

Thesis for the degree of Licentiate of Engineering

---

# **Practical methods for Gaussian mixture filtering and smoothing**

Abu Sajana Rahmathullah



**CHALMERS**

Department of Signals and Systems  
Chalmers University of Technology  
Gothenburg, Sweden

---

Abu Sajana Rahmathullah  
Practical methods for Gaussian mixture filtering and smoothing

Technical Report No. R012/2014  
ISSN 1403-266X  
Department of Signals and Systems  
Chalmers University of Technology  
SE-412 96 Gothenburg, Sweden  
Telephone: + 46 (0)31-772 1000

©Abu Sajana Rahmathullah, 2014.

This thesis has been prepared using LyX.

Printed by Chalmers Reproservice  
Gothenburg, Sweden, May 2014.

*To my family*



# Abstract

In many applications, there is an interest in systematically and sequentially estimating quantities of interest in a dynamical system, using indirect and inaccurate sensor observations. There are three important sub-problems of sequential estimation: prediction, filtering and smoothing. The objective in the prediction problem is to estimate the future states of the system, using the observations until the current point in time. In the filtering problem, we seek to estimate the current state of the system, using the same information and in the smoothing problem, the aim is to estimate a past state. The smoothing estimate has the advantage that it offers the best performance on average compared to filtering and prediction estimates. Often, the uncertainties regarding the system and the observations are modeled using Gaussian mixtures (GMs). The smoothing solutions to GMs are usually based on pruning approximations, which suffer from the degeneracy problem, resulting in inconsistent estimates. Solutions based on merging have not been explored well in the literature. We address the problem of GM smoothing using both pruning and merging approximations.

We consider the two main smoothing strategies of forward-backward smoothing (FBS) and two-filter smoothing (TFS), and develop novel algorithms for GM smoothing which are specifically tailored for the two principles. The FBS strategy involves forward filtering followed by backward smoothing. The existing literature provides pruning-based solutions to the forward filtering and the backward smoothing steps involved. In this thesis, we present a novel solution to the backward smoothing step of FBS, when the forward filtering uses merging methods. The TFS method works by running two filtering steps: forward filtering and backward filtering. It is not possible to apply the pruning or merging strategies to the backward filtering, as it is not a density function. To the best of our knowledge, there does not exist practical approximation techniques to reduce the complexity of the backward filtering. Therefore, in this thesis we propose two novel techniques to approximate the output of the backward filtering, which we call intragroup approximation and smoothed posterior pruning. We also show that the smoothed posterior pruning technique is applicable to forward filtering as well.

The FBS and TFS solutions based on the proposed ideas are implemented for a single target tracking scenario and are shown to have similar performance with respect to root mean squared error, normalized estimation error squared, computational complexity and track loss. Compared to the FBS based on  $N$ -scan pruning, both these algorithms provide estimates with high consistency and low complexity.



# List of Included Publications

This licentiate thesis is based on the following appended papers:

## Paper A

A. S. Rahmathullah, L. Svensson, D. Svensson and P. Willett, “**Smoothed probabilistic data association filter**”, in *Proceedings of the 16th International Conference on Information Fusion*, Istanbul, July 2013.

## Paper B

A. S. Rahmathullah, L. Svensson and D. Svensson, “**Two-filter Gaussian mixture smoothing with posterior pruning**”, to be published in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, July 2014.

## Paper C

A. S. Rahmathullah, L. Svensson and D. Svensson, “**Merging-based forward-backward smoothing on Gaussian mixtures**”, to be published in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, July 2014.

## Paper D

A. S. Rahmathullah, L. Svensson and D. Svensson, “**Efficient strategies for Gaussian mixture smoothing**”, *to be submitted*.



## Acknowledgments

So much learning in the past three years! I owe thanks to many many people who have had an impact on me and made this journey so very amazing.

Thanks to Prof. Mats Viberg and Chalmers for offering me this opportunity to pursue my doctoral studies here in Sweden.

I would like to express my immense gratitude to my supervisors, Lennart Svensson and Daniel Svensson. Thanks for the enormous amount of guidance you have been giving me the last three years. You have made this work experience a pleasure. I am looking forward to the future collaborations.

Thanks goes to Prof. Peter Willet, for the valuable collaboration and discussions with him.

My sincere thanks to my colleagues at Chalmers and SAAB - Malin, Yinan, Yixiao, Lars, Erik, Ashkan, Raghav, Maryam, Rajet and Johannes - for their valuable time in proofreading my thesis.

Thanks to my colleagues in the Signals and Systems department - Astrid, Marie, Tomas M, Tomas B, Angel, Aidin, Ingemar, Christian, Christopher - who have made the workplace so much fun, with ski trips, wine tasting, Swedish lessons and much more.

Special thanks to my friends, Sathya, Naga, Srikar, Keerthi, Tilak and Rahul, for their support and the exhilarating lunch room debates and discussions on India.

For the fun, food and dancing, thanks goes to the dear Team India members - Sarang, Fereshteh, Annelie, Raneesh, Anamika, Dipti, Ninva, Jaga, Mahsa, Rameez, Gokul, Sandra, Babu. We share the same level of sanity and I am so glad I found you all.

To my dear friends, Deepthi and Ram, thanks for your valuable friendship, support and the interesting discussions on various topics.

Finally, my deepest gratitude to my family - Zerina, Rahmathullah, Ziana, Asan, Nibras, Hussain - for their unconditional love and support. I cannot thank you enough.

*Abu Sajana Rahmathullah*  
Gothenburg, May 2014.



# Contents

<b>Abstract</b>	<b>5</b>
<b>List of Included Publications</b>	<b>7</b>
<b>Acknowledgments</b>	<b>9</b>
<b>Contents</b>	<b>11</b>
<b>I. Introduction</b>	<b>13</b>
<b>1. Introduction</b>	<b>15</b>
<b>2. Filtering</b>	<b>19</b>
2.1. Representation . . . . .	19
2.2. Problem statement . . . . .	20
2.3. Conceptual solution . . . . .	20
2.4. Kalman filter for Gaussian densities . . . . .	21
2.5. Non-linear models . . . . .	21
2.5.1. Gaussian filters . . . . .	22
2.5.2. Particle filters . . . . .	22
2.5.3. Gaussian mixture filters . . . . .	23
<b>3. Smoothing</b>	<b>25</b>
3.1. Problem formulation . . . . .	25
3.2. Conceptual solution . . . . .	25
3.3. Gaussian example . . . . .	26
3.4. Non-linear models . . . . .	27
3.4.1. Sequential Monte Carlo methods . . . . .	28
3.4.2. Gaussian smoothers . . . . .	28
3.4.3. Gaussian mixture smoothers . . . . .	28
<b>4. Gaussian mixture filtering and smoothing</b>	<b>31</b>
4.1. Optimal solution . . . . .	31
4.1.1. Forward filtering . . . . .	32
4.1.2. Backward smoothing of FBS . . . . .	32
4.1.3. Backward filter of TFS . . . . .	32

---

4.2. Gaussian mixture reduction . . . . .	33
4.2.1. Pruning . . . . .	33
4.2.2. Merging . . . . .	33
4.2.3. Choice of GMR . . . . .	34
4.2.4. GMR for FBS and TFS . . . . .	34
4.3. Data association problem . . . . .	35
<b>5. Contributions and future work</b>	<b>37</b>
5.1. Contributions . . . . .	37
5.2. Future work . . . . .	39
<b>Bibliography</b>	<b>43</b>
<b>II. Included Papers</b>	<b>47</b>

# **Part I.**

## **Introduction**



# 1. Introduction

In many applications, the interest is to systematically and sequentially estimate quantities of interest from a dynamic system using indirect and inaccurate sensor observations. For instance, in radar tracking, the aim is in determining position and velocity of a moving or stationary aircraft or ship. In communication systems, the concern is to determine the messages transmitted through a noisy channel. In driver assistance systems, the interest is in monitoring several features about the driver, the vehicle and the surroundings. There are also several other applications such as forecasting weather or financial trends, predicting house prices, hand writing recognition, speaker identification, and positioning in navigation systems.

The sequential estimation problem can be categorized into three different problem formulations: prediction, filtering and smoothing. The prediction problem is to forecast the values of the parameters of interest, given information up to an earlier time, whereas the filtering problem is about estimating the parameter at the current time, given information up to and including that time. The smoothing problem is to estimate the past state of the parameter using all the observations made. An example from [1] can be used to explain these different problem formulations, in layman terms. Assume that we have received a garbled telegram and that the task is to read it word-by-word and make sense of what the telegram means. The filtering formulation would be to read each word and understand the meaning so far. The prediction formulation would be to guess the coming words, based on what have been read thus far. In the smoothing formulation, the reader is allowed to look ahead one or more words. Clearly, as the idiom quoted in the book goes “it is easy to be wise after the event”, the smoothing formulation will give the best result on an average, given that a delay can be tolerated.

In the above-mentioned applications, the aim is not only to estimate the parameters of interest, but also to describe the uncertainties in the estimates. The uncertainties is used to describe the reliability or trustworthiness of the produced estimates. Mathematically, an estimate and its associated uncertainty is quantified using a probability density function (for continuous states) or probability mass function (for discrete states). One of the most commonly used density functions is the Gaussian density function, which is commonly referred to as the ‘bell-shaped’ curve. The famous Kalman filter is developed as a solution to the filtering problem when the uncertainties are modeled using Gaussian density functions. There also exist solutions for the smoothing problem with Gaussian densities.

Even though the Gaussian density models and the Kalman filter solutions work well for a wide range of applications, this may not be enough for more complex systems. There are many applications where the evolution of the parameter or

the observation noise cannot be accurately modeled using Gaussian densities. For instance, the house prices varies differently in different localities, and in each locality, the prices are high around a center and decrease gradually, as one moves away from the center. Further, while tracking an aircraft, it can be in one of the possible modes — taxiing, taking off, cruising or landing — and its movement cannot be modeled using a single Gaussian model. In the clutter problem, observations are often received from objects that are not of interest, in which case uncertainty about the observations are clustered in several small regions. For instance, this happens in ship surveillance, when false measurements are received from reflections of the sea, or in air traffic surveillance, where extraneous observations from clouds and birds are received. In these kinds of scenarios, instead of a single Gaussian density, the system or the observations can be modeled using what is called a Gaussian-mixture density.

Simply put, Gaussian mixtures assign one Gaussian for each cluster/region, around which the uncertainty/data is centered, along with a weight that captures the intensity. The advantage of using a Gaussian mixture is that it is made of Gaussian components, which allows one to extend the Kalman filter solutions to these problems as well. However, in most problems, the number of Gaussian components in the mixture grows with time, which can add to the complexity burden. So, one challenge is to reduce the number of components in the Gaussian mixture. The traditional approach has been to remove a few components that are insignificant. This may work well for problems where there are good measurements, or where the system is highly likely to be in one mode. However, the estimation can stray off when removed components could have become significant with more observations collected. So, another strategy, called merging, is often used. Then the components in the mixture are merged, i.e., clusters of data that are close by can be approximated using one Gaussian density, instead of many. This method is well-studied and established for prediction and filtering solutions, but not for smoothing problems. The objective of this thesis is to use these merging strategies, for Gaussian mixtures, in smoothing problems.

The research towards writing this thesis has been sponsored by the Swedish Governmental Agency for Innovation Systems (VINNOVA), under the program “National Aviation Engineering Research Programme” part 6 (NFFP6). The project within the program is entitled “Target tracking in complex scenarios including jamming conditions for airborne sensor systems”, and is a joint collaboration between Chalmers University of Technology, Electronic Defence Systems (Saab AB) and Saab Bofors Dynamics.

## Outline of the thesis

The thesis is divided into two parts. Part I presents the theoretical background of the Gaussian mixture smoothing problem. The Chapters 2 and 3 provide the mathematical formulations of the filtering and smoothing problems, respectively.

The existing solutions for the simple Gaussian model is also presented. It is further discussed how more complex systems are handled. Then, Chapter 4 discusses in detail the Gaussian mixture smoothing problem and the difficulties involved. Finally in Chapter 5 of Part I, a summary of the contributions is presented along with a discussion of few ideas for possible future research. In Part II, the main contributions of the thesis are presented as Papers A, B, C and D.



# 2. Filtering

In the applications of the Gaussian mixture (GM) discussed in Chapter 1, the challenges are to sequentially estimate the unobserved variable and to quantify the uncertainty associated with the estimate, as we obtain the observed data. The Bayesian estimation principle is commonly used for these problems. The idea is that based on our prior knowledge of the process, we predict the variable with some uncertainty. The prediction is updated to get the ‘posterior’ as new data or likelihood becomes available. The term ‘posterior’ takes different meaning based on the available measurements, which leads to the different problem statements of prediction, filtering and smoothing.

In this chapter, we present the mathematical representations of the variables, the measurements and the relations between them. The conceptual solution, using the Bayesian principle, for the filtering and the prediction problems is presented in this chapter together with the closed form solution, in case of linear models and Gaussian noise. When the exact solution is intractable, which happens in case of non-linear models, approximations are inevitable. We also discuss the different approximation strategies that are commonly employed.

## 2.1. Representation

In the state-space representation, the unobserved variable is termed the ‘state’. The state variable at time  $k$  is denoted as  $x_k \in \mathbb{R}^n$  and the observed data as  $z_k \in \mathbb{R}^m$ . The time variability of the state is described by a motion model while the relation between the state and the measurements are given by a sensor model. Both these models capture the noise statistics. To enable sequential estimation, it is often convenient to assume a Markov model for the motion model. That is, the state  $x_k$  at time  $k$ , given all the states until time  $k - 1$ , depends only on the state  $x_{k-1}$  at time  $k - 1$ . The motion model can then be written as

$$x_k = g_k(x_{k-1}, v_k) \quad (2.1)$$

where  $v_k$  is the process noise. The Markov relation is expressed using the conditional density of the state  $x_k$  as

$$f_k(x_k | x_{k-1}, x_{k-2}, \dots, x_1) = f_k(x_k | x_{k-1}). \quad (2.2)$$

It is assumed that we have some knowledge of the state at time 0, defined by the prior density  $p(x_0)$ .

The measurement  $z_k$  is according to the sensor model

$$z_k = h_k(x_k, w_k) \quad (2.3)$$

where  $w_k$  is the measurement noise random variable. The sensor model is used to obtain the likelihood function  $p(z_k|x_k)$ . In the remainder of the thesis, the subscript  $k$  in the notation of the functions  $g_k(\cdot)$ ,  $h_k(\cdot)$ ,  $f_k(\cdot|\cdot)$  and  $p_k(\cdot|\cdot)$  will be dropped without loss of generality and for ease of writing, and represented as  $g(\cdot)$ ,  $h(\cdot)$ ,  $f(\cdot|\cdot)$  and  $p(\cdot|\cdot)$ .

## 2.2. Problem statement

The objective is to recursively compute the posterior density of the state vector  $x_k$  using the Bayesian principle (cf. [1]). In the prediction problem, the goal is to obtain the density  $p(x_k|z_{1:l})$  for  $l < k$ , given the measurements obtained from time 1 to time  $l$ , denoted  $z_{1:l}$ . In the filtering problem, the goal is to obtain the posterior density  $p(x_k|z_{1:k})$  of the state  $x_k$ . In smoothing problem, we are interested in computing the posterior density  $p(x_k|z_{1:K})$ , where  $K > k$ . The details of the smoothing problem will be discussed in Chapter 3.

## 2.3. Conceptual solution

The prediction and filtering densities can be obtained recursively in two steps, namely, prediction and update, using the prior  $p(x_0)$ , the process model  $f(x_k|x_{k-1})$  and the likelihood  $p(z_k|x_k)$ . The one-step prediction (where  $l = k - 1$ ) gives the prediction density at time  $k$  by evaluation of the integral,

$$p(x_k|z_{1:k-1}) = \int p(x_{k-1}|z_{1:k-1})f(x_k|x_{k-1})dx_{k-1} \quad (2.4)$$

and an update on this prediction gives the filtering density which is given by

$$p(x_k|z_{1:k}) \propto p(x_k|z_{1:k-1})p(z_k|x_k). \quad (2.5)$$

The constant of proportionality in the above equation is  $\frac{1}{p(z_k|z_{1:k-1})}$  where

$$p(z_k|z_{1:k-1}) = \int p(x_k|z_{1:k-1})p(z_k|x_k)dx_k. \quad (2.6)$$

It should be mentioned here that the equations in (2.4), (2.5) and (2.6) provide the theoretical solutions but, in practice, these equations are generally not tractable.

## 2.4. Kalman filter for Gaussian densities

Assume that the prior,  $p(x_0)$ , is a Gaussian density, and that the motion and the sensor models are linear functions of the state vector  $x_k$ , with additive Gaussian noise, i.e.,

$$x_k = F_k x_{k-1} + v_k \quad (2.7)$$

and

$$z_k = H_k x_k + w_k \quad (2.8)$$

where  $F_k \in \mathbb{R}^{n,n}$ ,  $H_k \in \mathbb{R}^{m,n}$ ,  $v_k \sim \mathcal{N}(0, Q_k)$  and  $w_k \sim \mathcal{N}(0, R_k)$ . Then, it can be shown that the posterior densities are Gaussian and have closed form expressions. Again, for convenience of writing, the subscript  $k$  will be dropped from the matrix notations.

Let the prediction density,  $p(x_k|Z_{1:k-1})$ , and filtering density,  $p(x_k|Z_{1:k})$ , be denoted as  $\mathcal{N}(x_k; \mu_{k|k-1}, P_{k|k-1})$  and  $\mathcal{N}(x_k; \mu_{k|k}, P_{k|k})$ , respectively. The notation  $\mathcal{N}(x; \mu, P)$  denotes a Gaussian density in the variable  $x$  with mean  $\mu$  and covariance  $P$ . The goal of prediction and filtering is then to find the first two moments of the corresponding Gaussian densities. The ubiquitous Kalman filter equations (cf. [20]) provide closed form expressions for the first two moments of the prediction and filtering densities in (2.4) and (2.5). The prediction equations are given by

$$\mu_{k|k-1} = F\mu_{k-1|k-1} \quad (2.9)$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \quad (2.10)$$

and the update equations by

$$S_k = HP_{k|k-1}H^T + R \quad (2.11)$$

$$K_k = P_{k|k-1}H^T S_k^{-1} \quad (2.12)$$

$$\tilde{z}_k = z_k - H\mu_{k|k-1} \quad (2.13)$$

$$\mu_{k|k} = \mu_{k|k-1} + K_k \tilde{z}_k \quad (2.14)$$

$$P_{k|k} = (I_n - K_k H)P_{k|k-1}. \quad (2.15)$$

where  $I_n$  is an  $n$ -by- $n$  identity matrix.  $\tilde{z}_k$  called the innovation and  $S_k$ , called the innovation covariance, describe the measurement distributions.  $K_k$  is the Kalman gain, which is the cross covariance matrix between the state  $x_k$  and the measurement  $z_k$ .

## 2.5. Non-linear models

When the motion model,  $g(\cdot)$ , and/or the measurement model,  $h(\cdot)$ , are non-linear or when the noise is not additive Gaussian, the posterior density is, in general, not

a Gaussian density. One example for this is when one gets the range and bearing measurements from a radar and want to track the position and the velocity of the target. The optimal solution then becomes intractable and loses the recursive property. There are several sub-optimal approaches to estimate the posterior density in this case, some of which are discussed in this section.

### 2.5.1. Gaussian filters

One approach to handle non-linear models is to approximate the posterior density as a Gaussian density. The methods that use this approach are called Gaussian filters, named appropriately. There are several methods to make the Gaussian approximation of the posterior and to compute its first two moments. One method is based on linearizations of the functions,  $g(\cdot)$  and  $h(\cdot)$ , after which the Kalman filter equations in (2.9) to (2.15) can be used to obtain the mean and covariance of the Gaussian approximation of the posterior density. The famous extended Kalman filter [1] and the many variants of it are based on this approach. Though these algorithms work for a good number of models, their performance deteriorates when the functions are highly non-linear.

Another type of methods used to obtain a Gaussian approximation of the posterior is based on sigma-points. In these methods, a handful of points, termed sigma-points, are chosen deterministically based on the first two moments of the prior density. The sigma points are then propagated through the non-linear models to obtain the means and covariances used to compute the moments of the Gaussian approximation of the posterior density.

### 2.5.2. Particle filters

Particle filters, or sequential Monte Carlo filters (cf. [12]), on the other hand, are based on representing the densities with a set of particles along with their corresponding weights. The particles define the positions of the Dirac delta functions such that the weighted sum of the Dirac delta functions of the particles provides a good approximation of the true density. These methods use the concept of importance sampling, where the particles are generated from a proposal density, which is simpler to generate the samples from, instead of the true density. The particles are propagated through the process model and the weights are updated using the likelihood to obtain the posterior density. The choice of the proposal density is crucial to the particle filters. The proposal density should have the same support as the true density and should be as similar to the true density as possible. The advantage of particle filters is that the performance of the filter is unaffected by the severity of the non-linearity in  $g(\cdot)$  and  $h(\cdot)$ , and that the methods are asymptotically optimal also when the functions are nonlinear. The particle filters are easy to implement. However, they can be computationally demanding as the dimension of the state vector increases. Another problem with particle filters is that they degenerate, which mean that the weights of most particles go to zero. This can be overcome by re-sampling

frequently, where multiple copies of the ‘good’ particles with significant weights are retained and the ‘poor’ particles are removed.

### 2.5.3. Gaussian mixture filters

Like in data association problem, which will be discussed in Chapter 4, the transition density and the likelihood can be multi-modal functions. One way of approximating multi-modal densities is to use a Gaussian mixture which is weighted sum of Gaussian densities. In the case of data association problems, the true posterior is itself a Gaussian mixture. The filtering of Gaussian mixture densities is handled by using the Kalman filter equations for each Gaussian component in the mixture. Based on other assumptions in the model, the weights are also updated. When the number of components in the GM model is large, approximations such as removing insignificant components and merging of similar components are often used. The details of this will be discussed in Chapter 4.



# 3. Smoothing

In the solutions to the filtering and prediction problems, there are no delays between the receipt of the last measurement and the state to be estimated. There are several applications where a delay, during which further data can be collected, is permissible. Smoothing can be used in these applications, where the delay and complexity can be compromised to get a more accurate estimate. The smoothing problem arises in different applications, such as estimating the initial state of a process or for offline processing of data. Smoothing can take three different problem formulations — fixed-point, fixed-lag and fixed-interval smoothing (cf. [1]). In this thesis, we focus on two of the Bayesian solutions to these smoothing problems, namely, two-filter smoothing and forward-backward smoothing.

In this chapter, we present the mathematical formulation of the three different smoothing problems together with the two conceptual solutions. Closed-form expressions for the posterior density under linear-Gaussian model assumptions are also presented. The different sub-optimal approaches for non-linear models are also discussed in this chapter.

## 3.1. Problem formulation

In smoothing, the goal is to compute  $p(x_k|z_{1:K})$ , i.e., the density of the state  $x_k$ , given the measurements from time 1 to time  $K$ , where  $K > k$ . Based on the value of  $K$ , the smoothing problem can be formulated in three ways as mentioned above. For the fixed-interval smoothing, as the name suggests, the interval is fixed, i.e.,  $K$  is fixed, for all  $k$ . For fixed-lag smoothing, the interval varies as a function of  $k$  with a defined lag, say  $\delta$ , i.e.,  $K = k + \delta$  for each  $k$ . In case of fixed-point smoothing,  $k$  is fixed and  $K$  increases with time, so the interval widens with time.

## 3.2. Conceptual solution

Similar to the filtering problem, sequential estimation of the smoothing posterior density can be obtained using the Bayesian principle. Though the approaches discussed here have been designed towards fixed-interval smoothing, they are in their contextual form, applicable to the other two smoothing problems as well.

The first approach is forward-backward smoothing (FBS) (cf. [9]). As the name suggests, the first step is forward filtering from time 1 to  $K$ , to obtain the filtering density  $p(x_k|z_{1:k})$  at each  $k$ . This is followed by backward smoothing from time  $K$

to time 1. The backward smoothing step at time  $k$  uses the smoothing density at time  $k + 1$  together with the filtering densities at time  $k$  as

$$p(x_k|z_{1:K}) = p(x_k|z_{1:k}) \int \frac{p(x_{k+1}|z_{1:K})}{p(x_{k+1}|z_{1:k})} f(x_{k+1}|x_k) dx_{k+1}. \quad (3.1)$$

The integral in the above equation is proportional to  $p(z_{k+1:K}|x_k)$ , termed as the backward likelihood (BL) in this thesis. Therefore, it is possible to interpret the division in the backward smoother as computing the backward likelihood implicitly.

The second approach to smoothing is the two-filter smoothing (TFS) method. To obtain the smoothing density at time  $k$  by this method, forward filtering is performed from time 1 to  $k$  to get the filtering density  $p(x_k|z_{1:k})$  and backward filtering is run from time  $K$  to time  $k$  to get the backward likelihood  $p(z_{k+1:K}|x_k)$ . The product of the two filter outputs gives the smoothing density,

$$p(x_k|z_{1:K}) \propto p(x_k|z_{1:k})p(z_{k+1:K}|x_k). \quad (3.2)$$

The backward filtering, similar to the forward filter, is performed recursively using two steps: update and retrodiction. The update step computes the likelihood

$$p(z_{k+1:K}|x_{k+1}) = p(z_{k+1}|x_{k+1})p(z_{k+2:K}|x_{k+1}) \quad (3.3)$$

and the retrodiction step computes the BL as

$$p(z_{k+1:K}|x_k) = \int p(z_{k+1:K}|x_{k+1})f(x_{k+1}|x_k)dx_{k+1}. \quad (3.4)$$

### 3.3. Gaussian example

When the motion model and the sensor model are linear with additive Gaussian noise, it can be shown that the smoothing posterior density is Gaussian. The closed form expressions for the mean and covariance of the smoothing posterior obtained using the TFS and FBS solutions, depend on the filtering densities.

For the FBS method, the Rauch-Tung-Striebel (RTS) smoother gives the closed-form expressions for the mean and covariance of the smoothing density. Using notations similar to the ones for the prediction and filtering densities, the smoothing density at time  $k$  is denoted as  $\mathcal{N}(x_k; \mu_{k|K}, P_{k|K})$ . The RTS equations are

$$\mu_{k|K} = \mu_{k|k} + C_k (\mu_{k+1|K} - \mu_{k+1|k}) \quad (3.5)$$

$$P_{k|K} = P_{k|k} + C_k (P_{k+1|K} - P_{k+1|k}) C_k^T, \quad (3.6)$$

where

$$C_k = P_{k|k} F^T P_{k+1|k}^{-1} \quad (3.7)$$

is similar to the Kalman gain in (2.12) of the Kalman filter equations.

For the TFS method, the work in [16] provides the closed-form solution for the moments of the smoothing density. Let the likelihoods be denoted as  $p(Z_{k+1:K}|x_{k+1}) = \mathcal{N}(U_{k+1}x_{k+1}; \psi_{k+1}, G_{k+1})$  and  $p(Z_{k+1:K}|x_k) = \mathcal{N}(J_kx_k; \eta_k, B_k)$ . Starting with the terminal conditions at time  $K$  be  $J_K = []$ ,  $\eta_K = []$  and  $B_K = []$  at  $k = K$ , the update step in (3.3) of the backward filter is then given by

$$U_{k+1} = \begin{bmatrix} J_{k+1} \\ H \end{bmatrix} \quad (3.8)$$

$$\psi_{k+1} = \begin{bmatrix} \eta_{k+1} \\ z_{k+1} \end{bmatrix} \quad (3.9)$$

$$G_{k+1} = \begin{bmatrix} B_{k+1} & \mathbf{0} \\ \mathbf{0} & R \end{bmatrix} \quad (3.10)$$

while the retrodiction step in (3.4) is given by

$$J_k = U_{k+1}F \quad (3.11)$$

$$\eta_k = \psi_{k+1} \quad (3.12)$$

$$B_k = U_{k+1}QU_{k+1}^T + G_{k+1}. \quad (3.13)$$

Using the outputs of the forward filter and the backward filter at time  $k$ , the smoothing density in (3.2) is given by

$$\mu_{k|K} = \mu_{k|k} + W_k(\eta_k - J_k\mu_{k|k}) \quad (3.14)$$

$$P_{k|K} = (I_n - W_kJ_k)P_{k|k}, \quad (3.15)$$

where the gain term,

$$W_k = P_{k|k}J_k^T (J_kP_{k|k}J_k^T + B_k)^{-1}. \quad (3.16)$$

## 3.4. Non-linear models

Similar to the filtering problem, getting closed-form expressions for the true smoothing posterior, when it is non-Gaussian, is generally impossible. There are additional challenges in the smoothing problem. First, the equation in the FBS method involves division of densities, which is difficult to compute for arbitrary densities. Second, the accuracy of the approximations in the forward filtering highly affects the backward smoothing and the smoothing density. The TFS method, on the other hand, does not involve density divisions and the two filters can ideally be run independently of each other. However, the likelihood  $p(z_{k+1:K}|x_k)$  is not, in general, a normalizable density function, which limits the possibilities to apply conventional approximation techniques for densities during the backward filtering. Due to these additional complications, applying the techniques used for non-linear filtering to non-linear smoothing do not always produce fruitful results. In this section, we discuss the challenges in extending techniques such as sequential Monte Carlo sampling, linearization, sigma-point methods and Gaussian mixtures to the smoothing problem.

### 3.4.1. Sequential Monte Carlo methods

Similar to the filtering method, sequential Monte Carlo smoothing is based on approximating the smoothing posterior density using a set of particles. In case of FBS based on these methods, a vanilla version works well when  $k \approx K$ . However, when  $k \ll K$  (cf. [22]), because of successive resampling steps, the marginal density becomes approximated by a single particle which leads to deteriorated performance. This is the degeneracy problem which is inherent in the particle filters (cf. [12]). One simple approach is to use the forgetting properties of the Markov model. That is, to approximate the fixed-interval smoothing density  $p(x_k|z_{1:K})$  using the fixed-lag smoothing density  $p(x_k|z_{1:k+\delta})$  (cf. [14] and [17]). Unfortunately, automatic selection of  $\delta$  is difficult.

In case of TFS, it is not straightforward to approximate the output of the backward filter using particles, as it is not a normalizable density. The artificial prior method [9] uses the auxiliary probability density  $\tilde{p}(x_k|z_{k+1:K})$  instead of the likelihood  $p(z_{k+1:K}|x_k)$ . The auxiliary density is obtained using what is called artificial prior densities. The choice of the artificial prior plays a major role in the performance of the TFS algorithm for particle methods.

### 3.4.2. Gaussian smoothers

The analogue of Gaussian filtering methods, such as the extended Kalman filter and the unscented Kalman filter, exists for TFS of non-linear models. The extended Kalman smoother, similar to its filtering counterpart, has poor performance when the non-linearity is severe. The unscented Kalman smoother needs the inverse of the dynamic model functions, which may not be feasible in all scenarios. The unscented RTS smoother [37] is the FBS version of a Gaussian smoother and is shown to have similar performance as the unscented Kalman smoother, but without the need of inverting the model functions.

### 3.4.3. Gaussian mixture smoothers

Filtering of multi-modal densities, modeled as Gaussian mixtures, was discussed briefly in the previous chapter. To limit the complexity, Gaussian mixture filtering involve pruning and/or merging approximations. When the forward filtering is based on pruning, it is trivial to perform the backward smoothing of the FBS, using the filtering densities. Starting from the last time instant, RTS is performed backward on the individual retained components. This method suffers from degeneracy similar to particle smoothing. Because of pruning, the number of components retained will be 1 for  $k \ll K$ . One way to tackle the degeneracy is to perform FBS based on merging, something that has not been explored much in the literature. So, is the TFS of Gaussian mixture densities. Then the difficulty is in using the Gaussian mixture reduction techniques in the backward filter, since its output is not a density function.

In this thesis, we argue that the Gaussian mixture smoothing is a problem that deserves more attention and that the standard solutions has considerable shortcomings. The main contribution of this thesis is an in-depth study of how to design FBS and TFS algorithms for Gaussian mixture smoothing, based on both merging and pruning.



# 4. Gaussian mixture filtering and smoothing

In many applications, the motion model captures that the state can be in one of many hidden classes. In these cases, the state transition density is a multi-modal density, where each mode corresponds to a hidden class. The motion model uncertainties can then be modeled using a Gaussian mixture (GM). There are also many applications, in which we receive a lot of measurements on the state variable, where the reliability of the measurements can vary. The likelihood in these applications are conveniently modeled as mixtures. Gaussian mixtures, as was mentioned before, are weighted sum of Gaussian densities. These usually make a good approximation for the multi-modal densities. It will be shown in this chapter that when the likelihood and/or the state transition density are Gaussian mixtures, the true posterior densities after filtering and smoothing are also Gaussian mixtures. The number of terms in the GM usually grows exponentially with time, and we therefore need to constrain the number of terms. In these situations, reduction algorithms can be used to approximate the posteriors. In this chapter, we provide a brief overview of the most commonly used mixture reduction methods and discuss the challenges in applying these to smoothing problems. Towards the end of this chapter, we also present the data association problem in tracking applications. The application of Gaussian mixture reduction (GMR) methods for the filtering problem in data association is also discussed.

## 4.1. Optimal solution

It was presented in last chapter that the forward-backward smoothing (FBS) method is based on forward filtering and backward smoothing while the two-filter smoothing (TFS) method involves forward filtering and backward filtering. These steps involve the prediction, update and retrodiction steps stated in (2.4), (2.5), (3.3) and (3.4). One can notice that all these equations involve products of functions, which in this case are Gaussian mixtures. When the state transition densities and the likelihoods are both GMs, one can use the fact that a product of GMs yields a GM and show that the posterior densities are all Gaussian mixtures. The number of components in the resulting GM is the product of the number of components in the individual mixtures, which explains why the number of components grow exponentially with time.

### 4.1.1. Forward filtering

One can show that, starting with a GM prior, the prediction and the update steps of forward filtering result in a Gaussian mixture posterior density. Evaluating these steps with GMs is equivalent to using Kalman filters, one for every triplet of Gaussian components in the prior  $p(x_{k-1}|z_{1:k-1})$ , the transition density  $f(x_k|x_{k-1})$  and the likelihood  $p(z_k|x_k)$ , yielding a Gaussian term in the posterior  $p(x_k|z_{1:k})$ . The term in the constant of proportionality  $p(z_k|z_{1:k-1})$  in (2.6), is not calculated explicitly in the update step of Kalman filter, which involves product of Gaussian densities. However, in the case with GMs, this constant of proportionality is used in the updated weight calculation. The updated weight for the resulting Gaussian component is given by the product of the individual weights of the components in the prediction density and the likelihood along with the constant of proportionality.

### 4.1.2. Backward smoothing of FBS

The backward smoothing of FBS involves a divisions of the smoothing and the prediction GM densities as in (3.1). Starting from time  $K$ , using the principle of mathematical induction, it can be shown that the division results in a GM. Instead of performing this division, one can use the observation that there is an association between the components in the filtering posterior GM at time  $k$  and time  $k+1$ . Extending this association to the smoothing densities, Rauch-Tung-Striebel (RTS) recursions can be used for every triple of associated components in the smoothing density at time  $k+1$ , the prediction at time  $k+1$  and in the filtering density at time  $k$ , to compute the smoothing density at time  $k$ . The weights of the smoothing density at time  $k$  are the same as the weights of the smoothing density at time  $k+1$ .

### 4.1.3. Backward filter of TFS

In the backward filter of TFS, we need to compute the backward likelihood as in (3.3) and (3.4). The ideas in forward filtering cannot be applied directly to the backward filter because often the likelihoods can be of the form

$$w_0 + \sum_i w_i \mathcal{N}(H_i x_k; \mu_i, P_i) \quad (4.1)$$

where different  $H_i$  can capture different features of the state  $x_k$ . Strictly speaking, these are not Gaussian mixture densities; they are neither Gaussian nor densities in  $x_k$ . We refer to them as reduced dimension Gaussian mixtures in this thesis. To compute the product of likelihoods, one can use the following general product rule:

$$w_i \mathcal{N}(H_i x; \mu_i, P_i) \times w_j \mathcal{N}(H_j x; \mu_j, P_j) = w_{ij} \mathcal{N}(H_{ij} x; \mu_{ij}, P_{ij}) \quad (4.2)$$

where  $w_{ij} = w_i w_j$ ,  $H_{ij} = \begin{bmatrix} H_i \\ H_j \end{bmatrix}$ ,  $\mu_{ij} = \begin{bmatrix} \mu_i \\ \mu_j \end{bmatrix}$  and  $P_{ij} = \begin{bmatrix} P_i & 0 \\ 0 & P_j \end{bmatrix}$ . Using this in (3.3) and (3.4), one can show that the output of the backward filter has a structure

similar to the inputs as in (4.1). The smoothing density is given by the product of the outputs of the two filters, that can be computed similarly to the update step in the forward filter of GMs, including the weight update using the proportionality constant as discussed in Section 4.1.1.

## 4.2. Gaussian mixture reduction

The number of components in the resulting GM, after update, prediction and retro-diction iterations, grows exponentially with time. Therefore, approximations are necessary to reduce the number of components. There are several GM reduction algorithms that are well studied in the literature, that can be used for filtering and smoothing. The GMR algorithms are based on pruning insignificant components from the GM and/or merging similar components.

### 4.2.1. Pruning

The number of components in the posterior GM can be prevented from growing exponentially by pruning some of the components after each iteration. There are several pruning strategies that can be adopted. Three methods that are commonly used are threshold-based pruning,  $M$ -best pruning and  $N$ -scan pruning. In threshold-based pruning, only the components that have a weight greater than a predefined threshold are retained and used for prediction in the next iteration. The number of components in the resulting GM can vary based on the threshold. The idea behind the  $M$ -best pruning algorithm is that only the nodes with the  $M$  highest weights (or association probabilities) are retained.

To explain the N-scan pruning [8], which is designed for reduction during filtering, let us say we are interested in perform pruning at time  $k$ . We pick the component that has the maximum weight. Starting from this component, we trace backwards  $N$  steps to find its parent component, at time  $k - N$ . Only the offspring at time  $k$ , of this parent node at time  $k - N$ , are retained. To be mentioned here is that the multiple hypothesis tracking (MHT) filtering is often based on  $N$ -scan pruning.

### 4.2.2. Merging

One can also use merging of similar components to reduce the number of components in a GM . There are several merging algorithms such as the Salmond's [36], Runnalls' [35] and Williams' [42] algorithms. These algorithms work based on the following three steps:

1. Find the most suitable pair of components to merge according to a ‘merging cost’ criterion.
2. Merge the similar pair and replace the pair with the merged Gaussian component.

3. Check if a stopping criterion is met. Otherwise, set the reduced mixture to the new mixture and go to step 1.

The merging cost in step 1 looks for similarity of the components and it can be different across algorithms. A few of the commonly used merging costs are the Kullback-Leibler divergence [26] and the integral-squared error. The merging of the components in step 2 is usually based on moment matching. That is, the moments of the GM before and after merging are the same. The stopping criterion can also vary across algorithms. It can be based on if the components in the reduced mixture is at a manageable number. In certain algorithms, it is checked based on that the components in the reduced GM are not similar.

#### 4.2.3. Choice of GMR

Two main criteria in choosing the appropriate GMR algorithm would be the computational complexity involved and the accuracy. Most of the pruning algorithms are usually simpler to implement, compared to merging. There is information about the uncertainty of the estimate in the covariance matrices of the pruned components. So, as a result of pruning, we might have underestimated uncertainties. Whereas in merging, the uncertainty is preserved because of moment-matching. However, the merging algorithms are more computationally intensive than pruning. As a tradeoff between complexity and accuracy of the uncertainty, it may be more feasible to use a combination of pruning and merging. Pruning ensures that the components with negligible weights are removed, without being aggressive. Merging reduces the number of components further, but keeping the moments the same.

#### 4.2.4. GMR for FBS and TFS

Applying GMR, both pruning and merging, for the forward filtering is straightforward. In case of backward smoothing of GMs, RTS uses the association between components in the filtering, prediction and smoothing densities across time. When the forward filtering is based on pruning, the backward smoothing can be performed similar to the optimal solution, on the retained components. This pruning-based FBS can suffer from degeneracy discussed in Section 3.4.3. One way to overcome degeneracy is to use merging approximations during the forward filtering. Then, for the backward smoothing, the associations across components is no longer simple, to use RTS directly and compute the weights of the smoothing density. In paper C of this thesis, the problem of FBS based on merging is investigated.

The two filters of the TFS can be run independently of each other. This allows the GMR algorithms to be used on both the filters. However, as was pointed out before, the output of the backward filter, is not a proper GM density. So, the GMR algorithms discussed here cannot be applied directly. In Paper B of this thesis, we propose a method called smoothed posterior pruning, through which pruning can be employed in the backward filter.

### 4.3. Data association problem

In this section, we describe how the GM filtering and smoothing are handled in the data association problem in target tracking in the presence of clutter. In this problem, one obtains a set of measurements, the origins of which are not known. The likelihood, in this scenario, can often be modeled as a reduced dimension Gaussian mixture as in (4.1). Therefore, under linear-Gaussian assumptions for the motion model, the optimal solution for filtering and smoothing involves GMs. The number of components in the optimal GMs is a function of the product of the number of measurements across time, and so increases exponentially with time. Thus, approximations are inevitable to reduce the complexity.

In the data association problem, at each time instant, hypotheses are defined which associates the target to the individual measurements or to no-detection hypothesis. Track hypotheses has a sequence of these target-measurement or no-detection hypothesis associated to it across time. Filtering on each track hypothesis results in a component in the filtering GM at each time instant. The optimal solution to filtering in the data association problem retains several track hypotheses for the target, along with a probability or weight for each hypothesis. The sequence of data associations across time is usually illustrated using a graphical structure, the hypothesis tree. Along each branch in the tree, we get a sequence of Gaussian densities across time, along with the probabilities for the corresponding hypothesis sequence. Many suboptimal algorithms that uses the GMR techniques for filtering have been designed for this data association problem, as will be discussed in the remainder of this section. However, smoothing based on merging, which is the focus of this thesis, has not been explored well.

Suboptimal data association approaches, such as the nearest neighbor algorithm [8] and the probabilistic data association algorithm [5, 3], retain only one track hypotheses for the target after each iteration. In the nearest neighbor algorithm, all but the component in the likelihood that is nearest to the prediction, are removed. Thus, the uncertainty in the data association is lost mostly. In the probabilistic data association (PDA) algorithm, the GM is reduced (or merged) to a single Gaussian density after each iteration using moment matching. Thus, the disadvantage being that the associated uncertainty can be quite large. Paper A in this thesis addresses the problem of improving the filtered estimates using smoothing in the PDA setting. The MHT algorithms [33, 7, 10] often employ N-scan pruning as a means to reduce the number of branches in the tree (or components in the GM) at each time. Therefore, the number of tracks retained for the target can be more than one. The uncertainty about the data association for the tracks can be underestimated if the pruning is aggressive.



# 5. Contributions and future work

The main objective of the thesis is to design algorithms for filtering and smoothing of Gaussian mixtures, based on merging and pruning. The contributions of each paper comprising the thesis are discussed briefly in this chapter. Furthermore, possible ideas for future research which arose during the writing of this thesis are presented.

## 5.1. Contributions

In the following section, the contributions of the four papers in the thesis, and the relations between them, are presented.

### Paper A

Paper A presents the smoothed probabilistic data association filter (SmPDAF) which aims at improving the Gaussian approximation of the filtering density  $p(x_k|Z_{1:k})$ , compared to the probabilistic data association (PDA) filter [3]. The idea behind SmPDAF is to improve the approximations of the filtering densities  $p(x_l|Z_{1:l})$  at past time instants  $l$  using the corresponding smoothing densities  $p(x_l|Z_{1:k})$ , where  $l < k$ . The likelihood  $p(Z_{l+1:k}|x_k)$ , involved in computing the smoothing density by the two-filter smoothing method, is approximated as a Gaussian density using the expectation propagation method. The smoothing density computed this way is then used to prune components in the filtering density at time  $l$ . This filtering density is thereafter propagated to the future time instants  $l + 1$  until time  $k$ , each time employing the pruning strategy during filtering approximations.

Simulations show that the root mean squared error (RMSE) for the SmPDAF is better than for the PDA filter when the probability of detection is 1. However, when  $P_D < 1$ , the track loss performance of the SmPDAF algorithm deteriorates. This reduction in performance is due to the fact that the likelihood  $p(Z_{l+1:k}|x_k)$  is multi-modal when  $P_D < 1$ , and thus a Gaussian density for the likelihood can be a poor approximation. Instead, a Gaussian mixture (GM) approximation of the multi-modal likelihood would enable performance improvements. Thus, the propagated densities should be GMs instead of Gaussian densities. To this end, an understanding of the smoothing of GM densities is necessary, which lead us to Papers B and C. Naturally, a GM approximation for the filtering densities  $p(x_k|Z_{1:k})$  is justified if a GM approximation for the likelihood is used. Papers B and C present algorithms for GM filtering and smoothing based on two-filter smoothing (TFS) and forward-backward smoothing (FBS), respectively.

## Paper B

The objective of this paper is to obtain an algorithm for two-filter smoothing of GM densities based on merging approximations. The TFS involves two filters, namely the forward filter and the backward filter, where the former has been studied extensively in the literature. The latter, i.e., the backward filter, has a structure similar to a GM, but is not a normalizable density. Therefore, the traditional Gaussian mixture reduction (GMR) algorithms cannot be applied directly in the backward filter. The existing literature, though providing an analysis of the backward filter, does not present a strategy for the involved GMR. This paper presents two strategies, which allows the Gaussian mixture reduction (GMR) to be applied to the backward filter. The first one is an intragroup approximation method that depends on the structure of the backward filter, and presents a way to apply GMR within certain groups of components. The second method is a smoothed posterior pruning method, that is similar to the pruning strategy for the (forward) filtering discussed in Paper A. In paper B, the posterior pruning is formulated and proved to be a valid operation for both the forward and the backward filters. When compared to FBS based on an N-scan pruning algorithm, the two-filter smoothing densities obtained using the proposed ideas are shown to have better track-loss, RMSE and normalized estimation error squared (NEES) for lower complexity.

## Paper C

In this paper, the problem of forward-backward smoothing of GM densities based on merging is addressed. The existing literature provides practical algorithms for the FBS of GMs that are based on pruning. The drawback of a pruning strategy is that as a result of excessive pruning, the forward filtering may lead to degeneracy. The backward smoothing on this degenerate forward filter can lead to highly underestimated data association uncertainties. To overcome this, we propose using merging of the GM during forward filtering as well as during backward smoothing. As mentioned before, the forward filter based on merging is well studied in the literature. A strategy to perform the backward smoothing on filtering densities with merged components is analyzed and explained in this paper. When compared to FBS based on an N-scan pruning algorithm, the presented method is shown to have better track-loss, RMSE and NEES for lower complexity.

## Paper D

In this paper, we compare the FBS and TFS algorithms that are based on pruning and merging approximations. It is shown that both the algorithms perform very similarly in the scenarios where the target is considered to move slowly. Further, we compare and discuss the two smoothing algorithms based on the possibilities of extending the algorithms to more complex scenarios.

## 5.2. Future work

Besides the results for GM smoothing based on merging, we also obtained a plethora of ideas to investigate in the future. In this section, we present and discuss the ideas, which range from the extensions of GM smoothing to more complex scenarios than single-target linear Gaussian process models, to computationally cheaper GM merging methods and message passing in generic graphs.

### **Smoothed filtering of Gaussian mixtures**

It is possible to extend the idea of SmPDAF to use GM approximations for the densities in the PDA case, instead of the Gaussian density approximations. That is, GM approximation of the filtering density  $p(x_k|Z_{1:k})$  can be improved by better approximations of the filtering densities  $p(x_l|Z_{1:l})$ , for  $l < k$ . The approximations for  $p(x_l|Z_{1:l})$  can be improved in two steps. One is that the number of Gaussian components in  $p(x_l|Z_{1:l})$  can be reduced by smoothed posterior pruning. Two, for this reduced GM, a simpler and better Gaussian mixture approximation of  $p(x_l|Z_{1:l})$  can be obtained by employing Gaussian mixture merging. For smoothed posterior pruning of  $p(x_l|Z_{1:l})$ , one needs to compute the backward likelihood  $p(Z_{l+1:k}|x_l)$ , which in turn requires the filtering density  $p(x_l|Z_{1:l})$  for approximations. Thus, the SmPDAF approximation of  $p(x_l|Z_{1:l})$  will depend on the initial approximation of  $p(x_l|Z_{1:l})$ . Therefore, the performance gain highly depends on the approximations initially made for  $p(x_l|Z_{1:l})$ . The trade-off between the performance gain and the complexity of the SmPDAF for Gaussian mixture densities is an interesting problem to study.

### **Relation between artificial prior method and smoothed posterior pruning**

In a broad sense, there are similarities between the artificial prior method used in the TFS of particle filtering [9], and the smoothed posterior pruning idea. In both cases, the backward filter is not a density, which limits the use of traditional methods. In case of a particle filter, one needs a density to be able to sample the particles, so artificial priors are used to transform the backward filter into a density from which particles can be drawn. In the backward filter of TFS, one needs a GM density to use traditional GMR algorithms, where the forward filter is used to obtain the smoothing density so that GMR can be performed. It would be interesting to investigate further the connection between the artificial prior method and the smoothed posterior pruning method.

## Merging algorithms

The TFS and FBS algorithms presented in the thesis are based on merging. There are several methods, such as Runnalls' [35], Salmond's [36] and variants of these [11, 42], which one can choose for GM merging implementation. However, the computational complexity of these methods is a serious limitation when it comes to practical implementations where merging is necessary at each time instant. In both the reduction algorithms, the merging cost must be computed for every pair of components, which involves expensive matrix multiplications. The cost function for merging two components denoted  $w_i \mathcal{N}(x; \mu_i, P_i)$  and  $w_j \mathcal{N}(x; \mu_j, P_j)$ , are shown in the following equations for the two methods:

$$c_{Runnall}(i, j) = \frac{1}{2} ((w_i + w_j) \log |P_{i,j}| - w_i \log P_i - w_j \log P_j) \quad (5.1)$$

$$c_{Salmond}(i, j) = \frac{w_i w_j}{w_i + w_j} (\mu_i - \mu_j)^T P^{-1} (\mu_i - \mu_j). \quad (5.2)$$

Here  $P_{i,j}$  and  $P$  are the covariances of the sum of the two components and the entire GM, respectively. Next, the pair that has the lowest cost is merged and replaced in the GM, and the procedure is repeated until a stopping condition is fulfilled. Therefore, the complexity of these algorithms is quadratic, if not exponential, in the number of components, which is still expensive considering prediction, update and retrodiction steps. For the results presented in the thesis, significant amount of effort went into devising practical merging algorithms, which resulted in two strategies. One is a combination of Runnalls' and Salmond's algorithms, which is used in the forward filter. The other method is a modified version of Salmond's algorithm. There are also other ideas that we have for possible investigation, which are based on making lesser number of cost computations than computing the cost for every pair  $(i, j)$ . Some of these ideas are presented in the following discussions.

In Salmond's algorithm, the cost function has the difference between the mean terms,  $|\mu_i - \mu_j|$ . The term is also included in the  $P_{i,j}$  matrix in the Runnalls' cost function. If  $|\mu_i - \mu_j|$  is small, then  $c(i, j)$  has a small value. Therefore, sorting the components according to  $|\mu_i - \mu_j|$  can help us in easily finding a good pair that has small cost. This sorting and searching has to be performed after each pair that is merged and replaced in the GM. The so-called 'kd-tree' data structure used in computer science is a good tool in performing such a multidimensional sorting and searching efficiently, in  $N \log N$  steps, instead of  $N^2$  steps. Further investigation of the optimality of this method is interesting and can make the GM merging computationally cheaper.

Another way of reducing the number of merging cost computations is by getting bounds on the cost function. Suppose there is an upper bound on the least possible cost. And suppose that for some group of pairs of components, we can compute a lower bound on the costs. If the group's lower bound is greater than the upper

bound on the lowest cost, the cost computation for the component pairs in the group can be avoided. The challenge is thus in obtaining the upper bound on the least cost, and selecting the group that can be eliminated. A closer analysis of the cost function is necessary to obtain these bounds and a good choice of groups.



# Bibliography

- [1] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Dover Publications, 2012.
- [2] Y. Bar-Shalom and A. Jaffer, “Adaptive nonlinear filtering for tracking with measurements of uncertain origin,” in *Proceeding of the elevneth International Conference on Decision and Control and Symposium on Adaptive Processes*, 1972, pp. 243–247.
- [3] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.
- [4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [5] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
- [6] I. Bilik and J. Tabrikian, “Maneuvering target tracking in the presence of glint using the nonlinear gaussian mixture kalman filter,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 246–262, 2010.
- [7] S. S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” *Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.
- [8] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House Norwood, MA, 1999, vol. 685.
- [9] M. Briers, A. Doucet, and S. Maskell, “Smoothing algorithms for state-space models,” *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010. [Online]. Available: <http://dx.doi.org/10.1007/s10463-009-0236-2>
- [10] 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.
- [11] D. Crouse, P. Willett, K. Pattipati, and L. Svensson, “A look at gaussian mixture reduction algorithms,” in *Proceedings of the fourteenth International Conference on Information Fusion*, 2011, pp. 1–8.

- 
- [12] 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.
  - [13] 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.
  - [14] P. Fearnhead, D. Wyncoll, and J. Tawn, “A sequential smoothing algorithm with linear computational cost,” *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.
  - [15] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, “Multi-target tracking using joint probabilistic data association,” in *IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, vol. 19. IEEE, 1980, pp. 807–812.
  - [16] 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.
  - [17] 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.
  - [18] J. R. Hershey and P. A. Olsen, “Approximating the Kullback Leibler divergence between Gaussian mixture models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4. IEEE, 2007, pp. IV–317.
  - [19] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Dover Publications, 2007.
  - [20] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
  - [21] G. Kitagawa, “The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother,” *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 4, pp. 605–623, 1994.
  - [22] ——, “Monte carlo filter and smoother for non Gaussian nonlinear state space models,” *Journal of computational and graphical statistics*, vol. 5, pp. 1–25, 1996.
  - [23] 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.

- 
- [24] R. Krishnan, G. Colavolpe, T. Eriksson *et al.*, “Algorithms for joint phase estimation and decoding for mimo systems in the presence of phase noise,” *arXiv preprint arXiv:1312.2232*, 2013.
  - [25] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.
  - [26] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
  - [27] T. Kurien, “Issues in the design of practical multitarget tracking algorithms,” *Multitarget-Multisensor Tracking: Advanced Applications*, vol. 1, pp. 43–83, 1990.
  - [28] T. Kurien and M. Liggins, “Report-to-target assignment in multisensor multi-target tracking,” in *Proceedings of the 27th IEEE Conference on Decision and Control*. IEEE, 1988, pp. 2484–2488.
  - [29] R. P. Mahler, “Multitarget Bayes filtering via first-order multitarget moments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
  - [30] T. P. Minka, “Expectation propagation for approximate Bayesian inference,” in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.
  - [31] J. Pearl, “Reverend bayes on inference engines: A distributed hierarchical approach,” in *AAAI*, 1982, pp. 133–136.
  - [32] H. Rauch, F. Tung, and C. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA Journal*, vol. 3, pp. 1445–1450, 1965.
  - [33] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
  - [34] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted Gaussian mixture models,” *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.
  - [35] 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.
  - [36] 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.

- 
- [37] S. Sarkka, “Unscented rauch–tung–striebel smoother,” *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, 2008.
  - [38] R. A. Singer, R. G. Sea, and K. Housewright, “Derivation and evaluation of improved tracking filter for use in dense multitarget environments,” *IEEE Transactions on Information Theory*, vol. 20, no. 4, pp. 423–432, 1974.
  - [39] B.-N. Vo, B.-T. Vo, and R. P. Mahler, “Closed-form solutions to forward–backward smoothing,” *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 2–17, 2012.
  - [40] E. A. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE, 2000, pp. 153–158.
  - [41] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press, 1964, vol. 8.
  - [42] J. L. Williams and P. S. Maybeck, “Cost-function-based Gaussian mixture reduction for target tracking,” in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 2, 2003, pp. 1047–1054.

## **Part II.**

# **Included Papers**



# Paper A

## **Smoothed probabilistic data association filter**

A. S. Rahmathullah, L. Svensson, D. Svensson and P. Willett

In *Proceedings of the 16th International Conference on Information Fusion*, Istanbul, July 2013.



# Smoothed Probabilistic Data Association Filter

Abu Sajana Rahmathullah\*, Lennart Svensson\*, Daniel Svensson<sup>†</sup>,  
Peter Willett<sup>‡</sup>

\*Department of Signals and Systems, Chalmers University of Technology, Sweden

<sup>†</sup>Electronic Defence Systems, SAAB AB, Sweden

Emails: {sajana, lennart.svensson, daniel.svensson}@chalmers.se

<sup>‡</sup>University of Connecticut, USA

Email:willett@engr.uconn.edu

## Abstract

This paper presents the Smoothed Probabilistic Data Association Filter (SmPDAF) that attempts to improve the Gaussian approximations used in the Probabilistic Data Association Filter (PDAF). This is achieved by using information from future measurements. Newer approximations of the densities are obtained by using a combination of expectation propagation, which provides the backward likelihood information from the future measurements, and pruning, which uses these backward likelihoods to reduce the number of components in the Gaussian mixture. Performance comparison between SmPDAF and PDAF shows us that the root mean squared error performance of SmPDAF is significantly better than PDAF under comparable track loss performance.

## Index Terms

PDA, filtering, smoothing, factor graph, Gaussian mixtures, message passing, expectation propagation, target tracking, pruning.

## I. INTRODUCTION

In target tracking, the sensors pick up signals not only from the target but also from unwanted sources. When tracking with radar, echoes are reflected from ground, sea, etc. which can be sources of clutter signals. Even while tracking a single target, there can be a large number of measurements observed. It can also be that, at times, the target is not observed.

Even under simple assumptions on the process and measurement model — linear with additive Gaussian noise — the problem is computationally

intractable. It is easy to show that the true posterior filtered density has the Gaussian Mixture (GM) form with an exponentially increasing number of components over time, thus making approximations inevitable. There are many suboptimal algorithms that have been proposed and been in use. The two major approaches to the problem are one that uses a random-set framework and another that uses a data association framework. Probabilistic Hypothesis Density (PHD) [10] filters belong to the former category, whereas the Multiple Hypothesis Tracking (MHT) filters [4, 13] and the PDAFs [3] belong to the latter. This paper focuses on the data association setting; it should be pointed out that the solution presented in this paper is extendable to scenarios with GMs in general.

In the data association setting, multiple hypotheses are formed, one for each data-to-target association. Under each hypothesis, and under linear-Gaussian assumptions, the filtered posterior is a Gaussian density, thus resulting in a GM for the overall filtered posterior density. This mixture, when propagated through time, has exponentially increasing number of components. Thus, the number of components in the GM has to be reduced. There are several GM reduction algorithms [14, 15, 16] that can be employed to reduce the number of components in the GM. These algorithms vary in terms of complexity, the divergence measure that is minimized, the number of components retained, etc. [7].

MHT uses pruning and/or merging techniques to keep the number of terms under control. The GM reduction at a certain time instant is based on the information from future measurements. But the complexity involved is typically exponential. Based on the parameters of choice, MHT can be quite close to the optimal solution, but the closer it is to the optimal solution, the higher the computational complexity. In PDAF, the GM posterior is reduced to one single Gaussian which has the least Kullback-Leibler (KL) divergence [9] from the mixture. Thus, the PDAF has linear complexity in the number of measurements received.

In this paper, we present the SmPDAF algorithm that uses a combination of PDAF, Expectation Propagation (EP) [11] and pruning to reduce the GM to a single Gaussian density. The approximation is different from the PDAF in the way that before reducing the GM to a single Gaussian, SmPDAF tries to improve the approximation of the prediction density. Likelihood from the future measurements are used to improve the approximations made for the GMs in the past time instants. This improves the prediction, thereby improving the current posterior. This involves iterating between filtering and smoothing [12]. EP is used in obtaining the likelihood from the future measurements, and these likelihoods are used to prune components in the

---

GM of the filtered density. The complexity of the algorithm is linear in the number of measurements and in the depth (or the lag) involved in the smoothing.

The proposed algorithm is compared with PDAF. The two performance measures that are used for comparison are the root mean squared error (RMSE) and the track loss (TL). It will be later shown in the results that although TL is almost the same for both PDAF and SmPDAF, the latter does show a significant improvement in the RMSE performance. The performance has been compared for varying measurement noise levels and probabilities of detection. Results are shown for a single target scenario. A comparison of complexities of PDAF and SmPDAF is also discussed in the results section.

The layout of the paper is as follows: Section II describes the model assumptions, the clutter measurement distribution, etc. The problem statement and a brief description of the idea behind the SmPDAF algorithm are also presented in this section. Section III gives a comprehensive background on filtering, PDAF, smoothing and EP. The SmPDAF algorithm is described in detail in Section IV. It details on how EP helps us to obtain the likelihoods from the future measurements. Also explained is how pruning exploits these backward likelihoods to improve the approximation of the GM densities. An algorithmic description of SmPDAF is also provided. Section V presents the results and a discussion on the interpretation of the performance for varying measures. Appendices A and B throw more light into the details of Gaussian divisions involved in EP and the backward predictions, respectively.

## II. PROBLEM FORMULATION AND IDEA

### A. Model assumption

The state at time  $k$  is given by the process model,

$$x_k = Fx_{k-1} + v_k$$

and the target measurement at time  $k$  is described by the measurement model,

$$z_k^t = Hx_k + w_k$$

where  $x_k \in \mathbb{R}^M$ ,  $z_k^t \in \mathbb{R}^N$ ,  $F \in \mathbb{R}^{M \times M}$ ,  $H \in \mathbb{R}^{N \times M}$ ,  $v_k \sim \mathcal{N}(0, Q)$  and  $w_k \sim \mathcal{N}(0, R)$ .

In this paper, we only consider single-target scenarios. On top of the target-generated measurement, there are also spurious measurements due to clutter. The collection of measurements at time  $k$  is represented by the set  $Z_k$  and the collection of measurement sets from time 1 to  $k$  is given by  $Z_{1:k}$ . The clutter measurements are assumed to be uniformly distributed over

an observed region of volume  $V$ , and independent of the target state. The number of clutter measurements is also a random variable that follows a Poisson distribution with parameter  $\lambda V$ , where  $\lambda$  is the clutter density. Also, it is assumed that the initial prior,  $p(x_1)$ , is known and Gaussian.  $P_D$  will be the probability of detection.

### B. Problem statement and Idea

Under the above-mentioned assumptions, it is easy to show that the filtered posterior density  $p(x_k|Z_{1:k})$  is a GM with exponentially increasing number of components. The goal of this paper is to find a better approximation for the filtered density,  $p(x_k|Z_{1:k})$ .

The filtered posterior density is given by

$$p(x_k|Z_{1:k}) \propto p(x_k|Z_{1:k-1})p(Z_k|x_k) \quad (1)$$

where  $p(x_k|Z_{1:k-1})$  is the prediction density and  $p(Z_k|x_k)$  is the likelihood from measurements observed at  $k$ . Any improvements on the approximation of  $p(x_k|Z_{1:k-1})$  will help in improving the approximation of  $p(x_k|Z_{1:k})$ . This is achieved by improving the approximations of the filtered densities  $p(x_l|Z_{1:l})$  at the past time instants  $l < k$ , using the backward likelihood messages  $p(Z_{l+1:k}|x_l)$  from the future measurements. These likelihoods are obtained using EP and are passed backward in time for a fixed number of time steps. The backward likelihoods available at each of these time instants  $l$ , are used for pruning some of the components in the GM corresponding to the filtered density  $p(x_l|Z_{1:l})$ .

## III. BACKGROUND

The focus of the paper is on improving the approximation for filtering. To achieve, this we use smoothing along with EP. A brief background on filtering, smoothing, PDAF and EP is provided in this section. For a more thorough explanation on these topics, readers are referred to [3], [12] and [11].

### A. Filtering and PDAF

Filtering, in short, is combining the inference about the current state variable obtained from the past measurements along with the inference made from the current measurements to obtain the filtered posterior density of the state variable. This is performed in a recursive manner so that only the filtered density of the state at past time instant can be used along with the current measurement to obtain the filtered density of the current state variable; not

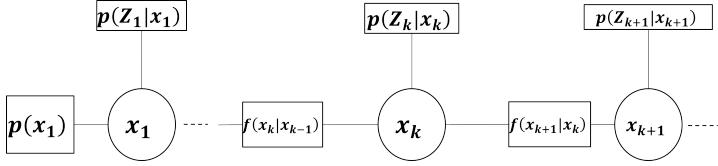


Fig. 1: State space model for filtering and smoothing.

all the past measurements have to be retained. But these operations are not always easy; the densities are complicated and it becomes computationally intractable to use recursion. Approximations are necessary. The PDAF is a filtering algorithm that uses Gaussian approximation at each time instant. The section provides some details on how this is performed. Also presented in this section is a means of involving future measurements to current inference: that is, a smoothed estimate.

Let the state transition density be  $f(x_k|x_{k-1})$  and the measurement likelihood be  $p(Z_k|x_k)$ . This state space model is illustrated in Fig. 1.

Filtering comprises two steps: prediction and update. Prediction is given by

$$p(x_k|Z_{1:k-1}) = \int p(x_{k-1}|Z_{1:k-1}) f(x_k|x_{k-1}) dx_{k-1} \quad (2)$$

In this equation, the filtered density at time  $k-1$  is propagated through the process model to obtain the prediction of the current state variable  $x_k$  from the past measurements  $Z_{1:k-1}$ . The prediction step is followed by the update step — the information from the current likelihood is updated along with the prediction to obtain the filtered density.

$$p(x_k|Z_{1:k}) \propto p(x_k|Z_{1:k-1}) p(Z_k|x_k). \quad (3)$$

The two components in equation (3) can be interpreted as:

- $p(x_k|Z_{1:k-1})$  is the information that the measurements from the past,  $Z_{1:k-1}$ , provide us about the current state  $x_k$ .
- $p(Z_k|x_k)$  is the information that the current measurement  $Z_k$  has about the current state  $x_k$ .

The update step puts together these pieces of information to give us the filtered posterior  $p(x_k|Z_{1:k})$ .

Under the model assumptions made in section II-A, the likelihood  $p(Z_k|x_k)$  is a GM.

$$p(Z_k|x_k) = w_0 + \sum_{i=1}^{m_k} w_i \mathcal{N}(Hx_k; z_{k,i}, R) \quad (4)$$

where  $\{z_k^i\}_{i=1,\dots,m_k}$  is the set of all measurements at time instant  $k$ ,  $w_i \propto \frac{1}{m_k}$ , for  $i = 1, \dots, m_k$  is the probability of the hypothesis that the measurement  $z_k^i$  correspond to the target, and  $w_0 \propto 1 - P_D P_G$  is the probability of the null hypothesis i.e., that the target did not yield any measurement.  $P_G$  is the probability of gating.

With a likelihood as in equation (4) and a Gaussian initial prior, the filtered density will be a Gaussian mixture. As this is propagated through time, all the filtered densities will be GMs and the number of terms in the GM grows exponentially with time. Thus, approximation of the GM is necessary. The PDAF approximates the filtered posterior,  $p(x_k|Z_{1:k})$ , with a single Gaussian. Assuming that the prediction  $p(x_k|Z_{1:k-1})$  is a Gaussian density  $\mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1})$  (which is typically due to the approximations made at the previous time instants), the filtered posterior will be:

$$\begin{aligned} p(x_k|Z_{1:k}) &\propto p(x_k|Z_{1:k-1}) p(Z_k|x_k) \\ &= w_0 \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) \\ &\quad + \sum_{i=1}^{m_k} w'_i \mathcal{N}(x_k; \hat{x}_{k|k,i}, P_{k|k,i}) \end{aligned} \quad (5)$$

which is again a GM with the same number of components as  $p(Z_k|x_k)$  but with updated means  $\hat{x}_{k|k,i}$ , covariances  $P_{k|k,i}$  and weights  $w'_i$ . The new weights  $w_0$  and  $w'_i$  are such that the weights shift towards the most likely component in the GM. The PDAF makes the following approximation of the GM:

$$\tilde{p}(x_k|Z_{1:k}) = \text{MM}(p(x_k|Z_{1:k})) \quad (6)$$

where  $\text{MM}(\psi(x))$  stands for moment matching that would return a Gaussian with the same first two moments as  $\psi(x)$ . This method minimizes the KL divergence,  $\text{KL}(p(x_k|Z_{1:k}) || \tilde{p}(x_k|Z_{1:k}))$ .

### B. Smoothing

Referring back to equation (3), if along with the information, if the information from the future measurements is also incorporated, that gives us the smoothed density. Let us consider smoothing for time instant  $l$  such that  $l < k$ . The filtered density at  $l$  will be

$$p(x_l|Z_{1:l}) \propto p(x_l|Z_{1:l-1}) p(Z_l|x_l).$$

The smoothed density at  $l$  will be

$$p(x_l|Z_{1:k}) \propto p(x_l|Z_{1:l-1}) p(Z_l|x_l) p(Z_{l+1:k}|x_l). \quad (7)$$

---

$p(Z_{l+1:k}|x_l)$  is the likelihood of the current state  $x_l$  from the future measurements  $Z_{l+1:k}$ . This likelihood is obtained in two steps, which are similar to the prediction and update step, but in a different order:

- Likelihood update:

$$p(Z_{l+1:k}|x_{l+1}) = p(Z_{l+1}|x_{l+1})p(Z_{l+2:k}|x_{l+1})$$

- Backward prediction:

$$p(Z_{l+1:k}|x_l) = \int p(Z_{l+1:k}|x_{l+1}) f(x_{l+1}|x_l) dx_{l+1} \quad (8)$$

### C. EP

In the algorithm presented in this paper, to compute the backward likelihood  $p(Z_{l+1:k}|x_{l+1})$ , EP [11] is used. In section IV-A, it will be explained how and why EP is employed to obtain an approximation of the backward likelihood. In this section, the general working of EP will be explained. EP is an approximation technique that can be used to obtain a Gaussian approximation of factors of a function. If the function is a product of several non-Gaussian factors, then a Gaussian approximation of each of the factors is obtained using the information from the other factors. This can be performed in an iterative fashion among all the factors in the function. This can also be used to get a Gaussian approximation of the function.

Let  $f(x)$  be a function that is made up of several factors  $f_i(x)$ . These factors are non-Gaussian and we would like to obtain a Gaussian approximation of it.

$$f(x) = \prod_{i=1}^N f_i(x).$$

Let us assume that we have an initial approximation of the factors,  $\tilde{f}_i(x)$ . We are interested in obtaining the approximation for the  $j^{th}$  factor  $f_j(x)$ . The new approximation provided by EP will be

$$\tilde{f}_j^{new}(x) = \frac{\text{MM} \left( f_j(x), \prod_{\substack{i=1 \\ i \neq j}}^N \tilde{f}_i(x) \right)}{\prod_{\substack{i=1 \\ i \neq j}}^N \tilde{f}_i(x)}.$$

The  $j^{th}$  factor is replaced with this new approximation and the procedure can be iterated for the remaining factors. The advantage is that the approximation of  $f_j(x)$  is made in the region where the product of all the factors has its mass. In the next section, there will be more explanation on the relevance of this region of interest.

#### IV. SMPDAF

The main goal of this paper is to obtain an accurate approximation of the filtered density. SmPDAF tries to make the approximation accurate in the region of interest. In this section we argue that the region of interest is where the smoothed density has its mass, which hence depends on all the available information. An important consequence is that future measurements influence where the density approximations should be accurate, i.e.,  $Z_{l+1:k}$  influence where our approximation of  $p(x_l|Z_{1:l})$  should be accurate, in order to yield an accurate approximation of  $p(x_k|Z_{1:k})$  (which is our objective). To be noted is that if we have an exact, and Gaussian, expression for the filtered density  $p(x_l|Z_{1:l})$ , in the first place, the information from the future is not going to change anything in the filtered density. The idea is to use the older approximations of the filtered density and the information from the future to obtain a more accurate approximation of the filtered density.

Under the model assumptions given in Section II-A, it was shown very briefly in Section III-A that the true filtered density  $p(x_k|Z_{1:k})$  is a GM. A PDAF makes a Gaussian approximation (by moment matching) of the filtered density at each stage. This filtered density depends on the prediction density  $p(x_k|Z_{1:k-1})$ , which in turn depends on (the approximation of) the filtered densities,  $p(x_l|Z_{1:l})$  at the past time instants  $l < k$ . Each of this filtered density is originally obtained through a PDAF. Thus before MM, each of these is a GM but with far lesser number of components than the true filtered density.

The idea is to improve on the GM in PDAF before MM. By improve, we mean that we want the GM to become more unimodal, with the mode corresponding to the true hypothesis gaining more weight. EP and pruning are used to perform this. This involves computing the backward likelihoods  $p(Z_{l+1:k}|x_l)$  ( $l < k$ ) from the future measurements, which again for computational simplicity and tractability, has to be approximated as a Gaussian density. EP is one way to obtain these backward likelihoods. It uses the prediction  $p(x_l|Z_{1:l-1})$  and the current likelihood  $p(Z_l|x_l)$  to approximate the backward likelihood at the previous time step. Based on this backward likelihood, we use pruning in order to remove components in the GM before MM.

---

Both EP and pruning use all the available information — prediction, current likelihood and the backward likelihood — to improve on the approximation of the backward likelihood and thereby, the approximation of the filtered density,  $p(x_l|Z_{1:l})$ . This will later be propagated forward to improve the filtered density  $p(x_k|Z_{1:k})$ . Thus, it becomes quite natural to make the approximation of these densities accurate in the region which uses the information from all measurements  $Z_{1:k}$ . Since the smoothed density is obtained using all the information, the region of interest is then the region where the smoothed density has most of its mass. EP and pruning provide accurate approximations of the backward likelihood and the filtered density in this region of interest. In the following sections, there will be more detailed discussions on the working of EP and pruning.

### A. EP

It was argued in the above section that it is crucial to obtain an accurate Gaussian approximation of the backward likelihood  $p(Z_{l+1:l}|x_l)$ . This can be obtained in two steps:

- 1) Starting with the Gaussian likelihood  $p(Z_{l+2:k}|x_{l+1})$  and the GM likelihood  $p(Z_{l+1}|x_{l+1})$  at time  $l + 1$ , EP is used to obtain a Gaussian approximation of  $p(Z_{l+1:l}|x_{l+1}) = p(Z_{l+1}|x_{l+1})p(Z_{l+2:k}|x_{l+1})$ ,
- 2) The Gaussian approximation from 1) is passed backwards through the process model to obtain the necessary likelihood  $p(Z_{l+1:l}|x_l)$ . (This step is referred as backward prediction in this paper).

It is emphasized that starting from time  $l + 1$ , these above two steps can be repeated for each  $l$ . The details of the backward prediction are provided in Appendix B. To be noted is that given the Gaussian density input, there are no additional approximations made in this backward prediction step.

For notational simplicity and without loss of generality, in this section we will discuss the possibilities to obtain the likelihood  $p(Z_{l:k}|x_l)$ , instead of  $p(Z_{l+1:k}|x_{l+1})$ , where

$$p(Z_{l:k}|x_l) = p(Z_l|x_l)p(Z_{l+1:k}|x_l). \quad (9)$$

The current likelihood has the form,

$$p(Z_l|x_l) = w_0 + \sum_{i=1}^{m_k} w_i \mathcal{N}(Hx_l; z_{l,i}, R).$$

If  $p(Z_{l+1:k}|x_l)$  is a proper Gaussian density, the product will yield a GM which can be moment matched and approximated to a Gaussian density. On the other hand, if  $p(Z_{l+1:k}|x_l)$  is of the form  $\mathcal{N}(Ux; \mu, P)$  where the

matrix  $U$  is not a square matrix, then the product in equation (9) will result in a function of the form,

$$p(Z_{l:k}|x_l) = w_0 \mathcal{N}(Ux_l; \mu, P) + \sum_{i=1}^{m_l} w'_i \mathcal{N}(U'x_l; \mu'_i, P').$$

This function is not a density in  $x_l$  and cannot be normalized. It is therefore difficult to find an approximation that is Gaussian-like and has the same moments. EP comes to the rescue. EP suggests that the prediction density can be used in obtaining the approximation. It serves two purposes: 1) the approximation is made accurate in the region where the smoothed density has most of its mass, and 2) it overcomes the problem of normalization and computing the moments. EP does the following:

$$p(Z_{l:k}|x_l) \approx \frac{\text{MM}(p(x_l|Z_{1:l-1})p(Z_l|x_l)p(Z_{l+1:k}|x_l))}{p(x_l|Z_{1:l-1})}. \quad (10)$$

The likelihood  $p(Z_{l:k}|x_l)$  is first multiplied by the prediction  $p(x_l|Z_{1:l-1})$ , to get the smoothed density  $p(x_l|Z_{1:k})$ . This will result in a GM which can be normalized. A Gaussian approximation of this density is made by moment matching. And the resulting Gaussian density is divided by the prediction to get an approximation of the desired likelihood  $p(Z_{l:k}|x_l)$ . This approximation for  $p(Z_{l:k}|x_l)$  is obtained so that the Kullback-Leibler divergence between the moment matched smoothed density,  $p(x_l|Z_{1:l-1})p(Z_{l:k}|x_l)$  and the GM smoothed density  $p(x_l|Z_{1:l-1})p(Z_l|x_l)p(Z_{l+1:k}|x_l)$  is minimized, i.e.,

$$\begin{aligned} p(Z_{l:k}|x_l) &\approx \underset{p(Z_{l:k}|x_l)}{\operatorname{argmin}} \text{KL} \{ &p(x_l|Z_{1:l-1})p(Z_l|x_l)p(Z_{l+1:k}|x_l) \\ &|| p(x_l|Z_{1:l-1})p(Z_{l:k}|x_l) \} . \end{aligned}$$

Thus the approximation is made such that the smoothed density is more accurate and thus yields an accurate approximation for the backward likelihood. This likelihood can be propagated back through the process model to obtain  $p(Z_{l:k}|x_{l-1})$ ; the details of this are in n Appendix B.

It can be observed in equation (10) that EP involves division of Gaussian densities. This operation is carried out similar to the Gaussian multiplication operation. There will be addition of covariance matrices in Gaussian multiplication to obtain the resultant (product) covariance. In case of Gaussian division, there is subtraction of covariance matrices. The problem is that this may result in a matrix with non-positive eigenvalues. The numerator in the Gaussian division is obtained after moment matching. Thus the covariance of the numerator will be much wider because of the spread of means term involved in moment matching a GM. If the covariance of the numerator is

---

wider in certain dimensions compared to the covariance of the denominator, the subtraction of covariance matrices will result in a matrix with negative eigenvalues. The approach taken in overcoming this is to use only the dimensions along which the eigenvalues are positive and to ignore the remaining dimensions. The details of how this is performed is explained in Appendix A.

### B. Pruning

EP provides us with a good approximation to the backward likelihood  $p(Z_{l+1:k}|x_l)$ . The question now is how this backward likelihood can be used to obtain an approximation of the filtered density  $p(x_l|Z_{1:l})$ .

One natural way of utilizing the backward likelihood would be to use EP as

$$p(x_l|Z_{1:l}) \approx \frac{\text{MM}(p(x_l|Z_{1:l-1})p(Z_l|x_l)p(Z_{l+1:k}|x_l))}{p(Z_{l+1:k}|x_l)}.$$

The problem with this method is that it involves Gaussian division and can result in a density with infinite covariance along certain state dimensions, which is not desirable.

Another way would be to use the smoothed density as an approximation,

$$p(x_l|Z_{1:l}) \approx \text{MM}(p(x_l|Z_{1:l-1})p(Z_l|x_l)p(Z_{l+1:k}|x_l)).$$

The disadvantage with this would be that as we iterate between the approximations of the backward likelihood  $p(Z_{l+1:k}|x_l)$  and the filtered density  $p(x_l|Z_{1:l})$ , the information in  $p(Z_{l+1:k}|x_l)$  will be overused (this is sometime referred to as data “incest”) and can result in an underestimated covariance.

The method that is suggested in this paper is to perform pruning on the filtered GM  $p(x_l|Z_{1:l})$  based on the smoothed density which uses the backward likelihood  $p(Z_{l+1:k}|x_l)$ . The filtered density and the smoothed density are GMs (but we assume that the prediction and the backward likelihood are Gaussian functions). Let them be denoted as:

$$\begin{aligned} p(x_l|Z_{1:l}) &= \sum_{i=0}^{m_l} w_i \mathcal{N}(x_l; \mu_l^i, P_l^i) \\ p(x_l|Z_{1:k}) &= \sum_{i=0}^{m'_l} w'_i \mathcal{N}(x_l; \eta_l^i, B_l^i) \end{aligned}$$

The different components in the GM correspond to the hypothesis that the measurement  $z_l^i$  at time  $l$  is due to the target, and  $i = 0$  refers to a missed detection. If the weights of some of the components in the GM  $p(x_l|Z_{1:k})$

are negligible compared to the weights of the filtered density GM, then it would imply that the backward likelihood indicates that the impact of those components become insignificant in the future time instants. If the impact of those components is negligible, these components may as well be pruned in the filtered GM density at time  $l$ . This helps in reducing the number of components in the GM. The lesser the number of components in the GM, the easier (or more accurate) can be the approximation of the GM to single Gaussian density. Thus, a better and an easier approximation for the filtered density  $p(x_l|Z_{1:l})$  would be to prune those components and moment-match the remaining components. Thus the idea is to prune the components in the filtered density GM that have negligible weights (or insignificant) in the smoothed density. The remaining components in the filtered density are moment-matched to obtain a Gaussian density approximation.

The pruning operation will be denoted as ‘Prune<sub>1:k</sub>’ in the algorithmic description 1.

### C. Algorithmic Description

An algorithmic description of SmPDAF is provided as Algorithm 1. It is assumed that the lag parameter  $L$  and the iteration parameter,  $J$  are defined. For each time instant,  $k$ , the procedure is explained in Algorithm 1. The filtered density at time  $k - 1$  is assumed to be available.

## V. RESULTS

Performance comparisons between SmPDAF and PDAF are presented in this section. Simulation comparisons of the root mean squared error and the track loss between the PDAF and SmPDAF are presented. The asymptotic complexity of the SmPDAF algorithm is discussed later.

### A. Root Mean Squared Error and Track Loss

The model considered for simulations is the constant velocity model with a two dimensional state vector containing position and velocity. The trajectory was generated with zero process noise. The initial speed was 50m/s. For the simulations, the acceleration noise had a standard deviation of around 4m/s<sup>2</sup>. The lag parameter  $L$ , for the SmPDAF was set as 5 time steps and for each time instant, the SmPDAF was performed once, i.e., the iteration parameter  $J$  was set as 1. The performance measures considered are RMSE and TL. The results are averaged over 1000 Monte Carlo iterations.

The RMSE performance is shown in Fig. 2 and Fig. 3. The RMSE in the position obtained with PDAF is 10% higher than that of SmPDAF for higher

---

**Algorithm 1** Algorithmic description of SmPDAF

---

- 1) Propagate the prior through the process model to get the prediction  $p(x_k|Z_{1:k-1})$  as in equation (2), compute the filtered density at  $k$  as in equation (5) and (6) and set  $b = k$ .
- 2) At  $b$ , compute the backward likelihood as in equation (10) with  $l = b$  and propagate it backward through the process model as in equation (8) with  $l = b - 1$ . Set  $b = b - 1$ .
- 3) Repeat step 2) until  $b = k - L$  and set  $f = b$ .
- 4) At  $f$ , compute the new prediction as in equation (2) and perform pruning on filtered density as

$$p(x_f|Z_{1:f}) = \text{Prune}_{1:k}(p(x_f|Z_{1:f-1}) p(Z_f|x_f)),$$

perform moment matching as

$$p(x_f|Z_{1:f}) = \text{MM}(p(x_f|Z_{1:f}))$$

and set  $f = f + 1$ .

- 5) Repeat step 4) until  $f = k$ .
  - 6) Repeat steps 2) to 5) the specified iteration ‘ $J$ ’ number of times.
- 

$P_D$  (See Fig. 2); for lower  $P_D$ , PDAF has 5% more RMSE in position than SmPDAF. With respect to the RMSE in velocity, PDAF has around 2% more RMSE than SmPDAF.

The TL decision was made by thresholding the Mahalanobis distance between the estimated state and the true state. The TL of SmPDAF is slightly better than the TL of the PDAF for higher  $P_D$  (See Fig. 4). For lower  $P_D$ , SmPDAF has higher track loss than PDAF. A closer analysis showed that the tracks were lost when the approximation of the backward likelihood made by EP was not accurate. This mainly happened when the target measurements were missed and were not part of the backward likelihood.

### B. Complexity

The complexity of SmPDAF is analyzed as a function of the number of terms in the GM filtered density. It will be argued that smoothing back until a certain lag and filtering forward have same complexity (asymptotically).

At each time instant, SmPDAF performs PDAF and smoothing back until a certain time step, followed by forward filtering employing pruning. Thus, the complexity of SmPDAF can be grouped into three parts: PDAF (Step 1 in Algorithm 1), smoothing back until a certain lag (Step 2), forward filtering with pruning (Step 4).

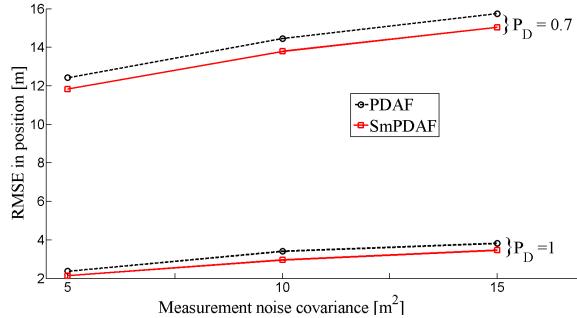


Fig. 2: Comparison of the RMSE in position between SmPDAF and PDAF

PDAF involves MM of the filtered GM density at each time instant. Thus PDAF has linear complexity in the number of components in the GM, which depends on the number of clutter measurements at each time instant. On an average, PDAF has a complexity linear in the expected number of measurements at a time instant, let's say ' $C$ '.

The smoothing step is very similar to the PDAF step. In smoothing, there is the GM backward likelihood instead of a GM filtered density; instead of direct MM, EP is performed. EP has an additional step of Gaussian division, which is independent of the number of components in GM. Thus, following a similar argument as before, smoothing for a time step has linear complexity in the expected number of measurements,  $C$ . This smoothing back is performed until certain lag, say ' $L$ '. Thus, this results in a complexity of  $L \times C$  for the smoothing step.

The filtering forward with pruning step is again very similar to PDAF, with additional pruning happening before MM. Thus, filtering forward with pruning for ' $L$ ' times steps has a complexity of  $L \times C$ . The smoothing and the filtering forward with pruning steps are repeated ' $J$ ' number of times (Step 6)). Putting all these together, SmPDAF has a complexity of  $(2 \times L \times J + 1)C$  for each time step, which is asymptotically linear in  $L \times C \times J$ .

## VI. CONCLUSIONS

In this paper, the Smoothed Probabilistic Data Association Filter algorithm was presented. The core idea of the algorithm is to exploit the information from the future to improve on the Gaussian approximation of the filtered density at the current time instant. The work done in this paper presented a way to use the idea of a combination of the probabilistic data association filter (PDAF), expectation propagation (EP) and pruning. To compute the

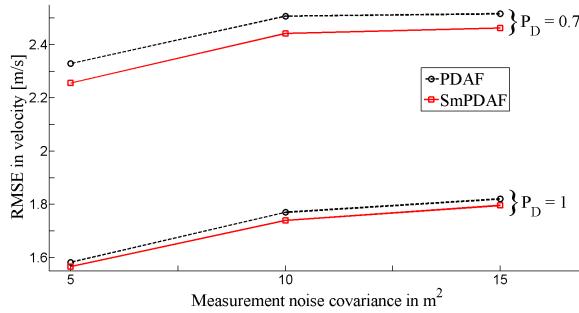


Fig. 3: Comparison of the RMSE in velocity between SmPDAF and PDAF

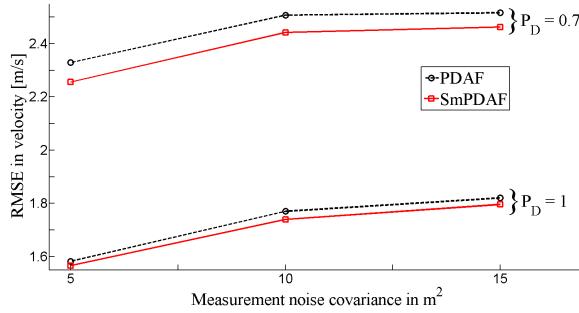


Fig. 4: TL comparison between SmPDAF and PDAF

backward likelihood from future measurements, EP was employed. Pruning was a way suggested to use this backward likelihood to improve the approximation of the filtered density. Both EP and pruning work in such a way that the approximations are made accurate in the region of interest. The RMSE performance of the SmPDAF algorithm was found to be significantly better than that of PDAF in a simulated single-target scenario with clutter and non-unity probability of detection.

## APPENDIX A GAUSSIAN DIVISIONS

When computing the backward likelihoods from the future measurements, EP is used as in equation (10). This involves division of Gaussian densities that leads to subtracting the inverses of the two covariance matrices, which may result in covariance matrices with negative eigenvalues. One way to overcome this is to exploit the dimensions along which the eigenvalues are positive, and propagate the corresponding density.

Let us assume that the densities involved in the Gaussian division are  $\mathcal{N}(x; \mu_a, P_a)$  and  $\mathcal{N}(x; \mu_b, P_b)$  and the interest is in computing the function  $g(x)$  such that it has a Gaussian form with positive definite covariance matrix.

$$\begin{aligned} g(x) &\propto \frac{\mathcal{N}(x; \mu_a, P_a)}{\mathcal{N}(x; \mu_b, P_b)} \\ &\propto \exp \left\{ -\frac{1}{2} [x^T (P_a^{-1} - P_b^{-1}) x \right. \\ &\quad \left. - 2x^T (P_a^{-1}\mu_a - P_b^{-1}\mu_b)] \right\}. \end{aligned}$$

Let the eigendecomposition of  $P_a^{-1} - P_b^{-1}$  be  $U^T \Lambda U$ .  $\Lambda$  is such that

$$\Lambda = \begin{bmatrix} \Lambda_p & \mathbf{0} \\ \mathbf{0} & \Lambda_{np} \end{bmatrix}$$

where  $\Lambda_p$  and  $\Lambda_{np}$  are diagonal matrices with positive eigenvalues and non-positive eigenvalues along their diagonals, respectively. The eigenvectors in the  $U$  matrix are sorted accordingly,

$$U = \begin{bmatrix} U_p \\ U_{np} \end{bmatrix}$$

and  $U_p U_p^T = I$ . The approximation made will be that  $P_a^{-1} - P_b^{-1}$  is replaced with  $U_p^T \Lambda_p U_p$ , thereby using only the dimensions along which the eigenvalues are positive. Thus,

$$g(x) \propto \mathcal{N}(U_p x; \mu_p, P_p)$$

where  $\mu_p = \Lambda_p^{-1} U_p (P_a^{-1}\mu_a - P_b^{-1}\mu_b)$  and  $P_p = \Lambda_p^{-1}$ .

## APPENDIX B BACKWARD PREDICTION

The backward likelihood functions used in SmPDAF have to be propagated back through the process model as in equation (8),

$$p(Z_{k+1:N}|x_k) = \int p(Z_{k+1:N}|x_{k+1}) f(x_{k+1}|x_k) dx_{k+1}.$$

This operation is very similar to the prediction step in filtering. The likelihood  $p(Z_{k+1:N}|x_{k+1})$  is a function obtained by using EP that involves Gaussian

---

division as in Appendix A. Thus, it will have the form  $\mathcal{N}(U_p x_{k+1}; \mu_p, P_p)$ . The state transition density is given by  $\mathcal{N}(x_{k+1}; Fx_k, Q)$ .

$$\begin{aligned} p(Z_{k+1:N}|x_k) &= \int \mathcal{N}(U_p x_{k+1}; \mu_p, P_p) \times \\ &\quad \mathcal{N}(x_{k+1}; Fx_k, Q) dx_{k+1} \\ &= \int \mathcal{N}(x_{k+1}; \bullet, \bullet) \times \\ &\quad \mathcal{N}(U_p Fx_k; \mu_p, U_p Q U_p^T + P_p) dx_{k+1}. \end{aligned}$$

The first term is a Gaussian density in  $x_{k+1}$  and integrates to unity. The second term in the above integral is independent of  $x_{k+1}$  and hence can be moved out of the integral. Thus,

$$p(Z_{k+1:N}|x_k) = \mathcal{N}(U_p Fx_k; \mu_p, U_p Q U_p^T + P_p).$$

#### REFERENCES

- [1] Y. Bar-Shalom and A. Jaffer, “Adaptive nonlinear filtering for tracking with measurements of uncertain origin,” in *Proceeding of the eleventh International Conference on Decision and Control and Symposium on Adaptive Processes*, 1972, pp. 243–247.
- [2] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *Control Systems, IEEE*, vol. 29, no. 6, pp. 82–100, 2009.
- [3] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
- [4] S. S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, 2004.
- [5] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House Norwood, MA, 1999, vol. 685.
- [6] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, “Multi-target tracking using joint probabilistic data association,” in *IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, vol. 19. IEEE, 1980, pp. 807–812.
- [7] J. R. Hershey and P. A. Olsen, “Approximating the Kullback Leibler divergence between Gaussian mixture models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4. IEEE, 2007, pp. IV–317.

- [8] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [9] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [10] R. P. Mahler, “Multitarget Bayes filtering via first-order multitarget moments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [11] T. P. Minka, “Expectation propagation for approximate Bayesian inference,” in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.
- [12] H. Rauch, F. Tung, and C. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA JOURNAL*, vol. 3, pp. 1445–1450, 1965.
- [13] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [14] 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.
- [15] 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.
- [16] J. L. Williams and P. S. Maybeck, “Cost-function-based Gaussian mixture reduction for target tracking,” in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 2, 2003, pp. 1047–1054.

## Paper B

# Two-filter Gaussian mixture smoothing with posterior pruning

A. S. Rahmathullah, L. Svensson and D. Svensson

To be published in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, July 2014.



# Two-filter Gaussian mixture smoothing with posterior pruning

Abu Sajana Rahmathullah\*, Lennart Svensson\*, Daniel Svensson†

\*Chalmers University of Technology, Sweden

†Electronic Defence Systems, SAAB AB, Sweden

Emails: {sajana, lennart.svensson, daniel.svensson}@chalmers.se

## Abstract

In this paper, we address the problem of smoothing on Gaussian mixture (GM) posterior densities using the two-filter smoothing (TFS) strategy. The structure of the likelihoods in the backward filter of the TFS is analyzed in detail. These likelihoods look similar to GMs, but are not proper density functions in the state-space since they may have constant value in a subspace of the state space. We present how the traditional GM reduction techniques can be extended to this kind of GMs. We also propose a posterior-based pruning strategy, where the filtering density can be used to make further approximations of the likelihood in the backward filter. Compared to the forward–backward smoothing (FBS) method based on  $N$ –scan pruning approximations, the proposed algorithm is shown to perform better in terms of track loss, normalized estimation error squared (NEES), computational complexity and root mean squared error (RMSE).

## Index Terms

filtering, smoothing, Gaussian mixtures, two-filter smoothing, backward likelihood, data association

## I. INTRODUCTION

In many sequential estimation problems, such as data association problem [1] and glint problem [2] in radar tracking and synchronization problem in communication systems, the uncertainties involved in the system are multi-modal and can be modeled using Gaussian mixtures (GMs). In these applications, the optimal solutions to the estimation problems of prediction, filtering and smoothing of GMs have closed form expressions, which can be obtained using the optimal solutions for the Gaussian densities [8, 14].

Often the optimal solutions have a complexity that increases exponentially with time. For instance, in the data association problem in target tracking, to obtain the optimal solution, each observed measurement is either associated

to a target or declared as clutter at each time resulting in sequences of data association hypotheses referred to as track hypotheses. Using these data association hypotheses during prediction, filtering and smoothing results in GM posterior densities in which the number of components grows exponentially with time.

The two main smoothing methods [6] are forward-backward smoothing (FBS) and two-filter smoothing (TFS). When the posterior densities are Gaussian, closed form solutions are available in [11] for FBS and in [7] for TFS. In this paper, we focus on the TFS method for GMs. The TFS algorithm works by running two independent filtering algorithms: the forward filtering (FF) and the backward filtering (BF). The smoothing density is obtained by multiplying the outputs of the FF and the BF. Ideally, the two filters are run independently of each other. The FF is the conventional GM filtering one (eg. a Multiple hypothesis tracking algorithm [12]). The BF, as the name suggests, is run in the opposite direction from the last time step to the start and can be obtained using a recursive procedure, similar to the FF. The likelihood returned by the BF, referred as reduced dimension GM (RDGM) in this paper, has a GM type of structure, but is not normalizable. The problem is that the traditional GM reduction (GMR) techniques, such as pruning and merging, cannot be used directly on the RDGM of the BF, as these techniques work on normalizable densities. Therefore, design of GMR techniques for the RDGM structure is necessary for a practical algorithm.

The TFS on GMs is discussed in [8], which provides the structure of the RDGM, but practical approximation techniques are not discussed in detail. In the particle filter setting of the TFS problem, the work in [4] proposes using an artificial prior in the BF, similar to the prior which we use in the FF, to make the output of the BF a proper density. The conventional techniques can then be used on the BF's output to sample particles. But extending this artificial prior concept to GM TFS has the problem that it involves division of densities, which can lead to indefinite covariances.

In this paper, we study the TFS of GM, with focus on the backward filter. Based on the analysis of the structure of the RDGM of the BF, we propose strategies where GMR techniques are used within certain groups of components of the RDGM. We also present the smoothed posterior pruning (SPP) method where we show that the filtering density from the FF can be used to approximate the RDGM. Simulations for a single target scenario show that the TFS based on proposed strategies outperforms the pruning-based FBS method. Additionally, the estimates from the TFS are consistent, with lower track loss and cheaper computational complexity.

---

## II. PROBLEM STATEMENT AND IDEA

We consider a single target moving in a cluttered background. The state vector  $x_k$  of dimension  $N_x$  at time  $k$  is according to the process model,

$$x_k = Fx_{k-1} + v_k, \quad (1)$$

where  $v_k$  is Gaussian with zero mean and covariance  $Q$  and is denoted as  $\sim \mathcal{N}(0, Q)$ . The target is detected with probability  $P_D$ . Measurement set  $Z_k$  includes the clutter measurements and the target measurements  $z_k^t$ , when detected. The target measurement is given by

$$z_k^t = Hx_k + w_k \quad (2)$$

where  $w_k \sim \mathcal{N}(0, R)$ . 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  $\beta V$ , where  $\beta$  is the clutter density. The number of measurements in  $Z_k$  is denoted by  $m_k$ . We assume that we have access to the measurements  $Z_{1:K}$  from time 1 to  $K$  where  $K \geq k$ .

The objective is to compute the smoothing density  $p(x_k|Z_{1:K})$  using the TFS method, which involves running two filters — the FF gives the filtering density  $p(x_k|Z_{1:k})$  and the BF gives the backward likelihood (BL)  $p(Z_{k+1:K}|x_k)$ . The FF is the same as the one in FBS, which is well studied in the literature. The BL, which is the output of the BF, looks similar to a GM density function but is not normalizable. Because of this structure of the likelihood, the traditional GMR techniques cannot be applied directly. The goal of this paper is to devise strategies for GMR of the BL to reduce the complexity of the BF.

### A. Idea

The BL of the BF has a mixture structure in which the components are densities in different subspaces of the state space. This structure is referred to as reduced dimension GM (RDGM) in this paper. By close investigation, one can observe that the components in the RDGM can be partitioned into groups such that the components within each group are density functions in the same subspace. Consequently, this grouping allows us to use traditional GMR techniques within each group. The details of how the intragroup approximations are performed are covered in Section IV. We also discuss the limitations of these intragroup methods.

Besides the intragroup approximations, the availability of the filtering density from the FF can be used to reduce the number of components further

in the RDGM of the BF. That is, based on the components in the smoothing density, further GMR can be performed on the RDGM, as will be shown in Section V.

### III. BACKGROUND

In this section, we present a background to the TFS method for GM densities. The conceptual solutions to the FF and the BF in the TFS are provided. The solution to the FF and BF of GM is described using a graphical illustration.

#### A. Two-filter smoothing

The goal of smoothing is to compute the smoothing posterior  $p(x_k|Z_{1:K})$ . The TFS method computes the smoothing density, at each  $k$ , according to

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

The filtering density  $p(x_k|Z_{1:k})$  and the BL  $p(Z_{k+1:K}|x_k)$  are obtained as outputs of the FF and the BF, respectively. It should be pointed out, that the counterpart of TFS in the factor graph world is the belief propagation algorithm [10], where the filtering steps are termed as message passing.

*1) Forward Filtering :* The FF involves two steps, namely the prediction and the update step, which are performed recursively from time  $k = 1$  to  $k = K$ . The prediction step computes the prediction density,

$$p(x_k|Z_{1:k-1}) = \int p(x_{k-1}|Z_{1:k-1})f(x_k|x_{k-1})dx_{k-1} \quad (4)$$

followed by the update step to compute the filtering density,

$$p(x_k|Z_{1:k}) \propto p(x_k|Z_{1:k-1})p(Z_k|x_k). \quad (5)$$

For the assumptions made in Section II, it can be shown that the filtering density is a GM and has the form

$$p(x_k|Z_{1:k}) = \sum_{i=1}^{M_k^f} w_{k,i}^f \mathcal{N}\left(x_k; \mu_{k,i}^f, P_{k,i}^f\right) \quad (6)$$

where  $\mathcal{N}\left(x_k; \mu_{k,i}^f, P_{k,i}^f\right)$  represents a Gaussian density in variable  $x_k$  with mean  $\mu_{k,i}^f$  and covariance  $P_{k,i}^f$  and  $w_{k,i}^f$  is the corresponding weight. The weight  $w_{k,i}^f$  in (6) is the probability  $\Pr\{\mathcal{H}_{k,i}^f|Z_{1:k}\}$  of the hypothesis  $\mathcal{H}_{k,i}^f$  which corresponds to a sequence of measurements or the missed-detection associations from time 1 to  $k$ . Kalman filtering under  $\mathcal{H}_{k,i}^f$  gives the mean  $\mu_{k,i}^f$  and the covariance  $P_{k,i}^f$ .

---

The number of components in the filtering GM is the product of the number of measurements observed over time. So, the number of components grow exponentially with time. Therefore, the number GMs have to be reduced using GMR techniques, such as pruning and merging. GMR techniques such as pruning and merging are used for practical implementations of FF. A few pruning methods to mention are gating [3],  $N$ -scan pruning, M-best pruning and threshold-based pruning. For merging algorithms, one can refer to [9], [13], [16] and [5]. The merging algorithms presented in these papers preserve the moments, i.e., the mean and covariance of the GM before and after merging are the same.

2) *Backward Filtering*: The BF in TFS also involves recursive steps, similar to the FF, from time  $k = K - 1$  to  $k = 1$ . The first step is the update step given by

$$p(Z_{k+1:K}|x_{k+1}) = p(Z_{k+1}|x_{k+1})p(Z_{k+2:K}|x_{k+1}), \quad (7)$$

followed by the retrodiction step to obtain the BL,

$$p(Z_{k+1:K}|x_k) \propto \int p(Z_{k+1:K}|x_{k+1}) f(x_{k+1}|x_k) dx_{k+1}. \quad (8)$$

For the assumptions made in Section II, similar to the FF, the BF can also be interpreted as running filtering operations under data association hypotheses. In the BF, each hypothesis, denoted  $\mathcal{H}_{k,j}^b$ , corresponds to a sequence of measurement or missed-detection associations made from time  $K$  until time  $k + 1$ , where  $k < K$ .

Assuming that there are only pruning approximations made during FF and BF, the TFS can be illustrated using graphical structures as shown in Fig. 1. Two hypothesis trees (to be precise, the BF needs a graph illustration, as there is not one root), one for each filter, are shown in where the nodes represent the components in the GM of  $p(x_k|Z_{1:k})$ . The details of the structure of the output of the BF and how the proposed algorithms can be used to make approximations in the BF are presented in the Sections IV and V.

#### IV. INTRAGROUP APPROXIMATIONS OF THE BACKWARD LIKELIHOOD

In this section, we analyze the details of the BL in the BF that has the RDGM structure. We discuss why the conventional pruning and merging strategies are not always suitable for the BL. Based on the analysis of the RDGM structure of the BL, we show that the components in the RDGM can be grouped and that the traditional GMR techniques can be used within the groups. It will also be shown that the number of groups is polynomial in the lag  $K - k$ , and that this can be a limitation of the intragroup GMR strategies, especially when the lag  $K - k$  is large.

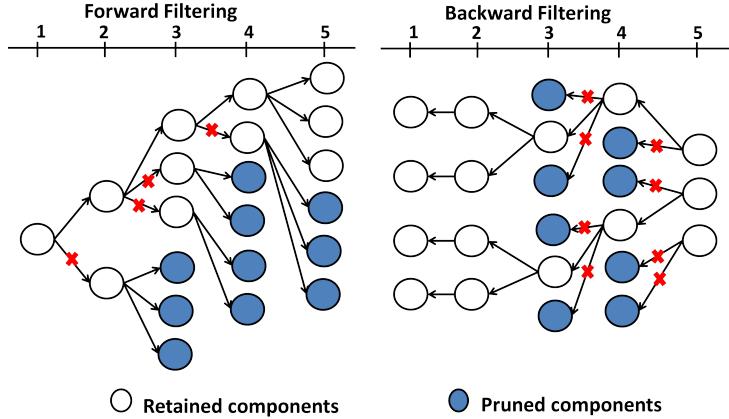


Fig. 1: Illustration of TFS: The nodes correspond to the GM components in the filter outputs. The ‘blue nodes’ represent the pruned nodes and they are not propagated further. At each time instant, the smoothing density is obtained as the product of the GMs, corresponding to all the nodes in the two filters.

### A. Structure of the backward likelihood

Under the assumptions made in Section II, starting with the likelihood of the form

$$p(Z_K|x_K) = \beta_{K,0} + \sum_{m=1}^{m_K} \beta_{K,m} \mathcal{N}(Hx_K; z_{K,m}, R) \quad (9)$$

at time  $K$  and evaluating the update and retrodiction recursions in (8) and (7), the BL can be written in the form,

$$p(Z_{k+1:K}|x_k) = w_{k,0}^b + \sum_{j=1}^{M_k^b} w_{k,j}^b \mathcal{N}\left(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b\right) \quad (10)$$

for any  $k < K$ . The parameters in (9) and (10) can be obtained using the extended observation model described in Section 3 of [8] or from Section II-C of [15]. Note that the term  $w_{k,0}^b$  is zero if the probability of detection is one. The expression in (10), which is the RDGM, is similar to a GM, with weights  $w_{k,j}^b$ , means  $\mu_{k,j}^b$  and covariances  $P_{k,j}^b$ . However, the terms  $\mathcal{N}\left(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b\right)$  are not generally densities in  $x_k$  and are not guaranteed to be normalizable since the integral  $\int \mathcal{N}\left(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b\right) dx_k$  may be infinite.

---

### B. Normalizability of the BL and intragroup approximations

Similar to the GM filtering density in the FF, the number of terms  $M_k^b$  in the RDGM in (10) of the BF grows exponentially with time. Thus, GMR approximations are necessary to reduce the complexity. The catch is that the conventional GMR techniques mentioned in Section III-A1 are developed for GM density functions and cannot be applied directly to RDGMs. For instance, the conventional pruning strategies are applied to GMs that are normalized, in which the weights of the components define the relative sizes of the Gaussian components. So, the weights of the components can be compared and the ones with insignificant weights can be removed. However, in the RDGM in (10), it is possible that a component with the smallest weight  $w_{k,j}^b$  is in fact among the largest components. So, the components in the RDGM cannot be compared based only on the weights unless the components  $\mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b)$  are normalized. Similarly, the conventional merging strategies applied to GMs involve moment matching, and to compute the moments, the components should be normalizable densities. In the trivial case, when the components in the RDGM are normalizable, i.e., when  $\text{rank}(H_{k,j}^b) = N_x$ , it is possible to rewrite the components in the RDGM  $\sum_j w_{k,j}^b \mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b)$  into the GM form  $\sum_j w_{k,j}^b \mathcal{N}(x_k; \mu_{k,j}^b, P_{k,j}^b)$  to which the GMR techniques can be applied.

The RDGM is in general not normalized, like in the data association problem, and the normalizability depends on the structure of the matrices  $H_{k,j}^b$ . As will be shown in Section IV-B1, in the RDGM, groups of components have identical  $H_{k,j}^b$  matrices. Thus, if the matrices  $H_{k,j}^b$  are also of full row rank in a group, then it is possible to compare the components within the group, and so, one can apply the GMR technique to that group.

Let  $I_g$  be the index set that contains the indices  $j$  of the components in a group  $g$  that have the same matrices, i.e.,  $H_{k,j}^b = H_g$  for  $j \in I_g$  and let  $\text{rank}(H_g) = m_g$ . Using this notation, the RDGM in (10) can be written as

$$p(Z_{k+1:K} | x_k) = \sum_{g=1}^{N_G} \sum_{j \in I_g} w_{k,j}^b \mathcal{N}(H_g x_k; \mu_{k,j}^b, P_{k,j}^b) \quad (11)$$

where  $N_G$  is the number of groups. The functions  $\mathcal{N}(H_g x_k; \mu_{k,j}^b, P_{k,j}^b)$  are scaled Gaussian functions in the row space of the matrix  $H_g$ , and have constant values in the null space of  $H_g$ . The idea is that the GMR can be applied to approximate group  $\sum_{j \in I_g} w_{k,j}^b \mathcal{N}(H_g x_k; \mu_{k,j}^b, P_{k,j}^b)$  of components in this row space and that no approximation needs to be performed in the null

space, as the values of the components are constant in the null space. An interpretation of this can be obtained using a change of basis matrix, say  $A_g = \begin{bmatrix} H_g \\ H_g^\perp \end{bmatrix}$  in  $\mathbb{R}^{N_x}$ , where the matrix  $H_g^\perp$  is such that the columns and rows of  $A_g$  span  $\mathbb{R}^{N_x}$ . Let us define a variable  $y_g = A_g x_k$  with respect to the new basis in  $\mathbb{R}^{N_x}$ . Using this variable  $y_g$ , the function  $w_{k,j}^b \mathcal{N}(H_g x_k; \mu_{k,j}^b, P_{k,j}^b)$  is written as  $w_{k,j}^b \mathcal{N}(y_g[1:m_g]; \mu_{k,j}^b, P_{k,j}^b)$ , where  $y_g[1:m_g]$  denotes the first  $m_g$  elements in the vector  $y_g$ . Clearly, functions  $\mathcal{N}(y_g[1:m_g]; \mu_{k,j}^b, P_{k,j}^b)$  are Gaussian in the variable  $y_g[1:m_g]$ . These functions are not integrable in  $y_g[m_g+1:N_x]$ , but are constant (or uniform) in that variable. This observation allows us to treat the group  $\sum_{j \in I_g} w_{k,j}^b \mathcal{N}(y_g[1:m_g]; \mu_{k,j}^b, P_{k,j}^b)$  as a GM in the variable  $y_g[1:m_g]$ . So, pruning and merging can be applied to approximate this mixture as a function of  $y_g[1:m_g] = H_g x_k$ . Overall, within each group  $g$ , the GMR can be applied as

$$\begin{aligned} & \sum_{g=1}^{N_G} \sum_{j \in I_g} w_{k,j}^b \mathcal{N}(y_g[1:m_g]; \mu_{k,j}^b, P_{k,j}^b) \\ & \approx \sum_{g=1}^{N_G} \sum_{j \in I'_g} \tilde{w}_{k,j}^b \mathcal{N}(y_g[1:m_g]; \tilde{\mu}_{k,j}^b, \tilde{P}_{k,j}^b) \\ & = \sum_{g=1}^{N_G} \sum_{j \in I'_g} \tilde{w}_{k,j}^b \mathcal{N}(H_g x_k; \tilde{\mu}_{k,j}^b, \tilde{P}_{k,j}^b) \end{aligned} \quad (12)$$

where  $I'_g$  refers to the index set corresponding to components in group  $g$  after GMR. From (12), it can be noticed that after the intragroup approximations, the number of components in the RDGM is at least the number of groups,  $N_G$ .

1) *Grouping:* As discussed earlier in this section, the grouping of the components in the BL in (10) is key in being able to apply the GMR techniques to approximate the RDGM. In this section, with the data association in target tracking as an example, we discuss why the grouping is possible. For this example, we also analyze how the number of groups grows with time to point out the limitation of the intragroup GMR methods.

By investigating (9) and (10) for the DA problem, one can see that many of the components in (10) have the same  $H_{k,j}^b$  matrices. It turns out that if two DA hypotheses,  $\mathcal{H}_{k,i}^b$  and  $\mathcal{H}_{k,j}^b$ , are such that they have measurement associations and missed-detection associations at the same time instants, then

---

$H_{k,i}^b = H_{k,j}^b$ . This observation allows us to partition the components in the RDGM into groups of components that have the same  $H_{k,j}^b$  matrices.

Consider the constant velocity model, in which the state contains the position and velocity components, and assume that only the position components are observed. One then needs measurements from at least two different time instants to estimate the velocity components. This implies that under the corresponding DA hypotheses  $\mathcal{H}_{k,j}^b$  in the BF, which have at least two measurement associations across time, the state can be estimated with finite covariance. So, the corresponding components  $\mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b)$  in the BL will be normalizable and therefore, the ranks of the corresponding matrices  $H_{k,j}^b$  is  $N_x = \dim(x_k)$ . These normalizable components will form one group. Additionally, there will also be components with  $H_{k,j}^b$  such that  $\text{rank}(H_{k,j}^b) < N_x$  which correspond to the hypothesis sequences  $\mathcal{H}_{k,j}^b$  that have less than two measurements associations across time.

We will now analyze how the number of groups grows with time. Let us assume that the matrices  $H$  and  $F$  in the motion model (1) and measurement model (2) are such that measurements from at least  $M$  different time instants (not necessarily consecutive) are needed to estimate all dimensions of the state  $x_k$ , i.e., to ensure that the corresponding component in the BL is normalizable. Using this parameter  $M$  and the time lag  $K - k$ , the number of groups can be analysed. As discussed for the constant velocity model, the components corresponding to the hypothesis sequences that have at least  $M$  measurement associations across time will be normalizable and form a group. Components that have hypothesis sequences with exactly  $l$  ( $l < M$ ) measurement associations at the same time instants belong to the same group. There are  $\binom{K-k}{l}$  ways of assigning  $l$  measurement associations in  $K - k$  time steps for each  $l = 0, 1 \dots M - 1$ . Thus, in total, the number of groups is given by

$$N_G(k, K) = 1 + \sum_{l=0}^{M-1} \binom{K-k}{l} \quad (13)$$

which grows in the order of  $\mathcal{O}((K-k)^{M-1})$  as a function of the lag  $K-k$ .

One can see that using the intragroup approximations, the number of components in the RDGM cannot be reduced to less than the number of groups, which grows as  $\mathcal{O}((K-k)^{M-1})$  with the lag  $K-k$ . Hence, these intragroup approximations are not sufficient to reduce the complexity due to the RDGM structure of the BL. It is therefore essential to present a sound manner to compare components across groups in order to enable us to reduce

the complexity further. In the next section, we present the smoothed posterior-based pruning with which the components in the BL can be compared and pruned.

## V. SMOOTHED POSTERIOR-BASED PRUNING

In this section, we show that it is valid to compare components across groups in the BL based on the smoothing probabilities of the corresponding hypotheses in the BF. Surprisingly enough, using these probabilities, large components or even groups of components can be pruned from the BL and propagated, without affecting the smoothing densities at any time instant. The smoothing probabilities of the hypotheses in the output of the BF can be calculated using the FF densities. We also show how this idea can be generalized for pruning components from the FF as well.

### A. Posterior-based pruning

The main objective is to compute the smoothing posterior density given in (3). The two terms in the right-hand side of (3) are the filtering density and the backward likelihood, whichs are the outputs of the FF and BF, respectively. The filtering density is reduced using traditional GMR techniques. The BL can be reduced using the intragroup approximations discussed before, but that may not be enough due to its limitations discussed in the previous section. We propose (and later prove) that the filtering density of the FF in (3) can be used to prune components in the BL of the BF, in regions where the smoothing posterior density is small, and therefore is not affected by this pruning. Fig. 2 shows an illustration of this strategy, referred to as smoothed posterior pruning (SPP), for one time instant. The GMs are in 2D and the curves shown in the figure are the contour plots of the GMs involved. More importantly, we can propagate the pruned version of the BL backwards in time without affecting the smoothing density at any other time instant.

The SPP idea can also be used the other way around. That is, it is possible to approximate the FF based on the BL. The BL of the BF, when available, can be used to prune components in the filtering density of the FF in the regions where the smoothing density is small. The bottom line is that the SPP idea presented in this section can be generalized to both the FF and the BF, depending on what functions are available to compute the posterior. In this section, we discuss the SPP idea for the FF, but the same arguments hold for SPP of BF as well.

For the pruning step in the SPP, any of the pruning strategies mentioned in Section III-A1 can be used, once we know the weights of the components

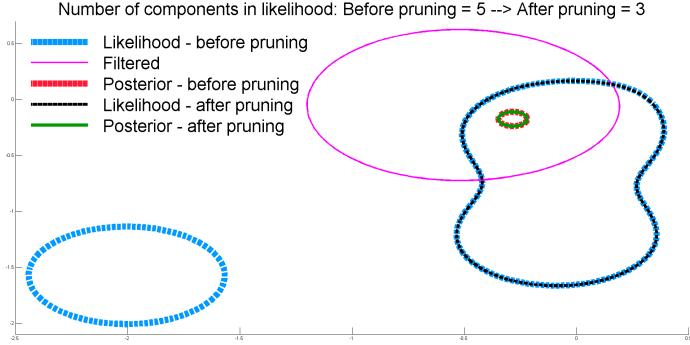


Fig. 2: Illustration of SPP: The green and the red curves, which represent the posterior density before and after the approximation of the likelihood, are very close to each other. This shows that the shape of the corresponding (smoothing) posterior is unaffected by the posterior pruning of the likelihood. Also, parts of the blue curve and black curve, which represent the likelihood before and after pruning, are very close to each other.

to be compared. Note that the weights relate to the probability of the data association hypothesis sequence  $\mathcal{H}_{k,i}^f$  (cf. Section III-A1). In SPP, as stated before, we want to perform the pruning based on the smoothing posterior. So, we do not use the filtering probability  $\Pr\{\mathcal{H}_{k,i}^f | Z_{1:k}\}$  of the hypothesis. Instead, we use the smoothing probability  $\Pr\{\mathcal{H}_{k,i}^f | Z_{1:K}\}$  of the hypothesis. As we will show in the following proposition, if the smoothing posterior probability  $\Pr\{\mathcal{H}_{k,i}^f | Z_{1:K}\}$  is zero, then the  $i^{th}$  Gaussian component can be pruned from the filtering density, without affecting the smoothing posterior distributions. The intuitive reason is that the smoothing probabilities of the offsprings of the pruned component, if it was propagated, would also be zero, which means that they would not influence the smoothing densities at later times either. In the following proposition, we also provide the mathematical justification for the same.

**Proposition 1.** Suppose that the objective is to compute the smoothing density  $p(x_l | Z_{1:K})$  for  $l = 1, \dots, K$ . If  $\Pr\{\mathcal{H}_{k,i}^f | Z_{1:K}\} = 0$ , then the  $i^{th}$  component can be pruned from  $p(x_k | Z_{1:k})$  and propagated to time  $k+1$  during FF without affecting the smoothing density  $p(x_l | Z_{1:K})$  for  $l \geq k$ .

Analogously, if  $\Pr\{\mathcal{H}_{k,j}^b | Z_{1:K}\} = 0$ , then the  $j^{th}$  component can be pruned from  $p(Z_{k+1:K} | x_k)$  and propagated to time  $k-1$  during BF without affecting the smoothing density  $p(x_l | Z_{1:K})$  for  $l \leq k$ .

*Proof:* In the following, we sketch the proof for the first part of the proposition, i.e., for SPP on the FF. It is possible to derive an analogous

proof for SPP on the BF.

Let us first consider how the suggested pruning of components in the FF affect the smoothing density  $p(x_k|Z_{1:K})$  at time  $k$ . At time  $k$ , the relation between the smoothing density  $p(x_k|Z_{1:K})$  and the smoothing probability  $\Pr\{\mathcal{H}_{k,i}^f|Z_{1:K}\}$  is given by

$$p(x_k|Z_{1:K}) = \sum_i p(x_k|Z_{1:K}, \mathcal{H}_{k,i}^f) \Pr\{\mathcal{H}_{k,i}^f|Z_{1:K}\} \quad (14)$$

$$\propto \sum_i \Pr\{\mathcal{H}_{k,i}^f|Z_{1:k}\} p(x_k|\mathcal{H}_{k,i}^f, Z_{1:k}) \\ \times p(Z_{k+1:K}|x_k) \quad (15)$$

where the product  $\Pr\{\mathcal{H}_{k,i}^f|Z_{1:k}\}p(x_k|\mathcal{H}_{k,i}^f, Z_{1:k})$  is the  $i^{\text{th}}$  component in the filtering density GM  $p(x_k|Z_{1:k})$ . Clearly, the  $i^{\text{th}}$  component in the summation of (14), is due to the product of the  $i^{\text{th}}$  component of the filtering density  $p(x_k|Z_{1:k})$  and the BL  $p(Z_{k+1:K}|x_k)$  as in (15). We note that the  $\Pr\{\mathcal{H}_{k,i}^f|Z_{1:K}\} = 0$ , for some value of  $i$ , implies that the  $i^{\text{th}}$  component can be removed from the filtering density without affecting the smoothing density  $p(x_k|Z_{1:K})$  at  $k$ .

We are now left to prove that the suggested pruning technique does not affect the smoothing density  $p(x_l|Z_{1:K})$  for  $l = k + 1, \dots, K$ . To this end, we use the fact that

$$\Pr\{\mathcal{H}_{k,i}^f|Z_{1:K}\} = \sum_{j \in S_{k \rightarrow l,i}} \Pr\{\mathcal{H}_{l,j}^f|Z_{1:K}\} \quad (16)$$

where the summation is over the set  $S_{k \rightarrow l,i}$  that contains the indices of the components  $j$  at time  $l$  which are offsprings of the component  $i$  at time  $k$ . So,  $\Pr\{\mathcal{H}_{k,i}^f|Z_{1:K}\} = 0 \iff \Pr\{\mathcal{H}_{l,j}^f|Z_{1:K}\} = 0 \forall j \in S_{k \rightarrow l,i}$ . Arguing as we did for pruning the  $i^{\text{th}}$  component from  $p(x_k|Z_{1:k})$  at time  $k$ , we can show that the components  $j \in S_{k \rightarrow l,i}$  with  $\Pr\{\mathcal{H}_{l,j}^f|Z_{1:K}\} = 0$  can be pruned from  $p(x_l|Z_{1:l})$ , without affecting the smoothing density  $p(x_l|Z_{1:K})$ . ■

The smoothing probability of a hypothesis  $\mathcal{H}_{k,i}^f$  in the FF can be evaluated from the weights of the components in the smoothing density using (15):

$$\Pr\{\mathcal{H}_{k,i}^f|Z_{1:K}\} = \int \Pr\{\mathcal{H}_{k,i}^f|Z_{1:k}\} p(x_k|\mathcal{H}_{k,i}^f, Z_{1:k}) \\ p(Z_{k+1:K}|x_k) dx_k$$

---


$$\begin{aligned}
&= \int w_{k,i}^f \mathcal{N}(x_k; \mu_{k,i}^f, P_{k,i}^f) \\
&\quad \times \sum_j w_{k,j}^b \mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b) dx_k \\
&= \sum_j w_{k,i}^f w_{k,j}^b \int \mathcal{N}(x_k; \mu_{k,i}^f, P_{k,i}^f) \\
&\quad \times \mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b) dx_k \\
&= \sum_j w_{k,i}^f w_{k,j}^b \\
&\quad \times \mathcal{N}(H_{k,j}^b \mu_{k,i}^f; \mu_{k,j}^b, H_{k,j}^b P_{k,i}^f H_{k,j}^{b^T} + P_{k,j}^b). \tag{17}
\end{aligned}$$

The product of the three terms in the summation of (17) is indeed the weights of the components in the smoothing density  $p(x_k|Z_{1:K})$ , obtained as the product of the  $i^{\text{th}}$  component in the filtering density and the BL. Similarly, the smoothing probability for the hypothesis  $\mathcal{H}_{k,j}^b$  during BF can be calculated from the smoothing posterior density as

$$\begin{aligned}
\Pr \left\{ \mathcal{H}_{k,j}^b | Z_{1:K} \right\} &= \sum_i w_{k,i}^f w_{k,j}^b \times \\
&\quad \mathcal{N}(H_{k,j}^b \mu_{k,i}^f; \mu_{k,j}^b, H_{k,j}^b P_{k,i}^f H_{k,j}^{b^T} + P_{k,j}^b). \tag{18}
\end{aligned}$$

Note that the summation is over the index  $i$  of the filtering density. Analogously, the product in the summation gives the weights of the components in the smoothing density obtained due to product of the  $j^{\text{th}}$  component in the BL and the filtering density.

## VI. ALGORITHM

In this section, we present the algorithmic description of the TFS algorithm performed in this paper. The computations can be divided into two parts — the FF and the BF. The FF algorithms are well studied in the literature. Therefore, only the algorithmic details of the BF are explained in this section (cf. Algorithm 1). The intragroup approximations described in Section IV and the SPP described in Section V are included in the BF algorithm. The smoothing density will be obtained as part of the SPP performed in the BF as in step 5 of Algorithm 1. We assume that at each  $k$ , the output of the FF will be the parameters of the filtering Gaussian mixture: weights  $w_{k,i}^f$ , means  $\mu_{k,i}^f$  and covariances  $P_{k,i}^f$ . Let  $M_k^f$ ,  $M_k^b$  and  $M_k^s$  be the number of components in the output of the FF, BF and the smoothing density respectively.

**Algorithm 1** Backward filter of the TFS

**Input:** Likelihoods:  $\beta_{k,i}$ ,  $H$ ,  $z_{k,i}$  and  $R$  for  $i = 0, \dots, m_k$  and  $k = 1, \dots, K$ .  
**Initialize**  $k \leftarrow K - 1$ ,  $w_{K,j}^b \leftarrow 1$ ,  $H_{K,j}^b \leftarrow []$ ,  $\mu_{K,j}^b \leftarrow []$  and  $P_{K,j}^b \leftarrow []$  for  $j = 0$ .

**Repeat**

- 1) **Update:** for every  $i$ ,  $j$ , compute  $\eta_{k+1,l} = \beta_{k+1,i} w_{k+1,j}^b$ ,
$$U_{k+1,l} = \begin{bmatrix} H \\ H_{k+1,j}^b \end{bmatrix}, \quad \psi_{k+1,l} = \begin{bmatrix} z_{k+1,i} \\ \mu_{k+1,j}^b \end{bmatrix} \text{ and } G_{k+1,l} = \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & P_{k+1,j}^b \end{bmatrix}.$$
    - a) **If**  $\text{rank}(U_{k+1,l}) = N_x$ , **then** set  $\eta_{k+1,l} = \frac{\eta_{k+1,l}}{\det(U_{k+1,l})}$ ,  $\psi_{k+1,l} = U_{k+1,l}^{-1} \psi_{k+1,l}$ ,  $G_{k+1,l} = U_{k+1,l}^{-1} G_{k+1,l} U_{k+1,l}^{-T}$ , and  $U_{k+1,l} = I_{N_x}$  **end**  - 2) **Grouping:** If components  $l_1$  and  $l_2$  are such that  $U_{k+1,l_1} = U_{k+1,l_2}$ , then the components belong to the same group.
  - 3) **Intragroup approximation:** Within each group, the traditional pruning and merging are performed.
  - 4) **Retrodiction:** Set  $w_{k,l}^b = \eta_{k+1,l}$ ,  $\mu_{k,l}^b = \psi_{k+1,l}$ ,  $H_{k,l}^b = U_{k+1,l} F$  and  $P_{k,l}^b = U_{k+1,l} Q U_{k+1,l}^T + G_{k+1,l}$ .
  - 5) **Smoothing density:** For every  $i$  and  $j$ , compute  $w_{k,l}^s$ ,  $\mu_{k,l}^s$  and  $P_{k,l}^s$  same way as in the Update step 1 with  $k = k - 1$ ,  $\beta_{k+1,i} = w_{k,i}^f$ ,  $H = I_{N_x}$ ,  $z_{k+1,i} = \mu_{k,i}^f$  and  $R = P_{k,i}^f$ . Note that the rank calculated will always be  $N_x$  for this case.
  - 6) **SPP:** Calculate  $w_{k,j}^{bs} = \sum_l w_{k,l}^s$  according to (18), for every  $j$ . Remove component  $j$  from the BL based on the weights,  $w_{k,j}^{bs}$  and renormalize.
  - 7)  $k \leftarrow k - 1$
- until**  $k = 1$
- 

## VII. IMPLEMENTATION AND SIMULATION RESULTS

### A. Simulation scenario

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

---

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

The TFS algorithm is compared with FBS based on an  $N$ -scan pruning algorithm where the FF is performed using  $N$ -scan pruning and Rauch-Tung-Striebel algorithm is used on each branch in the filtering hypothesis tree.

### B. Implementation details

In both the FF and BF of TFS, to reduce the complexity, extra gating is performed in addition to the conventionally used ellipsoidal gating. This extra gate is rectangular, with dimensions based on the measurement noise covariance and the center at the prediction density mean. The gating probability  $P_G$  and the pruning threshold  $P_P^f$  for the FF are set as  $(1 - 10^{-5})$  and  $10^{-4}$  respectively.

The merging algorithm in FF is a cheaper variant of Salmond's algorithm [13]. The original Salmond's algorithm looks for the minimum merging cost across every pair of components in the GM. Therefore, it has a quadratic complexity in the number of components. So, to reduce the complexity involved, instead of looking for the minimum cost, we use a heuristic algorithm in this paper. 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 the new reduced GM until there are no pairs that have costs lower than the threshold.

In the BF of the TFS, both the intragroup approximations and the SPP are used to reduce the components in the BL. The intragroup pruning is based on the maximum weight in the group. The components that have weights lesser than  $1/100^{\text{th}}$  of the maximum weight in the group are pruned. The intragroup merging is based on the variant of the Salmond's algorithm discussed above. To employ the SPP method followed by these intragroup approximations, first the smoothing density has to be computed. This involves taking the product of the filtering density GM and the RDGM BL which is an expensive operation. So, to reduce the number of operations involved in computing this smoothing density, we reduce the filtering density GM to a single Gaussian, and compute the smoothing density using this reduced filtering density and the BL. Then, using this interim smoothing density, SPP is employed to prune components from the BL.

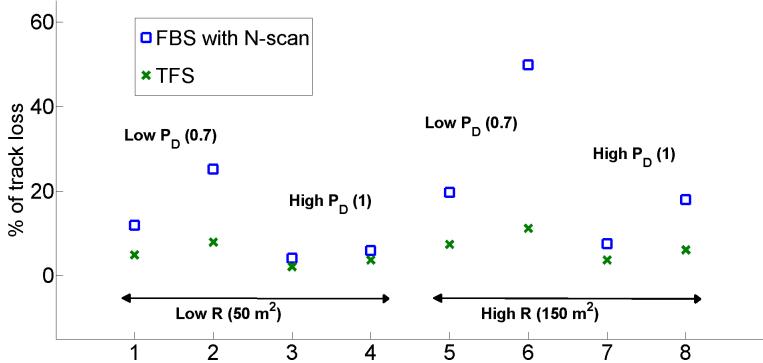


Fig. 3: Track Loss performance: Every odd point on x-axis (1, 3, 5, 7) is for low clutter intensity  $\beta = 0.0001$  and every even point (2,4,6,8) is for high  $\beta = 0.0002$ . The order of the eight scenarios is the same also for the others plots in Fig. 4, Fig. 5 and Fig. 6.

In case of  $N$ -scan pruning in the FBS algorithm, the parameter  $N$  for the various settings is chosen to be the largest possible  $N$  such that the complexity (run-time) for a single run is within the threshold of 2 s. The rectangular gating and ellipsoidal gating are also used here.

The performance measures used for comparison are root mean squared error (RMSE), normalized estimation error squared (NEES), complexity and track loss. A track is considered lost if the true state is more than three standard deviations (obtained from the estimated covariance) away from the estimated state for five consecutive time steps. The computational complexity is calculated as the average time taken during MATLAB simulations to run each algorithm on the entire trajectory of 40 time steps. The graphs are obtained by averaging over 1000 Monte Carlo iterations.

### C. Results

The results of the simulation are presented in Fig. 3 to 6. It can be seen that the TFS performs significantly better than the FBS with  $N$ -scan for most of the scenarios. From the Fig. 3 for track loss performance, one can notice that the performance gain is higher for TFS compared to FBS when  $P_D$  is low and the measurement noise  $R$  and the clutter intensity  $\beta$  are high (point 6 on the x-axis in Fig. 3). 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 can be quite aggressive resulting in the undesirable ‘degeneracy’ problem in the FBS. The impact of this degeneracy problem can also be observed in the

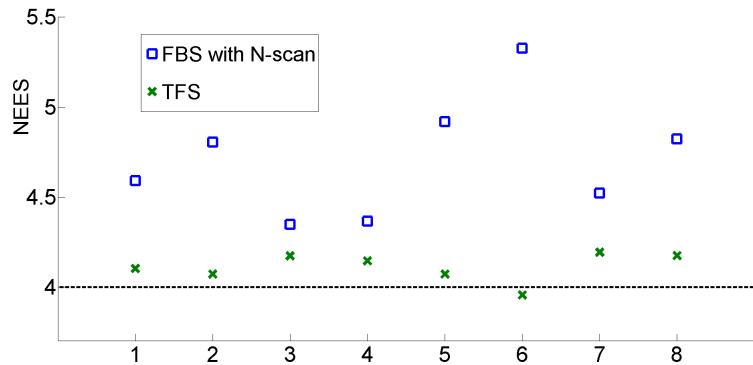


Fig. 4: NEES performance: Compared to the FBS, the values of the NEES for the TFS are very close to the optimal value of 4 in all the scenarios.

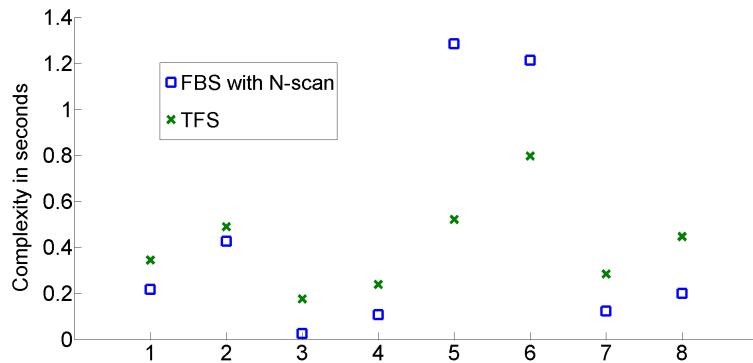


Fig. 5: Computational complexity: The TFS algorithm with intra group approximations and SPP is computationally cheaper compared to the FBS with  $N$ -scan.

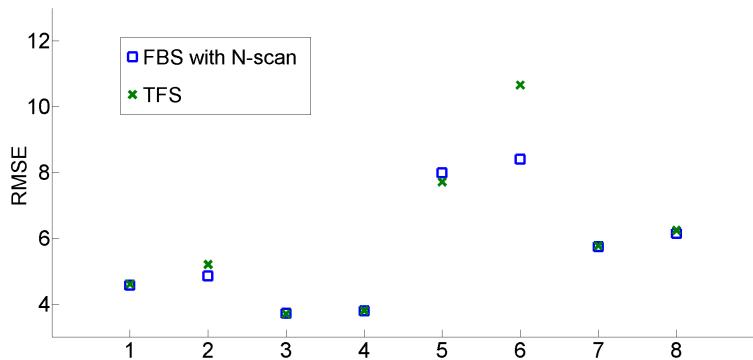


Fig. 6: RMSE performance: The results are very similar for the TFS and the FBS.

NEES performance plot in Fig. 4 (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 performance, TFS based on intragroup approximations and SPP offers a computationally cheaper solution compared to the FBS based on  $N$ -scan pruning as can be observed in Fig. 5. However, the RMSE performance of the TFS and FBS are very similar in most scenarios as seen in Fig. 6.

### VIII. CONCLUSION

In this paper, we present how two-filter smoothing can be performed when the posterior densities are Gaussian mixtures (GMs). GM reduction (GMR) techniques such as pruning and merging are used in the forward filter of the two-filter smoothing method. The structure of the backward likelihood at the output of the backward filter is a reduced dimension GM (RDGM). GMR techniques are also applied within groups of components in the RDGM. Since this strategy has limitations in the reduction, we have proposed the smoothed posterior pruning, where components in the backward likelihood are pruned based on the smoothing posterior weights of those components. The proposed algorithm is shown to have better track loss, root mean squared error, normalized estimation error squared as well as lower computationally complexity compared to a forward-backward smoothing algorithm based on  $N$ -scan pruning.

### REFERENCES

- [1] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *Control Systems, IEEE*, vol. 29, no. 6, pp. 82–100, 2009.
- [2] I. Bilik and J. Tabrikian, “Maneuvering target tracking in the presence of glint using the nonlinear gaussian mixture kalman filter,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 246–262, 2010.
- [3] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House Norwood, MA, 1999, vol. 685.
- [4] M. Briers, A. Doucet, and S. Maskell, “Smoothing algorithms for state-space models,” *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010. [Online]. Available: <http://dx.doi.org/10.1007/s10463-009-0236-2>
- [5] D. Crouse, P. Willett, K. Pattipati, and L. Svensson, “A look at gaussian mixture reduction algorithms,” in *Proceedings of the fourteenth International Conference on Information Fusion*, 2011, pp. 1–8.

- 
- [6] 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.
  - [7] 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.
  - [8] G. Kitagawa, “The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother,” *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 4, pp. 605–623, 1994.
  - [9] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
  - [10] J. Pearl, “Reverend bayes on inference engines: A distributed hierarchical approach,” in *AAAI*, 1982, pp. 133–136.
  - [11] H. Rauch, F. Tung, and C. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA JOURNAL*, vol. 3, pp. 1445–1450, 1965.
  - [12] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
  - [13] 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.
  - [14] R. A. Singer, R. G. Sea, and K. Housewright, “Derivation and evaluation of improved tracking filter for use in dense multitarget environments,” *IEEE Transactions on Information Theory*, vol. 20, no. 4, pp. 423–432, 1974.
  - [15] B.-N. Vo, B.-T. Vo, and R. P. Mahler, “Closed-form solutions to forward–backward smoothing,” *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 2–17, 2012.
  - [16] J. L. Williams and P. S. Maybeck, “Cost-function-based Gaussian mixture reduction for target tracking,” in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 2, 2003, pp. 1047–1054.



# Paper C

## Merging-based forward-backward smoothing on Gaussian mixtures

A. S. Rahmathullah, L. Svensson and D. Svensson

To be published in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, July 2014.



# Merging-based forward-backward smoothing on Gaussian mixtures

Abu Sajana Rahmathullah\*, Lennart Svensson\*, Daniel Svensson†

\*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,

$$x_k = Fx_{k-1} + v_k, \quad (1)$$

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

$$z_k^t = Hx_k + w_k \quad (2)$$

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  $\beta V$ , where  $\beta$  is the clutter density. The number of measurements obtained at time  $k$  is denoted  $m_k$ .

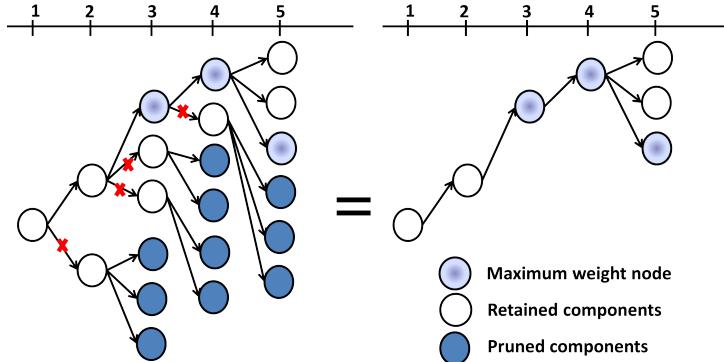


Fig. 1: Example of a degenerate tree after FF with  $N$ -scan pruning with  $N = 2$ . To the left, the figure shows the tree with the nodes that are pruned. In the figure to the right, the pruned branches are removed. One can see that eventually only one branch is retained from time 1 to 4.

The objective is to find the smoothing density  $p(x_k|Z_{1:K})$  for  $k = 1, \dots, K$  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.

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

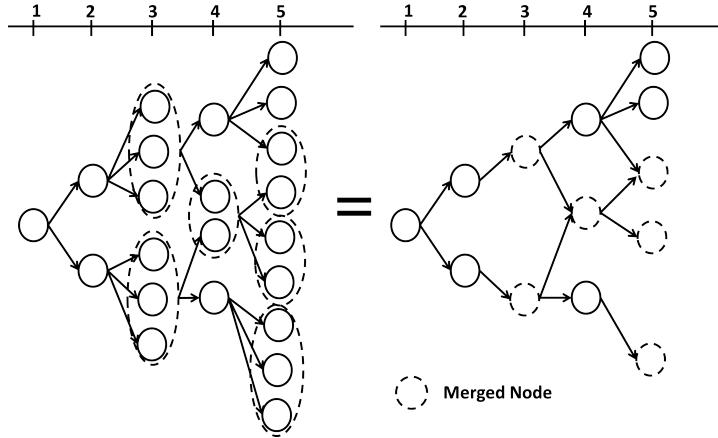


Fig. 2: Same example as in Fig. 1. Here, instead of pruning, merging of components is performed. One can see that the DA uncertainties are still retained in the merged nodes.

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,

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

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

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

where

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

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

$$p(x_k|Z_{1:k}) = \sum_{n=0}^{M_k^f} p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f) \Pr\left\{\mathcal{H}_{k,n}^f | Z_{1:k}\right\}, \quad (6)$$

where  $M_k^f$  is the number of components in the GM (for the optimal case,  $M_k^f = \prod_{i=1}^k (m_i + 1)$ ). The hypothesis  $\mathcal{H}_k^n$  represents a unique sequence of measurements or missed-detection hypotheses assignments from time 1 to time  $k$ , under which the density  $p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f)$  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  $p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f)$  and a probability  $\Pr\left\{\mathcal{H}_{k,n}^f | Z_{1:k}\right\}$ .

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

$$p(x_k|Z_{1:K}) = \sum_{a=0}^{M_K^f} p(x_k|Z_{1:K}, \mathcal{H}_{K,a}^f) \Pr\left\{\mathcal{H}_{K,a}^f | Z_{1:K}\right\}, \quad (7)$$

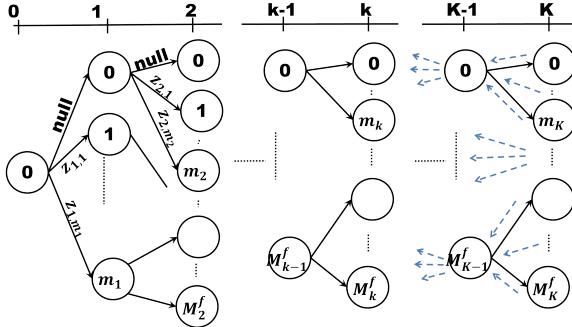


Fig. 3: Optimal Gaussian mixture filtering and smoothing are illustrated in a hypothesis tree. At each time, each node represents a component in the filtering GM at the corresponding time. The solid arrowed lines represent the data associations across time. The dashed arrowed lines represent the paths 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.

where each component  $p(x_k|Z_{1:K}, \mathcal{H}_{K,a}^f)$  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  $\Pr\{\mathcal{H}_{K,a}^f|Z_{1:K}\}$  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

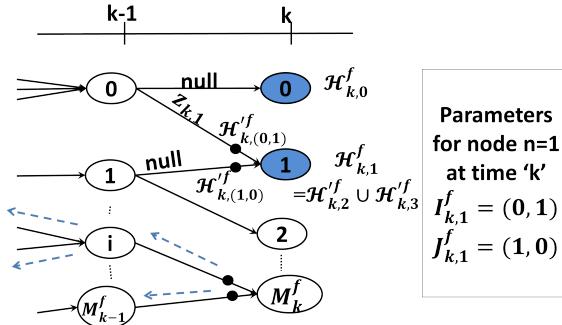


Fig. 4: A part of an f-graph is shown to illustrate merging during forward filtering (FF) (solid lines) and backward smoothing (BS) through merged nodes (dashed lines). The solid lines represent the branches in the FF. The small filled circles represent the components before merging. The dashed lines 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.

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 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  $\mathcal{H}_{k,n}^f$ , 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  $p(x_{k+1}|Z_{1:K}, \mathcal{H}_{K,a}^f)$  in the smoothing density at time  $k+1$  and a component  $p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f)$  in the filtering density at  $k$ , such that  $p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f) = p(x_k|Z_{1:k}, \mathcal{H}_{K,a}^f)$ , and returns a component  $p(x_k|Z_{1:K}, \mathcal{H}_{K,a}^f)$  in the smoothing density at  $k$ . In other words, the RTS algorithm combines the smoothing density component with hypothesis  $\mathcal{H}_{K,a}^f$  and the filtering density component with hypothesis  $\mathcal{H}_{k,n}^f$  if the DA subsequence in  $\mathcal{H}_{K,a}^f$  from time 1 to time  $k$  is identical to the DA sequence in  $\mathcal{H}_{k,n}^f$ . It should be noted that due to pruning during FF, the number of filtering hypotheses  $\mathcal{H}_{K,a}^f$  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  $\mathcal{H}_{k,n}^f$  for  $k = 1, \dots, K$ , represent. In the former, as a result of merging during the FF, the hypotheses  $\mathcal{H}_{K,a}^f$  are sets of DA sequences, whereas in the latter,  $\mathcal{H}_{K,a}^f$  corresponds to one DA sequence. As each DA sequence in the set  $\mathcal{H}_{K,a}^f$  corresponds to a Gaussian component, the term  $p(x_k|Z_{1:K}, \mathcal{H}_{K,a}^f)$  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  $\mathcal{H}_{K,a}^f$ . The idea in this paper is that the DA sequences in each hypothesis  $\mathcal{H}_{K,a}^f$  can be partitioned to form hypotheses  $\mathcal{H}_{k,l}^s$  such that  $p(x_k|Z_{1:K}, \mathcal{H}_{k,l}^s)$  is a Gaussian density. During BS, each of these hypotheses  $\mathcal{H}_{k,l}^s$  can be related to a hypothesis  $\mathcal{H}_{k,n}^f$  from the FF, during the BS, enabling us to employ RTS recursions on these new hypotheses  $\mathcal{H}_{k,l}^s$ . 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  $\mathcal{H}_{k,l}^s$  and the GMR of the components in the smoothing density, we use a hypothesis graph called the s-graph.

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

$$p(x_k|Z_{1:K}) = \sum_{l=0}^{M_k^s} p(x_k|Z_{1:K}, \mathcal{H}_{k,l}^s) \Pr \{ \mathcal{H}_{k,l}^s | Z_{1:K} \}, \quad (8)$$

where  $p(x_k|Z_{1:K}, \mathcal{H}_{k,l}^s)$  is a Gaussian density function  $\mathcal{N}(x_k; \mu_{k,l}^s, P_{k,l}^s)$ , with weight  $w_{k,l}^s = \Pr \{ \mathcal{H}_{k,l}^s | Z_{1:K} \}$ . Starting with  $\mathcal{H}_{K,a}^s = \mathcal{H}_{K,a}^f$  at  $k = K$ , the hypotheses  $\mathcal{H}_{k,l}^s$  (partitioned from  $\mathcal{H}_{K,a}^f$ ) can be obtained recursively by defining the hypotheses  $\mathcal{H}_{k+1,p}^s$  at time  $k + 1$  and  $\mathcal{H}_{k,n}^f$ , 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  $\cup$  and  $\cap$  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  $n = 1, \dots, M_k^f$  in the f-graph (after pruning and merging of the filtering GM) at time  $k$ , the following parameters are defined (cf. Fig. 4):

- $\mathcal{H}_{k,n}^f$ , the hypothesis that corresponds to node  $n$  (after merging). If  $\mathcal{H}_{k,(i,j)}^f$  represent the set of disjoint hypotheses formed by associating hypothesis  $\mathcal{H}_{k-1,i}^f$  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

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

Note that hypotheses  $\mathcal{H}_{k,(i,j)}'^f$  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  $\mathcal{H}_{k,(i,j)}'^f$  is to indicate that it is the hypothesis before merging. Similar notation will be used in the s-graph as well.

- $\mu_{k,n}^f$  and  $P_{k,n}^f$  are the mean and the covariance of the Gaussian density  $p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f)$  (after merging).
- $I_{k,n}^f$  is a vector that contains indices  $i$  of the hypotheses  $\mathcal{H}_{k-1,i}^f$  at node  $i$  at time  $k-1$ . An interpretation of this is that for each  $i$  in  $I_{k,n}^f$ , there is a branch between node  $i$  at time  $k-1$  and node  $n$  at time  $k$ .
- $w_{k,n}^f$  is a vector that contains the probabilities  $\Pr\{\mathcal{H}_{k,(i,j)}'^f | Z_{1:k}\}$  of the DA hypotheses  $\mathcal{H}_{k,(i,j)}'^f$  before merging. Using (9), it can be shown that

$$\Pr\{\mathcal{H}_{k,n}^f | Z_{1:k}\} = \sum_{i,j} \Pr\{\mathcal{H}_{k,(i,j)}'^f | Z_{1:k}\}. \quad (10)$$

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

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

- $\mathcal{H}_{k,l}^s, \mu_{k,l}^s$  and  $P_{k,l}^s$  are the hypothesis, mean and covariance of the  $l^{\text{th}}$  Gaussian component (after merging).
- $w_{k,l}^s$  is the probability  $\Pr\{\mathcal{H}_{k,l}^s | Z_{1:K}\}$ .
- $I_{k,l}^s$  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:  $\mathcal{H}_{K,l}^s = \mathcal{H}_{K,l}^f, \mu_{K,l}^s = \mu_{K,l}^f, P_{K,l}^s = P_{K,l}^f, w_{K,l}^s = \sum_r w_{K,l}^f(r)$  and  $I_{K,l}^s = l$ . 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  $\mathcal{H}_{k,l}^s$  are used to explain the

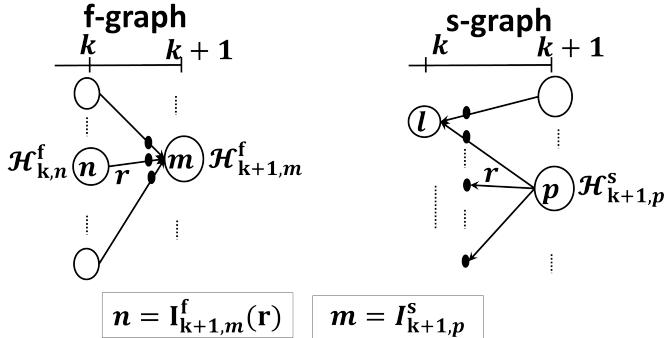


Fig. 5: Illustration of BS on a merged graph: A node  $p$  in the s-graph is shown along with the corresponding node  $n$  in the f-graph. The relations between the different parameters are as indicated. The filled black circles in the f-graph and s-graph represent the components before merging.

weight calculations. But, for implementation purposes, it suffices to update and store the parameters  $\mu_{k,l}^s$ ,  $P_{k,l}^s$ ,  $w_{k,l}^s$  and  $I_{k,l}^s$ .

### 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  $p^{\text{th}}$  component of  $p(x_{k+1}|Z_{1:K})$  and the  $n^{\text{th}}$  component of  $p(x_k|Z_{1:k})$ . Whether a pair  $(p, n)$  depends on if the hypothesis  $\mathcal{H}_{k,n}^f$  is in the history of the hypothesis  $\mathcal{H}_{k+1,p}^s$  or not. This information can be inferred from the relation between the f-graph and the s-graph.

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  $m = I_{k+1,p}^s$ . 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  $I_{k+1,m}^f$  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  $I_{k+1,m}^f$ , 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 hypothesis relations. The hypotheses involved in obtaining the component are  $\mathcal{H}_{k+1,p}^s$ ,  $\mathcal{H}_{k+1,m}^f$  and  $\mathcal{H}_{k,n}^f$ , where  $m = I_{k+1,p}^s$  as discussed before and node  $n$  is, say, the  $r^{\text{th}}$  element in the vector  $I_{k+1,m}^f$ , denoted  $n = I_{k+1,m}^f(r)$  (cf. Fig. 5). Using these hypotheses, the hypothesis corresponding to the resulting component is denoted  $\mathcal{H}_{k,(p,n)}^{fs}$  and is written as

$$\mathcal{H}_{k,(p,n)}^{fs} = \mathcal{H}_{k+1,p}^s \bigcap \mathcal{H}_{k,n}^f. \quad (11)$$

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

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

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  $I_{k+1,n}^f$ , 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  $\mathcal{H}_{k,n}^f$ , 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  $\mathcal{H}_{k,(p,n)}^{fs}$  for different  $(p, n)$ , the retained hypothesis are relabeled as  $\mathcal{H}_{k,l}^s$ , 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  $\mu_{k,n}^f$ ,  $P_{k,n}^f$ ,  $w_{k,n}^f$  and  $I_{k,n}^f$  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  $\mu_{k,l}^s$ ,  $P_{k,l}^s$ ,  $w_{k,l}^s$  and  $I_{k,l}^s$  in the list in Section IV-A2.

---

**Algorithm 1** Forward filtering

---

**Input:** Prior:  $\mu_{0|0}$ ,  $P_{0|0}$ .

Likelihoods:  $H$ ,  $z_{k,j}$ ,  $R$  and

$$\beta_{k,j} = \begin{cases} \beta(1 - P_D P_G) & j = 0 \\ P_D & j \neq 0 \end{cases}, \text{ for } j = 0, \dots m_k, k = 1, \dots K.$$

**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  $w_{LL,(i,j)} = \mathcal{N}(z_{k,j}; H\mu_{k|k-1,i}, HP_{k|k-1,i}H^T + R) > P_G$  and add  $G = G \cup \{(i, j)\}$  for the pairs that pass the threshold.
  - 3) **Pruning:** Initialize  $P = \{\}$ . For each pair  $(i, j) \in G$ , calculate the posterior weight  $w_{k,(i,j)} = w_{k-1,i}^f \beta_{k,j} w_{LL}(i, j)$  and re normalize. Check if  $w_{k,(i,j)} > P_P^f$  and add all pairs  $(i, j)$  that pass the threshold to  $P$ , i.e., set  $P = P \cup \{(i, j)\}$ .
  - 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  $\mu_{k,n}^f$  and  $P_{k,n}^f$ , each corresponding to a node  $n$  in the f-graph. Along with performing merging of components, the merging module also returns the vectors  $I_{k,n}^f$  and  $w_{k,n}^f$  that contains the indexes  $i$  and the weights  $w_{k,(i,j)}$ , respectively, of the components that are merged to form node  $n$ .
- 

## 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  $0.07 \text{ m/s}^2$ . The trajectory was generated for  $K = 40$  time steps with a sampling time of  $1 \text{ s}$ . The whole volume of the track was used for generating clutter data.

---

**Algorithm 2** Backward smoothing of FBS-GMM

---

**Input:** Filtering GM parameters:  $\mu_{k,n}^f, P_{k,n}^f, w_{k,n}^f, I_{k,n}^f$ , for  $n = 1, \dots, M_k^f$ .

**Initialize:** Set  $M_K^s = M_K^f$ ,  $\mu_{K,l}^s = \mu_{K,l}^f$ ,  $P_{K,l}^s = P_{K,l}^f$ ,  $I_{K,l}^s = l$  and  $w_{K,l}^s = \sum_r w_{K,l}^f(r)$  (summation is over the entire vector  $w_{K,l}^f$ ).

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

- 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  $\mu_{k,n}^f, P_{k,n}^f$  and  $\mu_{k+1,p}^s, P_{k+1,p}^s$  (Note, the parameters  $\mu_{k+1,p}^s$  and  $P_{k+1,p}^s$  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 (17). 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  $w_{k|K,(p,n)} > P_P^s$  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  $\{\mu_{k,l}^s, P_{k,l}^s, w_{k,l}^s\}$  along with the parameter  $I_{k,l}^s = n$ .
- 

The values for the measurement noise  $R$ , the probability of detection  $P_D$  and the clutter intensity  $\beta$ , 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  $\beta$  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  $P_P^f$  mentioned in step 2 and 3 of Algorithm 1 are  $(1 - 10^{-5})$  and  $10^{-4}$  respectively. The threshold  $P_P^s$  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 Salmon's algorithm [12] aimed at reducing the complexity compared to the original algorithm. The original Salmon'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 Salmon's algorithm and 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  $\beta$  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

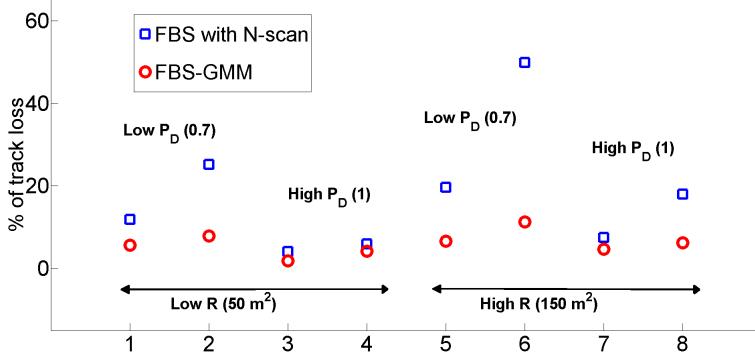


Fig. 6: Track Loss. Every odd point on x-axis (1,3,5,7) is for low clutter intensity  $\beta = 0.0001$  and every even point (2,4,6,8) is for high  $\beta = 0.0002$ . The order of the eight scenarios is the same for the others plots in Fig. 7, Fig. 8 and Fig. 9.

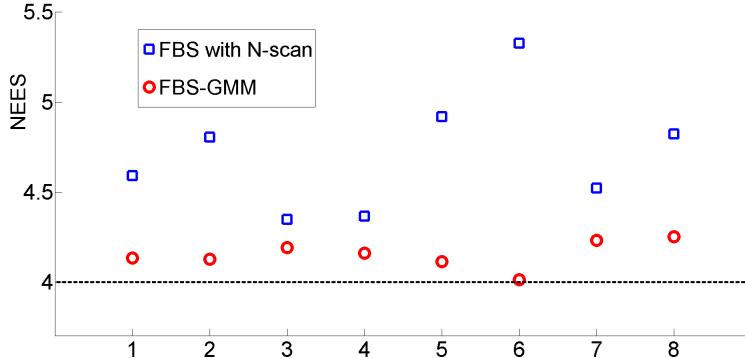


Fig. 7: NEES performance: Compared to the  $N$ -scan based FBS, the values of the NEES for the FBS-GMM are very close to the optimal value of 4 in all the scenarios.

components, the pruning during FBS with  $N$ -scan pruning can be quite aggressive resulting in the degeneracy problem. 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.

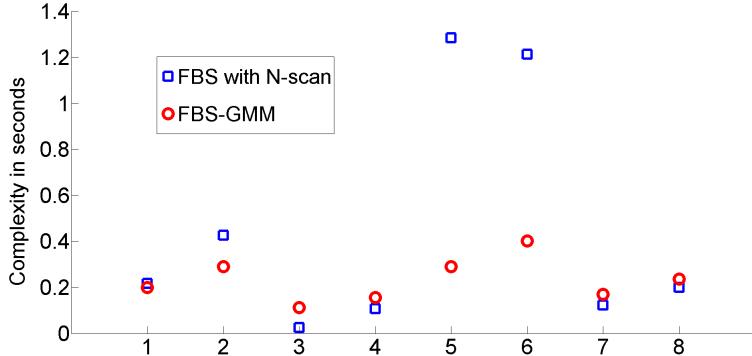


Fig. 8: Computational complexity: The FBS-GMM algorithm is computationally cheaper compared to the FBS with  $N$ -scan.

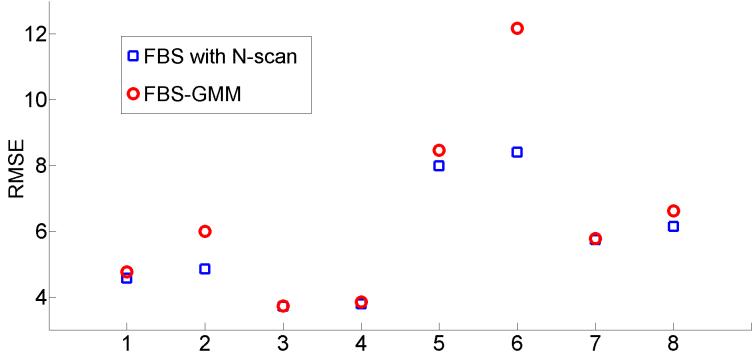


Fig. 9: RMSE performance: The results are very similar for the both the FBS algorithms.

## 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 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 pur-

- 
- pose 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 (12) can be obtained using the hypotheses definitions. Consider the hypothesis expression in (11), and the illustration

in Fig. 5. We are interested in calculating the probability of the hypothesis

$$\begin{aligned} \Pr \left\{ \mathcal{H}_{k,(p,n)}^{s'} | Z_{1:K} \right\} &= \Pr \left\{ \mathcal{H}_{k+1,p}^s \bigcap \mathcal{H}_{k,n}^f | Z_{1:K} \right\} \\ &\propto \Pr \left\{ \mathcal{H}_{k+1,p}^s | Z_{1:K} \right\} \Pr \left\{ \mathcal{H}_{k,n}^f | \mathcal{H}_{k+1,p}^s, Z_{1:k+1} \right\} \\ &\quad \times p(Z_{k+2:K} | \mathcal{H}_{k+1,p}^s, \mathcal{H}_{k,n}^f, Z_{1:k+1}). \end{aligned} \quad (13)$$

In the above equation, the factor  $\Pr \left\{ \mathcal{H}_{k+1,p}^s | Z_{1:K} \right\}$  is the weight  $w_{k+1,p}^s$ , 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  $n$ :

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

In (14), adding the hypothesis  $\mathcal{H}_{k+1,m}^f$  to the conditional statement does not make a difference as the hypothesis  $\mathcal{H}_{k+1,p}^s$  corresponding to the entire sequence of measurements masks it; but it does make the latter equations simpler to handle. The second factor in (13) is given by

$$\begin{aligned} \Pr \left\{ \mathcal{H}_{k,n}^f | \mathcal{H}_{k+1,p}^s, Z_{1:k+1} \right\} &= \Pr \left\{ \mathcal{H}_{k,n}^f | \mathcal{H}_{k+1,m}^f, Z_{1:k+1} \right\} \\ &\propto \frac{\Pr \left\{ \mathcal{H}_{k,n}^f, \mathcal{H}_{k+1,m}^f | Z_{1:k+1} \right\}}{\Pr \left\{ \mathcal{H}_{k+1,m}^f | Z_{1:k+1} \right\}}. \end{aligned} \quad (16)$$

The numerator term in (16) is the same as the probability  $w_{k+1,m}^f(r)$  of the  $r^{\text{th}}$  branch before merging to form node  $m$ . Consolidating (15) and (16) into (13), we get that

$$\Pr \left\{ \mathcal{H}_{k,(p,n)}^{s'} | Z_{1:K} \right\} \propto w_{k+1,p}^s \times w_{k+1,m}^f(r). \quad (17)$$

# Paper D

## Efficient strategies for Gaussian mixture smoothing

A. S. Rahmathullah, L. Svensson and D. Svensson

*To be submitted.*



# Efficient strategies for Gaussian mixture smoothing

Abu Sajana Rahmathullah\*, Lennart Svensson\*, Daniel Svensson†

## Abstract

In this paper, we consider the problem of Gaussian mixture smoothing using forward-backward smoothing (FBS) and two-filter smoothing strategies (TFS). We present techniques to apply pruning and merging approximations to the smoothing methods, while maintaining accurate representations of the smoothing densities. The key contributions include methods to enable merging in a FBS algorithm and a theory related to TFS that allows us to use the smoothed posterior to prune components in the two filters. The results using the proposed approximation techniques show that both FBS and TFS perform similarly for all the scenarios considered. Compared to FBS based on the  $N$ -scan pruning algorithm, both proposed algorithms perform significantly better.

## I. INTRODUCTION

In many applications, there is interest in sequentially estimating different parameters to determine the state of the underlying dynamical system, using indirect and inaccurate observations made from sensors. In certain applications, the observed system can be in one of many possible modes, such that the uncertainties cannot be modeled as uni-modal. It is also possible that several observations are made at a single time instant, each with unknown origin. In these scenarios, the motion and/or the sensor model are multi-modal and are appropriately modeled using mixtures densities, the famous of which is Gaussian mixtures (GMs). For the GM models, the optimal solutions to the posterior densities as a result of prediction, filtering and smoothing are GMs and the closed-form expressions for the optimal GMs are easily obtained by extending the solutions for Gaussian densities [8, 10, 11, 16, 21].

\* Abu Sajana Rahmathullah and Lennart Svensson are with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden, E-mail: {sajana, lennart.svensson}@chalmers.se.

† Daniel Svensson is with the Department of Safety Electronics, Volvo Car Group, Göteborg, Sweden, E-mail: daniel.svensson.2@volvocars.com. He is also with the Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden.

The problem with mixture densities is that the number of components in the resulting GM posterior densities grows exponentially with time. Therefore, for practical implementations, there is a need to reduce the number of components in the GM. For the prediction and filtering of GMs, there exist suboptimal solutions based on Gaussian mixture reduction (GMR) techniques, such as pruning and merging. For example, in target tracking application, the multiple hypothesis tracking (MHT) [5, 15, 17] solution is typically based on pruning, whereas the probabilistic data association (PDA) [2] solution is based on merging approximations.

In smoothing problem, the goal is to estimate the past values of the state using all the observations made. In general, the smoothing solution is more accurate than the filtering and prediction solutions, as it uses more observations than the other two. There are two main strategies to perform smoothing, namely, the forward-backward smoothing (FBS) [1, 4] and the two-filter smoothing (TFS) [11]. In theory, both these methods provide identical solutions, but in practice, they can differ due to the approximations involved. Since the smoothing result depends on the filtering, it is not straightforward to apply GMR to the smoothing problem. For both FBS and TFS, it is hence not clear how approximations should be made, and research within the area is scarce. The aim of this paper is to fill in these blanks, by addressing approximations within Gaussian mixture smoothing.

The FBS method involves forward filtering (FF) and backward smoothing (BS). The FF is similar to traditional filtering, and in the existing implementations of FBS, it is based on pruning approximations [12]. The BS operates on the output of the FF to recursively return smoothing densities. Pruning-based forward-backward smoothing works well for many scenarios, but in complex scenarios such as high clutter density with unreliable measurements, one may face with degeneracy problem in the FF. The consequence of this degeneracy is that during BS, the smoothing density is statistically similar to a uni-modal Gaussian, resulting in highly underestimated uncertainty and potential track loss. In essence, the pruning-based FBS, when applied to complicated scenarios, suffers from inferior performance.

The TFS involves forward filtering and backward filtering (BF) steps. The FF is similar to traditional filtering, and is based on pruning and merging approximations in existing algorithms [11, 22]. The BF, in principle, is similar to the FF but is difficult to handle since it is not a density function [11]. Therefore, direct application of the pruning or merging approximations to the BF is not valid. To our knowledge, there does not exist any practical approximation techniques for the backward filter in a Gaussian mixture two-filter smoother.

---

In this paper, our focus is on obtaining practical methods to mixture reduction in Gaussian mixture smoothing problems. More precisely, we present a novel FBS algorithm for GMs, which makes use of merging approximations both during FF and BS, thus avoiding the problem of degeneracy. To be specific, we propose a strategy to apply the BS recursions on filtering densities that are obtained using merging approximations in the FF. For Gaussian mixture two-filter smoothing, we propose two approximation techniques for the backward filter. The first one, called intragroup approximation, shows that it is possible to apply GMR techniques within groups of components in the BF. The second technique, called smoothed posterior pruning, proves that it is valid to use the FF posterior to prune components in the BF without affecting the smoothing density. Additionally, we show that this smoothed posterior pruning technique is also applicable to the forward filter, i.e., the components in the forward filter can be pruned using the backward filter output.

The proposed ideas for the FBS and TFS methods have been evaluated on the data association problem in single target tracking, under linear-Gaussian state-space model assumptions. The results show that the proposed methods provide consistent estimates with low track loss, compared to the FBS using the N-scan pruning method. Additionally, the proposed algorithms are computationally cheaper. The root mean squared error (RMSE) performance, however, is very similar across all evaluated algorithms, but it should be noted that the instances when the  $N$ -scan pruning-based FBS algorithm loses tracks are not considered in the calculations of the RMSE.

The outline of the paper is as follows: In Section II, we present the assumptions made for the state-space model and the problem formulation for the FBS and TFS methods. In Section III, we present the background theory for the two smoothing problems at hand. Section IV and Section V present the contributions of the paper towards the FBS and TFS problems, respectively. The algorithmic descriptions of the proposed ideas given are in Section VI. The implementation details and the results of the simulations are provided in Section VII. In Section VIII, we discuss how the two algorithms can be extended to complex scenarios.

## II. PROBLEM FORMULATION

We consider a single target moving in a cluttered background. The state vector  $x_k$  at time  $k$  is of dimension  $N_x$  and is varying according to the process model,

$$x_k = Fx_{k-1} + v_k, \quad (1)$$

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

$$z_k^t = Hx_k + w_k, \quad (2)$$

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  $\beta V$ , where  $\beta$  is the clutter density. The number of measurements obtained at time  $k$  is denoted  $m_k$ .

The objective is to develop novel and practical algorithms to compute the smoothing density  $p(x_k|Z_{1:K})$  for  $k = 1, \dots, K$  under data association uncertainty, using the forward-backward smoothing and two-filter smoothing methods. Our interest is also in comparing the performance of these two methods. For the assumptions made, the optimal smoothing density, as will be shown in Section III, is a GM with exponentially increasing number of components. Therefore, for complexity reasons, GMR algorithms, such as pruning and merging, are required in the filtering and smoothing algorithms. The main focus of the paper is to develop techniques to reduce GMs to manageable sizes while maintaining an accurate representation of the smoothing density at all times.

### III. BACKGROUND

In this section, we present a background to the two main smoothing strategies: forward-backward smoothing and two-filter smoothing. It is shown that the filtering and smoothing densities for the data association (DA) problem are Gaussian mixtures. It is also shown how these posterior densities are obtained by defining DA hypotheses and using the solutions for Gaussian densities. The existing methods and approximation for the two smoothing strategies are explained using graphical illustrations.

#### A. Two smoothing strategies

The smoothing density is given by

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

The interpretation of (3) is that the filtering density  $p(x_k|Z_{1:k})$  is updated with the likelihood  $p(Z_{k+1:K}|x_k)$ , referred to as the backward likelihood (BL), from future measurements to obtain the smoothing posterior  $p(x_k|Z_{1:K})$ .

In theory, the filtering density  $p(x_k|Z_{1:k})$  is given by forward filtering from time 1 to  $k$  using prediction and update recursions, which is similar

---

between the two main smoothing strategies, FBS and TFS. However, they differ in the way in which the likelihood,  $p(Z_{k+1:K}|x_k)$ , in (3) is calculated. In the FBS method, the BL is obtained by backward smoothing, which uses the filtering and smoothing densities recursively, in two steps:

$$p(Z_{k+1:K}|x_{k+1}) \propto \frac{p(x_{k+1}|Z_{1:K})}{p(x_{k+1}|Z_{1:k})} \quad (4)$$

and

$$p(Z_{k+1:K}|x_k) \propto \int p(Z_{k+1:K}|x_{k+1})f(x_{k+1}|x_k)dx_{k+1}. \quad (5)$$

Note that (4) involves division of densities, which does not have a closed-form expression for arbitrary densities. In the TFS method, the BL is obtained by running a so-called backward filtering step from time  $K$  to  $k$ . The BF involves an update step, similar to the one in the FF, given by

$$p(Z_{k+1:K}|x_{k+1}) = p(Z_{k+1}|x_{k+1})p(Z_{k+2:K}|x_{k+1}) \quad (6)$$

and a retrodiction step as in (5). The output of the BF, i.e., the BL, is in general not a density function in the state  $x_k$ . This makes it difficult to apply techniques from the FF to the BF.

For linear and Gaussian state space models (in the absence of clutter and  $P_D = 1$ ), the filtering and smoothing densities are Gaussian. For the FF, the classical Kalman filter [10] gives the equations for the optimal Gaussian density. For the BS of FBS, the Rauch-Tung-Striebel smoothing method in [16] directly gives a closed-form expression for the likelihood on the left-hand side in (5). The BF solution of the TFS is presented in [8].

For the assumption made in Section II, it can be shown that the smoothing densities are Gaussian mixtures of the form

$$p(x_k|Z_{1:K}) = \sum_{l=1}^{M_k^s} w_{k,l}^s \mathcal{N}(x_k; \mu_{k,l}^s, P_{k,l}^s), \quad (7)$$

where the notation  $\mathcal{N}(x; \mu, P)$  represents a Gaussian density in random variable  $x$  with mean  $\mu$  and covariance  $P$ . These GM solutions are obtained using Gaussian solutions to the smoothing problems.

### B. Data association problem

For the data association problem with the assumptions made in Section II, the optimal filtering and smoothing Gaussian mixture posterior densities are obtained by defining DA hypotheses and applying the Gaussian solutions for each hypothesis. Each hypothesis  $\mathcal{H}_{k,n}^f$  represents a sequence of measurement

associations and/or the missed-detection events from time 1 to  $k$  (cf. Fig. 1). Analogously to  $\mathcal{H}_{k,n}^f$ , hypothesis  $\mathcal{H}_{k,j}^b$  denotes a sequence of measurement assignments from time  $K$  to  $k+1$ , where  $k < K$  (cf. Fig. 2). Under each hypothesis, the state-space model is linear and Gaussian. Therefore, it is possible to apply the Gaussian solutions for each hypothesis to obtain the components in the GMs. It should be noted that the number of possible hypotheses is given by  $M_k^f = \prod_{i=1}^k (m_i + 1)$  and  $M_k^b = \sum_{i=k+1}^K (m_i + 1)$ , where  $m_i$  is the number of measurements at time  $i$ . Throughout the remainder of the paper, we consider smoothing for the DA problem .

### C. Forward filtering of FBS and TFS

Using the hypothesis definitions, the filtering density ([5, 15, 17, 21]) is written as

$$p(x_k | Z_{1:k}) = \sum_{n=1}^{M_k^f} \Pr\left\{\mathcal{H}_{k,n}^f | Z_{1:k}\right\} p(x_k | Z_{1:k}, \mathcal{H}_{k,n}^f) \quad (8)$$

$$= \sum_{n=1}^{M_k^f} w_{k,n}^f \mathcal{N}\left(x_k; \mu_{k,n}^f, P_{k,n}^f\right), \quad (9)$$

where  $w_{k,n}^f$  is the hypothesis probability and  $\mu_{k,n}^f$  and  $P_{k,n}^f$  are the mean and covariance of the  $n^{\text{th}}$  component respectively.

In the optimal case, the number of components  $M_k^f$  increase exponentially as  $k$  increases. Therefore, to reduce the computational burden, Gaussian mixture reduction techniques such as pruning ([3], [5]) and merging ([14], [19], [24], [6]) are used for practical implementations. In pruning, components with insignificant weights are removed, and in merging, similar components are approximated by a single Gaussian density. It should be noted that retained hypotheses  $\mathcal{H}_{k,n}^f$  include sets of DA sequences after merging.

A hypothesis tree is a useful tool to illustrate how the hypotheses propagate over time. In Fig. 1 and Fig. 2, hypothesis trees for FBS and TFS under pruning approximations are shown. The nodes represent components in the GM, and the branches show the relation between the components across time.

### D. Backward smoothing of FBS

The optimal BS for FBS involves GM division as in (4), which is difficult to compute directly. This difficulty can be overcome by running Rauch-Tung-Striebel iterations under the hypotheses  $\mathcal{H}_{K,a}^f$  [12]. Using the hypotheses

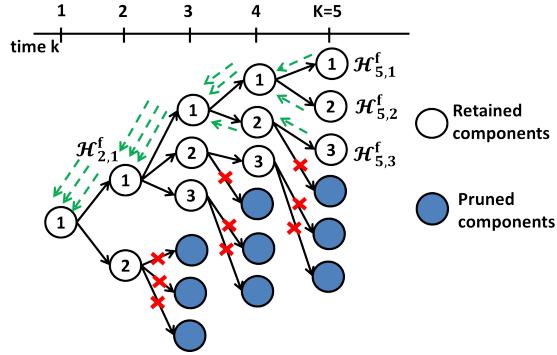


Fig. 1: Illustration of forward-backward smoothing: In this example  $K = 5$ . The nodes correspond to the GM components in the FF. The black solid lines represent the relation between the GM components across time, during FF. At each  $k$ , at most 3 components are retained, as a result of pruning. The green dashed lines represent the paths taken during BS. It can be seen that surviving hypotheses  $\mathcal{H}_{5,l}^f$  for  $l = 1, 2, 3$  at time 5, are related to the same hypothesis  $\mathcal{H}_{2,1}^f$  at time 2, resulting in degeneracy.

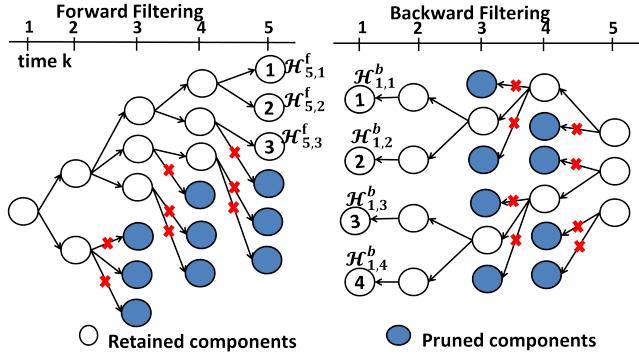


Fig. 2: Illustration of two-filter smoothing: The nodes correspond to the GM components in the filter outputs. The ‘blue nodes’ represent the pruned nodes, which are not propagated further. At each time instant, the smoothing density is obtained as the product of the GMs from the FF and BF respectively, corresponding to all the nodes in the two filters.

$\mathcal{H}_{K,a}^f$ , the smoothing density can be obtained as

$$p(x_k|Z_{1:K}) = \sum_{a=0}^{M_K^f} \Pr \left\{ \mathcal{H}_{K,a}^f | Z_{1:K} \right\} p(x_k|Z_{1:K}, \mathcal{H}_{K,a}^f). \quad (10)$$

In the optimal scenario, the history of each  $\mathcal{H}_{K,a}^f$  at time  $K$  contains a unique sequence of hypotheses  $\mathcal{H}_{k,n}^f$  for  $k = 1, \dots, K-1$ . Conversely, for each  $\mathcal{H}_{k,n}^f$

at time  $k$ , there is a hypothesis  $\mathcal{H}_{K,a}^f$  that has an identical DA sub-sequence from time 1 to  $k$ . BS uses this relation between  $\mathcal{H}_{K,a}^f$  and  $\mathcal{H}_{k,n}^f$  and Rauch-Tung-Striebel (RTS) iteration is performed using the sequence of filtering density components corresponding to the hypotheses  $\mathcal{H}_{k,n}^f$  from  $k = 1 \dots K$ , to obtain each term in (10). To be precise, RTS iteration is recursively applied to the smoothing density component at  $k+1$  under hypothesis  $\mathcal{H}_{K,a}^f$  and the filtering density component under  $\mathcal{H}_{k,n}^f$  at  $k$ , to obtain the smoothing density component at  $k$ . The BS is also illustrated in the hypothesis tree in Fig. 1.

When the FF is based on pruning, the RTS iterations of BS are applied on the surviving hypotheses  $\mathcal{H}_{K,a}^f$ . The number of hypothesis  $\mathcal{H}_{K,a}^f$  is fewer and manageable. As a consequence of pruning, it happens that in the history of  $\mathcal{H}_{K,a}^f \forall a$ , there exists only one hypothesis  $\mathcal{H}_{k,n}^f$  for  $k \ll K$  (see Fig. 1). This phenomenon is called degeneracy and leads to underestimated covariances.

1) *Backward filtering of TFS*: The optimal BF of TFS [11] is performed similar to the FF. Using DA hypotheses  $\mathcal{H}_{k,j}^b$ , the BL from the BF is written as

$$p(Z_{k+1:K}|x_k) = \sum_{j=1}^{M_k^b} \Pr\left\{\mathcal{H}_{k,j}^b | Z_{k+1:K}\right\} \times p(x_k | Z_{k+1:K}, \mathcal{H}_{k,j}^b) \quad (11)$$

$$= \sum_{j=1}^{M_k^b} w_{k,j}^b \mathcal{N}\left(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b\right). \quad (12)$$

There is a one-to-one correspondence between the terms in (12) and (11). The parameters in (11) can be obtained using the extended observation model described in Section 3 of [11] or from Section IIC of [22]. Each  $j^{\text{th}}$  component is obtained by performing Gaussian recursions for the BF as in [8], under the hypothesis  $\mathcal{H}_{k,j}^b$ . These Gaussian recursions in the BF involve the update and retrodiction steps similar to the update and prediction steps in the Kalman filter. Note that when the smoothing density at  $k$  is computed using the outputs of the FF and the BF, hypotheses  $\mathcal{H}_{k,n}^f$  and  $\mathcal{H}_{k,j}^b$  together form one DA hypothesis  $\mathcal{H}_{K,a}^f$  from time 1 to  $K$ .

The BF of TFS is also illustrated using a graphical structure, as shown in Fig. 2. The nodes and branches represent the components in the BL and their relations across time, similar to the FF.

In the optimal scenario, the number of components  $M_k^b$  increases exponentially with time. Thus, approximations are necessary to reduce the computational complexity. The difficulty is that the components in the BL are not Gaussian densities in  $x_k$ . Therefore, the GM reduction techniques

---

mentioned in Section III-C for the FF, are not directly applicable to the BF, as those techniques are developed for GM with components that are densities in  $x_k$ .

One good aspect of the TFS is that it does not suffer from the degeneracy problem even when the FF and the BF are based on pruning. This is because the smoothing density in TFS is computed based on the components corresponding to the surviving hypotheses  $\mathcal{H}_{k,n}^f$  in the FF and  $\mathcal{H}_{k,j}^b$  in the BF at time  $k$ , whereas in the FBS, only the hypotheses  $\mathcal{H}_{k,i}^f$  which are in the history of  $\mathcal{H}_{K,a}^f$  are used.

#### IV. FORWARD-BACKWARD SMOOTHING

In the previous section, it was discussed how optimal Gaussian mixture smoothing using the forward-backward smoothing strategy can be performed. It was also discussed how approximations such as merging and pruning are employed during forward filtering. 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 backward smoothing 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 FF. During BS, each component in the smoothing density in (10) is obtained by using a Rauch-Tung-Striebel smoother, which combines the smoothing density component with hypothesis  $\mathcal{H}_{K,a}^f$  and the filtering density component with hypothesis  $\mathcal{H}_{k,n}^f$ , if the DA sub-sequence in  $\mathcal{H}_{K,a}^f$  from time 1 to time  $k$  is identical to the DA sequence in  $\mathcal{H}_{k,n}^f$ . In other words, it combines a component  $p(x_{k+1}|Z_{1:K}, \mathcal{H}_{K,a}^f)$  in the smoothing density at time  $k+1$  and a component  $p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f)$  in the filtering density at  $k$ , such that  $p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f) = p(x_k|Z_{1:k}, \mathcal{H}_{K,a}^f)$ , and returns a component  $p(x_k|Z_{1:K}, \mathcal{H}_{K,a}^f)$  in the smoothing density at  $k$ . It should be noted that due to pruning during FF, the number of filtering hypotheses  $\mathcal{H}_{K,a}^f$  at time  $K$  is manageable. Therefore, the number of components in the smoothing density  $p(x_k|Z_{1:K})$  in (10) 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  $\mathcal{H}_{k,n}^f$  represent. In the former, as a result of merging during the FF, the hypotheses  $\mathcal{H}_{K,a}^f$  are sets of DA sequences, whereas in the latter,  $\mathcal{H}_{K,a}^f$  corresponds to one DA sequence. As each DA sequence in the set  $\mathcal{H}_{K,a}^f$  corresponds to a Gaussian component,

the term  $p(x_k|Z_{1:K}, \mathcal{H}_{K,a}^f)$  in (10) represents a GM in the merging-based setting. It is therefore not obvious how to use the RTS algorithm for the merged hypotheses  $\mathcal{H}_{K,a}^f$ . The idea in this paper is that the DA sequences in each hypothesis  $\mathcal{H}_{K,a}^f$  can be partitioned to form hypotheses  $\mathcal{H}_{k,l}^s$  such that  $p(x_k|Z_{1:K}, \mathcal{H}_{k,l}^s)$  is a Gaussian density. During the BS, each of these hypotheses  $\mathcal{H}_{k,l}^s$  can be related to a hypothesis  $\mathcal{H}_{k,n}^f$  from the FF, enabling us to employ RTS recursions on these new hypotheses  $\mathcal{H}_{k,l}^s$ . 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 GM reduction during the BS. To represent these hypotheses  $\mathcal{H}_{k,l}^s$  and the GMR of the components in the smoothing density, we use a hypothesis graph called the s-graph.

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

$$p(x_k|Z_{1:K}) = \sum_{l=0}^{M_k^s} p(x_k|Z_{1:K}, \mathcal{H}_{k,l}^s) \Pr\{\mathcal{H}_{k,l}^s|Z_{1:K}\}, \quad (13)$$

where  $p(x_k|Z_{1:K}, \mathcal{H}_{k,l}^s)$  is the Gaussian density  $\mathcal{N}(x_k; \mu_{k,l}^s, P_{k,l}^s)$ , with weight  $w_{k,l}^s = \Pr\{\mathcal{H}_{k,l}^s|Z_{1:K}\}$ . Starting with  $\mathcal{H}_{K,n}^s = \mathcal{H}_{K,n}^f$  at  $k = K$ , the hypotheses  $\mathcal{H}_{k,l}^s$  (partitioned from  $\mathcal{H}_{K,a}^f$ ) can be obtained recursively by defining the hypotheses  $\mathcal{H}_{k+1,p}^s$  at time  $k + 1$  and  $\mathcal{H}_{k,n}^f$ , 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  $\cup$  and  $\cap$  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  $n = 1, \dots, M_k^f$  in the f-graph (after pruning and merging of the filtering GM) at time  $k$ , the following parameters are defined:

- $\mathcal{H}_{k,n}^f$ , the hypothesis that corresponds to node  $n$  (after merging). If  $\mathcal{H}_{k,(i,j)}^f$  represent the set of disjoint hypotheses formed by associating hypothesis  $\mathcal{H}_{k-1,i}^f$  of node  $i$  at time  $k - 1$  to measurement  $z_{k,j}$  at time  $k$ ,

---

and they correspond to the Gaussian components that have been merged to form the component at node  $n$ , then

$$\mathcal{H}_{k,n}^f = \bigcup_{i,j} \mathcal{H}_{k,(i,j)}^{'f}. \quad (14)$$

Note that hypotheses  $\mathcal{H}_{k,(i,j)}^{'f}$  do not represent any node in the f-graph. However, they can be associated to the branches incident on node  $n$  before merging. The prime in the notation of  $\mathcal{H}_{k,(i,j)}^{'f}$  is to indicate that it is the hypothesis before merging. Similar notation will be used in the s-graph as well.

- $\mu_{k,n}^f$  and  $P_{k,n}^f$  are the mean and the covariance of the Gaussian density  $p(x_k|Z_{1:k}, \mathcal{H}_{k,n}^f)$  (after merging).
- $I_{k,n}^f$  is a vector that contains indexes  $i$  of the hypotheses  $\mathcal{H}_{k-1,i}^f$  at time  $k-1$ . An interpretation of this is that for each  $i$  in  $I_{k,n}^f$ , there is a branch between node  $i$  at time  $k-1$  and node  $n$  at time  $k$ .
- $w_{k,n}^f$  is a vector that contains the probabilities  $\Pr\{\mathcal{H}_{k,(i,j)}^{'f}|Z_{1:k}\}$  of the DA hypotheses  $\mathcal{H}_{k,(i,j)}^{'f}$  before merging. Using (14), it can be shown that  $\Pr\{\mathcal{H}_{k,n}^f|Z_{1:k}\} = \sum_{i,j} \Pr\{\mathcal{H}_{k,(i,j)}^{'f}|Z_{1:k}\}$ .

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

2) *s-graph*: At time  $k$ , the s-graph parameters corresponding to the  $l^{\text{th}}$  component of the smoothing density in (13) are:

- $\mathcal{H}_{k,l}^s, \mu_{k,l}^s$  and  $P_{k,l}^s$  are the hypothesis, mean and covariance of the  $l^{\text{th}}$  Gaussian component (after merging).
- $w_{k,l}^s$  is the probability  $\Pr\{\mathcal{H}_{k,l}^s|Z_{1:K}\}$ .
- $I_{k,l}^s$  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:  $\mathcal{H}_{K,l}^s = \mathcal{H}_{K,l}^f, \mu_{K,l}^s = \mu_{K,l}^f, P_{K,l}^s = P_{K,l}^f, w_{K,l}^s = \sum_r w_{K,l}^f(r)$  and  $I_{K,l}^s = l$ . 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  $\mathcal{H}_{k,l}^s$  are used to explain the

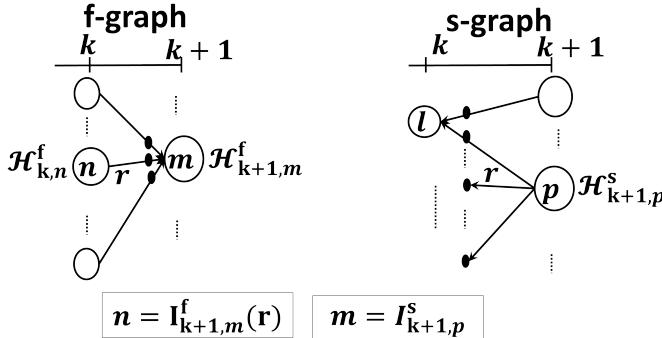


Fig. 3: Illustration of backward smoothing on a merged graph: A node  $p$  in the s-graph is shown along with the corresponding node  $n$  in the f-graph. The relations between the different parameters are as indicated. The filled black circles in the f-graph and the s-graph represent the components before merging.

weight calculations. But, for implementation purposes, it suffices to update and store the parameters  $\mu_{k,l}^s$ ,  $P_{k,l}^s$ ,  $w_{k,l}^s$  and  $I_{k,l}^s$ .

### 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  $p^{\text{th}}$  component of  $p(x_{k+1}|Z_{1:K})$  and the  $n^{\text{th}}$  component of  $p(x_k|Z_{1:k})$ . Whether a pair  $(p, n)$  depends on if the hypothesis  $\mathcal{H}_{k,n}^f$  is in the history of the hypothesis  $\mathcal{H}_{k+1,p}^s$  or not. This information can be inferred from the relation between the f-graph and the s-graph.

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 analyzed using the parameters listed in Section IV-A (cf. Fig. 3). 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  $m = I_{k+1,p}^s$ . 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  $I_{k+1,m}^f$  contains the parent node index  $n$ , then the pair  $(p, n)$  can be formed. See Fig. 3 for an illustration. In fact, for every element  $n$  in the vector  $I_{k+1,m}^f$ , 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 (13), for which we now wish to compute the mean, covariance and weight using an RTS iteration and the hypothesis relations. The hypotheses involved in obtaining the component are  $\mathcal{H}_{k+1,p}^s$ ,  $\mathcal{H}_{k+1,m}^f$  and  $\mathcal{H}_{k,n}^f$ , where  $m = I_{k+1,p}^s$  as discussed before and node  $n$  is, say, the  $r^{\text{th}}$  element in the vector  $I_{k+1,m}^f$  denoted  $n = I_{k+1,m}^f(r)$  (cf. Fig. 3). Using these hypotheses, the hypothesis, corresponding to the resulting component, denoted  $\mathcal{H}_{k,(p,n)}^{fs}$ , is written as

$$\mathcal{H}_{k,(p,n)}^{fs} = \mathcal{H}_{k+1,p}^s \bigcap \mathcal{H}_{k,n}^f. \quad (15)$$

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

$$\begin{aligned} \Pr \left\{ \mathcal{H}_{k,(p,n)}^{fs} | Z_{1:K} \right\} &\propto \Pr \left\{ \mathcal{H}_{k+1,p}^s | Z_{1:K} \right\} \\ &\quad \times \Pr \left\{ \mathcal{H}_{k+1,m}^f \bigcap \mathcal{H}_{k,n}^f | Z_{1:k+1} \right\} \\ &= w_{k+1,p}^s w_{k+1,m}^f(r). \end{aligned} \quad (16)$$

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  $I_{k+1,n}^f$ , 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  $\mathcal{H}_{k,n}^f$ , 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  $\mathcal{H}_{k,(p,n)}^{fs}$ , for different  $(p, n)$ , the retained hypothesis are relabeled as  $\mathcal{H}_{k,l}^s$ , and the corresponding components form the nodes  $l$  in the s-graph at time  $k$ .

## V. TWO-FILTER SMOOTHING

In this section, we propose approximation techniques for the backward filtering of the two-filtering smoothing. First technique, called intragroup approximation, is obtained by analyzing the details of the structure of the backward likelihood. In the second technique, called smoothed posterior pruning, we propose that the filtering densities can be used to make pruning on the BL of the BF.

### A. Intragroup approximations

In this section, we analyze the structure of the backward likelihood and discuss why the conventional pruning and merging strategies are not always suitable for the BL. Based on this analysis, we show that the BL has a reduced dimension Gaussian mixture structure. The components in the BL can be grouped and the traditional Gaussian mixture reduction techniques can be used within the groups. It will also be shown that the number of groups is polynomial in the lag  $K - k$ , and that this can be a limitation of the intragroup GMR strategies, especially when the lag  $K - k$  is large.

1) *Normalizability of the BL*: Consider the BL in (12), which we refer to as the reduced dimension Gaussian mixture (RDGM) in this paper. It is similar to a Gaussian mixture, with weights  $w_{k,j}^b$ , means  $\mu_{k,j}^b$  and covariances  $P_{k,j}^b$ . However, the terms  $\mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b)$  are not generally densities in  $x_k$  and are not guaranteed to be normalizable since the integral  $\int \mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b) dx_k$  may be infinite.

Similar to the GM filtering density in the forward filtering (FF), the number of terms  $M_k^b$  in the RDGM in (12) of the BF grows exponentially with time. Thus, GMR approximations are necessary to reduce the complexity. The catch is that the conventional GMR techniques mentioned in Section III-C are developed for GM density functions and cannot be applied directly to RDGMs. For instance, the conventional pruning strategies are applied to GMs that are normalized, in which the weights of the components define the relative sizes of the Gaussian components. So, the weights of the components can be compared and the ones with insignificant weights can be removed. However, in the RDGM in (12), it is possible that a component with the smallest weight  $w_{k,j}^b$  is in fact among the largest components. So, the components in the RDGM cannot be compared based only on the weights unless the components  $\mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b)$  are normalized. Similarly, the conventional merging strategies applied to GMs involve moment matching, and to compute the moments, the components should be normalizable densities. In the trivial case, when the components in the RDGM are normalizable, i.e., when  $\text{rank}(H_{k,j}^b) = N_x$ , it is possible to rewrite the components in the RDGM  $\sum_j w_{k,j}^b \mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b)$  into the GM form  $\sum_j w_{k,j}'^b \mathcal{N}(x_k; \mu_{k,j}'^b, P_{k,j}'^b)$  to which the GMR techniques can be applied.

The RDGM is in general not normalized, like in the data association problem, and the normalizability depends on the structure of the matrices  $H_{k,j}^b$ .

---

As will be shown in Section V-A2, in the RDGM, groups of components have identical  $H_{k,j}^b$  matrices. Thus, if the matrices  $H_{k,j}^b$  are also of full row rank in a group, then it is possible to compare the components within the group, and so, one can apply the GMR technique to that group.

Let  $I_g$  be the index set that contains the indexes  $j$  of the components in a group  $g$  that have the same matrices, i.e.,  $H_{k,j}^b = H_g$  for  $j \in I_g$  and let  $\text{rank}(H_g) = m_g$ . Using this notation, the RDGM in (12) can be written as

$$p(Z_{k+1:K}|x_k) = \sum_{g=1}^{N_G} \sum_{j \in I_g} w_{k,j}^b \mathcal{N}\left(H_g x_k; \mu_{k,j}^b, P_{k,j}^b\right)$$

where  $N_G$  is the number of groups. The functions  $\mathcal{N}\left(H_g x_k; \mu_{k,j}^b, P_{k,j}^b\right)$  are scaled Gaussian functions in the row space of the matrix  $H_g$ , and have constant values in the null space of  $H_g$ . The idea is that GMR can be applied to approximate the group  $\sum_{j \in I_g} w_{k,j}^b \mathcal{N}\left(H_g x_k; \mu_{k,j}^b, P_{k,j}^b\right)$  of components in this row space and that no approximation needs to be performed in the null space, as the values of the components are constant in the null space. An interpretation of this can be obtained using a change of basis matrix, say  $A_g = \begin{bmatrix} H_g \\ H_g^\perp \end{bmatrix}$  in  $\mathbb{R}^{N_x}$ , where the matrix  $H_g^\perp$  is such that the columns and rows of  $A_g$  span  $\mathbb{R}^{N_x}$ . Let us define a variable  $y_g = A_g x_k$  with respect to the new basis in  $\mathbb{R}^{N_x}$ . Using this variable  $y_g$ , the function  $w_{k,j}^b \mathcal{N}\left(H_g x_k; \mu_{k,j}^b, P_{k,j}^b\right)$  is written as  $w_{k,j}^b \mathcal{N}\left(y_g[1:m_g]; \mu_{k,j}^b, P_{k,j}^b\right)$ , where  $y_g[1:m_g]$  denotes the first  $m_g$  elements in the vector  $y_g$ . Clearly, the functions  $\mathcal{N}\left(y_g[1:m_g]; \mu_{k,j}^b, P_{k,j}^b\right)$  are Gaussian in the variable  $y_g[1:m_g]$ . These functions are not integrable in  $y_g[m_g+1:N_x]$ , but are constant (or uniform) in that variable. This observation allows us to treat the group  $\sum_{j \in I_g} w_{k,j}^b \mathcal{N}\left(y_g[1:m_g]; \mu_{k,j}^b, P_{k,j}^b\right)$  as a GM in the variable  $y_g[1:m_g]$ . So, pruning and merging can be applied to approximate this mixture as a function of  $y_g[1:m_g] = H_g x_k$ . Overall, within each group  $g$ , the GMR can

be applied as

$$\begin{aligned}
 & \sum_{g=1}^{N_G} \sum_{j \in I_g} w_{k,j}^b \mathcal{N} \left( y_g[1:m_g]; \mu_{k,j}^b, P_{k,j}^b \right) \\
 & \approx \sum_{g=1}^{N_G} \sum_{j \in I'_g} \tilde{w}_{k,j}^b \mathcal{N} \left( y_g[1:m_g]; \tilde{\mu}_{k,j}^b, \tilde{P}_{k,j}^b \right) \\
 & = \sum_{g=1}^{N_G} \sum_{j \in I'_g} \tilde{w}_{k,j}^b \mathcal{N} \left( H_g x_k; \tilde{\mu}_{k,j}^b, \tilde{P}_{k,j}^b \right)
 \end{aligned} \tag{17}$$

where  $I'_g$  refers to the index set corresponding to components in group  $g$  after GMR. From (17), it can be noticed that after the intragroup approximations, the number of components in the RDGM is at least the number of groups,  $N_G$ .

2) *Grouping:* As discussed earlier in this section, the grouping of the components in the BL in (12) is key in being able to apply the GMR techniques to approximate the RDGM. In this section, with the data association in target tracking as an example, we discuss why the grouping is possible. For this example, we also analyze how the number of groups grows with time to point out the limitation of the intragroup GMR methods.

By investigating (12) for the DA problem, one can see that many of the components in (12) have the same  $H_{k,j}^b$  matrices. It turns out that if two DA hypotheses,  $\mathcal{H}_{k,i}^b$  and  $\mathcal{H}_{k,j}^b$ , are such that they have measurement associations and missed-detection associations at the same time instants, then  $H_{k,i}^b = H_{k,j}^b$ . This observation allows us to partition the components in the RDGM into groups of components that have the same  $H_{k,j}^b$  matrices.

Consider the constant velocity model, in which the state contains the position and velocity components, and assume that only the position components are observed. One then needs measurements from at least two different time instants to estimate the velocity components. This implies that under the corresponding DA hypotheses  $\mathcal{H}_{k,j}^b$  in the BF, which have at least two measurement associations across time, the state can be estimated with finite covariance. So, the corresponding components  $\mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b)$  in the BL, will be normalizable, and therefore, the ranks of the corresponding matrices  $H_{k,j}^b$  is  $N_x = \dim(x_k)$ . These normalizable components will form one group. Additionally, there will also be components with  $H_{k,j}^b$  such that  $\text{rank}(H_{k,j}^b) < N_x$ , which correspond to the hypothesis sequences  $\mathcal{H}_{k,j}^b$  that have less than two measurement associations across time.

---

We will now analyze how the number of groups grows with time. Let us assume that the matrices  $H$  and  $F$  in the motion model (1) and measurement model (2) are such that measurements from at least  $M$  different time instants (not necessarily consecutive) are needed to estimate all dimensions of the state  $x_k$ , i.e., to ensure that the corresponding component in the BL is normalizable. Using this parameter  $M$  and the time lag  $K - k$ , the number of groups can be analyzed. As discussed for the constant velocity model with positional measurements, the components corresponding to the hypothesis sequences that have at least  $M$  measurement associations across time will be normalizable and form a group. Components that have hypothesis sequences with exactly  $l$  ( $l < M$ ) measurement associations at the same time instants belong to the same group. There are  $\binom{K-k}{l}$  ways of assigning  $l$  measurement associations in  $K - k$  time steps for each  $l = 0, 1, \dots, M-1$ . Thus, in total, the number of groups is given by

$$N_G(k, K) = 1 + \sum_{l=0}^{M-1} \binom{K-k}{l} \quad (18)$$

which grows in the order of  $\mathcal{O}\left((K-k)^{M-1}\right)$  as a function of the lag  $K - k$ .

One can see that using the intragroup approximations, the number of components in the RDGM cannot be reduced to less than the number of groups, which grows as  $\mathcal{O}\left((K-k)^{M-1}\right)$  with the lag  $K - k$ . Hence, these intragroup approximations are not sufficient to reduce the complexity due to the RDGM structure of the BL. It is therefore essential to present a sound manner to compare components across groups in order to enable us to reduce the complexity further. In the next section, we present the smoothed posterior-based pruning with which the components in the BL can be compared and pruned.

### B. Smoothed posterior pruning

In this section, we present the smoothed posterior pruning strategy to approximate the BL. We show that it is valid to compare components across groups in the backward likelihood based on the smoothing probabilities of the corresponding hypotheses in the backward filtering. Surprisingly enough, using these probabilities, large components or even groups of components can be pruned from the BL and propagated, without affecting the smoothing densities at any time instant. The smoothing probabilities of the hypotheses in the output of the BF can be calculated using the forward filtering densities.

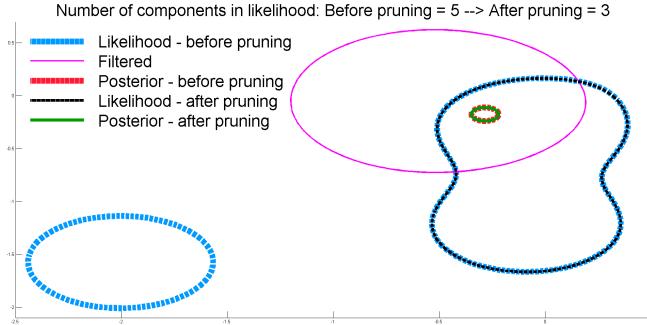


Fig. 4: Illustration of smoothed posterior pruning: The green and the red curves, which represent the posterior density before and after the approximation of the likelihood, are very close to each other. This shows that the shape of the corresponding (smoothing) posterior is unaffected by the posterior pruning of the likelihood. Also, parts of the blue curve and black curve, which represent the likelihood before and after pruning, are very close to each other.

We also show how this idea can be generalized for pruning components from the FF as well.

The main objective is to compute the smoothing posterior density given in (3). The two terms in the right-hand side of (3) are the filtering density and the backward likelihood, which are the outputs of the FF and BF, respectively. The filtering density is reduced using traditional Gaussian mixture reduction techniques. The BL can be reduced using the intragroup approximations discussed before, but that may not be enough due to its limitations discussed in the previous section. We propose (and later prove) that the filtering density of the FF in (3) can be used to prune components in the BL of the BF, in regions where the smoothing posterior density is small, and therefore is not affected by this pruning. Fig. 4 shows an illustration of this strategy, referred to as smoothed posterior pruning, for one time instant. The Gaussian mixtures are in 2D and the curves shown in the figure are the contour plots of the GMs involved. More importantly, we can propagate the pruned version of the BL backwards in time without affecting the smoothing density at any other time instant.

The SPP idea can also be used the other way around. That is, it is possible to approximate the FF based on the BL. The BL of the BF, when available, can be used to prune components in the filtering density of the FF in the regions where the smoothing density is small. The bottom line is that the SPP idea presented in this section can be generalized to both the FF and the BF, depending on what functions are available to compute the posterior. In this section, we discuss the SPP idea for the FF, but the same arguments

---

hold for SPP of BF as well.

For the pruning step in the SPP, any of the pruning strategies mentioned in Section III-C can be used, once we know the weights of the components to be compared. Note that the weights relate to the probability of the data association hypothesis sequence  $\mathcal{H}_{k,n}^f$  (cf. Section III-C). In SPP, as stated before, we want to perform the pruning based on the smoothing posterior. So, we do not use the filtering probability  $\Pr\{\mathcal{H}_{k,n}^f | Z_{1:k}\}$  of the hypothesis. Instead, we use the smoothing probability  $\Pr\{\mathcal{H}_{k,n}^f | Z_{1:K}\}$  of the hypothesis. As we will show in the following proposition, if the smoothing posterior probability  $\Pr\{\mathcal{H}_{k,n}^f | Z_{1:K}\}$  is zero, then the  $n^{\text{th}}$  Gaussian component can be pruned from the filtering density, without affecting the smoothing posterior distributions. The intuitive reason is that the smoothing probabilities of the offspring of the pruned component, if it was propagated, would also be zero, which means that they would not influence the smoothing densities at later times either. In the following proposition, we also provide the mathematical justification for the same.

**Proposition 1.** *Suppose that the objective is to compute the smoothing density  $p(x_m | Z_{1:K})$  for  $m = 1, \dots, K$ . If  $\Pr\{\mathcal{H}_{k,n}^f | Z_{1:K}\} = 0$ , then the  $n^{\text{th}}$  component can be pruned from  $p(x_k | Z_{1:k})$  and propagated to time  $k+1$  during FF without affecting the smoothing density  $p(x_m | Z_{1:K})$  for  $m \geq k$ .*

*Analogously, if  $\Pr\{\mathcal{H}_{k,j}^b | Z_{1:K}\} = 0$ , then the  $j^{\text{th}}$  component can be pruned from  $p(Z_{k+1:K} | x_k)$  and propagated to time  $k-1$  during BF without affecting the smoothing density  $p(x_m | Z_{1:K})$  for  $m \leq k$ .*

*Proof:* See Appendix B. ■

The smoothing probability of a hypothesis  $\mathcal{H}_{k,n}^f$  in the FF can be evaluated from the weights of the components in the smoothing density using (27) in Appendix B:

$$\Pr\{\mathcal{H}_{k,n}^f | Z_{1:K}\} = \int \Pr\{\mathcal{H}_{k,n}^f | Z_{1:k}\} p(x_k | \mathcal{H}_{k,n}^f, Z_{1:k}) p(Z_{k+1:K} | x_k) dx_k$$

$$\begin{aligned}
&= \int w_{k,n}^f \mathcal{N}(x_k; \mu_{k,n}^f, P_{k,n}^f) \\
&\quad \times \sum_j w_{k,j}^b \mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b) dx_k \\
&= \sum_j w_{k,n}^f w_{k,j}^b \int \mathcal{N}(x_k; \mu_{k,n}^f, P_{k,n}^f) \\
&\quad \times \mathcal{N}(H_{k,j}^b x_k; \mu_{k,j}^b, P_{k,j}^b) dx_k \\
&= \sum_j w_{k,n}^f w_{k,j}^b \\
&\quad \times \mathcal{N}(H_{k,j}^b \mu_{k,n}^f; \mu_{k,j}^b, H_{k,j}^b P_{k,n}^f H_{k,j}^{b^T} + P_{k,j}^b). \tag{19}
\end{aligned}$$

The product of the three terms in the summation of (19) is indeed the weights of the components in the smoothing density  $p(x_k|Z_{1:K})$ , obtained as the product of the  $n^{\text{th}}$  component in the filtering density and the BL. Similarly, the smoothing probability for the hypothesis  $\mathcal{H}_{k,j}^b$  during BF can be calculated from the smoothing posterior density as

$$\begin{aligned}
\Pr \left\{ \mathcal{H}_{k,j}^b | Z_{1:K} \right\} &= \sum_n w_{k,n}^f w_{k,j}^b \times \\
&\quad \mathcal{N}(H_{k,j}^b \mu_{k,n}^f; \mu_{k,j}^b, H_{k,j}^b P_{k,n}^f H_{k,j}^{b^T} + P_{k,j}^b). \tag{20}
\end{aligned}$$

Note that the summation is over the index  $n$  of the filtering density. Analogously, the product in the summation gives the weights of the components in the smoothing density obtained due to product of the  $j^{\text{th}}$  component in the BL and the filtering density.

## VI. ALGORITHMIC DESCRIPTION

The algorithmic descriptions of the FBS and the TFS, using the ideas proposed in Section IV and Section V, respectively, 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:  $\beta$ ,  $P_D$ ,  $P_G$ ,  $P_P^f$  and  $P_P^s$ .

The forward filtering algorithm is common to both smoothing methods and involves both pruning and merging approximations. It is presented in Algorithm 1. The f-graph parameters described in Section IV are updated during the FF. During the BS of FBS, the s-graph parameters in the list in Section IV are computed (cf. Algorithm 2). The BF algorithm of the TFS is described in Algorithm 3. The intragroup approximations and smoothed posterior pruning described in Section V are included in the BF algorithm.

---

The smoothing density will be obtained as part of the SPP performed in the BF as in step 5 of Algorithm 3.

---

**Algorithm 1** Forward filtering

---

**Input:** Prior:  $\mu_{0|0}$ ,  $P_{0|0}$ .

Likelihoods:  $H$ ,  $z_{k,j}$ ,  $R$  and

$$\beta_{k,j} = \begin{cases} \beta(1 - P_D P_G) & j = 0 \\ P_D & j \neq 0 \end{cases}, \text{ for } j = 0, \dots, m_k, k = 1, \dots, K.$$

**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  $w_{LL,(i,j)} = \mathcal{N}(z_{k,j}; H\mu_{k|k-1,i}, HP_{k|k-1,i}H^T + R) > P_G$  and add  $G = G \cup \{(i, j)\}$  for the pairs that pass the threshold.
  - 3) **Pruning:** Initialize  $P = \{\}$ . For each pair  $(i, j) \in G$ , calculate the posterior weight  $w_{k,(i,j)} = w_{k-1,i}^f \beta_{k,j} w_{LL}(i, j)$  and re normalize. Check if  $w_{k,(i,j)} > P_P^f$  and add all pairs  $(i, j)$  that pass the threshold to  $P$ , i.e., set  $P = P \cup \{(i, j)\}$ .
  - 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  $\mu_{k,n}^f$  and  $P_{k,n}^f$ , each corresponding to a node  $n$  in the f-graph. Along with performing merging of components, the merging module also returns the vectors  $I_{k,n}^f$  and  $w_{k,n}^f$  that contains the indexes  $i$  and the weights  $w_{k,(i,j)}$ , respectively, of the components that are merged to form node  $n$ .
- 

## VII. 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  $0.07 \text{ m/s}^2$ . The trajectory was generated for  $K = 40$  time steps with a sampling time of  $1 \text{ s}$ . The whole volume of the track was used for generating clutter data.

**Algorithm 2** Backward smoothing of FBS-GMM**Input:** Filtering GM parameters:  $\mu_{k,n}^f, P_{k,n}^f, w_{k,n}^f, I_{k,n}^f$ , for  $n = 1, \dots, M_k^f$ .**Initialize:** Set  $M_{K,s}^s = M_K^f$ ,  $\mu_{K,l}^s = \mu_{K,l}^f$ ,  $P_{K,l}^s = P_{K,l}^f$ ,  $I_{K,l}^s = l$  and  $w_{K,l}^s = \sum_r w_{K,l}^f(r)$  (summation is over the entire vector  $w_{K,l}^f$ ).**Iterate** for  $k = K - 1, \dots, 1$ 

- 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  $\mu_{k,n}^f, P_{k,n}^f$  and  $\mu_{k+1,p}^s, P_{k+1,p}^s$  (Note, the parameters  $\mu_{k+1,p}^s$  and  $P_{k+1,p}^s$  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 (25). 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  $w_{k|K,(p,n)} > P_P^s$  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  $\{\mu_{k,l}^s, P_{k,l}^s, w_{k,l}^s\}$  along with the parameter  $I_{k,l}^s = n$ .
- 

The values for the measurement noise  $R$ , the probability of detection  $P_D$  and the clutter intensity  $\beta$ , 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  $\beta$  were 0.0001 and 0.0002. Thus, there are 8 sets of parameters for which the simulation results are compared.

The proposed forward-backward smoothing with Gaussian mixture merging (FBS-GMM) and the two-filter smoothing algorithm with the intra-group approximations and the smoothed posterior pruning are compared with forward-backward smoothing based on an  $N$ -scan pruning algorithm where the forward filtering is performed using  $N$ -scan pruning.

### B. Implementation details

During FF, the merging algorithm used in step 5 in Algorithm 1 is a variant of Salmond's algorithm [19] 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 Gaussian mixture

---

**Algorithm 3** Backward filter of TFS

**Input:** Likelihoods:  $\beta_{k,i}$ ,  $H$ ,  $z_{k,i}$  and  $R$  for  $i = 0, \dots, m_k$  and  $k = 1, \dots, K$ .

**Initialize:**  $k \leftarrow K - 1$ ,  $w_{K,j}^b \leftarrow 1$ ,  $H_{K,j}^b \leftarrow []$ ,  $\mu_{K,j}^b \leftarrow []$  and  $P_{K,j}^b \leftarrow []$  for  $j = 0$ .

**Repeat**

- 1) **Update:** for every  $i$ ,  $j$ , compute  $\eta_{k+1,l} = \beta_{k+1,i} w_{k+1,j}^b$ ,  
 $U_{k+1,l} = \begin{bmatrix} H \\ H_{k+1,j}^b \end{bmatrix}$ ,  $\psi_{k+1,l} = \begin{bmatrix} z_{k+1,i} \\ \mu_{k+1,j}^b \end{bmatrix}$  and  $G_{k+1,l} = \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & P_{k+1,j}^b \end{bmatrix}$ .  
 a) If  $\text{rank}(U_{k+1,l}) = N_x$ , then set  $\eta_{k+1,l} = \frac{\eta_{k+1,l}}{\det(U_{k+1,l})}$ ,  $\psi_{k+1,l} = U_{k+1,l}^{-1} \psi_{k+1,l}$ ,  $G_{k+1,l} = U_{k+1,l}^{-1} G_{k+1,l} U_{k+1,l}^{-T}$ , and  $U_{k+1,l} = I_{N_x}$   
 end
- 2) **Grouping:** If components  $l_1$  and  $l_2$  are such that  $U_{k+1,l_1} = U_{k+1,l_2}$ , then the components belong to the same group.
- 3) **Intragroup approximation:** Within each group, the traditional pruning and merging are performed.
- 4) **Retrodiction:** Set  $w_{k,l}^b = \eta_{k+1,l}$ ,  $\mu_{k,l}^b = \psi_{k+1,l}$ ,  $H_{k,l}^b = U_{k+1,l} F$  and  $P_{k,l}^b = U_{k+1,l} Q U_{k+1,l}^T + G_{k+1,l}$ .
- 5) **Smoothing density:** For every  $n$  and  $j$ , compute  $w_{k,l}^s$ ,  $\mu_{k,l}^s$  and  $P_{k,l}^s$  same way as in update step 1 with  $k = k - 1$ ,  $n = i$ ,  $\beta_{k+1,i} = w_{k,n}^f$ ,  $H = I_{N_x}$ ,  $z_{k+1,i} = \mu_{k,n}^f$  and  $R = P_{k,n}^f$ . Note that the rank calculated will always be  $N_x$  for this case.
- 6) **SPP:** Calculate  $w_{k,j}^{bs} = \sum_l w_{k,l}^s$  according to (20). Remove component  $j$  from the BL based on the weights,  $w_{k,j}^{bs}$  and re-normalize.
- 7)  $k \leftarrow k - 1$

**until**  $k = 1$

---

(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 smallest weights, we compute the cost of merging pairs of components and if the cost is lower than a threshold ( $0.001 \times \text{state dimension}$ ), 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 Runnalls' algorithm [18]. The additional Runnalls' algorithm is necessary to ensure that the number of components in the GM during BS is within a threshold (50 components).

In the BF of the TFS, both the intragroup approximations and the smoothed posterior pruning are used to reduce the components in the backward likelihood. The intragroup pruning is based on the maximum weight in the group. The components that have weights lesser than  $1/100^{\text{th}}$  of the maximum weight in the group are pruned. The intragroup merging is based on the variant of the Salmond's algorithm discussed above. To employ the SPP method followed by these intragroup approximations, first the smoothing density has to be computed. This involves taking the product of the filtering density GM and the RDGM BL, which is an expensive operation. So, to reduce the number of operations involved in computing this smoothing density, we reduce the filtering density GM to a single Gaussian, and compute the smoothing density using this reduced filtering density and the BL. Then, using this interim smoothing density, SPP is employed to prune components from the BL.

In case of  $N$ -scan pruning in the FBS algorithm, the parameter  $N$  for the various settings is chosen to be the largest possible  $N$  such that the complexity (run-time) for a single run is 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. This rectangular gating is also employed in the FF and the BF of TFS. Besides the model parameters, the gating probability  $P_G$  and the pruning threshold  $P_P^f$  mentioned in step 2 and step 3 of Algorithm 1 are  $(1 - 10^{-5})$  and  $10^{-4}$  respectively. The threshold  $P_P^s$  for BS in step 3 of Algorithm 2 is  $10^{-3}$ . It should be mentioned here that the most of our effort has been into optimizing the parameters for the FBS with N-scan.

The performance measures used for comparison are the RMSE, normalized estimation error squared (NEES), computational time 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 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.

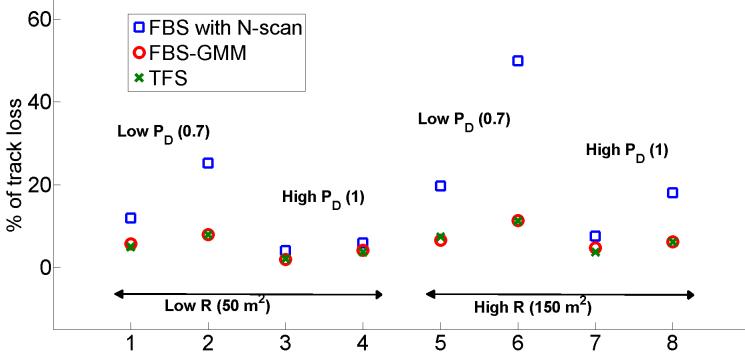


Fig. 5: Track Loss performance: Every odd point on the x-axis (1, 3, 5, 7) is for low clutter intensity  $\beta = 0.0001$  and every even point (2,4,6,8) is for high  $\beta = 0.0002$ . The order of the eight scenarios is the same also for the other plots in Fig. 6, Fig. 7 and Fig. 8.

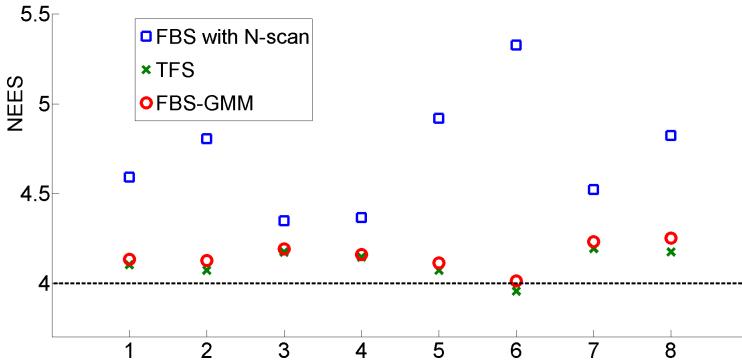


Fig. 6: NEES performance: Compared to the FBS based on  $N$ -scan pruning, the values of the NEES for the proposed FBS and TFS algorithms are very close to the optimal value of 4 in all the scenarios.

### C. Results

The results of the simulation are presented in Fig. 5 to 8. It can be seen that the proposed FBS and TFS algorithms perform significantly better than the FBS with  $N$ -scan pruning for most of the scenarios. From the Fig. 5 for track loss performance, one can notice that the performance gain is higher for the proposed algorithms compared to FBS with  $N$ -scan pruning, when  $P_D$  is low and the measurement noise  $R$  and the clutter intensity  $\beta$  are high (point 6 on the x-axis in Fig. 5). The reason for this is the degeneracy in the FBS with  $N$ -scan pruning. The impact of this degeneracy problem can

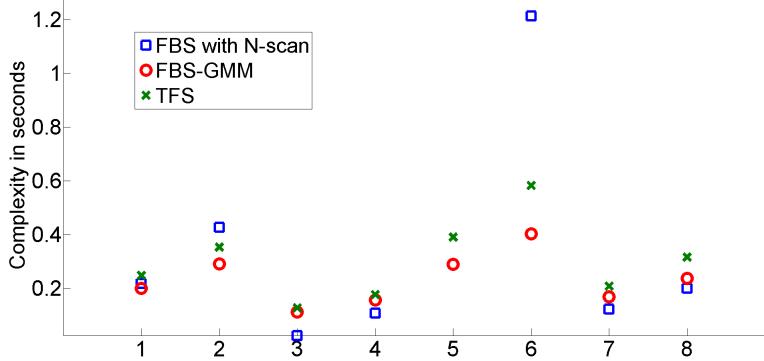


Fig. 7: Computational complexity: The proposed FBS and TFS algorithms are computationally cheaper compared to the FBS with  $N$ -scan pruning.

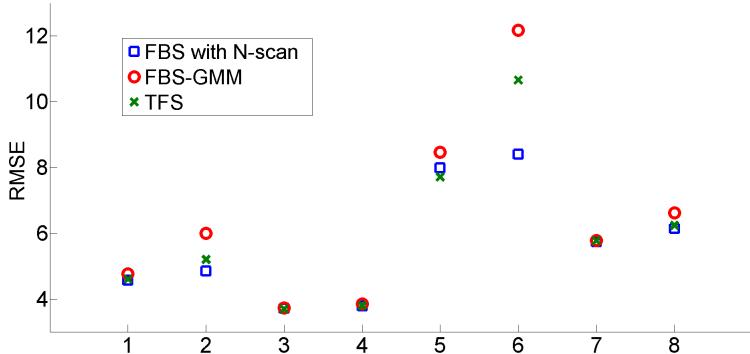


Fig. 8: RMSE performance: The results are very similar for the TFS and the FBS algorithms except for the complex scenario with low  $P_D$ , high intensity  $\beta$  and high measurement noise  $R$  (point 6 on the x-axis).

also be observed in the NEES performance plot in Fig. 6 (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 ones, resulting in a larger value for the NEES compared to the expected value of 4. In addition to the better track loss and NEES performance, TFS based on intragroup approximations and SPP, as well as FBS-GMM, offers a computationally cheaper solution compared to the FBS based on  $N$ -scan pruning as can be observed in Fig. 7. However, the RMSE performance of the TFS and FBS are very similar in most scenarios as seen in Fig. 8, except for point 6 where the track loss is high for FBS based on  $N$ -scan pruning. It should be noted that the instances when the  $N$ -scan pruning-based FBS algorithm

---

loses tracks are not considered in the calculations of the RMSE.

## VIII. DISCUSSION

The simulations for the data association problem in the single target scenarios show that the proposed two-filter smoothing and the forward-backward smoothing algorithms have very similar performance. In this section, we compare and discuss the two smoothing algorithms based on the possibilities of extending the algorithms to more complex scenarios. We also suggest some ideas on how some of the existing problems with the algorithms can be handled.

### A. Data association with multiple targets

Let us consider how the FBS and TFS can be applied to the data association problem in multiple target tracking. For simplicity, we limit our discussion to situations where the number of targets is fixed and known. The algorithms studied in this paper can be viewed as single-target multiple hypothesis tracking algorithms, and there are two common types of MHT algorithms for multiple target tracking: hypothesis-oriented MHT, [17] and track-oriented MHT [15] algorithms. The hypothesis-oriented MHT can be conveniently combined with both the algorithms that we have proposed above, where there are only global hypotheses. The popular track-oriented MHT, on the other hand, is normally combined with a pruning-based reduction technique, since merging complicates the treatment of DA conflicts between different tracks. Therefore, pruning-based FBS and TFS are applicable to the problem. However, as discussed before, this pruning-based FBS solution does not perform well due to the degeneracy problem. Fortunately, this degeneracy problem does not exist in the pruning-based TFS. To implement the pruning-based TFS, one has to run two track-oriented MHTs, one for the FF and one for BF, along with the track-conflict handling. It would be interesting to see how the two MHTs can be combined, along with their corresponding track hypotheses to obtain the smoothing density.

### B. Fixed-lag FBS smoothing to avoid degeneracy

As we have explained above, the FBS based on pruning for the fixed-interval smoothing problem suffers from the degeneracy problem. That is, the number of branches in the hypothesis tree is just one except for the latest few time steps. As BS is performed backwards in this degenerate tree, the components in the smoothing density become more and more similar,

eventually resulting in a single Gaussian density. One of the serious consequences of degeneracy is underestimated uncertainties during backward smoothing in the hypothesis tree. One approach to reduce the impact of the degeneracy in particle filter smoothing, where degeneracy is an even bigger problem, is the fixed-lag approximation idea [7]. That is, as an approximation of fixed-interval smoothing, fixed-lag smoothing can be used, in which the BS is performed only on the latest part of the hypothesis tree instead of the entire tree. In other words,  $p(x_k|Z_{1:K}) \approx p(x_k|Z_{1:k+\delta})$  for  $k \ll K$ , where  $\delta$  is chosen based on the level of degeneracy. For instance, in the FBS with  $N$ -scan pruning, the ideal choice of  $\delta$  is such that  $\delta \leq N$ . Our preliminary simulations for this  $N$ -scan pruning based FBS have shown that the NEES and the track-loss performances of the fixed-lag algorithm, which depends on the estimated covariances, is better than the fixed-interval smoothing and that the performance improves with smaller  $\delta$  such that  $\delta \leq N$ . However, the RMSE performance has shown the opposite trend. In summary, these results suggest that there is a trade-off between the estimated mean and covariance from the fixed-lag approximation. So, the choice of  $\delta$  should be made based on the requirements of the application in hand.

### C. Non-linear models

There are applications in which the mode-specific measurement and motion models are non-linear. A common technique to handle non-linearity is to linearize the functions, either analytically as in an extended Kalman filter [9] or using statistical linear regression as in, e.g., an unscented Kalman filter [23], in order to obtain a Gaussian approximation for each mode in the mixture. These techniques have been successfully combined with FBS in both Gaussian filtering settings [20] as well as in Gaussian mixture filtering.

It is possible to extend the TFS and FBS of GMs to non-linear models. This can be achieved in many ways. One way is to extend the extended Kalman filter idea, in which the functions involved are linearized. That is, by making linearizations of the functions in the process and sensor models, one can get GM densities. Traditionally, the linearization points for the FF is based on the prior density. For the BS of FBS, it is straightforward, and no extra linearization is necessary since the smoothing densities only depend on the FF outputs, which are already obtained after linearization. For the BF of the TFS, one way to choose the linearization points can be based on the components in the forward filter density. This way, there will be several linearization points for each component in the backward filter, based on each component in the forward filter. Strategies on how these several linearizations for the same component can be handled is worth investigating.

---

#### D. TFS for generic factor graphs

The FF, BS and BF recursions of FBS and TFS can be viewed as message passing algorithms in a factor graph representation [13] for the state-space model. The factor graph representation for the state-space model is a simple tree in which the state variables are all lined up in one branch. It would be interesting to extend the FBS and TFS ideas presented in this paper to factor graphs with general graphical structures, for example, to trees with nodes that have more than two branches and to graphs with loops. It is trivial to extend the idea of TFS with smoothed posterior pruning to message passing on generic graphs. For each branch in the factor graph, separate filtering recursions are run, and the messages along each branch can then be approximated based on the smoothing density before propagation.

In case of the FBS method for the state space model, a closer analysis reveals that there is a defined order in the two steps involved, i.e., first FF on the whole of the tree and then BS of messages that are computed based on the FF densities. This suggests that to extend the idea of FBS to generic graphs, one needs to choose an order in which message passing is performed in the different branches of the graph. There are normally several choices for the ordering, since there can be several leaf nodes in the factor graph. It would be interesting to investigate the advantages of one order over another. In summary, the TFS seems simpler and more straightforward to extend to general factor graphs, whereas FBS might be feasible and there are several implementation choices one has to make.

## IX. CONCLUSION

In this paper, we presented algorithms for forward-backward smoothing and two-filter smoothing on single-target Gaussian mixtures. We presented how the the forward filtering and backward smoothing steps of forward-backward smoothing can be performed using data association hypotheses including merging approximations. For the backward filtering of the two-filter smoothing, two different approximation techniques — intragroup approximation and smoothed posterior pruning — were proposed. Evaluations of root-mean squared error and track loss were performed on a simulated scenario. The results showed that both proposed algorithms perform similarly. Compared to FBS based on  $N$ -scan pruning, the proposed algorithms showed improved performance for low complexity and high credibility (normalized estimation error squared).

## REFERENCES

- [1] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Dover Publications, 2012.
- [2] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.
- [3] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House Norwood, MA, 1999, vol. 685.
- [4] M. Briers, A. Doucet, and S. Maskell, “Smoothing algorithms for state-space models,” *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010. [Online]. Available: <http://dx.doi.org/10.1007/s10463-009-0236-2>
- [5] 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.
- [6] D. Crouse, P. Willett, K. Pattipati, and L. Svensson, “A look at gaussian mixture reduction algorithms,” in *Proceedings of the fourteenth International Conference on Information Fusion*, 2011, pp. 1–8.
- [7] 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.
- [8] 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.
- [9] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Dover Publications, 2007.
- [10] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [11] G. Kitagawa, “The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother,” *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 4, pp. 605–623, 1994.
- [12] 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.
- [13] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and

- 
- the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [14] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
  - [15] T. Kurien and M. Liggins, “Report-to-target assignment in multisensor multitarget tracking,” in *Proceedings of the 27th IEEE Conference on Decision and Control*. IEEE, 1988, pp. 2484–2488.
  - [16] H. Rauch, F. Tung, and C. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA Journal*, vol. 3, pp. 1445–1450, 1965.
  - [17] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
  - [18] 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.
  - [19] 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.
  - [20] S. Sarkka, “Unscented rauch–tung–striebel smoother,” *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, 2008.
  - [21] R. A. Singer, R. G. Sea, and K. Housewright, “Derivation and evaluation of improved tracking filter for use in dense multitarget environments,” *IEEE Transactions on Information Theory*, vol. 20, no. 4, pp. 423–432, 1974.
  - [22] B.-N. Vo, B.-T. Vo, and R. P. Mahler, “Closed-form solutions to forward–backward smoothing,” *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 2–17, 2012.
  - [23] E. A. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE, 2000, pp. 153–158.
  - [24] J. L. Williams and P. S. Maybeck, “Cost-function-based Gaussian mixture reduction for target tracking,” in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 2, 2003, pp. 1047–1054.

## APPENDIX A WEIGHT CALCULATION FOR FBS

The weight calculation in (16) can be obtained using the hypothesis definitions. Consider the hypothesis expression in (15), and the illustration in

Fig. 3. We are interested in calculating the probability of the hypothesis

$$\begin{aligned} \Pr \left\{ \mathcal{H}_{k,(p,n)}^{s'} | Z_{1:K} \right\} &= \Pr \left\{ \mathcal{H}_{k+1,p}^s \bigcap \mathcal{H}_{k,n}^f | Z_{1:K} \right\} \\ &\propto \Pr \left\{ \mathcal{H}_{k+1,p}^s | Z_{1:K} \right\} \Pr \left\{ \mathcal{H}_{k,n}^f | \mathcal{H}_{k+1,p}^s, Z_{1:k+1} \right\} \\ &\quad \times p(Z_{k+2:K} | \mathcal{H}_{k+1,p}^s, \mathcal{H}_{k,n}^f, Z_{1:k+1}). \end{aligned} \quad (21)$$

In the above equation, the factor  $\Pr \left\{ \mathcal{H}_{k+1,p}^s | Z_{1:K} \right\}$  is the weight  $w_{k+1,p}^s$ , 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  $n$ :

$$\begin{aligned} &p(Z_{k+2:K} | \mathcal{H}_{k+1,p}^s, \mathcal{H}_{k,n}^f, Z_{1:k+1}) \\ &= p(Z_{k+2:K} | \mathcal{H}_{k+1,p}^s, \mathcal{H}_{k+1,m}^f, \mathcal{H}_{k,n}^f, Z_{1:k+1}) \end{aligned} \quad (22)$$

$$\begin{aligned} &= \int p(x_{k+1} | \mathcal{H}_{k+1,p}^s, \mathcal{H}_{k+1,m}^f, \mathcal{H}_{k,n}^f, Z_{1:k+1}) \\ &\quad \times p(Z_{k+2:K} | \mathcal{H}_{k+1,p}^s, \mathcal{H}_{k+1,m}^f, \mathcal{H}_{k,n}^f, Z_{1:k+1}, x_{k+1}) dx_{k+1} \\ &= \int p(x_{k+1} | \mathcal{H}_{k+1,m}^f, Z_{1:k+1}) p(Z_{k+2:K} | \mathcal{H}_{k+1,p}^s, x_{k+1}) dx_{k+1}. \end{aligned} \quad (23)$$

In (22), adding the hypothesis  $\mathcal{H}_{k+1,m}^f$  to the conditional statement does not make a difference as the hypothesis  $\mathcal{H}_{k+1,p}^s$  corresponding to the entire sequence of measurements masks it; but it does make the latter equations simpler to handle. The second factor in (21) is given by

$$\begin{aligned} \Pr \left\{ \mathcal{H}_{k,n}^f | \mathcal{H}_{k+1,p}^s, Z_{1:k+1} \right\} &= \Pr \left\{ \mathcal{H}_{k,n}^f | \mathcal{H}_{k+1,m}^f, Z_{1:k+1} \right\} \\ &\propto \frac{\Pr \left\{ \mathcal{H}_{k,n}^f, \mathcal{H}_{k+1,m}^f | Z_{1:k+1} \right\}}{\Pr \left\{ \mathcal{H}_{k+1,m}^f | Z_{1:k+1} \right\}}. \end{aligned} \quad (24)$$

The numerator term in (24) is the same as the probability  $w_{k+1,m}^f(r)$  of the  $r^{\text{th}}$  branch before merging to form node  $m$ . Consolidating (23) and (24) into (21), we get that

$$\Pr \left\{ \mathcal{H}_{k,(p,n)}^{s'} | Z_{1:K} \right\} = \frac{w_{k+1,p}^s \times w_{k+1,m}^f(r)}{\sum_r w_{k+1,m}^f(r)}. \quad (25)$$

---

APPENDIX B  
PROOF FOR SMOOTHED POSTERIOR PRUNING

In the following, we sketch the proof for the first part of the proposition 1, i.e., for SPP on the FF. It is possible to derive an analogous proof for SPP on the BF.

Let us first consider how the suggested pruning of components in the FF affect the smoothing density  $p(x_k|Z_{1:K})$  at time  $k$ . At time  $k$ , the relation between the smoothing density  $p(x_k|Z_{1:K})$  and the smoothing probability  $\Pr\{\mathcal{H}_{k,n}^f|Z_{1:K}\}$  is given by

$$p(x_k|Z_{1:K}) = \sum_n p(x_k|Z_{1:K}, \mathcal{H}_{k,n}^f) \times \Pr\{\mathcal{H}_{k,n}^f|Z_{1:K}\} \quad (26)$$

$$\propto \sum_n \Pr\{\mathcal{H}_{k,n}^f|Z_{1:k}\} p(x_k|\mathcal{H}_{k,n}^f, Z_{1:k}) \times p(Z_{k+1:K}|x_k) \quad (27)$$

where the product  $\Pr\{\mathcal{H}_{k,n}^f|Z_{1:k}\} p(x_k|\mathcal{H}_{k,n}^f, Z_{1:k})$  is the  $n^{\text{th}}$  component in the filtering density GM  $p(x_k|Z_{1:k})$ . Clearly, the  $n^{\text{th}}$  component in the summation of (26), is due to the product of the  $n^{\text{th}}$  component of the filtering density  $p(x_k|Z_{1:k})$  and the BL  $p(Z_{k+1:K}|x_k)$  as in (27). We note that the  $\Pr\{\mathcal{H}_{k,n}^f|Z_{1:K}\} = 0$ , for some value of  $n$ , implies that the  $n^{\text{th}}$  component can be removed from the filtering density without affecting the smoothing density  $p(x_k|Z_{1:K})$  at  $k$ .

We are now left to prove that the suggested pruning technique does not affect the smoothing density  $p(x_m|Z_{1:K})$  for  $m = k + 1, \dots, K$ . To this end, we use the fact that

$$\Pr\{\mathcal{H}_{k,n}^f|Z_{1:K}\} = \sum_{j \in S_{k \rightarrow m,n}} \Pr\{\mathcal{H}_{m,j}^f|Z_{1:K}\} \quad (28)$$

where the summation is over the set  $S_{k \rightarrow m,n}$  that contains the indexes of the components  $j$  at time  $m$  which are offspring of the component  $n$  at time  $k$ . So,  $\Pr\{\mathcal{H}_{k,n}^f|Z_{1:K}\} = 0 \iff \Pr\{\mathcal{H}_{m,j}^f|Z_{1:K}\} = 0 \forall j \in S_{k \rightarrow m,n}$ . Arguing as we did for pruning the  $n^{\text{th}}$  component from  $p(x_k|Z_{1:k})$  at time  $k$ , we can show that the components  $j \in S_{k \rightarrow m,n}$  with  $\Pr\{\mathcal{H}_{m,j}^f|Z_{1:K}\} = 0$  can be pruned from  $p(x_m|Z_{1:m})$ , without affecting the smoothing density  $p(x_m|Z_{1:K})$ .

