Merging-based forward-backward smoothing on Gaussian mixtures

Abu Sajana Rahmathullah *, Lennart Svensson*, Daniel Svensson†
*Department of Signals and Systems, Chalmers University of Technology, Sweden
†Electronic Defence Systems, SAAB AB, Sweden
Emails: {sajana, lennart.svensson, daniel.svensson}@chalmers.se

***Abstract*—Conventional forward-backward smoothing (FBS) for Gaussian mixture (GM) problems are based on pruning methods which yield a degenerate hypothesis tree and often lead to underestimated uncertainties. To overcome these shortcomings, we propose an algorithm that is based on merging components in the GM during filtering and smoothing. Compared to FBS based on the $N-$scan pruning, the proposed algorithm offers better performance in terms of track loss, root mean squared error (RMSE) and normalized estimation error squared (NEES) without increasing the computational complexity.**

***Index Terms*—filtering, smoothing, Gaussian mixtures, forward-backward smoothing, data association**

I. Introduction

Gaussian mixture (GM) densities appear naturally in a range of different problems. One such problem is target tracking in the presence of clutter, which is a challenging problem in many situations. Under linear and Gaussian assumptions for the motion and sensor models, there exists a closed form optimal solution for this problem. The optimal solution for filtering and smoothing involves GMs, with an exponentially increasing number of components as a function of the product of the number of measurements across time. Thus, approximations are inevitable to reduce the complexity.

The optimal solution to GM filtering retains several track hypotheses for the target, along with a probability or weight for each hypothesis. In target tracking, each track hypothesis has a sequence of data association (DA) hypotheses associated to it, and corresponds to a component in the GM. The sequence of DAs across time is usually illustrated using a graphical structure, the hypothesis tree. Along each branch in the hypothesis tree, there is a sequence of Gaussian densities across time, along with the probability for the corresponding hypothesis sequence. Within the family of multiple hypothesis tracking (MHT) algorithms [1, 10], a large quantity employ pruning as a means to reduce the number of branches in the tree (or components in the GM) at each time. As a result, the uncertainty about the DA hypotheses is significantly underestimated if the pruning is aggressive. If the track with the correct DAs was pruned during the pruning step, it can also lead to incorrect DAs (being retained after pruning) during the subsequent time instants, eventually leading to track loss.

In fixed-interval smoothing, the goal is to estimate a sequence of state variables given the data observed at all times, i.e., given a batch of data. There are two main approaches to smoothing: forward-backward smoothing (FBS) [9] and two-filter smoothing [6]. In theory, these two methods are optimal and thus identical, but in practice they often differ due to approximations made in order to obtain practical implementations. In GM smoothing, we are normally forced to use approximations during both the filtering and smoothing steps. Therefore, the performance depends on the approximations made during both the filtering and smoothing steps.

In the conventional FBS implementations [4, 6, 8], the forward filtering (FF) method is performed using a traditional pruning-based filtering algorithm, such as an MHT. For the backward smoothing (BS) step, Rauch-Tung-Striebel (RTS) smoothing is used along each branch in the hypothesis tree to obtain the smoothing posterior. Along with the Gaussian densities, the weights of all branches are also computed. The obtained solution is optimal, if there is no pruning (or merging) employed during FF. If the FF algorithm is based on pruning, we usually perform BS on a degenerated hypothesis tree. That is, the tree usually indicates that there is a large number of DA hypotheses at the most recent times, but only one hypothesis at earlier times. Degeneracy is not necessarily an issue for the filtering performance as that mainly depends on the description of the most recent DA hypotheses. However, during BS we also make extensive use of the history of DA hypotheses, which is often poorly described by a degenerate hypothesis tree. There is always a risk that a correct hypothesis is pruned at least at some time instances. BS on this degenerate tree is not going to help in improving the accuracy of the DA. At times when the degenerate tree contains incorrect DAs, the mean of the smoothing posterior may be far from the correct value. Even worse, since the smoother is unaware of the DA uncertainties, it will still report a small posterior covariance matrix indicating that there are little uncertainties in the state.

The degeneracy of the hypothesis tree in GM forward filtering is closely related to the well known degeneracy of the number of surviving particles in a particle filter. There has been a lot of work done in the existing literature regarding degeneracy in particle filter smoothing ([3, 5, 7]). Most of these discuss the degeneracy issue extensively. However, the ideas proposed in the particle smoothing literature have not yet led to any improvements in the treatment of the degeneracy in the GM smoothing context.

In this paper, we propose merging during FF as a way to avoid the occurrence of degenerate trees. We consider the problem of tracking a single target in a clutter background, in

which case the posterior densities are GMs. We present a strategy for FF and BS on GMs that involves merging and pruning approximations which we refer to as FBS-GMM (stands for forward-backward smoothing with Gaussian mixture merging). Once merging is introduced, we get a hypothesis graph instead of a hypothesis tree. Using the graphical structure, we discuss in detail how BS can be performed on the GMs that are obtained as a result of merging (and pruning) during FF.

The FBS-GMM is compared to an FBS algorithm that uses $N-$scan pruning during FF. The performance measures compared are root mean squared error (RMSE), track loss, computational complexity and normalized estimation error squared (NEES). When it comes to track loss, NEES and complexity, the merging based algorithm performs significantly better than the pruning based one for comparable RMSE.

II. Problem formulation and idea

We consider a single target moving in a clutter background. The state vector x_k is varying according to the process model,

$$x_k = Fx_{k-1} + v_k, \tag{1}$$

where $v_k \sim \mathcal{N}(0, Q)$. The target is detected with probability P_D. The target measurement, when detected, is given by

$$z_k^t = Hx_k + w_k \tag{2}$$

where $w_k \sim \mathcal{N}(0, R)$. The measurement set Z_k is the union of the target measurement (when detected) and a set of clutter detections. The clutter measurements are assumed to be uniformly distributed in the observation region of volume V. The number of clutter measurements is Poisson distributed with parameter βV, where β is the clutter density. The number of measurements obtained at time k is denoted m_k.

The objective is to find the smoothing density $p(x_k|Z_{1:K})$ for $k = 1, ..., K$ using FBS, and to compare different implementation strategies. If pruning is the only method used for mixture reduction during FF, it is straightforward to use RTS iterations to perform BS. However, if the FF approximations involve merging as well, then we need to show how the RTS iterations can be used for the merged components.

A. Idea

The shortcoming of performing BS after pruning-based FF is that the DA uncertainties are typically completely ignored for the early parts (often the majority) of the time sequence, due to pruning in the FF. This can be easily illustrated using a graphical structure, called an hypothesis tree. The hypothesis tree captures different DA sequences across time. The nodes in the graph correspond to the components of the filtering GM density. Naturally, a pruned node will not have any offsprings. Let us consider the example in Fig. 1. To the left, the figure shows the tree during FF using $N-$scan pruning (with $N = 2$). To the right in the figure is the resulting degenerate tree after FF is completed. This example can be extended to generalize that, for $k \ll K$, the tree after pruning-based FF will be degenerate. By performing BS on the degenerate tree, the DA uncertainties can be underestimated greatly.

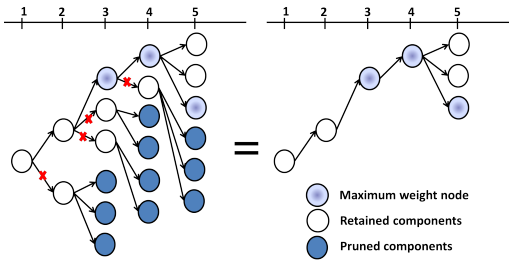

Fig. 1: Example of a degenerate tree after FF with $N-$scan pruning with N = 2. To the left, the figure shows the tree with the nodes that are pruned. In the figure to the right, the pruned branches are removed. One can see that eventually only one branch is retained from time 1 to 4.

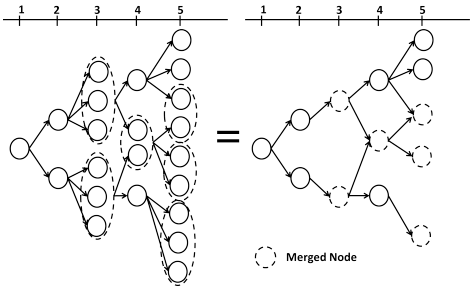

Fig. 2: Same example as in Fig. 1. Here, instead of pruning, merging of components is performed. One can see that the DA uncertainties are still retained in the merged nodes.

To overcome this weakness, one can perform merging of the nodes in the hypothesis tree instead of pruning. One can also illustrate the merging procedure during FF using a hypothesis graph, herein called the f-graph (not a tree in this case). Consider the same example shown in Fig. 2, with merging instead of pruning. To the left, one can see the graph with merging performed at several places and to the right, the result after FF that uses merging. It is clear that there is no degeneracy after merging-based FF. The idea in this paper is to develop a BS algorithm on the graph obtained after FF with merging. The smoothing density is also a GM, where GM reduction (GMR) can be employed. We use graphical structures to illustrate the FF and BS and define hypotheses corresponding to each node in the graphs to calculate the weights of different branches.

III. Background

In this section, we discuss the optimal FBS of GMs and how FBS can be performed with approximations based on pruning strategies. Towards the end of the section, FF using merging approximations is also discussed. The graphical structures in all of these scenarios are explained. In the next section, it will be shown how a similar graph structure can be created to illustrate the BS in a merging-based filter.

A. Forward-Backward Smoothing

In smoothing, we are interested in finding the smoothing posterior, $p(x_k|Z_{1:K})$. This density can be written as,

$$p(x_k|Z_{1:K}) \propto p(x_k|Z_{1:k})p(Z_{k+1:K}|x_k). \tag{3}$$

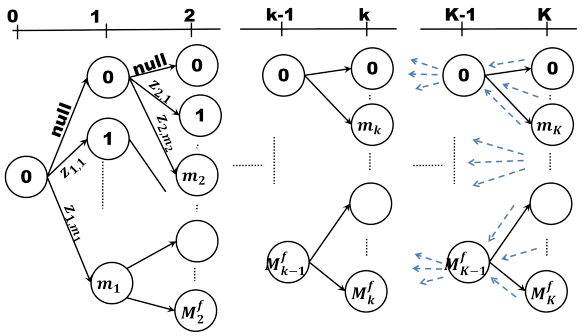

Fig. 3: Optimal Gaussian mixture filtering and smoothing are illustrated in a hypothesis tree. At each time, each node represents a component in the filtering GM at the corresponding time. The solid arrowed lines represent the data associations across time. The dashed arrowed lines represent the paths taken while performing backward smoothing from each leaf node back to the nodes at time $K-1$. During BS, these paths continue until the root node.

In (3), the smoothing posterior is obtained by updating the filtering posterior $p(x_k|Z_{1:k})$ with the likelihood $p(Z_{k+1:K}|x_k)$. This update step is analogous to the update step in the Kalman filter. The filtering density $p(x_k|Z_{1:k})$ is obtained using a forward filter. In the FBS formulation, the likelihood is obtained from the filtering density recursively as

$$p(Z_{k+1:K}|x_k) \propto \int p(Z_{k+1:K}|x_{k+1})\, f(x_{k+1}|x_k)\, dx_{k+1}, \tag{4}$$

where

$$p(Z_{k+1:K}|x_{k+1}) \propto \frac{p(x_{k+1}|Z_{1:K})}{p(x_{k+1}|Z_{1:k})}. \tag{5}$$

A problem with (5) is that it involves division of densities. In the simple case, when the two densities $p(x_{k+1}|Z_{1:k})$ and $p(x_{k+1}|Z_{1:K})$ are Gaussian, the division is straightforward. Then, the RTS smoother [9] gives a closed-form expression for the likelihood in (5) and the smoothing posterior in (3). But, for GM densities, this division does not have a closed-form solution, in general.

B. Optimal Gaussian mixture FBS

In a target-tracking problem with clutter and/or $P_D < 1$, it can be shown that the true filtering and smoothing posterior densities at any time k are GMs [8]. The filtering density GM is

$$p(x_k|Z_{1:k}) \;=\; \sum_{n=0}^{M^f_k} p(x_k|Z_{1:k}, \mathcal{H}^f_{k,n}) \Pr\left\{\mathcal{H}^f_{k,n}|Z_{1:k}\right\}, \tag{6}$$

where M^f_k is the number of components in the GM (for the optimal case, $M^f_k = \prod_{i=1}^{k}(m_i+1)$). The hypothesis $\mathcal{H}^n_k$ represents a unique sequence of measurements or missed-detection hypotheses assignments from time 1 to time k, under which the density $p(x_k|Z_{1:k}, \mathcal{H}^f_{k,n})$ is obtained. The missed-detection assignment can also be viewed as a measurement assignment and will be treated so in the rest of this paper.

The FF procedure in the optimal case is illustrated in the hypothesis tree in Fig. 3. Each node n at time k in the tree represents a Gaussian density $p(x_k|Z_{1;k}, \mathcal{H}^f_{k,n})$ and a probability $\Pr\left\{\mathcal{H}^f_{k,n}|Z_{1:k}\right\}$.

As was pointed out in Section III-A, BS involves division of densities. Since the densities involved here are GMs, we want to avoid the division of densities as it is difficult to handle in most situations. To overcome this difficulty, the smoothing density $p(x_k|Z_{1:K})$ is represented as

$$p(x_k|Z_{1:K}) = \sum_{a=0}^{M^f_K} p(x_k|Z_{1:K}, \mathcal{H}^f_{K,a}) \Pr\left\{\mathcal{H}^f_{K,a}|Z_{1:K}\right\}, \tag{7}$$

where each component $p(x_k|Z_{1:K}, \mathcal{H}^f_{K,a})$ is obtained by performing RTS using the filtering densities along every branch of the hypothesis tree (cf. Fig. 3).

C. Forward filter based on pruning and merging approximations

In this section, we discuss the different existing suboptimal strategies that are used in performing FF when the posterior densities are GMs. These methods are based on merging and pruning the branches in the hypothesis tree.

1) Pruning-based filter: In pruning, the nodes that have low values for $\Pr\left\{\mathcal{H}^f_{K,a}|Z_{1:K}\right\}$ are removed. The way the low values are identified can vary across different pruning strategies. One advantage of pruning is that it is simple to implement, even in multiple targets scenarios. A few commonly used pruning strategies are threshold-based pruning, M-best pruning and N-scan pruning [1].

The disadvantage of any pruning scheme is that in complex scenarios, it can happen that we have too many components with significant weights, but we are forced to prune some of them to limit the number of components for complexity reasons. For instance, if many components corresponding to the validated measurements have approximately the same weights, then it may not be desirable to prune some of the components. However, the algorithm might prune the components that correspond to the correct hypothesis, leading to track loss. One other drawback of pruning is that the covariance of the estimated state can be underestimated because the covariances of the pruned components are lost. This can lead to an inconsistent estimator, with a bad NEES performance. Also, as was shown in Fig. 1, pruning during FF often returns degenerate trees.

BS on trees obtained after pruning-based FF is performed in a similar way as in the optimal case discussed in Section III-B. However, the number of branches in this tree is lesser, because of which the number of RTS smoothers run is also less. The readers are referred to [10] and [1] for more details of FBS on GM with pruning-based approximations.

2) Merging-based filter: To overcome the degeneracy problem in pruning-based FF, one can use a merging (or a combination of merging and pruning) algorithm to reduce the number of components in the filtering density, instead of only pruning the components. In merging, the components which are similar are approximated as identical and replaced with one Gaussian component that has the same first two moments. Merging can be represented in the hypothesis tree with several

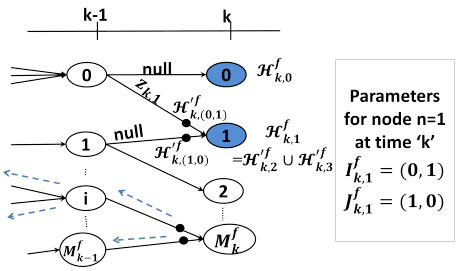

Fig. 4: A part of an f-graph is shown to illustrate merging during forward filtering (FF) (solid lines) and backward smoothing (BS) through merged nodes (dashed lines). The solid lines represent the branches in the FF. The small filled circles represent the components before merging. The dashed lines illustrate that each incident component on the merged node i while smoothing can split into many components. The hypothesis and the parameters in the box are discussed in Section IV-A1.

components being incident on the same node as shown in Fig. 4. Therefore, the structure of the hypothesis tree changes from a tree to a generic graph, which we refer as the f-graph.

There are several merging strategies discussed in [12], [11] and [2], which are used for GMR. Two main criteria for choosing the appropriate GMR algorithm are the computational complexity involved and the accuracy. Merging strategies will be discussed briefly in Section VI-B.

As a tradeoff between complexity and track loss performance, it is more feasible to use pruning along with merging, since pruning can be performed quickly. Pruning ensures that the components with negligible weights are removed, without being aggressive. Merging reduces the number of components further. This combination of pruning and merging ensures that the computations are under control without compromising too much on the performance.

IV. Backward smoothing through merged nodes

In the previous section, it was discussed how optimal GM filtering and smoothing can be performed. It was also discussed how approximations such as merging and pruning are employed during FF. We also observed that FBS is simple to apply when the FF algorithm uses pruning but not merging. In this section, we discuss in detail how BS can be performed after an FF step that involves pruning and merging approximations.

The idea behind BS on a hypothesis graph with merged nodes can be understood by first analyzing how BS works on a hypothesis tree obtained after pruning-based forward filter. As mentioned in Section III-B, in the pruning-based FF, each component in the filtering density $p(x_k|Z_{1:k})$ corresponds to a hypothesis $\mathcal{H}^f_{k,n}$, which is a unique DA sequence from time 1 to time k. BS is then performed to obtain the smoothing density as described in (7). Each component in (7) is obtained by using an RTS smoother, which combines a component $p(x_{k+1}|Z_{1:K}, \mathcal{H}^f_{K,a})$ in the smoothing density at time $k+1$ and a component $p(x_k|Z_{1:k}, \mathcal{H}^f_{k,n})$ in the filtering density at k, such that $p(x_k|Z_{1:k}, \mathcal{H}^f_{k,n}) = p(x_k|Z_{1:k}, \mathcal{H}^f_{K,a})$, and returns a component $p(x_k|Z_{1:K}, \mathcal{H}^f_{K,a})$ in the smoothing density at k. In other words, the RTS algorithm combines the smoothing density component with hypothesis $\mathcal{H}^f_{K,a}$ and the filtering density component with hypothesis $\mathcal{H}^f_{k,n}$ if the DA subsequence in $\mathcal{H}^f_{K,a}$ from time 1 to time k is identical to the DA sequence in $\mathcal{H}^f_{k,n}$. It should be noted that due to pruning during FF, the number of filtering hypotheses $\mathcal{H}^f_{K,a}$ at time K is manageable. Therefore, the number of components in the smoothing density $p(x_k|Z_{1:K})$ in (7) is also manageable and approximations are not normally needed during BS.

The key difference between the FBS that makes use of merging and the pruning-based FBS is in what the hypotheses $\mathcal{H}^f_{k,n}$ for $k = 1, \ldots, K$, represent. In the former, as a result of merging during the FF, the hypotheses $\mathcal{H}^f_{K,a}$ are sets of DA sequences, whereas in the latter, $\mathcal{H}^f_{K,a}$ corresponds to one DA sequence. As each DA sequence in the set $\mathcal{H}^f_{K,a}$ corresponds to a Gaussian component, the term $p(x_k|Z_{1:K}, \mathcal{H}^f_{K,a})$ in (7) represents a GM in the merging-based setting. It is therefore not obvious how to use the RTS algorithm for the merged hypotheses $\mathcal{H}^f_{K,a}$. The idea in this paper is that the DA sequences in each hypothesis $\mathcal{H}^f_{K,a}$ can be partitioned to form hypotheses $\mathcal{H}^s_{k,l}$ such that $p(x_k|Z_{1:K}, \mathcal{H}^s_{k,l})$ is a Gaussian density. During BS, each of these hypotheses $\mathcal{H}^s_{k,l}$ can be related to a hypothesis $\mathcal{H}^f_{k,n}$ from the FF, during the BS, enabling us to employ RTS recursions on these new hypotheses $\mathcal{H}^s_{k,l}$. Clearly, this strategy results in an increase in the number of hypotheses, leading to an increase in the number of components in the smoothing density. Therefore, there is typically a need for GMR during the BS. To represent these hypotheses $\mathcal{H}^s_{k,l}$ and the GMR of the components in the smoothing density, we use a hypothesis graph called the s-graph.

Using the new hypotheses $\mathcal{H}^s_{k,l}$, the smoothing density is

$$p(x_k|Z_{1:K}) = \sum_{l=0}^{M^s_k} p(x_k|Z_{1:K}, \mathcal{H}^s_{k,l}) \Pr\left\{\mathcal{H}^s_{k,l}|Z_{1:K}\right\}, \quad (8)$$

where $p(x_k|Z_{1:K}, \mathcal{H}^s_{k,l})$ is a Gaussian density $\mathcal{N}\left(x_k; \mu^s_{k,l}, P^s_{k,l}\right)$, with weight $w^s_{k,l} = \Pr\left\{\mathcal{H}^s_{k,l}|Z_{1:K}\right\}$. Starting with $\mathcal{H}^s_{K,a} = \mathcal{H}^f_{K,a}$ at $k = K$, the hypotheses $\mathcal{H}^s_{k,l}$ (partitioned from $\mathcal{H}^f_{K,a}$) can be obtained recursively by defining the hypotheses $\mathcal{H}^s_{k+1,p}$ at time $k+1$ and $\mathcal{H}^f_{k,n}$, as will be shown in Section IV-B. In the following sections, we introduce the two graphs, the relations between them and how these relations are used to obtain the weights of the components during BS.

A. Notation

In this subsection, we list the parameters corresponding to each node in the f-graph and s-graph. There is a one-to-one mapping between nodes in the graphs and components in the corresponding GM densities (after GMR). The symbols $\bigcup$ and $\bigcap$ used in the hypothesis expressions represent union and intersection of the sets of DA sequences in the involved hypotheses, respectively.

1) f-graph: For each node $n = 1, \ldots, M^f_k$ in the f-graph (after pruning and merging of the filtering GM) at time k, the following parameters are defined (cf. Fig. 4):

- $\mathcal{H}^f_{k,n}$, the hypothesis that corresponds to node n (after merging). If $\mathcal{H}^{'f}_{k,(i,j)}$ represent the set of disjoint hypotheses formed by associating hypothesis $\mathcal{H}^f_{k-1,i}$ of node i at time $k-1$ to measurement $z_{k,j}$ at time k, and if they correspond to the Gaussian components that have been merged to form the component at node n, then

$$\mathcal{H}^f_{k,n} = \bigcup_{i,j} \mathcal{H}^{'f}_{k,(i,j)}. \tag{9}$$

  Note that hypotheses$\mathcal{H}^{'f}_{k,(i,j)}$ do not represent any node in the f-graph. However, they can be associated to the branches incident on node n before merging (cf. Fig. 4). The prime in the notation of $\mathcal{H}^{'f}_{k,(i,j)}$ is to indicate that it is the hypothesis before merging. Similar notation will be used in the s-graph as well.
- $\mu^f_{k,n}$ and $P^f_{k,n}$ are the mean and the covariance of the Gaussian density $p(x_k|Z_{1;k}, \mathcal{H}^f_{k,n})$ (after merging).
- $I^f_{k,n}$ is a vector that contains indices i of the hypotheses $\mathcal{H}^f_{k-1,i}$ at node i at time $k-1$. An interpretation of this is that for each i in $I^f_{k,n}$, there is a branch between node i at time $k-1$ and node n at time k.
- $w^f_{k,n}$ is a vector that contains the probabilities $\Pr\left\{\mathcal{H}^{'f}_{k,(i,j)}|Z_{1:k}\right\}$ of the DA hypotheses $\mathcal{H}^{'f}_{k,(i,j)}$ before merging. Using (9), it can be shown that $\Pr\left\{\mathcal{H}^f_{k,n}|Z_{1:k}\right\} = \sum_{i,j} \Pr\left\{\mathcal{H}^{'f}_{k,(i,j)}|Z_{1:k}\right\}$.

It should be noted that the parameters $I^f_{k,n}$, $\forall n, k$, capture all the information regarding the nodes and their connections in the f-graph. Therefore, for implementation purposes, it suffices to store the parameter $I^f_{k,n}$ along with GM parameters $\mu^f_{k,n}$, $P^f_{k,n}$ and $w^f_{k,n}$, instead of storing the exponential number of DA sequences, corresponding to $\mathcal{H}^f_{k,n}$.

2) s-graph: At time k, the s-graph parameters corresponding to the l^{th} component of the smoothing density in (8) are:

- $\mathcal{H}^s_{k,l}$, $\mu^s_{k,l}$ and $P^s_{k,l}$ are the hypothesis, mean and covariance of the l^{th} Gaussian component (after merging).
- $w^s_{k,l}$ is the probability $\Pr\left\{\mathcal{H}^s_{k,l}|Z_{1:K}\right\}$.
- $I^s_{k,l}$ is a scalar that contains the index of the node (or component) in the f-graph at time k that is associated to the node l in the s-graph. This parameter defines the relation between the two graphs.

At time K, these parameters are readily obtained from the f-graph parameters: $\mathcal{H}^s_{K,l} = \mathcal{H}^f_{K,l}$, $\mu^s_{K,l} = \mu^f_{K,l}$, $P^s_{K,l} = P^f_{K,l}$, $w^s_{K,l} = \sum_r w^f_{K,l}(r)$ and $I^s_{K,l} = l$. Starting from time K, the parameters in the list can be recursively obtained at each time k as discussed in Section IV-B. In the discussion in Section IV-B, the hypotheses $\mathcal{H}^s_{k,l}$ are used to explain the weight calculations. But, for implementation purposes, it suffices to update and store the parameters $\mu^s_{k,l}$, $P^s_{k,l}$, $w^s_{k,l}$ and $I^s_{k,l}$.

B. Smoothing on merged nodes

The goal is to recursively find the smoothing density from time K to time 1. We assume that the smoothing density is available at time $k+1$, or equivalently that the nodes and the branches in the s-graph are updated until $k+1$. The components of the smoothing density $p(x_k|Z_{1:K})$ at time k are obtained by applying the RTS iterations to every possible pair, say (p, n), of the p^{th} component of $p(x_{k+1}|Z_{1:K})$ and the n^{th} component of $p(x_k|Z_{1:k})$. Whether a pair (p, n) depends on if the hypothesis $\mathcal{H}^f_{k,n}$ is in the history of the hypothesis $\mathcal{H}^s_{k+1,p}$ or not. This information can be inferred from the relation between the f-graph and the s-graph.

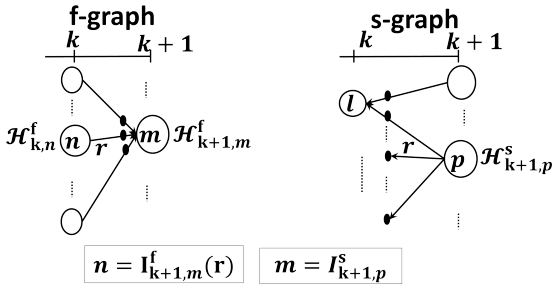

Fig. 5: Illustration of BS on a merged graph: A node p in the s-graph is shown along with the corresponding node n in the f-graph. The relations between the different parameters are as indicated. The filled black circles in the f-graph and s-graph represent the components before merging.

The possibility of forming a pair (p, n) from node p in the s-graph at time $k+1$ and node n in the f-graph at time k can be analysed using the parameters listed in Section IV-A (cf. Fig. 5). It always holds that node p in the s-graph, at time $k+1$, corresponds to one specific node m in the f-graph at time $k+1$, where $m = I^s_{k+1,p}$. A pair (p, n) can be formed whenever node m at time $k+1$ and node n at time k are connected in the f-graph. That is, if the vector $I^f_{k+1,m}$ contains the parent node index n, then the pair (p, n) can be formed. See Fig. 5 for an illustration. In fact, for every element n in the vector $I^f_{k+1,m}$, the pair (p, n) is possible.

If the pair (p, n) is 'possible', we form a node in the s-graph at time k, corresponding to that pair which is connected to node p at time $k+1$. The new node in the s-graph corresponds to a component in (8), for which we now wish to compute the mean, covariance and weight using an RTS iteration and the hypothesi relations. The hypotheses involved in obtaining the component are $\mathcal{H}^s_{k+1,p}$, $\mathcal{H}^f_{k+1,m}$ and $\mathcal{H}^f_{k,n}$, where $m = I^s_{k+1,p}$ as discussed before and node n is, say, the r^{th} element in the vector $I^f_{k+1,m}$, denoted $n = I^f_{k+1,m}(r)$ (cf. Fig. 5). Using these hypotheses, the hypothesis corresponding to the resulting component is denoted $\mathcal{H}'^s_{k,(p,n)}$ and is written as

$$\mathcal{H}'^s_{k,(p,n)} = \mathcal{H}^s_{k+1,p} \bigcap \mathcal{H}^f_{k,n}. \tag{10}$$

It can be shown that (See Appendix A for details.)

$$\begin{aligned} \Pr\left\{\mathcal{H}'^s_{k,(p,n)}|Z_{1:K}\right\} &\propto \Pr\left\{\mathcal{H}^s_{k+1,p}|Z_{1:K}\right\} \\ &\quad \times \Pr\left\{\mathcal{H}^f_{k+1,m} \bigcap \mathcal{H}^f_{k,n}|Z_{1:k+1}\right\} \\ &= w^s_{k+1,p} w^f_{k+1,m}(r). \end{aligned} \tag{11}$$

After applying the RTS iterations to every possible pair (p, n), it can happen that we have many components in the smoothing density at k. Starting with the node p at time $k+1$, we form a pair for every element n in the vector

$I^f_{k+1,n}$, resulting in a component for each pair. Therefore, the number of components in the smoothing density at time k can possibly increase, depending on how many components have been merged to form the node m at time k. Thus, to reduce the complexity, we use pruning and merging strategies during the BS step. For simplicity, merging is only allowed among the components which have the same $\mathcal{H}^f_{k,n}$, i.e, only the components that correspond to the same node n in the f-graph will be merged. After merging and pruning of the hypothesis $\mathcal{H}^{'s}_{k,(p,n)}$ for different (p, n), the retained hypothesis are relabeled as $\mathcal{H}^s_{k,l}$, and the corresponding components form the nodes l in the s-graph at time k.

V. Algorithm description

The algorithmic descriptions of the FF and the BS of the proposed FBS-GMM algorithm are presented in this section. We assume that we know the prior $p(x_0)$ at time 0 and also that we have the parameters for gating, pruning and merging. Given a set of measurements $Z_{1:K}$, we first perform FF (cf. Algorithm 1) from time $k = 1$ to $k = K$. We form the f-graph and at each node n, at each time k, store the parameters $\mu^f_{k,n}$, $P^f_{k,n}$, $w^f_{k,n}$ and $I^f_{k,n}$ described in the list in Section IV-A1. After the FF until time K, we start smoothing backwards (cf. Algorithm 2). We form the s-graph. For each time k, we get a GM, with components corresponds to a node l in the s-graph. For each of the Gaussian components, we store the parameters $\mu^s_{k,l}$, $P^s_{k,l}$, $w^s_{k,l}$ and $I^s_{k,l}$ in the list in Section IV-A2.

VI. Implementation and simulation Results

A. Simulation scenario

As mentioned in the problem formulation, we consider the problem of tracking a single target moving in a cluttered environment. The model used for simulation is a constant-velocity model with positions and velocities along x and y dimensions in the state vector. The target is assumed to be a slowly accelerating target with acceleration noise standard deviation of $0.07\ m/s^2$. The trajectory was generated for $K = 40$ time steps with a sampling time of $1\ s$. The whole volume of the track was used for generating clutter data.

The values for the measurement noise R, the probability of detection P_D and the clutter intensity β, were varied for the simulations. The measurement noise R was set to $50 \times I$ or $150 \times I$. P_D was either 0.7 or 1. The values used for β were 0.0001 and 0.0002. Thus, there are 8 sets of parameters for which the simulation results are compared.

The proposed FBS-GMM algorithm was compared with FBS based on an $N-$scan pruning algorithm. The FF was performed using $N-$scan pruning and RTS smoother was used on each branch in the filtering hypothesis tree.

B. Implementation details

The parameter N of the $N-$scan algorithm for the various settings was chosen to be the largest possible N such that the complexity (run-time) for a single run was within the threshold of $2\ s$. To reduce the complexity, extra gating was performed before the ellipsoidal gating mentioned in step 2 of Algorithm 1. This extra gate is rectangular, with dimensions based on the measurement noise covariance and the center at the prediction density mean. Besides the model parameters, the gating probability P_G and the pruning threshold P^f_P mentioned in step 2 and 3 of Algorithm 1 are $(1 - 10^{-5})$ and 10^{-4} respectively. The threshold P^s_P in step 3 of Algorithm 2 is 10^{-3}.

The merging algorithm used in step 5 during FF in Algorithm 1 is a variant of Salmond's algorithm [12] aimed at reducing the complexity compared to the original algorithm. The original Salmond's algorithm looks for the minimum merging cost across every pair of components in the GM. Thus, it has a quadratic complexity in the number of components. But to reduce the complexity of the merging algorithm, in this paper, instead of looking for the minimum cost, we use a heuristic algorithm. Starting with the components that have the least weights, we compute the cost of merging pairs of components and if the cost is lower than a threshold ($0.001 \times$ state dimension), then the components are merged and replaced in the GM. The procedure is continued with this new GM until there are no pairs of components that have a merging cost lower than the threshold.

The merging algorithm used in step 5 during BS (Algorithm 2) is a combination of the alternative Salmond's algorithm and

---

Algorithm 1 Forward filtering

Input: Prior: $\mu_{0|0}$, $P_{0|0}$.
Likelihoods: H, $z_{k,j}$, R and
$\beta_{k,j} = \begin{cases} \beta(1 - P_D P_G) & j = 0 \\ P_D & j \neq 0 \end{cases}$, for $j = 0, \dots m_k$, $k = 1, \dots K$.

Iterate for $k = 1, \dots, K$

1) **Prediction**: For each node i at time $k-1$, perform prediction to compute $\mu_{k|k-1,i}$ and $P_{k|k-1,i}$ from $\mu_{k-1|k-1,i}$ and $P_{k-1|k-1,i}$.
2) **Gating**: Initialize $G = \{\}$. For each pair of node i at $k-1$ and measurement $z_{k,j} \in Z_k$, check if $w_{LL,(i,j)} = \mathcal{N}\left(z_{k,j};\ H\mu_{k|k-1,i},\ HP_{k|k-1,i}H^T + R\right) > P_G$ and add $G = G \bigcup \{(i, j)\}$ for the pairs that pass the threshold.
3) **Pruning**: Initialize $P = \{\}$. For each pair $(i, j) \in G$, calculate the posterior weight $w_{k,(i,j)} = w^f_{k-1,i}\beta_{k,j}w_{LL}(i, j)$ and re normalize. Check if $w_{k,(i,j)} > P^f_P$ and add all pairs (i, j) that pass the threshold to P, i.e., set $P = P \bigcup \{(i, j)\}$.
4) **Update**: For each $(i, j) \in P$, update the predicted density with the measurement innovation to get $\mu_{k,(i,j)}$, $P_{k,(i,j)}$ and $w_{k,(i,j)}$.
5) **Merging**: The GM from step 4 is passed to a merging module. This module returns a reduced GM with components $\mu^f_{k,n}$ and $P^f_{k,n}$, each corresponding to a node n in the f-graph. Along with performing merging of components, the merging module also returns the vectors $I^f_{k,n}$ and $w^f_{k,n}$ that contains the indexes i and the weights $w_{k,(i,j)}$, respectively, of the components that are merged to form node n.

Algorithm 2 Backward smoothing of FBS-GMM

Input: Filtering GM parameters: $\mu^f_{k,n}$, $P^f_{k,n}$, $w^f_{k,n}$, $I^f_{k,n}$, for $n = 1, \ldots, M^f_k$.

Initialize: Set $M^s_{K,} = M^f_K$, $\mu^s_{K,l} = \mu^f_{K,l}$, $P^s_{K,l} = P^f_{K,l}$, $I^s_{K,l} = l$ and $w^s_{K,l} = \sum_r w^f_{K,l}(r)$ (summation is over the entire vector $w^f_{K,l}$).

Iterate for $k = K-1, \ldots, 1$

1) **RTS**: For each node p at time $k+1$ in the s-graph, form pairs, (p, n), as described in Section IV-B. Calculate the smoothing density mean $\mu_{k|K,(p,n)}$ and covariance $P_{k|K,(p,n)}$ using RTS on $\mu^f_{k,n}$, $P^f_{k,n}$ and $\mu^s_{k+1,p}$, $P^s_{k+1,p}$ (Note, the parameters $\mu^s_{k+1,p}$ and $P^s_{k+1,p}$ are the same for different n's).
2) **Weight calculation**: For each pair (p, n), the weight $w_{k|K,(p,n)}$ is calculated as in (16). After this, we have a bunch of triplets $\{\mu_{k|K,(p,n)}, P_{k|K,(p,n)}, w_{k|K,(p,n)}\}$ that form a GM.
3) **Pruning**: Pruning can be performed on the GM based on $w_{k|K,(p,n)} > P^s_P$ after which the GM is re-normalized.
4) **Grouping**: The components in the pruned GM are sorted into groups G_n such that all the components in the group have a common parent n at time $k-1$. The grouping is performed across all p's.
5) **Merging**: Merging can be performed within each group G_n. The output of this merging module is $\left\{\mu^s_{k,l}, P^s_{k,l}, w^s_{k,l}\right\}$ along with the parameter $I^s_{k,l} = n$.

Runnalls' algorithm [11]. The additional Runnalls' algorithm is necessary to ensure that the number of components in the GM during BS is within a threshold (50 components).

The performance measures used for comparison are the RMSE, NEES, complexity and track loss. A track was considered lost if the true state was more than three standard deviations (obtained from the estimated covariance) away from the estimated state for five consecutive time steps. The track loss was calculated only on the BS results. The complexity results presented is the average time taken during MATLAB simulations on an Intel i5 at 2.5GHz to run each algorithm on the entire trajectory of 40 time steps. The graphs were obtained by averaging over 1000 Monte Carlo iterations.

C. Results

The results of the simulations are presented in Fig. 6 to 9. It can be seen that the FBS-GMM performs significantly better than the FBS with $N-$scan pruning for most of the scenarios. From the Fig. 6 for track loss performance, one can notice that the performance gain is higher for FBS-GMM compared to FBS with $N-$scan pruning when P_D is low and the measurement noise R and the clutter intensity β are high (point 6 on the x-axis in Fig. 6). The reason for this is that in these scenarios, the number of components in the filtering GMs before approximations is quite large. To limit the number of components, the pruning during FBS with $N-$scan pruning can be quite aggressive resulting in the degeneracy problem.

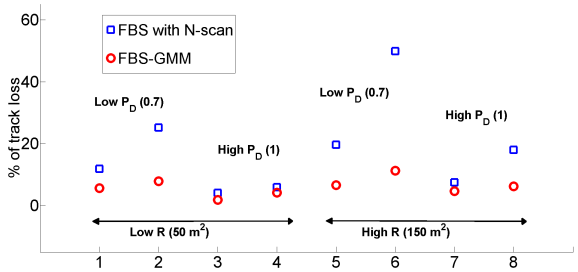

Fig. 6: Track Loss. Every odd point on x-axis (1,3,5,7) is for low clutter intensity $\beta = 0.0001$ and every even point (2,4,6,8) is for high $\beta = 0.0002$. The order of the eight scenarios is the same for the others plots in Fig. 7, Fig. 8 and Fig. 9.

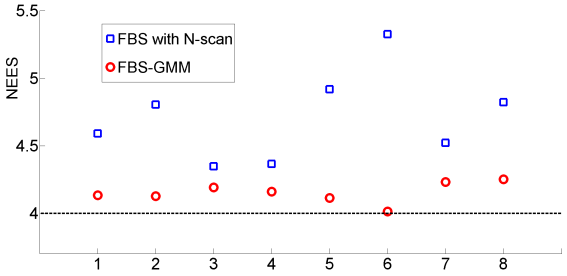

Fig. 7: NEES performance: Compared to the $N-$scan based FBS, the values of the NEES for the FBS-GMM are very close to the optimal value of 4 in all the scenarios.

The impact of this degeneracy problem can also be observed in the NEES performance plot in Fig. 7 (point 6 on the x-axis). In the degeneracy case, the uncertainties are underestimated, i.e., the estimated covariances are smaller compared to the optimal, resulting in a larger value for the NEES compared to the expected value of 4. In addition to the better track loss and NEES performances, FBS-GMM offers a computationally cheaper solution compared to the FBS based on $N-$scan pruning as can be observed in Fig. 8. However, the RMSE performance of the two algorithms are very similar in most scenarios, as seen in Fig. 9.

VII. Conclusion

In this paper, we presented an algorithm for forward-backward smoothing on single-target Gaussian mixtures (GMs) based on a merging algorithm. The weight calculation of the components in the GM during filtering and smoothing were explained by defining hypotheses. Evaluations of root-mean squared error and track loss were performed on a

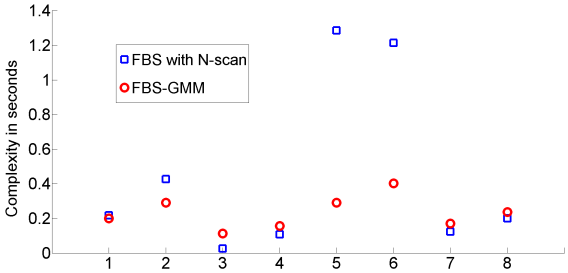

Fig. 8: Computational complexity: The FBS-GMM algorithm is computationally cheaper compared to the FBS with $N-$scan.

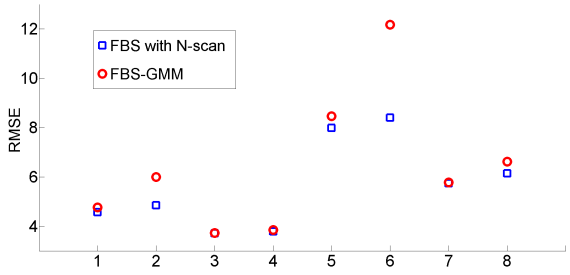

Fig. 9: RMSE performance: The results are very similar for the both the FBS algorithms.

simulated scenario. The results showed improved performance of the proposed algorithm compared to forward-backward smoothing on an $N-$scan pruned hypothesis tree, for low complexity and high credibility (normalized estimation error squared).

REFERENCES

[1] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.

[2] D. Crouse, P. Willett, K. Pattipati, and L. Svensson, "A look at gaussian mixture reduction algorithms," in *Proceedings of the Fourteenth International Conference of Information Fusion*, 2011, pp. 1–8.

[3] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of Nonlinear Filtering*, vol. 12, pp. 656–704, 2009.

[4] O. E. Drummond, "Multiple target tracking with multiple frame, probabilistic data association," in *Optical Engineering and Photonics in Aerospace Sensing*. International Society for Optics and Photonics, 1993, pp. 394–408.

[5] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.

[6] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387–390, 1969.

[7] S. J. Godsill, A. Doucet, and M. West, "Monte carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, 2004.

[8] W. Koch, "Fixed-interval retrodiction approach to bayesian imm-mht for maneuvering multiple targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 2–14, 2000.

[9] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, pp. 1445–1450, 1965.

[10] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[11] A. R. Runnalls, "Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 989–999, 2007.

[12] D. Salmond, "Mixture reduction algorithms for point and extended object tracking in clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 2, pp. 667–686, 2009.

APPENDIX A
WEIGHT CALCULATION

The weight calculation in (11) can be obtained using the hypotheses definitions. Consider the hypothesis expression in (10), and the illustration in Fig. 5. We are interested in calculating the probability of the hypothesis

$$\begin{aligned}\Pr\left\{\mathcal{H}'^{s}_{k,(p,n)}|Z_{1:K}\right\} &= \Pr\left\{\mathcal{H}^{s}_{k+1,p}\bigcap\mathcal{H}^{f}_{k,n}|Z_{1:K}\right\}\\ \propto\ &\Pr\left\{\mathcal{H}^{s}_{k+1,p}|Z_{1:K}\right\}\Pr\left\{\mathcal{H}^{f}_{k,n}|\mathcal{H}^{s}_{k+1,p},Z_{1:k+1}\right\}\\ &\times p(Z_{k+2:K}|\mathcal{H}^{s}_{k+1,p},\mathcal{H}^{f}_{k,n},Z_{1:k+1}).\end{aligned}\tag{12}$$

In the above equation, the factor $\Pr\left\{\mathcal{H}^{s}_{k+1,p}|Z_{1:K}\right\}$ is the weight $w^{s}_{k+1,p}$, which is available from the last iteration of BS at time $k+1$. With respect to the third factor, the following set of equations show that it is actually independent of n:

$$\begin{aligned}&p(Z_{k+2:K}|\mathcal{H}^{s}_{k+1,p},\mathcal{H}^{f}_{k,n},Z_{1:k+1})\\ &= p(Z_{k+2:K}|\mathcal{H}^{s}_{k+1,p},\mathcal{H}^{f}_{k+1,m},\mathcal{H}^{f}_{k,n},Z_{1:k+1})\qquad(13)\\ &= \int p(x_{k+1}|\mathcal{H}^{s}_{k+1,p},\mathcal{H}^{f}_{k+1,m},\mathcal{H}^{f}_{k,n},Z_{1:k+1})\\ &\times p(Z_{k+2:K}|\mathcal{H}^{s}_{k+1,p},\mathcal{H}^{f}_{k+1,m},\mathcal{H}^{f}_{k,n},Z_{1:k+1},x_{k+1})\,dx_{k+1}\\ &= \int p(x_{k+1}|\mathcal{H}^{f}_{k+1,m},Z_{1:k+1})p(Z_{k+2:K}|\mathcal{H}^{s}_{k+1,p},x_{k+1})\,dx_{k+1}.\qquad(14)\end{aligned}$$

In (13), adding the hypothesis $\mathcal{H}^{f}_{k+1,m}$ to the conditional statement does not make a difference as the hypothesis $\mathcal{H}^{s}_{k+1,p}$ corresponding to the entire sequence of measurements masks it; but it does make the latter equations simpler to handle. The second factor in (12) is given by

$$\begin{aligned}\Pr\left\{\mathcal{H}^{f}_{k,n}|\mathcal{H}^{s}_{k+1,p},Z_{1:k+1}\right\} &= \Pr\left\{\mathcal{H}^{f}_{k,n}|\mathcal{H}^{f}_{k+1,m},Z_{1:k+1}\right\}\\ &\propto \frac{\Pr\left\{\mathcal{H}^{f}_{k,n},\mathcal{H}^{f}_{k+1,m}|Z_{1:k+1}\right\}}{\Pr\left\{\mathcal{H}^{f}_{k+1,m}|Z_{1:k+1}\right\}}.\end{aligned}\tag{15}$$

The numerator term in (15) is the same as the probability $w^{f}_{k+1,m}(r)$ of the r^{th} branch before merging to form node m. Consolidating (14) and (15) into (12), we get that

$$\Pr\left\{\mathcal{H}'^{s}_{k,(p,n)}|Z_{1:K}\right\} \propto w^{s}_{k+1,p}\times w^{f}_{k+1,m}(r).\tag{16}$$