into account the predicted density and the received measurements to group potential targets that may have given rise to a common measurement, which is computationally advantageous compared to spatial clustering techniques.

Our second contribution is a Bernoulli merging approach for local hypotheses corresponding to the same potential target. We compute the similarity between Bernoulli components via the KLD and merge the most similar ones, as presented in [34]. The proposed method allow us to obtain an accurate representation of the filter posterior, merging similar local hypotheses with different data association history.

Our third contribution aims to improve the efficiency of the clustering algorithm for situations in which targets move in close proximity and then separate. In this setting, some Bernoulli components of different potential targets may overlap even though the actual locations of the potential targets are already well separated, which hinders clustering. To address this, we propose to swap certain Bernoulli components of different potential targets, keeping the PMBM representation unaltered, so that all the Bernoulli components of the same potential target are found in the same region, and clustering can be done efficiently. This approach bears resemblance to the particle swapping approach used in the particle filter for track-before-detect in [35] and also to the set JPDA algorithm [36] and variational PMB filters [15].

The rest of this article is organized as follows. Background on the PMBM filter is provided in Section II. In Section III, given the clusters of potential targets, we provide the clustered posterior density of the filter through KLD minimization. The measurement-driven clustering algorithm is presented in Section IV. Section V introduces two strategies to decrease the number of Bernoulli components via intratrack and intertrack Bernoulli merging. In Section VI, we evaluate filter performance via simulations. Finally, Section VII concludes this article.

II. BACKGROUND: PMBM FILTERING

In this section, we briefly review the standard dynamic model and the standard point target measurement model in Section II-A. Section II-B provides an overview of the PMBM filter; for a more extensive description see [11] and

[12]. Finally, we introduce auxiliary variables in the PMBM in Section II-C.

A. Multitarget System Modeling

In the context of multitarget systems, we regard filtering as the estimation of the states of a time-varying number of targets at the current time step k. We denote a single target state as $x_k \in \mathbb{R}^{n_x}$, and the set of target states at time k as $X_k \in \mathcal{F}(\mathbb{R}^{n_x})$, where $\mathcal{F}(\mathbb{R}^{n_x})$ is the set of all finite subsets of $\mathbb{R}^{n_x}$. The set X_k is modeled as an RFS, meaning that both the cardinality $|X_k|$ and the elements of the set (i.e., the target states) are random variables [10].

At each time step, a target with state x survives with probability $p_S(x)$, or departs with probability $1 - p_S(x)$ independently of the rest of the targets. The evolution of a surviving target can be defined by a Markov transition density $g(\cdot|x)$. At time step $k+1$, the multitarget state X_{k+1} is the union of the surviving targets and the independent new targets, which are modeled by a PPP with intensity $\lambda(\cdot)$.

The set of measurements at time k is denoted by $Z_k \in \mathcal{F}(\mathbb{R}^{n_z})$ and is the union of target-generated measurements and independent PPP clutter with intensity $\lambda_C(\cdot)$. At each time step, an existing target x_k is detected with probability $p_D(x_k)$, or misdetected with probability $1 - p_D(x_k)$. Each detected target $x_k \in X_k$ generates a measurement z_k with density $l(z_k|x_k)$.

B. PMBM Density

For the models described in Section II-A, the density $f_{k'|k}(\cdot)$ of the set of targets at time step $k' \in \{k, k+1\}$ given measurements up to time step k is a PMBM [11]. That is, it results from the union of two independent RFSs: a PPP with density $f^p_{k'|k}(\cdot)$, and a MBM RFS with density $f^{\text{mbm}}_{k'|k}(\cdot)$. The PMBM density is expressed as

$$f_{k'|k}(X_{k'|k}) = \sum_{Y \uplus W = X_{k'|k}} f^p_{k'|k}(Y) f^{\text{mbm}}_{k'|k}(W) \tag{1}$$

where the sum goes over all mutually disjoint sets Y and W, such that their union is $X_{k'|k}$.

The PPP density represents the targets that exist at the current time, but have not yet been detected. Its density is

$$f^p_{k'|k}(X) = e^{-\int \lambda_{k'|k}(x)dx} \prod_{x \in X} \lambda_{k'|k}(x) \tag{2}$$

where $\lambda_{k'|k}(\cdot)$ is the intensity. In the PPP, the cardinality is Poisson distributed and targets are independent, and identically distributed. The MBM part represents the potentially detected targets, and it can be described as [11]

$$f^{\text{mbm}}_{k'|k}(X) = \sum_{a \in \mathcal{A}_{k'|k}} w^a_{k'|k} \sum_{\uplus_{j=1}^{n^a_{k'|k}} X^j = X} \prod_{i=1}^{n_{k'|k}} f^{i,a^i}_{k'|k}(X^i) \tag{3}$$

where i is the index over the Bernoulli components, $a = (a^1, \ldots, a^{n_{k'|k}}) \in \mathcal{A}_{k'|k}$ represents a specific data association hypothesis, $a^i \in \{1, \ldots, h^i_{k'|k}\}$ is an index over the $h^i_{k'|k}$ single target hypotheses for the ith potential target, and $n_{k'|k}$ is the number of potentially detected targets. Each set of single target hypothesis $a \in \mathcal{A}_{k'|k}$ is also called a global hypothesis (whose mathematical expression is provided in [11]), and it is associated to a weight $w^a_{k'|k}$ satisfying $\sum_{a \in \mathcal{A}_{k'|k}} w^a_{k'|k} = 1$.

The Bernoulli density corresponding to the ith potential target, $i \in \{1, \ldots n_{k'|k}\}$, and the a_i single target hypothesis density $f^{i,a^i}_{k'|k}(X)$ can describe a newly detected target, a previously detected target or clutter. It efficiently models both the uncertainty regarding target existence and state. Mathematically, it can be expressed as

$$f^{i,a_i}_{k'|k}(X) = \begin{cases} 1 - r^{i,a^i}_{k'|k} & X = \emptyset \\ r^{i,a^i}_{k'|k} p^{i,a^i}_{k'|k}(x) & X = \{x\} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $r^{i,a^i}_{k'|k} \in [0, 1]$ is the probability of existence and $p^{i,a^i}_{k'|k}(\cdot)$ is the state density given that it exists. We often refer to a potential target as a track, which is defined as a collection of single target hypotheses corresponding to the same potential target [11].

The prediction and update steps of the PMBM filter to obtain (1) are given in [11] and [12].

C. PMBM With Auxiliary Variables

In order to perform clustering on the PMBM density (1) based on KLD minimization, which will be done in Section III, we require to introduce auxiliary variables in the density (1), as done in [11] and [33] for PMB filters. Auxiliary variables do not change the PMBM/MBM distribution, they just make implicit information in the posterior explicit. Similar approaches to introduce auxiliary/hidden variables in mixtures of densities can be found in the particle filtering and expectation maximization literature [37], [38].

Given (1), the target state space is augmented with an auxiliary variable $u \in \mathbb{U}_{k'|k} = \{0, 1, .., n_{k'|k}\}$, such that $(u, x) \in \mathbb{U}_{k'|k} \times \mathbb{R}^{n_x}$. Variable $u = 0$ implies that the target has not yet been detected, so it corresponds to the PPP, and $u = i$ indicates that the target corresponds to the ith Bernoulli component. We denote a set of target states with auxiliary variables as $\widetilde{X}_{k'} \in \mathcal{F}(\mathbb{U}_{k'|k} \times \mathbb{R}^{n_x})$.

DEFINITION 1 Given $f_{k'|k}(\cdot)$ of the form (1), the density $\widetilde{f}_{k'|k}(\cdot)$ on the space $\mathcal{F}(\mathbb{U}_{k'|k} \times \mathbb{R}^{n_x})$ of sets of target states with auxiliary variable is [33]

$$\begin{aligned} \widetilde{f}_{k'|k}\left(\widetilde{X}_{k'}\right) &= \sum_{\uplus_{l=1}^{n_{k'|k}} \widetilde{X}^l \uplus \widetilde{Y} = \widetilde{X}_{k'}} \widetilde{f}^p_{k'|k}\left(\widetilde{Y}\right) \sum_{a \in \mathcal{A}_{k'|k}} w^a_{k'|k} \prod_{i=1}^{n_{k'|k}} \left[\widetilde{f}^{i,a^i}_{k'|k}\left(\widetilde{X}^i\right)\right] \\ &= \widetilde{f}^p_{k'|k}\left(\widetilde{Y}_{k'}\right) \sum_{a \in \mathcal{A}_{k'|k}} w^a_{k'|k} \prod_{i=1}^{n_{k'|k}} \left[\widetilde{f}^{i,a^i}_{k'|k}\left(\widetilde{X}^i_{k'}\right)\right] \end{aligned} \tag{5}$$

where, for a given $\widetilde{X}_{k'}$, $\widetilde{Y}_{k'} = \{(u, x) \in \widetilde{X}_{k'} | u = 0\}$ and $\widetilde{X}^i_{k'} = \{(u, x) \in \widetilde{X}_{k'} | u = i\}$, and

$$\widetilde{f}^p_{k'|k}\left(\widetilde{X}_{k'}\right) = e^{-\int \lambda_{k'|k}(x)dx} \prod_{(u,x) \in \widetilde{X}_{k'}} \widetilde{\lambda}_{k'|k}(u, x) \tag{6}$$

$$\widetilde{\lambda}_{k'|k}(u,x) = \delta_0[u]\,\lambda_{k'|k}(x) \tag{7}$$

$$\widetilde{f}^{i,a^i}_{k'|k}\left(\widetilde{X}_{k'}\right) = \begin{cases} 1 - r^{i,a^i}_{k'|k} & \widetilde{X} = \emptyset \\ r^{i,a^i}_{k'|k} p^{i,a^i}_{k'|k}(x)\delta_i[u] & \widetilde{X} = \{(u,x)\} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where the Kronecker delta $\delta_i[u] = 1$ if $u = i$ and $\delta_i[u] = 0$, otherwise. The introduction of the auxiliary variables allows us to remove the sum over the sets in (5), as there is only one term in the sum that provides a nonzero density.

III. CLUSTERED PMBM APPROXIMATION VIA KLD MINIMIZATION

In this section, we are given clusters of potential targets, and our aim is to approximate a PMBM density as a clustered PMBM density based on KLD minimization with auxiliary variables. The clustered PMBM density ensures that potential targets belonging to different clusters are independent, and corresponds to the union of an independent PPP and independent MBMs, one for each cluster. The results in this section hold for any clustering algorithm and show how to obtain the parameters of the clustered PMBM posterior once the clusters are defined. The specific data-driven clustering algorithm we propose is explained in Section IV.

A. Clustered Density

Suppose we perform clustering at each update and denote the set of clusters as $\mathcal{C}_k = \{C^1_k, \ldots, C^{n^c_k}_k\}$, where each element C^i_k is the set of auxiliary variables corresponding to the tracks assigned to the cluster $c \in \{1, \ldots, n^c_k\}$. The set $\mathcal{C}_k$ is a partition of the auxiliary variable space without 0, $\mathbb{U}_{k'|k} \setminus \{0\}$, that meets the following properties [10].

1) Each cluster C^c_k is a subset of auxiliary variables $C^c_k \subset \mathbb{U}_{k'|k} \setminus \{0\}$.
2) The union of the clusters is the auxiliary variable space without 0; i.e., $\cup_{c=1}^{n^c_k} C^c_k = \mathbb{U}_{k'|k} \setminus \{0\}$.
3) The intersection of any two distinct clusters with indexes c_1 and c_2, $c_1 \neq c_2$, is empty; i.e., $C^{c_1}_k \cap C^{c_2}_k = \emptyset$.

Given $\mathcal{C}_k$, we can approximate the density of the set of targets with independent clusters (including auxiliary variables) as

$$\widetilde{q}_{k'|k}\left(\widetilde{X}_{k'}\right) = \widetilde{q}^0_{k'|k}\left(\widetilde{Y}_{k'}\right) \prod_{c=1}^{n^c_k} \widetilde{q}^c_{k'|k}\left(\cup_{i\in C^c_k} \widetilde{X}^i_{k'}\right) \tag{9}$$

where $\widetilde{q}^0_{k'|k}(\cdot)$ represents the density on the set of undetected targets $\widetilde{Y}_{k'}$, and the density on the detected target set is expressed as the multiplication of the cluster densities $\widetilde{q}^c_{k'|k}(\cdot)$. The form (9) implies that targets belonging to different clusters are independent. If there is exactly one potential target in each cluster, all potential targets are independent, and (9) becomes PMB, with auxiliary variables.

B. Clustered PMBM With Auxiliary Variables

We calculate the clustered PMBM density (9) by minimizing the KLD between $\widetilde{f}_{k'|k}(\cdot)$ in (5) and $\widetilde{q}_{k'|k}(\cdot)$ in (9). The KLD is defined as the set integral [10]

$$\mathrm{D}\left(\widetilde{f}_{k'|k} \,\|\, \widetilde{q}_{k'|k}\right) = \int \widetilde{f}\left(\widetilde{X}_{k'|k}\right) \log \frac{\widetilde{f}\left(\widetilde{X}_{k'|k}\right)}{\widetilde{q}\left(\widetilde{X}_{k'|k}\right)} \delta\widetilde{X}. \tag{10}$$

LEMMA 2 Let $\widetilde{f}_{k'|k}(\cdot)$ be the PMBM density with auxiliary variables in (5). The densities $\widetilde{q}^0_{k'|k}(\cdot)$, $\widetilde{q}^1_{k'|k}(\cdot)$,..., $\widetilde{q}^{c_{k'|k}}_{k'|k}(\cdot)$ in (9) that minimize the KLD $\mathrm{D}(\widetilde{f}_{k'|k} \| \widetilde{q}_{k'|k})$ are

$$\widetilde{q}^0_{k'|k}\left(\widetilde{Y}_{k'}\right) = \widetilde{f}^p_{k'|k}\left(\widetilde{Y}_{k'}\right) \tag{11}$$

$$\widetilde{q}^c_{k'|k}\left(\cup_{i\in C^c_k} \widetilde{X}^i_{k'}\right) \propto \sum_{a\in\mathcal{A}_{k'|k}} w^a_{k'|k} \prod_{i\in C^c_k} \left[\widetilde{f}^{i,a^i}_{k'|k}\left(\widetilde{X}^i_{k'}\right)\right] \tag{12}$$

where $\widetilde{Y}_{k'}$ and $\widetilde{X}_{k'}$ are given in Definition 1, and $\widetilde{q}^c_{k'|k}(\cdot)$ is the cluster density of the cluster c.

Lemma 2 builds on (1), which was derived in [11], and computes the clustered PMBM given the clusters. See Appendix A (Supplementary material) for the proof of Lemma 2. We can see that the density of each cluster is a multi-Bernoulli mixture for the set of targets in the cluster, and the density for the undetected targets is a PPP. Therefore, (9) with (11) and (12) define a clustered PMBM, with auxiliary variables.

Since index i only goes through the potential targets in the cluster in (12), there can be repeated terms in the sum, which can be merged into one. To do so, we can define a cluster alphabet $\mathcal{A}^c_{k'|k}$ by only considering the entries of the $\mathcal{A}_{k'|k}$ that correspond to this cluster, adding a level of indirection between the cluster density and the Bernoulli components that constitute them. Then, we can define a weight for the a_c cluster hypothesis that is the sum over all the weights $w^{a_c}_{k'|k}$ with the same local hypotheses for the potential targets in the cluster. Thus, we can rewrite the cluster density (12) as

$$\widetilde{q}^c_{k'|k}\left(\cup_{i\in C^c_k} \widetilde{X}^i_{k'}\right) = \sum_{a_c\in\mathcal{A}^c_{k'|k}} w^{a_c}_{k'|k} \prod_{i\in C^c_k} \left[\widetilde{f}^{i,a^i_c}_{k'|k}\left(\widetilde{X}^i_{k'}\right)\right]. \tag{13}$$

C. Clustered PMBM

The clustered density without auxiliary variables can be obtained by integrating out the auxiliary variables in (9).

LEMMA 3 Let $\widetilde{q}_{k'|k}(\cdot)$ be the clustered density with auxiliary variables in (9) defined in $\mathcal{F}(\mathbb{U}_{k'|k} \times \mathbb{R}^{n_x})$. The corresponding density $q_{k'|k}(\cdot)$ in $\mathcal{F}(\mathbb{R}^{n_x})$ is derived by integrating out the auxiliary variables, obtaining

$$\sum_{u_{1:n}\in\mathbb{U}^n_k} \widetilde{q}_{k'|k}\left(\{(u_1,x_1),\ldots,(u_n,x_n)\}\right) = q_{k'|k}\left(\{x_1,\ldots,x_n\}\right) \tag{14}$$

where

$$q_{k'|k}\left(X_{k'}\right) = \sum_{Y^0\uplus X^1\uplus\ldots\uplus X^{c_{k'|k}} = X_{k'}} q^0_{k'|k}\left(Y^0\right) \prod_{c=1}^{n^c_k} q^c_{k'|k}\left(X^c\right) \tag{15}$$

and

$$q^{c}_{k'|k}(\{x_1,\ldots,x_n\}) = \sum_{u_{1:n}\in\mathbb{U}^n_k} \widetilde{q}^{c}_{k'|k}(\{(u_1,x_1),\ldots,(u_n,x_n)\}). \quad (16)$$

The proof of Lemma 3 in reported in Appendix B (Supplementary material). If $q^0_{k'|k}(\cdot)$ and $q^c_{k'|k}(\cdot)$ are obtained via the KLD minimization on a PMBM in (11)–(12), the density $q_{k'|k}(\cdot)$ is the union of $c_{k'|k}+1$ independent RFS [10] (as its density is obtained through the convolution formula). One RFS represents undetected targets, and each of the rest of them corresponds to the RFS in a cluster, whose density is an MBM. Therefore, a clustered PMBM is the union of an independent PPP and $c_{k'|k}$ independent MBMs.

It can be shown that the KLD between the PMBM (5) and clustered PMBM (9) with auxiliary variables is an upper bound of the KLD distance between the PMBM and clustered PMBM densities without auxiliary variables [33]

$$\mathrm{D}\left(f_{k'|k}\,\|\,q_{k'|k}\right) \le \mathrm{D}\left(\widetilde{f}_{k'|k}\,\|\,\widetilde{q}_{k'|k}\right) \quad (17)$$

where $q_{k'|k}$ denotes the clustered PMBM density in the form of (15) without auxiliary variables. Therefore, Lemma 2 minimizes an upper bound of the KLD between $f_{k'|k}$ and $q_{k'|k}$, which is the one of primary interest.

D. Recursive Clustered PMBM Approximation

So far, we have explained how to obtain a clustered PMBM density from a PMBM density. In order to apply these results to the filtering recursion, in this section, we explain how to obtain a clustered PMBM from a previously clustered PMBM, in which the clusters may differ.

After the prediction, we obtain a clustered PMBM density $\widetilde{f}_{k|k-1}$ of the form (9), where $\widetilde{q}^0_{k|k-1}$ and $\widetilde{q}^c_{k|k-1}$ are defined, respectively, in (11) and (13) on the set of clusters $\mathcal{C}_{k-1}$. At time k we use a new set of clusters $\mathcal{C}_k$ (e.g., following the procedure that will be described in Section IV-B). We compute a new clustered PMBM density $\widetilde{q}^{c'}_{k|k-1}$ based on the new set of clusters $\mathcal{C}_k$ via KLD minimization, where c' is the cluster index in the set $\mathcal{C}_k$.

LEMMA 4 Let us assume the predicted density with auxiliary variables $\widetilde{f}_{k|k-1}(\cdot)$ is a clustered PMBM density with clusters $C^1_{k-1},\ldots,C^{n^c_{k-1}}_{k-1}$ such that

$$\widetilde{f}_{k|k-1}\left(\widetilde{X}_k\right) = \widetilde{f}^0_{k|k-1}\left(\widetilde{Y}_k\right)\prod_{c=1}^{n^c_{k-1}}\widetilde{f}^c_{k|k-1}\left(\cup_{i\in C^c_{k-1}}\widetilde{X}^i_k\right) \quad (18)$$

where

$$\widetilde{f}^c_{k|k-1}\left(\cup_{i\in C^c_{k-1}}\widetilde{X}^i_k\right) = \sum_{a_c\in\mathcal{A}^c_{k|k-1}} w^{a_c}_{k|k-1}\prod_{i\in C^c_{k-1}}\widetilde{f}^{i,a^i}_{k|k-1}\left(\widetilde{X}^i_k\right). \quad (19)$$

If the clusters at time step k are $C^1_k,\ldots,C^{n^c_k}_k$, the predicted clustered density $\widetilde{q}_{k|k-1}(\widetilde{X}_k)$ of the form (9) that minimizes $D(\widetilde{f}_{k|k-1}||\widetilde{q}_{k|k-1})$ is a clustered PMBM characterized by

$$\widetilde{q}^{c'}_{k|k-1}\left(\cup_{i\in C^{c'}_k}\widetilde{X}_k\right) \propto \prod_{c=1:C^{c'}_k\cap C^c_{k-1}\neq\emptyset}^{n^c_{k-1}}\sum_{a_c\in\mathcal{A}^c_{k|k-1}} w^{a_c}_{k|k-1} \times \prod_{i\in C^{c'}_k\cap C^c_{k-1}}\widetilde{f}^{i,a^i}_{k|k-1}\left(\widetilde{X}^i_k\right) \quad (20)$$

and $\widetilde{q}^0_{k|k-1}$ as defined in (11).

See Appendix C (Supplementary material) for the proof of Lemma 4. In (20), potential targets that belong to the same cluster at time step $k-1$ and k retain their statistical dependencies (modeled by an MBM). The PMBM update is performed independently for each cluster c' on the basis of the predicted cluster density (20), as described in [11] and [12].

IV. MEASUREMENT-DRIVEN CLUSTERING

In this section, we describe a novel procedure to efficiently cluster the potential targets on the basis of the current set of measurements Z_k. At each time step k, tracks and measurements are linked through the gating procedure based on efficient data structures, described in Section IV-A. The cluster formation is performed by the algorithm described in Section IV-B. Finally, an efficient method to prune the global hypotheses in the new clustered PMBM density is presented in Section IV-C.

A. Gating via Efficient Data Structures

Gating can significantly reduce the complexity of the data association problem by avoiding computing low-weight hypotheses [39]. In the PMBM update, each measurement can be associated to a previous Bernoulli or to the PPP. For each previous Bernoulli, we calculate its predicted measurement $\widehat{z}$ and its covariance matrix S [12]. In ellipsoidal gating, we can evaluate if a received measurement z_j is likely to be produced by the hypothesis a^i of the potential target i, (i,a^i), by computing its Mahalanobis distance with the covariance matrix S, to each $\widehat{z}$. For each hypothesis (i,a^i), $z_j\in Z_k$ belongs to $\mathcal{G}_k(i,a^i)$ if the Mahalanobis distance between z_j and the predicted measurement of (i,a^i) is below the threshold γ_G.

Denoting as N^h_k the sum of the number of Bernoulli components and the number of Gaussian components in the PPP intensity [12] in the filter at time instant k, the evaluation of all these possible pairs has a complexity $\mathcal{O}(|Z_k|\cdot N^h_k)$. It is possible to lower the complexity of this process by using efficient data structures; we proceed to discuss how k-d trees [40] and R-trees [41] can be used in this context.

1) *k-D Tree:* The k-d tree is a binary space-partitioning tree, which recursively divides the k-dimensional space to organize the entries and perform fast range searches. At each time step k, the computational cost of building a k-d tree based on the set of measurements Z_k is $\mathcal{O}(|Z_k|\log|Z_k|)$. For each hypothesis (i,a^i), we define the mean variance

across dimensions in the innovation covariance $(\sigma_k^{i,a^i})^2 = \text{tr}(S_k^{i,a^i})/n_z$. Then, we perform N_k^h range queries on the expected target measurements $\widehat{z}_k^{i,a^i}$ for a range defined by $\gamma_G \sigma_k^{i,a^i}$. Thus, the gating procedure queries the k-d tree in a computational time $\mathcal{O}(N_k^h(|Z_k|^{1-1/n_z} + s))$ [42] at each time instant, where n_z is the number of dimensions of the search space, and s is the average number of measurements returned by each query. Note that in our setting $k = n_z$ as the k-d tree operates on the single measurement space.

2) *R-Tree:* The R-tree is a hierarchical data structure in which every entry is represented by a minimum bounding d-dimensional rectangle (MBR). The internal nodes of the tree organise the leaf nodes into larger MBR, allowing efficient retrievement of the entries that intersect a window (or a point) in the d-dimensional space [43].

In the R-tree implementation, the tree is built on the N_k^h predicted single target states, with an overall computational time $\mathcal{O}(N_k^h \log N_k^h)$. The predicted measurement from the Bernoulli f^{i,a^i}, and its covariance matrix are represented by an n_z-dimensional rectangle $\mathcal{R}^{i,a^i}$ with centre in $\widehat{z}_k^{i,a^i}$ and dimensions proportional to the standard deviation σ_k^{d,i,a^i} of the innovation covariance S_k^{i,a^i} for each axis. That is, the gating area is defined by

$$\mathcal{R}^{i,a^i} = \left\{ z : \left| z_k^d - \widehat{z}_k^{d,i,a^i} \right| \leq \gamma_G \sigma_k^{d,i,a^i} \forall d \right\} \tag{21}$$

where $d \in \{1, \ldots, n_z\}$ indicates the dimensions, $z_k = [z_k^1, \ldots, z_{z_k}^n]^T$, and γ_G is the gating threshold. We define the set of measurements $\mathcal{G}_k(i, a^i)$ selected to update the hypothesis $a^i \in \{1, \ldots, h_{k'|k}\}$ as the subset of the measurements Z_k which belong into the rectangle $\mathcal{R}^{i,a^i}$. The gating procedure can be implemented by inserting the hyperrectangles $\mathcal{R}^{i,a^i}$ in the R-tree, and it provides an efficient approximation of the ellipsoidal gating [5]. The entire gating procedure is performed with $|Z_k|$ queries in a computational time $\mathcal{O}(|Z_k|((\log N_k^h)^{n_z-1} + s))$, where s is the number of elements returned at each query.

The capability of the R-tree to efficiently store hyperrectangles allows us to perform fewer queries than with the k-d tree, exploiting the efficiency provided by the logarithmic query time on a greater number of elements in the tree [24]. On the other side, the efficiency of the R-tree is reduced if the hyperrectangles in the structure show a high degree of overlap, as more edges need to be inspected to complete a query [44].

B. Clustering

Potential targets that have local hypotheses with common measurements at the current or past time steps are not independent and, in principle, should belong to the same cluster. Nevertheless, the dependencies in the distributions of the potential targets tend to weaken if there are no common measurements in recent time steps. In this work, we propose a clustering algorithm that only accounts for the data associations at the current time step. The algorithm does not explicitly maintain cluster information from scan to scan, as it defines new clusters at each time step. It only retrieves information from the previous time step to cluster misdetected tracks. Its main benefit is the computational efficiency and ease of implementation, though it discards possible target dependencies lingering from past time steps.

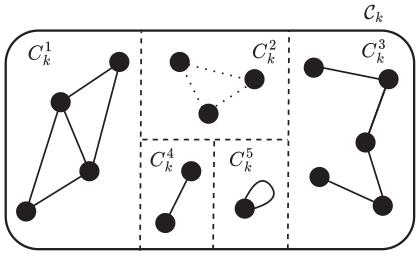

Fig. 2. Example of disjoint union of graphs. The 15 nodes represent the potential targets arranged in five clusters, on the basis of the common measurements represented by the edges. Adjacent nodes depict potential targets associated to the same measurement, while loops represent measurements related to a single potential target. Cluster C_k^2 contains three misdetected targets, which belonged to the same cluster at the previous time instant. The dashed lines in C_k^2 represent the dummy measurement assigned by the clustering algorithm (Algorithm 1).

Suppose $\mathcal{C}_k = \{C_k^1, \ldots, C_k^{n_k^c}\}$ is a partition of the auxiliary variable set $\mathbb{U}_{k'|k} \backslash \{0\} = \{1, .., n_{k'|k}\}$. For each cluster C_k^c we can determine the set of associated measurements $\mathcal{S}_k^c$ as the union of the sets of the gated measurements for the targets in the cluster

$$\mathcal{S}_k^c = \bigcup_{i \in C_k^c} \mathcal{G}_k^i \tag{22}$$

where $\mathcal{G}_k^i = \{\mathcal{G}^k(i, a^i) | a^i \in \{1, \ldots, h_{k'|k}^i\}\}$ is the set of measurements related to the track i.

The relation between targets and measurements can be represented by a graph, where the nodes denote the targets, and the edges connect targets that have at least one measurement in their gates in common. The partitioning of the tracks into clusters is defined by the connected components of the graph, which can be considered as a disjoint union of graphs (see Fig. 2). The connected components of the graph can be determined by an algorithm for traversing or searching graph data structures, as depth-first search or breadth-first search [45]. This approach obtains sets of clusters such that the intersection of any two distinct measurements sets is empty; i.e., $\mathcal{S}_{k'|k}^{c_1} \cap \mathcal{S}_{k'|k}^{c_2} = \emptyset, c_1 \neq c_2, \{c_1, c_2\} \subset \{1, \ldots n_k^c\}$.

The isolated nodes of the graph, i.e., those that are not an endpoint of any edge, represent the misdetected targets at the current time step k, as they are not associated with any measurement. The algorithm clusters the misdetected targets according to their cluster membership at the previous time instant $k-1$. A dummy measurement $*_j$, $j \in \{1, \ldots n_{k-1}^c\}$, is generated for each cluster at time $k-1$, and the targets misdetected at time k are associated with the corresponding measurement $*_j$, where j represent the cluster membership at $k-1$. Fig. 2 shows an example of this kind for cluster C_k^2, where it can be noted that the misdetected targets are clustered due to the links provided by the dummy measurement, represented with dashed lines.

Algorithm 1: Measurement-Driven Clustering.

Input: Pairs target-measurements $A = \{(i, \mathcal{G}_k^i) | i \in \mathbb{U}_{k'|k}\}$. Set of clusters at the previous time step $\mathcal{C}_{k-1} = \{C^1, \ldots, C^{n_{k-1}^c}\}$

Output: Set of clusters at the current time step $\mathcal{C}_k$

1: Obtain $U_0 = \{i : i \in \mathbb{U}_{k|k} \setminus \{0\}, \mathcal{G}_k^i = \emptyset\}$: the set of indexes of misdetected tracks.
2: Retrieve the indexes of misdetected tracks at time step k in each cluster defined at the previous time step $\{C_0^1, \ldots, C_0^{n_{k-1}^c}\}$ such that $C_0^i \subseteq C^i$, $\cup_{i=1}^{n_{k-1}^c} C_0^i = U_0$.
3: **for** $j \in \{1 \ldots, n_{k-1}^c\}$ **do**
4: **for** $i \in C_0^j$ **do**
5: $\mathcal{G}_k^i \leftarrow \mathcal{G}_k^i \cup \{*_j\}$ ▷ Assign a dummy measurement.
6: **end for**
7: **end for**
8: Generate the graph G on A.
9: Assign the track indexes of the connected components of G to a cluster to obtain $\mathcal{C}_k = \{C^1, \ldots, C^{n_k^c}\}$.
10: **return** Set of clusters $\mathcal{C}_k$.

The pseudocode of the clustering algorithm is provided in Algorithm 1.

C. Efficient Pruning for the New Clustered PMBM

After we have obtained the clusters via Algorithm 1, we can use Lemma 2 or 4 to obtain the clustered PMBM, and perform the update for each cluster independently. Due to the product over the clusters in (20), if previously independent clusters are merged, the resulting MBM can contain a high number of multi-Bernoulli components. We propose an efficient method to prune the least likely components by computing only the K best merged global hypotheses in each cluster $C_k^{c'} \in \mathcal{C}_k$, where K is adaptively determined by the normalise merged weights.

Suppose c' is the index of the new cluster and assume we merge the previous clusters with indexes in the set $\mathcal{M} = \{c | C_k^{c'} \cap C_{k-1}^c \neq \emptyset\}$. We define the binary decision variables $v_{c,j} \in \{0, 1\}$, where $v_{c,j} = 1$ if the global hypothesis of index $j \in \{1, \ldots, |\mathcal{A}_{k|k-1}^c|\}$ in the cluster c contributes to the solution. According to the same logic, we denote the weight selected by the variable $v_{c,j}$ as $w_{c,j}$. We can describe the problem of finding the best merged global hypothesis as an optimization problem

$$\text{maximize} \sum_{c \in \mathcal{M}} \sum_{j=1}^{\left|\mathcal{A}_{k|k-1}^c\right|} v_{c,j} \log w_{c,j} \tag{23}$$

$$\text{subject to} \sum_{j=1}^{\left|\mathcal{A}_{k|k-1}^c\right|} v_{c,j} = 1, \quad c \in \mathcal{M}. \tag{24}$$

The solution to (23) corresponds to taking the maximum weight $w_{c,j}$ over j for each c. Here, we take the K best global hypothesis until the normalised weight of the K-th hypothesis is below a pruning threshold Γ_{mbm}.

Assuming the set of global hypotheses is sorted by descending weight in each cluster $c \in \mathcal{M}$, the problem can be solved in $\mathcal{O}(|\mathcal{M}|K \max(\log|\mathcal{M}|, \log K))$ using a branch-and-bound approach implemented with a priority queue [46, Ch. 6]. Starting from the best hypothesis, defined as the product of the best hypothesis in each cluster, we determine the other $K-1$ merged hypotheses by iteratively extracting and expanding the best solution in the tree of all the possible combinations of weights.

Once we obtain the set of merged global hypotheses for each cluster $c \in \mathcal{M}$, we can define the posterior density on a partition $\mathcal{C}_k$, as shown in Lemmas 2 and 4. If the $n_{k'|k}$ tracks are partitioned into a set of n_k^c clusters, the simplified data association problem can be split into n_k^c independent subproblems. This approximation enables a remarkable reduction of the computational time and it enables the direct use of parallelization techniques in the update step.

V. Merging of Bernoulli Densities

In this section, we present two merging strategies to reduce the number of Bernoulli components in the clustered PMBM posterior. In Section V-A, we propose to merge the most similar single target hypotheses corresponding to the same potential target according to the KLD between Bernoulli RFSs [34]. In Section V-B, we present an algorithm that can rearrange the Bernoulli components across different potential targets that is useful in cluster formation and to lower the number of Bernoulli components in situations where targets get in close proximity and then separate.

A. Intra-track Bernoulli Merging

This section deals with merging of different local hypotheses corresponding to the same potential target. The aim is to detect the Bernoulli components that are sufficiently similar in terms of KLD for each potential target. The similar Bernoulli components, each in a different local hypothesis, are substituted by a single local hypothesis, reducing the overall number of single target hypotheses and decreasing the computational burden of the update step.

The algorithm results in an heuristic mixture reduction procedure which iteratively decrease the number of mixture components by merging the most similar Bernoulli components at each iteration. Note that, unlike in [32], the merging is performed at the Bernoulli components level, and it does not affect the global hypotheses weights directly. Nevertheless, after the update of the global hypotheses with the new local hypotheses indexes due to merging, the list of global hypotheses can present duplicates, which can be simplified by summing the weights of the identical global hypotheses.

We perform merging in two ways. First, the Bernoulli components (of the same potential target) associated with the same measurement are merged. Second, we use the KLD

to determine similar Bernoulli components that should be merged. Note that a distance between the Bernoulli densities based on the Rényi divergence has been presented in [47].

1) *Bernoulli Merging:* Let us consider a potential target i, its $h^i_{k|k}$ single target hypotheses with index $a^i \in \{1, \ldots, h^i_{k|k}\}$, and the subset of global hypotheses $\mathcal{A}^i_{k'|k} \subseteq \mathcal{A}_{k'|k}$ in which the target i is supposed to exist. The potential target state can be described by the mixture of Bernoulli densities

$$\widetilde{f}^i_{k'|k}(\widetilde{X}^i) = \sum_{a \in \mathcal{A}^i_{k'|k}} W^{i,a^i}_{k'|k} \widetilde{f}^{i,a^i}_{k'|k}(\widetilde{X}^i) \tag{25}$$

where $W^{i,a^i}_{k'|k}$ is the component weight defined as the sum of the weights associated to the global hypotheses in which a specific Bernoulli component $\widetilde{f}^{i,a^i}_{k|k}$ appears, i.e.,

$$W^{i,a^i}_{k'|k} = \sum_{a \in \mathcal{A}^i_{k'|k}} w^a_{k'|k}. \tag{26}$$

Suppose $p^{i,a^i}_{k'|k}(x)$, the single target density of $\widetilde{f}^{i,a^i}_{k'|k}(\widetilde{X}^i)$, is Gaussian, e.g., $p^{i,a^i}_{k'|k}(x) = \mathcal{N}(x; \mu^{i,a^i}_{k'|k}, P^{i,a^i}_{k'|k})$. Assume that we aim to merge the components of indexes $\mathcal{A}^{i_m}_{k'|k} \subseteq \{1, \ldots, h^i_{k|k}\}$ in the Bernoulli mixture $\widetilde{f}^i_{k'|k}(\widetilde{X}^i)$ into a single Bernoulli density $\widehat{f}^i_{k'|k}(\widetilde{X}^i)$, with single target density $\widehat{p}^i_{k'|k}(x) = \mathcal{N}(x; \widehat{\mu}^i_{k'|k}, \widehat{P}^i_{k'|k})$. The approximated Bernoulli density $\widehat{f}^i_{k'|k}(\widetilde{X}^i)$ that minimizes the KLD $D(\widetilde{f}|\widehat{f})$ is characterized by

$$\widehat{W}^{i,a^i}_{k'|k} = \sum_{a^i \in \mathcal{A}^{i_m}_{k'|k}} W^{i,a^i}_{k'|k} \tag{27}$$

and it is expressed by [11], [32]

$$\widehat{f}^i_{k'|k}\left(\widetilde{X}^i\right) = \begin{cases} 1 - \widehat{r}^i_{k'|k} & \widetilde{X}^i = \emptyset \\ \widehat{r}^i_{k'|k} \mathcal{N}(x; \widehat{\mu}^i_{k'|k}, \widehat{P}^i_{k'|k}) \delta_i[u] & \widetilde{X}^i = \{(u, x)\} \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

where

$$\widehat{r}^i_{k'|k} = \frac{\sum_{a^i \in \mathcal{A}^{i_m}_{k'|k}} W^{i,a^i}_{k'|k} r^{i,a^i}_{k'|k}}{\sum_{a^i \in \mathcal{A}^{i_m}_{k'|k}} W^{i,a^i}_{k'|k}} \tag{29}$$

$$\widehat{\mu}^i_{k'|k} = \frac{\sum_{a^i \in \mathcal{A}^{i_m}_{k'|k}} W^{i,a^i}_{k'|k} r^{i,a^i}_{k'|k} \mu^{i,a^i}_{k'|k}}{\sum_{a^i \in \mathcal{A}^{i_m}_{k'|k}} W^{i,a^i}_{k'|k} r^{i,a^i}_{k'|k}} \tag{30}$$

$$\widehat{P}^i_{k'|k} = \frac{\sum_{a^i \in \mathcal{A}^{i_m}_{k'|k}} W^{i,a^i}_{k'|k} r^{i,a^i}_{k'|k} \left(P^{i,a^i}_{k'|k} + \mu^{i,a^i}_{k'|k} (\mu^{i,a^i}_{k'|k})^T\right)}{\sum_{a^i \in \mathcal{A}^{i_m}_{k'|k}} W^{i,a^i}_{k'|k} r^{i,a^i}_{k'|k}} - \widehat{\mu}^i_{k'|k} (\widehat{\mu}^i_{k'|k})^T. \tag{31}$$

2) *KLD Between Bernoulli Distributions:* We aim to find an approximation of (25) by merging the most similar Bernoulli components. We evaluate the similarity between two Bernoulli distributions using the closed-form of the KLD between two Bernoulli distributions presented in Lemma 5. The proof and other distances for Bernoulli merging are available in [34].

LEMMA 5 Let $\widetilde{f}_1(\widetilde{X})$ and $\widetilde{f}_2(\widetilde{X})$ be two Bernoulli RFS distributions with Gaussian single target densities. The ith Bernoulli RFS has probability of existence r_i, mean $\bar{x}_i$, and covariance matrix P_i. If $r_2 \notin \{0, 1\}$, the KLD of $\widetilde{f}_2$ from $\widetilde{f}_1$ exists and it is a finite value, given by

$$\begin{aligned} \mathrm{D}_{KL}\left(\widetilde{f}_1 \,\|\, \widetilde{f}_2\right) = {} & (1 - r_1) \log \frac{1 - r_1}{1 - r_2} + r_1 \log \frac{r_1}{r_2} \\ & + \frac{r^1}{2} \left[\mathrm{tr}\left((P_2)^{-1} P_1\right) - \log\left(\frac{|P_1|}{|P_2|}\right) - n_x \right. \\ & \left. + (\bar{x}_2 - \bar{x}_1)^T (P_2)^{-1} (\bar{x}_2 - \bar{x}_1) \right]. \end{aligned} \tag{32}$$

If $r_1 = r_2 \in \{0, 1\}$, the KLD is

$$\begin{aligned} \mathrm{D}_{KL}(\widetilde{f}_1 \,\|\, \widetilde{f}_2) = {} & \frac{r^1}{2} \left[\mathrm{tr}\left((P_2)^{-1} P_1\right) - \log\left(\frac{|P_1|}{|P_2|}\right) - n_x \right. \\ & \left. + (\bar{x}_2 - \bar{x}_1)^T (P_2)^{-1} (\bar{x}_2 - \bar{x}_1) \right]. \end{aligned} \tag{33}$$

3) *Identification of Similar Bernoulli Components:* The proposed intratrack merging algorithm consists of two main steps. First, for each potential target i, the algorithm reduces the $h^i_{k|k-1}$ Bernoulli components associated with z^j_k to one single Bernoulli component $\widehat{f}^{i,j}_{k|k}$ by moment-matching, see (29)–(31). The output of this step is a set of m_k single target hypotheses resulting from the merging algorithm, $h^i_{k|k-1}$ Bernoulli components associated with a misdetection hypothesis, and their relative weights in the mixture.

Second, the single target hypotheses are iteratively merged by following a greedy merge procedure based on the KLD defined in Lemma 5, as described in [34]. The procedure considers the distances between all the elements of the Bernoulli set, and merges the two most similar Bernoulli components at each iteration, i.e., those which show the minimum distance. The merging is performed only if the distance is below a specified threshold Γ_m. Otherwise, the algorithm breaks the loop and returns the current set of Bernoulli components. The algorithm is similar to that proposed by Runnalls in [48], and the pseudocode is available in [34]. Note that an appropriate choice of the threshold Γ_m allows the filter to keep well-spaced mixture components in the mixture, providing a more adequate representation of complex scenarios.

Fig. 3 shows an example of the different steps of the intratrack merging algorithm. At time $k = 3$, the potential target $i = 1$ is described by three Bernoulli components, of which two updated with the same measurement z^1_3. We can assume that both $f^{1,1}_3$ and $f^{1,2}_3$ are sufficiently similar, and merge them in a new hypothesis according to the first step of the procedure. At time $k = 4$, the KLD between the two target hypotheses is lower than a predefined threshold, enabling the reduction via moment-matching to one single component.

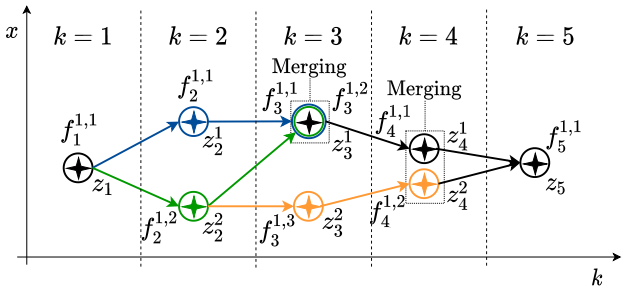

Fig. 3. Example of a superposition of the Bernoulli components for the same potential target. The stars indicate the measurements z_k at time k, while the circles represent the Bernoulli components associated to the corresponding measurement at each time step. At $k = 3$, $f_3^{1,1}$ and $f_3^{1,2}$ are updated with the same measurement z_3^1, and they are merged into the new Bernoulli $\widehat{f}_3^{1,1}$(not displayed in the figure). At the next time instant, the KLD between $f_4^{1,1}$ and $f_4^{1,2}$ results below the threshold, leading to the merging for the local hypothesis. The notation for the density $\widetilde{f}(\cdot)$ has been simplified in the figure.

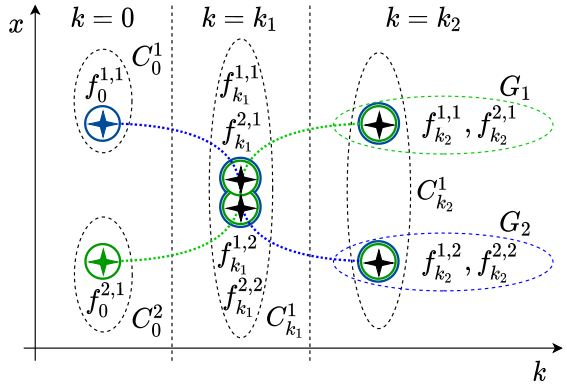

Fig. 4. Example of two targets crossing. The stars indicate the measurements at time $k = \{0, k_1, k_2\}$, and the colored circles represent the Bernoulli components associated to the tracks at each time step, where the tracks $i = \{1, 2\}$ are depicted, respectively, in blue and green. The tracks are composed of a single Bernoulli at $k = 0$, and then updated with common measurements at $k = k_1$. When the targets move away, the tracks are represented at different locations, and the related Bernoulli components can be clustered into two local clusters, namely, G_1 and G_2. The notation for the density $\widetilde{f}(\cdot)$ has been simplified in the figure.

B. Intertrack Bernoulli Swapping

It is known that after targets get in close proximity and then separate, local targets hypotheses from different targets get mixed up. For example, in an area where there is only one target, we may have hypotheses from multiple targets [35]. A standard clustering algorithm cannot put this target into a single cluster.

In this section, we propose a strategy to swap the Bernoulli components across different potential targets to improve clustering after targets get in close proximity and separate. There are two computational benefits of swapping the Bernoulli components when targets get in close proximity and then separate.

1) We avoid performing repeated prediction and update steps that are similar for different potential targets in standard PMBM filtering (see Fig. 4).
2) We increase the number of clusters.

We first note that the PMBM posterior (without auxiliary variables) remains unchanged by permuting the Bernoulli indexes in each global hypothesis. That is, the clustered MBM in (13) expressed without auxiliary variables is equivalent to

$$\widehat{q}^c_{k'|k}\left(\cup_{i\in C^c_k} X^i_{k'}\right) = \sum_{a_c\in\mathcal{A}^c_{k'|k}} w^{a_c}_{k'|k} \sum_{\cup_{j\in C^c_k} X^j = X} \prod_{i\in C^c_k} \left[f^{\sigma_{a_c}(i),a_c^{\sigma_{a_c}(i)}}_{k'|k}\left(X^i_{k'}\right)\right] \tag{34}$$

where $\sigma_{a_c} = (\sigma_{a_c}(1), \ldots, \sigma_{a_c}(n^c_{k'|k}))$ is a permutation of $(1, \ldots, n^c_{k'|k})$ applied to the global hypothesis a_c. The idea is then to use the flexibility introduced in (34) to design a fast algorithm that swaps the candidate Bernoulli indexes in specific global hypotheses to improve clustering. In addiction, these candidates will then likely be merged by the intratrack Bernoulli merging algorithm, described in Section V-A, at the next time step. In the following, we propose a computationally efficient method to exploit this flexibility through four steps.

1) *Candidate Tracks:* We identify the tracks with divergent data association histories indirectly by computing the KLD between the Gaussian component of the single target hypotheses of each track in a cluster. If a pair of components of track i has a KLD greater than a defined threshold Γ_s, we consider that the data association history is divergent and the track i is considered as candidate for the intertrack swapping. Given the Gaussian component $p^{i,a^i}_{k'|k}$ of the single target hypothesis a^i of track i, we define the set of candidate tracks in the cluster $c \in \{1, \ldots, n^c_k\}$ as

$$T_c = \left\{ i | \mathrm{D}_{KL}\left(p^{i,a^i}_{k'|k} \,\middle\|\, p^{i,b^i}_{k'|k}\right) > \Gamma_s \right\} \tag{35}$$

where $i \in C^c_k$ and $a^i, b^i \in \{1, \ldots h^i_{k'|k}\}$. Note that this procedure can be implemented as an extension of the intratrack merging procedure presented in Section V-A with no extra computational time.

In Fig. 4, we consider $T_c = \{1, 2\}$, as we suppose that the KLD between the Bernoulli components of the tracks 1 and 2 exceeds the threshold at time $k = k_2$.

2) *Bernoulli Local Clustering:* We seek to represent each potential target $i \in T_c$ by means of a set of similar hypotheses, i.e., Bernoulli components located in the same area. We apply the K-means algorithm [49] on the posterior mean positions of the candidate tracks to obtain a partition of the Bernoulli components, where K= $|T_c|$. We indicate the resulting local cluster associated with the Bernoulli component $f^{i,a^i_c}_{k'|k}(\cdot)$ with the index $j^{i,a^i_c} \in \{1, \ldots, |T_c|\}$. The clustering requires a low increase in the computational burden of the algorithm, as the number of local hypotheses to cluster is usually low due to the pruning and merging procedures applied before this point. The example in Fig. 4 shows the partition of the Bernoulli components in the cluster $C^1_{k_2}$ into two subclusters G_1 and G_2, where the subscripts $\{1, 2\}$ represent the indexes j^{i,a^i_c}.

3) *Track Assignment to Local Clusters:* At this point, each candidate track belongs to multiple local clusters. We assign each track $i \in T_c$ to one single local cluster, in order to locate it in a specific area. We express the assignment by the vector $s = (s(1), \ldots, s(|T_c|))$, $s(j^{i,a_c^i}) \in T_c$, with $s(j^{i,a_c^i})$ being the index of the reference track for the local cluster j^{i,a_c^i}. This procedure allows us to split the original cluster into several subclusters, correspondent to the local clusters, reducing the data association problem into smaller ones.

4) *Bernoulli Swapping:* At this stage, we allocate the Bernoulli components in each local cluster $l \in \{1, \ldots, |T_c|\}$ to the track $s(l)$ assigned to the local cluster l.

For example, assume the track 1 is assigned to the local cluster G_1 and track 2 is assigned to G_2 in Fig. 4. The swapping procedure aims to allocate $f_{k_2}^{2,1}$ to track 1 and $f_{k_2}^{1,2}$ to track 2.

We represent the Bernoulli swapping procedure by considering the MBM cluster density of the current cluster c expressed in (13). We determine an equivalent MBM cluster density, as in (34), by defining σ_{a_c} according to the following rules.

1) $\sigma_{a_c}(i) = i$, for $i \notin T_c$.
2) $\sigma_{a_c}(i) = s(j^{i,a_c^i})$ for $i \in T_c$.

Once we have selected σ_{a_c}, we should note that there is a rearrangement between Bernoulli components and tracks that carries along to the following time steps.

VI. SIMULATIONS

In this section, we proceed to assess the accuracy and computational time of the clustered PMBM filter and the proposed Bernoulli merging strategies in two scenarios. We also compare the standard PMBM filter implementation [12], the track-oriented PMB filter [11], and the PMBM filter with intratrack Bernoulli merging [34] against their clustered versions.

The filter implementations use a threshold for pruning the Poisson components $\Gamma_p = 10^{-5}$, a threshold for pruning global hypotheses $\Gamma_{\text{mbm}} = 10^{-4}$, and a threshold for pruning the Bernoulli components $\Gamma_b = 10^{-5}$. The maximum number of global hypotheses is $N_h = 200$ for the standard PMBM and PMB filters, while the limit on the number of global hypotheses is $N_h^c = 20|C^c|$ for each cluster c in the clustered versions of the filters. The ellipsoidal gating is performed with a k-d tree of threshold $\Gamma_g = 4.5\sigma_k^{i,a^i}$, while the estimation is performed selecting the global hypothesis with the highest weight and reporting the Bernoulli components whose existence probability is above 0.4 [12, Sec. VI.A]. The intratrack Bernoulli merging procedure has threshold $\Gamma_m = 0.25$ to determine similar Bernoulli components, and the intertrack Bernoulli has its threshold set at $\Gamma_s = 50$. These parameters have been determined empirically for good performance, and they represent a reasonable tradeoff between computational burden and accuracy. Note that the clustering and merging methods only add two parameters to those required by the standard PMBM filter implementation.

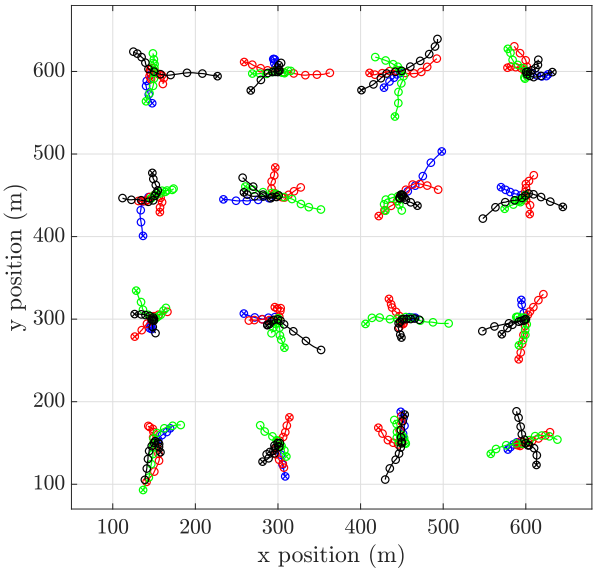

Fig. 5. Example of a simulation $N_{\text{sim}} = 2$ in Scenario 1. Each group is composed by four targets, and each target is depicted with a different colour within the group. All the targets are born at time step $k = 1$ and survive for 101 time steps, except the blue targets that die at time step $k = 50$ in all the groups. The target positions at $k = 1$ are indicated by a cross, and the circles show the target positions every ten time steps.

Furthermore, we provide a comparison with the efficient implementation of the δ-GLMB [13] with joint prediction and update, with and without adaptive birth. The number of global hypotheses and birth model in δ-GLMB are matched with the standard PMBM filter [33]. All filters have been implemented using Murty's algorithm [50].

In the simulations, target motion follows a nearly constant velocity model [51]. The target state is described in a 2-D Cartesian coordinate system by $s_k = [p_{x,k}, v_{x,k}, p_{y,k}, v_{y,k}]^T$, where the first two components represent position and velocity of the target on the x-axis, and the last two those on the y-axis. The parameters of the linear and Gaussian motion and measurement models are

$$F = I_2 \otimes \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}, \qquad Q = qI_2 \otimes \begin{pmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{pmatrix}$$

$$H = I_2 \otimes \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad R = I_2$$

where $\otimes$ is the Kronecker product, $T = 1$ is the sampling period, and $q = 0.01$ or $q = 0.2$ in Scenario 1 and 2, respectively. The clutter model is Poisson, uniformly distributed in the area of interest, with a mean number of clutter measurements per scan λ_c dependent on the area of interest in each scenario. We set the probability of survival of the targets $p_S = 0.99$, and the probability of detection $p_D = 0.9$ for all the simulations. To evaluate the performance of the algorithm, we consider the root mean square (rms) of the GOSPA error ($\alpha = 2$, $c = 10$, $p = 2$) [52], which allows us to decompose the total error into localization error, missed target error, and false target error.

We consider two scenarios based on different parameters and structure, as shown in Figs. 5 and 9. For each

TABLE I
Simulations Parameters for Scenario 1 and 2

	Scenario 1				Scenario 2			
N_{sim}	1	2	3	4	1	2	3	4
N_g	4	16	64	256	N/D	N/D	N/D	N/D
N_b	16	64	256	1024	16	64	256	1024
N_a	14	56	224	895	6	24	96	374
d_A	400	750	1350	2550	600	1200	1800	2400
λ_c	2.25	6.25	20.25	72.25	24	96	216	384
N^b_{GH}	267	284	300	296	143	242	258	259
N^a_{GH}	126	163	191	195	33	61	88	91

The number of groups of targets N_g is not defined in Scenario 2, as the targets are born in the same area of interest.

scenario, we perform four simulations denoted by the index N_{sim} and defined by the number of groups of targets N_g, the mean number of targets born during the simulation N_b, the mean number of targets alive at each time step N_a, the side length of the area of interest d_A, and the mean number of clutter measurements per scan λ_c. All units in this section are expressed in the international system and omitted for notational clarity. Table I reports the parameters of the simulations and the mean number of global hypotheses before and after pruning, N^b_{GH} and N^a_{GH}, respectively, in the standard PMBM for each simulation and scenario.

The simulations have been performed on a laptop equipped with Intel (R) Core(TM) i7-8850H @ 2.60 GHz and 16 GB of memory. All the codes are written in MATLAB, except for Murty's algorithm and R-tree, which are written in C++,[1] and the priority queue, which is based on a Python implementation. The results are based on the average on 50 Monte Carlo (MC) runs, except for those related to the simulations $N_{\text{sim}} = 4$, which are based on 30 MC runs due to the long execution times for the standard PMBM filter.

Note that all filters use suboptimal estimators and different approximations, like pruning, merging, and clustering. While the PMBM filter without approximations and an optimal estimator provides optimal estimates of the set of targets, a PMB filter implementation can perform better than a PMBM filter implementation with a suboptimal estimator and approximations.

A. Gating

In Fig. 6, we compare the mean gating times of several gating procedures. The gating time is defined as the time to update the single target hypotheses in the prediction density $\widetilde{f}_{k|k-1}(\cdot)$, and it includes the time to build and query the space partitioning data structure. It also considers the time to generate the misdetection hypotheses and to compute the expected target measurements and the innovation covariances. The gating thresholds for the ellipsoidal, k-d tree, and R-Tree gating are $\gamma_G = 20$, $\gamma_G = 4.5$, and $\gamma_G = 8$, respectively, and they provide equivalent results.

[1]We used the Murty's algorithm implementation in the tracker component library [53], and a modification of the R-Tree algorithm by Antonin Guttman available on https://github.com/nushoin/RTree.

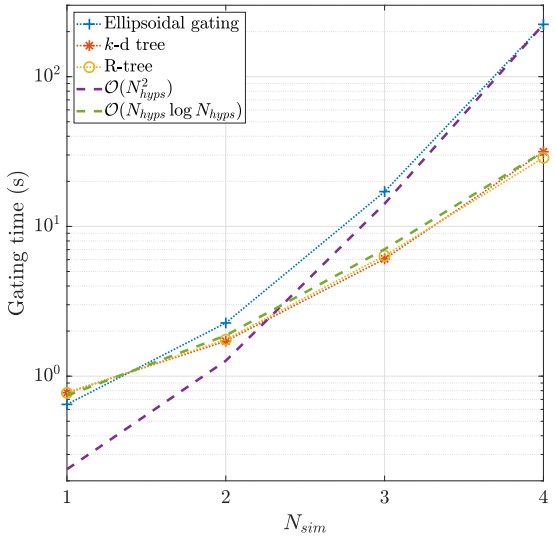

Fig. 6. Comparison between the computational time of the gating procedure using different data structures (5MC runs). The dashed lines represents the asymptotic computational complexity based on N_{hyps}.

Fig. 6 highlights the asymptotic computational complexity based on the mean of the total number of single target hypotheses N_{hyps} generated throughout all the time instants of each simulation. The use of k-d trees or R-trees has a computational burden associated with the initialization step, which overcomes the benefits of using data structures for $N_{\text{sim}} = 1$. This computational effort is rewarded with faster queries, resulting in a relevant speed-up of the gating procedure as the number of targets increases in the simulations. The comparison between k-d tree and R-tree yields a limited difference in terms of gating time, and results independent on the number of targets.

B. Scenario 1

Scenario 1 is an extension of the base scenario proposed in [12], which consists of four targets, all born at time step 1 and alive throughout the simulation of 101 time steps, except one which dies at time step 50 (the blue ones in Fig. 5). The base scenario is considered challenging, as all the targets get close at time step 50, when the blue one dies. We extended the base scenario by generating N_g groups of four targets in the area of interest, as shown in Fig. 5. Each group of four targets is generated at a distance $d_{\text{offset}} = 150$ from the centre of the adjacent groups, within a square area of side length $d_a = 300$. The total area of interest is $A = [0, d_A(N_g)] \times [0, d_A(N_g)]$, where $d_A(N_g) = d_a + d_{\text{offset}} \cdot (N_g - 1)$.

We test the scenario with four configurations based on different numbers of groups N_g, as indicated in Table I. In each configuration, the targets are born according to a PPP of intensity $\lambda \cdot u_A(z)$ at the first time step, where $u_A(z)$ is a uniform density in its area of interest and $\lambda = 3N_g$; the PPP intensity decreases to 0.005 at the next time steps. The intensity is Gaussian with mean $[d_A(N_g)/2, 0, d_A(N_g)/2, 0]^T$, and covariance $\text{diag}([(1.1d_A(N_g))^2, 1, (1.1d_A(N_g))^2, 1])$.

TABLE II
Performance and Computation Time of the Clustered PMBM With Bernoulli Merging and Swapping Techniques Based on Different Simulations in Scenario 1

	RMS GOSPA error				Time (s)			
N_{sim}	1	2	3	4	1	2	3	4
Standard PMBM	5.28	9.37	18.65	38.35	14.71	80.78	630.05	37523.1
Clustered PMBM	5.29	9.33	18.46	37.09	3.36	12.86	57.33	285.61
Clustered PMBM intra-track merging	5.28	9.31	18.46	36.84	2.97	11.16	44.25	212.50
Clustered PMBM merging and swapping	5.27	9.27	18.42	37.08	2.83	10.28	42.69	198.73

The simulations $N_{\text{sim}} = [1, 2, 3, 4]$ have mean number of targets born during the simulation $N_b = [16, 64, 256, 1024]$.

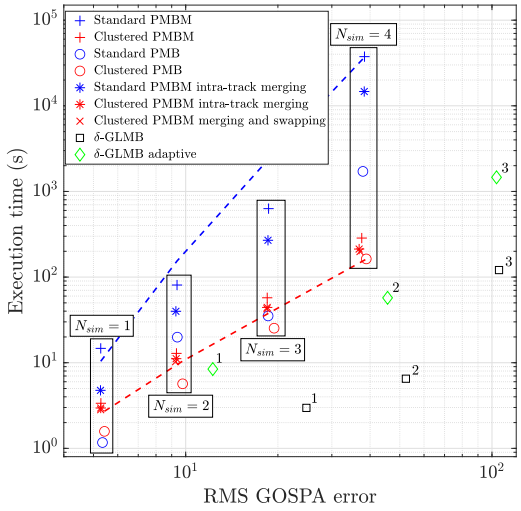

Fig. 7. Comparison between performance and execution times between the standard PMBM and PMB filters (in blue), their clustered versions (in red) and δ-GLMB in Scenario 1. Each version of the filter is represented with a different marker. The simulations $N_{\text{sim}} = [1, 2, 3, 4]$ have mean number of targets born during the simulation $N_b = [16, 64, 256, 1024]$, and they indicated with a box or a superscript number (δ-GLMB). The blue and red dashed lines represent the asymptotic computational complexity $\mathcal{O}(N_a^2)$ and $\mathcal{O}(N_a)$, respectively.

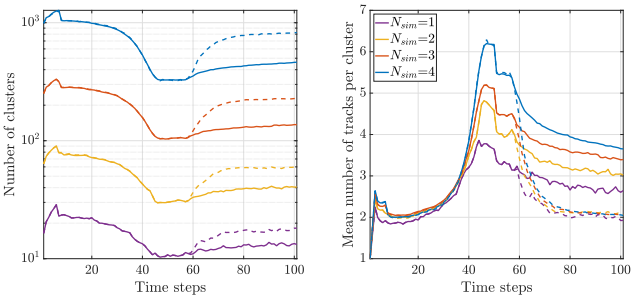

Fig. 8. Comparison of the mean number of clusters and mean number of tracks per cluster for the clustered PMBM with intratrack and intertrack Bernoulli merging. The simulations are based on different simulations in Scenario 1. Solid lines correspond to the clustered PMBM performing only intratrack merging, while dashed lines correspond to the clustered PMBM performing both the intra and inter track Bernoulli merging and swapping procedures.

1) *Clustering:* In Fig. 7 the results of the simulations based on Scenario 1 are indicated using different markers, and the asymptotic computational complexity based on the mean number of targets alive at each time step N_a is expressed by the two dashed lines. Note that the computational gains are highly dependent on the scenario, and we cannot draw general conclusions.

The outcome shows reduced computational time for both the clustered PMBM and PMB compared to their standard implementations and the δ-GLMB filter with adaptive birth. Notably, the clustered PMBM and its variations based on the Bernoulli merging and swapping show the same performance than the standard filter, where the two Bernoulli reduction techniques provide even lower execution times, as indicated in more detail in Table II. The δ-GLMB filter without adaptive birth results faster than PMBM, but it shows a high GOSPA error, especially in the misdetected target error, due to the great number of targets born simultaneously at the beginning of the simulations.

The clustered PMB filter is usually faster than the correspondent standard implementation, although it results less accurate. The management cost of the clusters overcomes the benefits of our approach in scenarios with a low number of targets, e.g., $N_{\text{sim}} = 1$. Similar conclusions can be drawn for $p_D = 0.7$, see Appendix D (Supplementary material).

2) *Intertrack Bernoulli Merging:* Table II compares the results of the simulations based on Scenario 1 of the clustered PMBM filter approximated with the intra and intertrack Bernoulli merging and swapping procedures. As already noticed, these techniques presents a greater reduction of the computational time in the most challenging scenarios, e.g., $N_{\text{sim}} = \{3, 4\}$. This observation is supported by the statistics reported in Fig. 8, where it is possible to notice a significant increase in the number of clusters using the intertrack Bernoulli swapping procedure. Moreover, the mean number of tracks per cluster falls to two regardless of the number of targets in the simulation, which suggests the formation of efficient clusters comprising an updated track and a new one related by the same measurement.

C. Scenario 2

Scenario 2 considers targets that appear and disappear at different time instants in an area $A = [0, d_A(N_{\text{sim}})] \times [0, d_A(N_{\text{sim}})]$, $d_A = 600 N_{\text{sim}}$. The target state at the appearing time is Gaussian with mean $[d_A(N_{\text{sim}})/2, 0, d_A(N_{\text{sim}})/2, 0]^T$ and covariance $\text{diag}([(60N_{\text{sim}})^2, 1, (60N_{\text{sim}})^2, 1])$. The probability of survival is $p_D = 0.99$ and the expected number of targets born at each time step is $N_b/100$, where N_b is

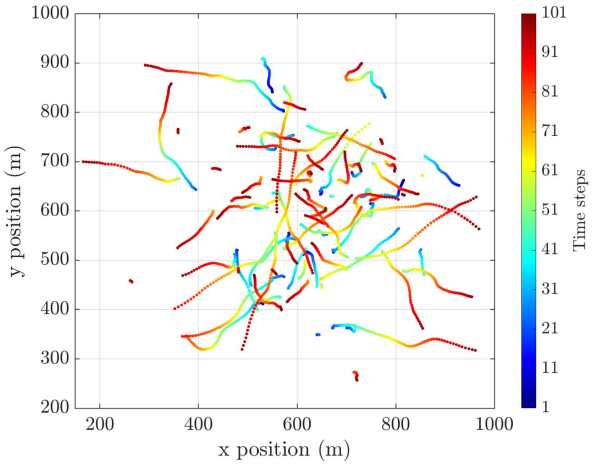

Fig. 9. Example of a simulation $N_{\text{sim}} = 2$ in Scenario 2. The colors represent the evolution over time of the target positions in the field of view.

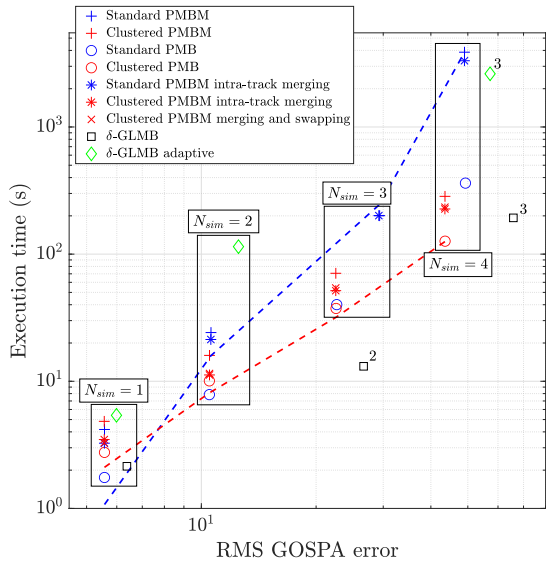

Fig. 10. Comparison between performance and execution times between the standard PMBM and PMB filters (in blue), their clustered versions (in red) and δ-GLMB in Scenario 2. Each version of the filter is represented with a different marker. The simulations $N_{\text{sim}} = [1, 2, 3, 4]$ have mean number of targets born during the simulation $N_b = [16, 64, 256, 1024]$, and they indicated with a box or a superscript number (δ-GLMB). The blue and red dashed lines represent the asymptotic computational complexity $\mathcal{O}(N_a^2)$ and $\mathcal{O}(N_a)$, respectively.

indicated in Table I. The PPP intensity is Gaussian with mean $[d_A(N_{\text{sim}})/2, 0, d_A(N_{\text{sim}})/2, 0]^T$, and covariance $\text{diag}([(1.1d_A(N_{\text{sim}}))^2, 1, (1.1d_A(N_{\text{sim}}))^2, 1])$.

In Fig. 10, we show the results of the simulations based on Scenario 2. The RMS GOSPA error of the PMBM filter results higher than the respective clustered version in simulations $N_{\text{sim}} \in \{3, 4\}$. The reason is related to the high clutter rate in these simulations, which generates a significant number of global hypotheses. Most of these hypotheses are pruned in the PMBM filter due to the cap on the maximum number of global hypotheses. The distributed representation of the global hypotheses implemented by the clustered PMBM allows us to express a higher number of global hypotheses in a more efficient way, resulting in better performance in reduced computational time. Note that, as in Scenario 1, if the number of targets is low, the management cost of the clusters overcomes the benefits of our approach for some filters.

VII. CONCLUSION

In this article, we have proposed several algorithms to reduce the complexity of the PMBM filter, enabling its use in complex scenarios with high number of targets. We have introduced the clustered PMBM density and a clustering algorithm based on the measurements associated with the potential targets. We have also proposed two techniques to decrease the number of similar Bernoulli components that arise while filtering, namely, intratrack Bernoulli merging and intertrack Bernoulli swapping.

We have evaluated the filters in two simulated scenarios showing the advantages of PMBM clustering for a high number of targets. Considering clustering, tracking well-separated targets splits the problem into different targets reaching linear complexity with a clustered PMBM filter. If all targets are in close proximity, it is not possible to split targets into clusters, and the PMBM and clustered PMBM should have similar complexity.

Future work is the development of a hybrid message passing interface and open multiprocessing (OpenMP) [54] implementation, such that the algorithm can be scaled beyond the capabilities of the shared memory systems for which OpenMP alone is applicable. Future lines of research can also include the use of the proposed merging strategies to perform multisensor average fusion in the PMBM/PMB framework [55]. Furthermore, other clustering techniques can be explored to perform clustered PMBM filtering.

REFERENCES

[1] B.-N. Vo et al., "Multitarget tracking," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–22, 2015.
[2] K. Chang, C.-Y. Chong, and S. Mori, "Analytical and computational evaluation of scalable distributed fusion algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 4, pp. 2022–2034, Oct. 2010.
[3] F. Meyer et al., "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.
[4] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.
[5] T. Kurien, *Issues in the Design of Practical Multitarget Tracking Algorithms, Multitarget-Multisensor Tracking*. Norwood, MA, USA: Artech House, 1990.
[6] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA, USA: Artech House, 1991.
[7] E. Brekke and M. Chitre, "Relationship between finite set statistics and the multiple hypothesis tracker," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 4, pp. 1902–1917, Aug. 2018.
[8] S. Coraluppi and C. Carthel, "If a tree falls in the woods, it does make a sound: Multiple-hypothesis tracking with undetected target births," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 3, pp. 2379–2388, Jul. 2014.
[9] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, no. 3, pp. 173–184, Jul. 1983.
[10] R. P. S. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, 2014.

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

[12] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: Direct derivation and implementation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 4, pp. 1883–1901, Aug. 2018.

[13] 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.

[14] J. L. Williams, "Hybrid Poisson and multi-Bernoulli filters," in *Proc. IEEE 15th Int. Conf. Inf. Fusion*, 2012, pp. 1103–1110.

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

[16] 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, Dec. 2014.

[17] S. Maskell, M. Briers, and R. Wright, "Fast mutual exclusion," in *Signal and Data Processing of Small Targets*, O. E. Drummond Ed., vol. 5428. Bellingham, WA, USA: SPIE, 2004, pp. 526–536.

[18] J. Collins and J. Uhlmann, "Efficient gating in data association with multivariate gaussian distributed states," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 3, pp. 909–916, Jul. 1992.

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

[20] H. Wang, T. Kirubarajan, and Y. Bar-Shalom, "Precision large scale air traffic surveillance using IMM/assignment estimators," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 1, pp. 255–266, Jan. 1999.

[21] M. A. Campbell, D. E. Clark, and F. de Melo, "An algorithm for large-scale multitarget tracking and parameter estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 4, pp. 2053–2066, Aug. 2021.

[22] O. E. Drummond and G. Frenkel, "Target tracking glossary of the SDI Panels on Tracking," in *Proc. SPIE Signal Data Process. Small Targets*, vol. 1954, 1993, pp. 606–619.

[23] J. R. Werthmann, "Step-by-step description of a computationally efficient version of multiple hypothesis tracking," in *Signal and Data Processing of Small Targets*, O. E. Drummond Ed. Bellingham, WA, USA: SPIE. Aug. 1992.

[24] J. K. Uhlmann, M. R. Zuniga, and J. Picone, "Efficient approaches for report/cluster correlation in multitarget tracking systems," Naval Res, Lab, Washington DC, USA, NRL Report 9281, 1990.

[25] H. de Waard, "An improved clustering concept for MHT applications," in *Proc. IEE Int. Seminar Target Tracking: Algorithms Appl.*, 2001, pp. 9/1–9/8.

[26] J. Roy, N. Duclos-Hindie, and D. Dessureault, "Efficient cluster management algorithm for multiple-hypothesis tracking," in *Proc. SPIE Signal Data Process. Small Targets*, 1997, vol. 3163. pp. 301–313.

[27] M. Beard, B. T. Vo, and B.-N. Vo, "A solution for large-scale multi-object tracking," *IEEE Trans. Signal Process.*, vol. 68, pp. 2754–2769, 2020.

[28] Á. F. García-Fernández, Y. Xia, and L. Svensson, "A comparison between PMBM Bayesian track initiation and labelled RFS adaptive birth," in *Proc. 25th Int. Conf. Inf. Fusion*, 2022, pp. 1–8.

[29] P. Boström-Rost, D. Axehill, and G. Hendeby, "Sensor management for search and track using the poisson multi-Bernoulli mixture filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 2771–2783, Oct. 2021.

[30] R. Singer, R. Sea, and K. Housewright, "Derivation and evaluation of improved tracking filter for use in dense multitarget environments," *IEEE Trans. Inf. Theory*, vol. 20, no. 4, pp. 423–432, Jul. 1974.

[31] Y. Xia, K. Granström, L. Svensson, Á. F. García-Fernández, and J. L. Williams, "Multi-scan implementation of the trajectory poisson multi-Bernoulli mixture filter," *J. Adv. Inf. Fusion*, vol. 14, no. 2, pp. 213–235, Dec. 2019.

[32] Y. Xia, K. Granström, L. Svensson, M. Fatemi, Á. F. García-Fernández, and J. L. Williams, "Poisson multi-Bernoulli approximations for multiple extended object filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 2, pp. 890–906, Apr. 2022.

[33] Á. F. García-Fernández, L. Svensson, J. L. Williams, Y. Xia, and K. Granström, "Trajectory poisson multi-Bernoulli filters," *IEEE Trans. Signal Process.*, vol. 68, pp. 4933–4945, 2020.

[34] M. Fontana, Á. F. García-Fernández, and S. Maskell, "Bernoulli merging for the Poisson multi-Bernoulli mixture filter," in *Proc. IEEE 23rd Int. Conf. Inf. Fusion*, 2020, pp. 1–8.

[35] C. Kreucher, K. Kastella, and A. Hero, "Multitarget tracking using the joint multitarget probability density," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.

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

[37] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, Jun. 1999.

[38] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[39] S. Challa, M. R. Morelande, D. Musicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[40] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.

[41] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1984, pp. 47–57.

[42] J. K. Uhlmann, "Algorithms for multiple-target tracking," *Amer. Scientist*, vol. 80, no. 2, pp. 128–141, 1992.

[43] Y. Manolopoulos, A. Nanopoulos, A. N. Papadopoulos, and Y. Theodoridis, *R-Trees: Theory and Applications*. London, U.K.: Springer2006.

[44] H. Alborzi and H. Samet, "Execution time analysis of a top-down R-tree construction algorithm," *Inf. Process. Lett.*, vol. 101, no. 1, pp. 6–12, Jan. 2007.

[45] M. T. Goodrich and R. Tamassia, *Algorithm Engineering*. Hoboken, NJ, USA: Wiley, 2001.

[46] R. Neapolitan, *Foundations of Algorithms*. Boston, MA, USA: Jones & Bartlett, Mar. 2014.

[47] B. Ristic and S. Arulampalam, "Bernoulli particle filter with observer control for bearings-only tracking in clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2405–2415, Jul. 2012.

[48] A. R. Runnalls, "Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 989–999, Jul. 2007.

[49] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.

[50] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Res.*, vol. 16, no. 3, pp. 682–687, 1968.

[51] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. Hoboken, NJ, USA: Wiley, 2001.

[52] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *Proc. IEEE 20th Int. Conf. Inf. Fusion*, 2017, pp. 1–8.

[53] D. F. Crouse, "The tracker component library: Free routines for rapid prototyping," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 32, no. 5, pp. 18–27, May 2017.

[54] R. Rabenseifner, G. Hager, G. Jost, and R. Keller, "Hybrid MPI and OpenMP parallel programming," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, (Lecture Notes in Computer Science), B. Mohr, J. L. Träff, J. Worringen, and J. Dongarra, Eds. Berlin, Heidelberg, Germany: Springer, 2006, Art. no. 11.

[55] T. Li, Y. Xin, Z. Liu, and K. Da, "Best fit of mixture for multi-sensor Poisson multi-Bernoulli mixture filtering," *Signal Process.*, vol. 202, Jan. 2023, Art. no. 108739.

Marco Fontana received the M.Sc. degree in telecommunication engineering from the Univeristà degli Studi di Padova, Padua, Italy, in 2018. He is currently working toward the Ph.D. degree in electrical engineering and electronics with the University of Liverpool, Liverpool, U.K..

His research interests include Bayesian estimation, multitarget tracking, and target detection methods.

Ángel F. García-Fernández received the telecommunication engineering degree and the Ph.D. degree in telecommunication engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 2007 and 2011, respectively.

He is currently a Lecturer with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, U.K. He previously held postdoctoral positions with the Universidad Politécnica de Madrid; Chalmers University of Technology, Gothenburg, Sweden; Curtin University, Perth, Australia; and Aalto University, Espoo, Finland. His research interests include Bayesian estimation, with emphasis on dynamic systems and multiple target tracking.

Dr. García-Fernández was the recipient of paper awards at the International Conference on Information Fusion in 2017, 2019, and 2021.

Simon Maskell (Member, IEEE) received the M.A., M.Eng., and Ph.D. degrees in engineering from the University of Cambridge, Cambridge, U.K., in 1998, 1999, and 2003, respectively.

Prior to 2013, he was Technical Manager of command, control, and information systems with QinetiQ, Farnborough, U.K. Since 2013, he has been a Professor of autonomous systems with the University of Liverpool, Liverpool, U.K. His research interests include Bayesian inference applied to signal processing, multitarget tracking, data fusion, and decision support with particular emphasis on the application of sequential Monte Carlo methods in challenging data science contexts.