A batch algorithm for estimating trajectories of point targets using expectation maximization

IEEE Transactions on Signal Processing (ISSN: 1053-587X)

Citation for the published paper:
Rahmathullah, A. ; Selvan, R. ; Svensson, L. (2016) "A batch algorithm for estimating trajectories of point targets using expectation maximization". IEEE Transactions on Signal Processing, vol. 64(18), pp. 4792 - 4804.

http://dx.doi.org/10.1109/TSP.2016.2572048

A batch algorithm for estimating trajectories of point targets using expectation maximization

Abu Sajana Rahmathullah, Raghavendra Selvan, Lennart Svensson

Abstract—**In this paper, we propose a strategy that is based on expectation maximization for tracking multiple point targets. The algorithm is similar to probabilistic multi-hypothesis tracking (PMHT), but does not relax the point target model assumptions. According to the point target models, a target can generate at most one measurement and a measurement is generated by at most one target. With this model assumption, we show that the proposed algorithm can be implemented as iterations of Rauch-Tung-Striebel (RTS) smoothing for state estimation, and the loopy belief propagation method for marginal data association probabilities calculation. Using example illustrations with tracks, we compare the proposed algorithm with PMHT and joint probabilistic data association (JPDA) and show that PMHT and JPDA exhibit coalescence when there are closely moving targets whereas the proposed algorithm does not. Furthermore, extensive simulations comparing the mean optimal sub-pattern assignment (MOSPA) performance of the algorithm for different scenarios averaged over several Monte Carlo iterations show that the proposed algorithm performs better than JPDA and PMHT. We also compare it to benchmarking algorithm: N-scan pruning based track-oriented multiple hypothesis tracking (TOMHT). The proposed algorithm shows a good trade-off between computational complexity and the MOSPA performance.**

Index Terms—**Data association, expectation maximisation, loopy belief propagation, probabilistic multi-hypothesis tracking (PMHT).**

I. Introduction

There are many interesting applications based on tracking multiple targets, and the range of applications are only increasing by the day. With the well justified enthusiasm in the domain of autonomous vehicular navigation [1], computer vision [2] and the classical radar based tracking applications [3], the need for tractable multi-target tracking algorithms that perform close to optimal in real-time, has risen in the last few years. The problem of multi-target tracking is widely studied [4]–[6], and newer paradigms are being investigated to solve existing bottlenecks, and to enable further possibilities [7].

The focus of this paper is on tracking multiple point targets, and one of the main challenges in such a tracking problem is to resolve uncertainties in the origin of measurements from targets [8], [9]. Discerning target-generated measurements from clutter, and finding correspondence between targets and measurements, in order to estimate target tracks in space and time forms the crux of the problem. The uncertainty in data association can be handled by maintaining data association hypotheses, which map partitions of measurement data to different targets [10], [11]. While this approach of maintaining data hypotheses might seem plausible to resolve measurement uncertainties, it scales poorly with number of targets and measurements, when aggregated over different time instants [12]. The number of hypotheses grows exponentially, and most existing multi-target tracking algorithms propose solutions which can circumvent handling massive number of data association hypotheses, making a trade-off between optimality and complexity [13].

One of the earliest multi-target tracking algorithms developed was multiple hypothesis tracking (MHT) [4], [9], [11], with a possibility of maintaining all data association hypotheses in a tree form, making it optimal and at the same time rendering it computationally intractable. To reduce complexity there have been many approximations to the original MHT, that do not maintain all possibilities in the hypothesis tree. One such common version of MHT is the N-scan pruning based track-oriented MHT (TOMHT) [12], which maintains a hypothesis tree only for the latest N scans of measurements, and prunes away all hypotheses before N scans, in effect leaving only a single data association hypothesis before N scans. While this strategy reduces complexity, low scan depths can considerably degrade performance of the algorithm.

Another strategy that has received considerable attention in the past is the joint probabilistic data association (JPDA) filter [11], [14], [15]. JPDA performs moment matching at each time instant to merge multiple hypotheses into a single hypothesis, which is equivalent to performing the filter update with composite measurements. The composite measurements are obtained by weighting measurements at each time instant with their corresponding marginal data association probabilities. Computing marginal probabilities for a high number of targets can be expensive, but there exist efficient approximate techniques like k-best hypotheses based on Murty's algorithm [11], [16], Monte Carlo sampling method [17], tree-based approach [18] and loopy belief propagation (LBP) [19] algorithm. Another disadvantage with JPDA is coalescence which is a consequence of moment matching when there are closely moving targets [20], [21].

Probabilistic multi-hypothesis tracking (PMHT) [22], [23], on the other hand, uses multiple scans of data and iteratively optimizes the maximum a posteriori (MAP) estimate of the target states, using the expectation maximization (EM) algorithm [24], [25]. While PMHT handles measurement data from all scans in its EM iterations, it does not maintain a hypothesis tree, resulting in reduced complexity when compared to MHT. Further, PMHT makes a model assumption that befits extended targets and not point targets [26], to avoid the computation of

A. S. Rahmathullah and L. Svensson are with the Signals and Systems Department, Chalmers University of Technology, Gothenburg, Sweden (emails:{sajana, lennart.svensson}@chalmers.se).

R. Selvan is with the Computer Science Department, University of Copenhagen, Copenhagen, Denmark (email:raghav@di.ku.dk)

marginal data association probabilities. This relaxation further reduces complexity, but makes PMHT susceptible to track coalescence.

In this paper, we propose a batch algorithm that uses EM [25] to obtain MAP estimates of target states, by treating the data association variable as a hidden variable, for the case of known number of targets. We show that the main steps involved in the algorithm are smoothing and computing marginal data association probabilities. It turns out that the algorithm we propose is similar to PMHT, as both are batch solutions and use EM to obtain MAP estimate of target states, but with an important difference in the model assumptions: our assumptions are consistent for point targets, meaning, each target is assumed to generate and hence be associated only to one measurement, whereas this constraint is relaxed in PMHT. Further, the need to compute marginal data association probabilities in the proposed algorithm is similar to JPDA, but differing in the fact that we do not perform moment matching. Our implementation of the algorithm uses Rauch-Tung-Striebel (RTS) smoother [27] for smoothing, and an extension to the LBP [19] to approximate the marginal probabilities. Although it is beyond the scope of the current article, the strategies used to turn PMHT into an online algorithm and to perform track-handling can be adopted to extend our algorithm.

We compare the performance of the proposed algorithm with MHT, PMHT and JPDA. We illustrate the conceptual differences with PMHT and JPDA, while comparing it to the benchmark algorithm, the N-scan pruning-based TOMHT algorithm. The proposed algorithm does not suffer from track coalescence, when there are closely moving or crossing targets, in contrast to the PMHT or the JPDA algorithms. Plus, a good initialisation can significantly improve the performance of the proposed algorithm compared to PMHT. We also compare these algorithms for different scenarios to highlight the accuracy and computational advantage of the proposed algorithm. In all the scenarios considered, the proposed algorithm shows improvements in mean optimal sub-pattern assignment (MOSPA) [28], [29] when compared to PMHT and JPDA, while being computationally cheaper than MHT.

The organization of the paper is as follows: In Section II, we present the model assumptions and main challenges addressed in this paper. In Sections III and IV, our main contributions are presented, where we elaborate on the role of expectation maximization and computing marginal probabilities in the presented algorithm.In Section V, algorithm-level details of the proposed solution are presented. In Section VI, the proposed solution is contrasted with two closely related methods, PMHT and JPDA. Performance evaluation of our solution is presented in Section VII. Finally, we present our conclusions and possible tracks for future work in Section VIII.

II. Problem formulation and background

In this section, we begin by presenting the model assumptions based on which the algorithm will be derived, followed by a discussion on the problem we are interested in solving. After that, we describe a few existing multi-target tracking algorithms, providing a context for our contribution.

A. Nomenclature

The following list consolidates the index notations used in this paper:

T	Batch duration
N_T	Number of targets
t	Time index
M_t	Number of measurements at time t
i	Target index
j	Measurement index
n	EM iteration index

B. Problem formulation

We consider multiple point targets moving in a cluttered background. The number of targets, N_T, is assumed to be known and all targets are assumed to be present in the observation region at all times. The state vector $X_{t,i}$ for target i at time t is varying according to the process model,

$$X_{t,i} = FX_{t-1,i} + V_{t,i}, \tag{1}$$

where $V_{t,i} \sim \mathcal{N}(0, Q)$. This model implies that targets move independently of other targets. We also assume we have a Gaussian prior distribution on each target i with mean $\mu_{0,i}$ and covariance $P_{0,i}$. A target is detected with probability P_D and the detected target gives a measurement according to

$$Y^{\text{target}}_{t,i} = HX_{t,i} + W_{t,i}, \tag{2}$$

where $W_{t,i} \sim \mathcal{N}(0, R)$. Thus, the measurement set Y_t at each time is the union of the target measurements (of the detected targets) and a set of clutter detections. The clutter measurements are assumed to be distributed according to a Poisson process with intensity $\beta_c V$ where V is the volume of the observation region and β_c is the clutter density per volume.

In this paper, we consider a batch problem in which we have access to the data $Y = (Y_1, \ldots, Y_T)$ and we seek to obtain the MAP estimate of the state $X = (X_{1,1}, \ldots, X_{1,N_T}, X_{2,1}, \ldots, X_{T,N_T})$, given as

$$X_{\text{MAP}} = \arg\max_X p(X|Y). \tag{3}$$

That is, the goal is to estimate the trajectories of all targets. Also, an estimate of the quality of the state estimates, often captured in a covariance matrix, is desirable.

The density involved in MAP estimation is $p(X|Y)$, and evaluating this density is tractable and straightforward if there is no uncertainty in the measurement origin, for the model assumptions described above. However, in the multi-target tracking problems, the measurement set comprises both the measurements from the targets that are detected and the clutter measurements. And the information regarding which measurements correspond to which targets is not available. To describe this uncertainty in the measurement origin, one traditional way is to introduce the data association variable $K = (k_{1,1}, \ldots, k_{T,N_T})$, where $k_{t,i} = j$ denotes that the measurement $Y_{t,j}$ was generated by the target i at time t. Note that these variables should take values that satisfy the point target assumptions, i.e., a target can generate at most one measurement, and a measurement can be assigned to at most one target. With these variables, the density of interest becomes $p(X, K|Y)$, using which the estimates can be computed. For instance, consider $X_{\text{MAP}} = \arg\max_X \sum_K p(X, K|Y)$ for MAP estimation of X. Here, introducing the data association variable makes it easier to represent the measurement uncertainty. However, the estimation problem is still intractable due to the sheer number of possibilities of K. The number of possible hypotheses, K, grows exponentially with the number of measurements and the trajectory length, thus making it intractable to marginalize or maximize with respect to K. In the remainder of this section, we will give a brief overview of some of the existing sub-optimal algorithms to estimate X. Adhering to the conventional terminology, we will refer to an instance of K as the data association hypothesis.

C. Background

For each K, the density $p(X|K, Y)$ is a Gaussian density based on our model assumptions. This implies that $p(X|Y)$ is a Gaussian mixture (GM) with exponential number of terms. Handling such mixture densities is prohibitive, and some of the sub-optimal multi-target tracking algorithms resort to estimating X by making approximations to the posterior density. In the remainder of this section, we will give a summary of a few existing algorithms, the approximations involved and their limitations. Two of the algorithms discussed here, namely, MHT and JPDA, are for filtering problems, wherein the algorithms obtain target states X_t recursively by approximating the posterior density $p(X_t|Y_{1:t})$ every time instant, where $Y_{1:t}$ denotes the set of measurements from time 1 to t. It is possible to extend these filtering algorithms to batch problems by performing smoothing on the filtered estimates. We also discuss a batch solution, PMHT, which is based on EM and is very similar to the proposed algorithm in this paper.

The JPDA algorithm, first introduced in [14], approximates the posterior density GM $p(X_t|Y_{1:t})$ as a single Gaussian using moment matching. For linear models, moment matching of the posterior density is equivalent to updating the prediction density with a single composite measurement [11] $\tilde{Y}_{t,i}$,

$$\tilde{Y}_{t,i} = \sum_{j=1}^{M_t} \omega_{t,i,j} Y_{t,j}. \tag{4}$$

Here, $\omega_{t,i,j}$, is the marginal probability that measurement $Y_{t,j}$ is associated to target i, given by,

$$\omega_{t,i,j} = \sum_{K:k_{t,i}=j} \Pr\{K_{1:t}|Y_{1:t}\}. \tag{5}$$

For linear-Gaussian point target models, (5) can be written as,

$$\omega_{t,i,j} \propto \sum_{K:k_{t,i}=j} \prod_{t,i} \left(\frac{P_D \mathcal{N}(Y_{t,j}; H\mu^p_{t,i}, HP^p_{t,i}H^\dagger + R)}{(1-P_D)\beta_c} \right)^{\mathbb{1}_{k_{t,i}}}$$

where $\mu^p_{t,i}$ and $P^p_{t,i}$ denote the mean and covariance of the predicted density at time t for target i, and the indicator function $\mathbb{1}_{k_{t,i}}$, is described as, $\mathbb{1}_{k_{t,i}} = \begin{cases} 1 & k_{t,i} \neq 0 \\ 0 & k_{t,i} = 0 \end{cases}$. As can be observed in (4), we need to know the marginal probabilities $\omega_{t,i,j}$ for the measurement update. A straightforward computation of these marginal probabilities is expensive, again because of the number of possibilities of K with $k_{t,i} = j$ and also due to the constraints on K for point targets. That is, for each $w_{t,i,j}$, all the possible hypotheses K that assigns target i to measurement j should be considered; in addition, these hypotheses should ensure point target assumptions and be such that a target is assigned to at most one measurement and a measurement is assigned to at most one target. In the literature, there are different strategies to compute these marginal probabilities by efficiently traversing through the hypotheses [17], [18], [30]. The methods proposed in [17] and [30] are approximations whereas the method in [18] returns the exact marginal probabilities, trading off complexity for accuracy.

A disadvantage of moment matching in JPDA is that it suffers from track coalescence problem when there are closely moving targets. In such scenarios, the priors and likelihoods are almost identical for the two targets. As a result, the posterior, which is a GM can have equally likely Gaussian components, which when moment matched yields almost identical tracks for the targets leading to coalescence [20], [21]. Note that this problem is also possible in many of the tracking algorithms [31], however it is prevalent in the JPDA case.

Like JPDA, MHT is also a filtering algorithm but maintains many Gaussian components in the posterior $p(X_t|K_{1:t})$. In other words, it maintains many hypotheses $K_{1:t}$ at each time instant. Before propagation to the next time, it prunes away hypotheses $K_{1:t}$ that are insignificant, which corresponds to reducing the number of global hypotheses in MHT nomenclature. For this pruning purpose, N-scan pruning [4], [9], [11] is one of the commonly implemented algorithms in MHT. The N-scan pruning algorithm traces back N time scans and prunes away all hypotheses at $t - N$, but the one that is the ancestor of the current best hypothesis $K^*_{1:t}$. Thus, the history before $t - N$ has just one data association possibility. The accuracy of the MHT algorithm depends on the depth N. Larger the N is, the better the accuracy is. However, larger N implies that at each t there will be many hypotheses retained which in turn implies higher computational complexity.

In case of PMHT algorithms [22], the MAP estimates of the states are obtained using a variational inference method, expectation maximization. This batch method is iterative and involves several local optimizations,

$$X^{(n+1)} = \arg\max_X \sum_K \Pr\{K|Y, X^{(n)}\} \ln p(X, K, Y) \tag{6}$$

where the superscripts (n) and $(n + 1)$ refer to the iteration indices. As one can observe from the above equation, there is the distribution $\Pr\{K|Y, X^{(n)}\}$ on K. With each iteration, this distribution on K varies and the state is estimated according to this new distribution. One important aspect regarding PMHT is that it allows a target to be associated to multiple measurements, which is more suitable for the extended target models than the point target model assumptions presented in the beginning of this section. As a consequence, PMHT allows the same measurement to be associated to multiple targets, which can lead to track coalescence when there are closely moving targets. In the next section we present an algorithm for tracking of multiple point targets which is based on EM similar to PMHT. But in contrast to PMHT, we derive the algorithm without relaxing the point target assumptions.

III. EM for target tracking

In this section, we apply EM to a batch version of multiple point target tracking problem and derive our solution. We show that E and M steps in the proposed algorithm mainly comprise of obtaining marginal data association probabilities, which serve as weights for the composite measurements, and smoothing using thus obtained composite measurements. We first present a background on EM and continue with the derivation for the model assumptions in Section II.

A. Expectation maximization for MAP estimation

Expectation maximization (EM), first discussed in [24], is an iterative technique, widely used to obtain, both, approximate maximum likelihood (ML) or MAP estimates of parameters from incomplete observed data. Equivalently, EM is a common choice for parameter estimation when the model has hidden variables, relating the data with parameters. We present a brief introduction to EM for MAP estimation, touching upon the idea of it being a strategy that increases a lower bound on the logarithm of the joint density.

The MAP estimation of state variable X, including the hidden data association variable can be written as,

$$\arg\max_X p(X,Y) \quad = \quad \arg\max_X \ln \sum_K p(X,Y,K). \quad (7)$$

The logarithm of summation over K in (7) has exponential complexity, making the MAP estimation problem NP-hard. EM tackles this complexity of working through all possible values of K by introducing a distribution over the data association variable, $q_K(K)$, and obtaining a term that can be interpreted as a lower bound on $\ln p(X,Y)$ [24], [32], shown below using Jensen's inequality,

$$\ln p(X,Y) \quad \geq \quad \sum_K q_K(K) \ln \frac{p(X,Y,K)}{q_K(K)}. \quad (8)$$

It is evident from (8) that the lower bound is a functional of q_K and X. Using them as parameters in the n-th iteration, EM alternates between finding the best estimate $q_K^{(n)}(K)$ given $X^{(n)}$, and finding the MAP estimate $X^{(n+1)}$ given $q_K^{(n)}(K)$. This alternative optimization is guaranteed to increase the lower bound $\ln p(X,Y)$ until convergence to an optimum [24], at which point the MAP estimate obtained would be the best approximation. Thus, using the lower bound (8) in (7), and collecting the terms dependent on X, the original MAP problem can now be written as,

$$\arg\max_X p(X,Y) \approx \qquad \arg\max_{X,q_K} \sum_K q_K(K) \ln p(X,Y,K).$$

From the above equation it is clear that there is an expectation (E-step) with respect to $q_K(K)$ and a maximization (M-step) with respect to X, which form the core of the EM algorithm iterations. It can be shown that the E and M step iterations can be uncoupled as [32]

$$q_K^{(n)}(K) \quad = \quad \Pr\{K|X^{(n)},Y\} \quad (9)$$

$$X^{(n+1)} \quad = \quad \arg\max_X \sum_K q_K^{(n)}(K) \ln p(X,Y,K). \quad (10)$$

The main computational advantage of using EM is that it converts the log-sum problem in (7), into the sum-log form in (10), as this gain can be often substantial. Further, in cases where the joint distribution can be expressed as a product of distributions from the exponential family, these factors reduce to polynomial terms, which further decreases the complexity.

B. Derivation of the EM algorithm for tracking multiple point targets

In this subsection, we derive the E and M steps for the point target tracking problem formulated in Section II. The E-step derivation is immediately obtained from the model description, whereas, we derive the M-step in some detail to show that the MAP estimate of states can be obtained by smoothing using composite measurements. We show that the composite measurements are weighted by the marginal probabilities of the data association variables, that are in turn distributed according to the density q_K returned by the E-step.

Starting with the expression for the E-step from (9),

$$q_K^{(n)}(K) = \Pr\{K|X^{(n)},Y\} \propto p(K,Y|X^{(n)}). \quad (11)$$

For the Gaussian model assumptions in Section II, it can be shown that [11], [23]

$$p(K,Y|X) \propto \prod_{i,t} p(Y_{t,j}, k_{t,i}=j|X_{t,i}), \quad (12)$$

where

$$p(Y_{t,j}, k_{t,i}=j|X_{t,i}) = \begin{cases} (1-P_D)\beta_c & k_{t,i}=0 \\ P_D \mathcal{N}(Y_{t,j}; HX_{t,i}, R) & k_{t,i}\neq 0 \end{cases}.$$

Incorporating the conditions described above using an indicator function $\mathbb{1}_{k_{t,i}}$, it follows that

$$q_K^{(n)}(K) \propto \prod_{t,i} \left(\frac{P_D \mathcal{N}(Y_{t,k_{t,i}}; HX_{t,i}^{(n)}, R)}{(1-P_D)\beta_c} \right)^{\mathbb{1}_{k_{t,i}}}, \quad (13)$$

where $X_{t,i}^{(n)}$ is the state estimate target i at time t.

In the M-step, we solve a convex maximization problem at iteration n. There are many ways to solve such problems, but our strategy is to show that the criterion can be rewritten as a Gaussian distribution for which the mean and the covariance have closed form expressions. Based on that result, it is trivial to see that the desired MAP estimate is the mean of that Gaussian distribution.

We begin by rewriting (10) as

$$X^{(n+1)} = \arg\max_X \exp\Big(\sum_K q_K^{(n)}(K) \ln p(X,Y,K) \Big). \quad (14)$$

Next, we make use of the factorization of $p(X,Y,K)$ in order to simplify the log term inside the summation in (14). One natural factorization is $p(X,Y,K) = p(K,Y|X)p(X)$, where the first factor $p(K, Y|X)$ has already been discussed in (12). Proceeding to the prior term $p(X)$, we notice that it factorizes across targets: $p(X) = \prod_i p(X_i)$, since the individual target motions are assumed independent of other targets' evolution. Further, the Markovian property of individual target states sequences can also be used to factorize the prior term as, $p(X) = \prod_{t,i} p(X_{t,i}|X_{t-1,i})$. Using these simplifications in (14), we get

$$X^{(n+1)} = \arg\max_X \Big(\prod_{t,i} p(X_{t,i}|X_{t-1,i}) \times \exp\Big(\sum_{K,t,i} q_K^{(n)}(K) \ln p(Y_{t,j}, k_{t,i} = j|X_{t,i}) \Big) \Big). \quad (15)$$

We next focus on simplifying the exponent term in (15), and observe that in summing over K, each $k_{t,i} \in \{0, 1, \ldots, M_t\}$. There can be several values of K, in which $k_{t,i} = j$, and we use the notation $K_{t,i,j}$ to denote these sets of hypotheses with assignment $k_{t,i} = j$. The implication of this analysis is that for a fixed value of j, $\ln p(Y_{t,j}, k_{t,i} = j|X_{t,i})$ appears in $K_{t,i,j}$ terms along with the weights $q_K^{(n)}$. Using this insight, the exponent in (15) can be rewritten as

$$\sum_{t,i} \sum_{j=0}^{M_t} \Big(\sum_{K_{t,i,j}} q_K^{(n)}(K) \Big) \ln p(Y_{t,j}, k_{t,i} = j|X_{t,i}). \quad (16)$$

The term $\left(\sum_{K_{t,i,j}} q_K^{(n)}(K) \right)$ is in fact the marginal probability of assigning target i to measurement j, according to the distribution $q_K^{(n)}$. We use the notation

$$w_{t,i,j}^{(n)} = \sum_{K_{t,i,j}} q_K^{(n)}(K) \quad (17)$$

to denote these marginal probabilities. Substituting the marginal probabilities in (15) and moving the summations out of the exponents, we get

$$X^{(n+1)} = \arg\max_X \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \times \exp\left(\sum_{j=0}^{M_t} w_{t,i,j}^{(n)} \ln p(Y_{t,j}, k_{t,i} = j|X_{t,i}) \right). \quad (18)$$

Using the Gaussian process model for state transition density $p(X_{t,i}|X_{t-1,i})$ from (1), and also using (13) in (18), it further simplifies to

$$X^{(n+1)} = \arg\max_X \prod_{i,t} \mathcal{N}\left(X_{t,i}; FX_{t-1,i}, Q\right) \times \mathcal{N}\left(\tilde{Y}_{t,i}^{(n)}; HX_{t,i}, \tilde{R}_{t,i}^{(n)}\right), \quad (19)$$

where, $\tilde{Y}_{t,i}^{(n)}$ and $\tilde{R}_{t,i}^{(n)}$ are the composite measurements and composite measurement covariances, given as

$$\tilde{Y}_{t,i}^{(n)} = \frac{\sum_{j=1}^{M_t} w_{t,i,j}^{(n)} Y_{t,j}}{1 - w_{t,i,0}^{(n)}} \quad (20)$$

$$\tilde{R}_{t,i}^{(n)} = \frac{R}{1 - w_{t,i,0}^{(n)}}. \quad (21)$$

Details of this simplification are provided in Appendix A. Note that $\sum_{j=0}^{M_t} w_{t,i,j}^{(n)} = 1$. When $w_{t,i,0}^{(n)} = 1$ in (20) and (21), the likelihood term $\mathcal{N}\left(\tilde{Y}_{t,i}^{(n)}; HX_{t,i}, \tilde{R}_{t,i}^{(n)}\right) = 1$ in (19).

From (19), it can be inferred that the objective function of each target state $X_{t,i}$ is proportional to a Gaussian density. First factor in (19) is the prior term, while the second factor is likelihood of the composite measurement. A straightforward method to obtain these marginal posterior densities is by performing forward-backward smoothing, and the resulting mean yields the MAP estimates of state variables.

From (13), (17) and (19), it can be seen that the link between the E and M steps are through the marginal probabilities. The state estimate $X^{(n+1)}$ in (19) depends on the distribution $q_K^{(n)}$ in (13) through the marginal probabilities $w_{t,i,j}^{(n)}$ in (17). In the next section, we elaborate on the role of marginal probabilities in (17), and show how the LBP method in [19] can be employed to approximate them.

IV. Marginal data association probabilities calculation using Loopy belief propagation

In this section, we discuss the challenges involved in computing the marginal data association probabilities $w_{t,i,j}^{(n)}$, that appear in (19) as part of the proposed algorithm. We also provide a brief overview of techniques to compute or approximate these probabilities. In this paper, we use loopy belief propagation (LBP) for computing these marginal probabilities. This LBP algorithm is inspired by the method employed in [19] for JPDA filtering. We extend it to compute marginal data association probabilities efficiently for batch problems.

To compute the marginal probabilities in (17), we make use of the expression for the data hypothesis probabilities in (13). The constraints that are implicitly in (13) due to the point target model assumptions pose the main challenge in computing the marginal probabilities in (17). At each time instant, there are constraints on the $k_{t,i}$ variables introducing dependency across targets. The constraint on $k_{t,i}$ at every time instant t ensures that no two targets share the same measurement at the same time. That is, $k_{t,i_1} \neq 0$ and $k_{t,i_2} \neq 0$ for $i_1 \neq i_2$ implies that $k_{t,i_1} \neq k_{t,i_2}$. However, no such constraints apply across time, that is, for targets i_1 and i_2, at different time instants, t_1 and t_2, the data association variables k_{t_1,i_1} and k_{t_2,i_2} are independent. Due to these constraints on the $k_{t,i}$ variables, $q_K(K)$ in (13) only factorizes across time, but not across targets within every time instant. Consequently, the marginalization of the data association variable can be performed at each time independently. Even so, the number of possible data associations at each time is very high, and it is computationally expensive to perform marginalization through an exhaustive listing of different possible values of K. Instead, there are less computationally intensive methods to obtain the marginal probabilities.

Techniques like k-best hypotheses based on Murty's algorithm [11], [16], Monte Carlo sampling method [17] and the tree-based approach [18] can be used to compute joint probabilities without explicit enumeration of all possible K. The method in [18] gives exact probability values whereas

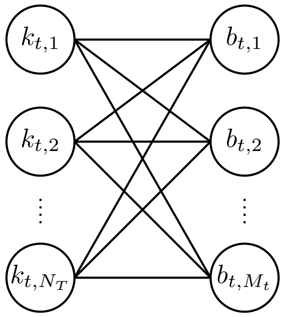

Figure 1: Belief propagation makes use of the dependencies across the (redundant) data association variables. As can be seen, the graph is bipartite which implies that the message flow is from the set of nodes on the left to the right and vice versa, when performing LBP.

the methods in [11], [16] and [17] give approximations. The method we have employed to compute $w_{t,i,j}$ in (17) in our implementation is inspired by the graph based approach, loopy belief propagation presented in [19], which has been shown to provide good accuracy versus computation time trade-off. In the next subsection, we present how the method proposed in [19] can be employed to the proposed algorithm. For the sake of brevity, we will drop the iteration indices.

A. Loopy belief propagation algorithm

The idea behind applying LBP is to use a factorization of $q_K(K)$ and apply the sum-product algorithm [33] to compute the marginal probabilities. In fact, in [19], a redundant representation for $q_K(K)$ is introduced and a factorization of this new representation is used to compute the marginals more accurately compared to what would have been possible with applying LBP on $q_K(K)$ directly.

The LBP method proposed in [19] is for computing the marginal probabilities for JPDA filtering, whereas in this paper, the marginal probabilities are calculated for the batch problem where estimates from the smoothed posterior density. Consider the distribution of data association in (13), with the point target constraints expressed explicitly in the equation using factors denoted $\phi_{t,i,i'}$:

$$q_K(K) \quad \propto \quad \prod_{t,i} \psi_{t,i}(k_{t,i}) \prod_{j=i+1}^{N_T} \phi_{t,i,i'}(k_{t,i}, k_{t,i'}) \quad (22)$$

where

$$\psi_{t,i}(k_{t,i}) \quad = \quad \left(\frac{P_D \mathcal{N}(Y_{t,k_{t,i}}; HX_{t,i}, R)}{(1-P_D)\beta_c} \right)^{1_{k_{t,i}}}, \quad (23)$$

$$\phi_{t,i,i'} \quad = \quad \begin{cases} 0 & k_{t,i} = k_{t,i'} \neq 0 \\ 1 & \text{otherwise} \end{cases}. \quad (24)$$

In our problem, the factors $\psi_{t,i}$ depend on the smoothed estimates of $X_{t,i}$, whereas in [19], they depend on the estimates from the prediction density. However, the constraints on the variable $k_{t,i}$ are the same in both the problems. So, the LBP method in [19] is applicable here wherein we use the smoothed estimates of $X_{t,i}$ instead of the predicted estimates.

Similar to [19], another set of variables $B = \{b_{t,j}, \forall t, j\}$ is introduced such that we have a redundant representation $q_{K,B}(K,B)$ instead of $q_K(K)$. The $b_{t,j}$ variables are the measurement-oriented data association variables, i.e., $b_{t,j}$ stands for assigning the measurement $Y_{t,j}$ to the target $b_{t,j}$ at time t. Using these variables, $q_{K,B}(K,B)$ can be comprehensively expressed, including the constraints as follows:

$$q_{K,B}(K,B) \propto \prod_{t,i} \psi_{t,i}(k_{t,i}) \prod_{j=1}^{Mt} \psi_{t,i,j}(k_{t,i}, b_{t,j}), \quad (25)$$

where

$$\psi_{t,i,j}(k_{t,i}, b_{t,j}) = \begin{cases} 0, & {\scriptstyle k_{t,i}=j, b_{t,j}\neq i} \\ & {\scriptstyle k_{t,i}\neq j, b_{t,j}=i} \\ 1, & \text{otherwise} \end{cases}. \quad (26)$$

Interpretation of the function $\psi_{t,i,j}(k_{t,i}, b_{t,j})$ is that it takes the value '1' when the target-oriented variable $k_{t,i}$ and the measurement-oriented variable $b_{t,j}$ agree regarding the association and the value '0' when there is a disagreement. So, these $\psi_{t,i,j}$ functions capture the constraints that a target can be assigned to at most one measurement and a measurement can be assigned to at most one target. The important advantage of introducing these new functions in the representation of $q_K(K)$ is that the dependency among the variables $k_{t,i}$ is only through the variables $b_{t,j}$. This dependency is best illustrated in the factor graph for (25) in Figure 1. The graph as can be seen is bipartite. This is advantageous because the possible message flow in LBP will be from the nodes on the left side of the graph to the right side of the graph and vice versa.

Once the factorization with $\psi_{t,i}$ and $\psi_{t,i,j}$ functions is established, one can follow the procedures discussed in Sections III-A and III-B of [19] to arrive at the expressions for the messages. The messages passed during the iteration l from $k_{t,i}$ variables to $b_{t,j}$ variables are

$$\eta^{(l)}_{t,i\to j} = \frac{\psi_{t,i}(k_{t,i} = j)}{1 + \sum\limits_{j'\neq j, j'>0} \psi_{t,i}(k_{t,i} = j')\nu^{(l-1)}_{t,j'\to i}} \quad (27)$$

and the messages passed from $b_{t,j}$ to $k_{t,i}$ variables are

$$\nu^{(l)}_{t,j\to i} = \frac{1}{1 + \sum\limits_{i'\neq i, i'>0} \eta^{(l)}_{t,i'\to j}}. \quad (28)$$

The messages are passed in both directions until convergence. Upon convergence, it follows from [19] that the marginal probabilities are calculated according to

$$w_{t,i,j} = \frac{\psi_{t,i}(k_{t,i} = j)\nu^{(l)}_{t,j\to i}}{\sum\limits_{j'} \psi_{t,i}(k_{t,i} = j')\nu^{(l)}_{t,j'\to i}}. \quad (29)$$

A step-by-step algorithmic description for every time instant t is given in Algorithm 1.

V. Algorithm

In this section, we present implementation details of the proposed algorithm for the Gaussian model assumptions. We discuss how the equations and methods derived in Section III, and Section IV can be interpreted, and implemented using

Algorithm 1 Loopy belief propagation algorithm for calculating marginal data association probabilities

```
Require: $\psi_{t,i}(j)$, $i = 1, \ldots, N_T$, $j = 1, \ldots, M_t$
Set $l = 0$.
Initialize $\nu^{(l)}_{t,j\to i} = 1$.
while Not converged do
    Set $l = l + 1$.
    for $i = 1, \ldots, N_T$, $j = 1, \ldots, M_t$ do
        Compute $\eta^{(l)}_{t,i\to j}$ according to (27).
    end for
    for $i = 1, \ldots, N_T$, $j = 1, \ldots, M_t$ do
        Compute $\nu^{(l)}_{t,j\to i}$ according to (28).
    end for
end while
for $i = 1, \ldots, N_T$, $j = 0, \ldots, M_t$ do
    Compute $w_{t,i,j}$ according to (29).
end for
Ensure: $w_{t,i,j}$, $i = 1, \ldots, N_T$, $j = 0, \ldots, M_t$
```

Algorithm 2 Proposed multi-target tracking algorithm based on EM

```
Input: Measurements $Y$
Initialize $X^{(n)}_{t,i}$, $t = 1, \ldots, T$, $i = 1, \ldots, N_T$ for $n = 0$.
while not converged do
    E step
    for $t = 1, \ldots, T$ do
        Compute $\psi^{(n)}_{t,i}(k_{t,i})$ $\forall$ $i$ and $\forall$ $k_{t,i} = 1, \ldots, M_t$
        according to (23).
        Marginal probabilities: Compute $w^{(n)}_{t,i,j}$ according
        to (29) using loopy belief propagation, described in
        Algorithm 1.
    end for
    M step
    Composite measurements: With inputs $w^{(n)}_{t,i,j}$, compute
    $\tilde{Y}_{t,i}$, $\tilde{R}_{t,i}$ $\forall\, t, i$, according to (20) and (21).
    for $i = 1, \ldots, N_T$ do
        RTS Smoothing: With inputs $X^{(n)}_{0,i}, P^{(n)}_{0,i}, \tilde{Y}_{t,i}, \tilde{R}_{t,i}$
        compute the means $X^{(n)}_{t,i}$ $\forall$ $t$ using RTS smoother
        described in Algorithm 3.
    end for
    Set $X^{(n+1)}_{t,i} = X^{(n)}_{t,i}, \forall t, i$.
    Set $n = n + 1$.
end while
Output: $X^{(n)}_{t,i}, w^{(n)}_{t,i,j}$ $\forall\, t, i, j$
```

existing algorithms in the literature. We present a step-by-step algorithm that describes the proposed solution.

The E and M steps in the proposed algorithm involve obtaining the marginal data association probabilities and MAP estimates of the state variable, as described in Section III. To initialize the algorithm, we assume that the estimates $X^{(0)}_{t,i}$, $\forall\ t, i$ are known. In our implementation, we use the true target states at time zero, i.e., for $X^{(0)}_{0,i}$ and run Kalman filters using the composite measurements, whose weights are obtained using the LBP, as described in Section IV and obtain $X^{(0)}_{t,i}$ for all t and i.

In the E-step, we compute $\psi_{t,i}$ according to (23). Given these values, the marginal probabilities $w^{(n)}_{t,i,j}$ are computed using the LBP method in Algorithm 1. With these marginal probabilities, the composite measurements and covariances in (20) and (21), respectively, are computed. The M-step in (19) involves forward-backward smoothing for each target i with the likelihood $\mathcal{N}\left(\tilde{Y}_{t,i}; HX_{t,i}, \tilde{R}_{t,i}\right)$ of these composite measurements and covariances. In our implementation, forward-backward smoothing is carried out using RTS smoother (described in Appendix B) which provide the desired MAP estimates $X^{(n)}_{t,i}$ in (19). Note that the complexity of running the RTS smoothers, thus, scales linearly with the number of targets. The new estimates $X^{(n)}_{t,i}$ from the RTS smoother are passed on to the next iteration of E-step, and the algorithm is run until convergence.

One can use common ways of checking for convergence, for instance, to check for the change in estimates after each iteration. In our implementation, when the change is sufficiently small, we terminate the algorithm. The entire algorithm is summarized in Algorithm 2.

VI. Comparison with PMHT and JPDA

In this section, we compare the proposed method with PMHT and JPDA to highlight some of the similarities and differences using example illustrations. The comparison with PMHT is presented based on marginal probabilities, where we show that PMHT can coalesce trajectories as it does not enforce the point target constraints. Similar coalescence property is also exhibited by JPDA [21], but in this case it is due to moment matching.

A. Comparison with PMHT

As pointed in Section II-C, the proposed algorithm is similar to PMHT, as both algorithms are based on EM and treat the data association variables as the hidden variables. The similarity becomes even more pronounced in the algorithmic details, as both perform smoothing with composite measurements of the form in (20) and covariances as in (21). Further, local hypothesis probability calculation is identical as in (23). However, the two algorithms differ in the computation of marginal probabilities for composite measurements from the local hypothesis probabilities.

PMHT assumes that a target can be assigned to multiple measurements, implying that there are no constraints on the data association variables $k_{t,i}$, and distribution over the data association variables, $q_K(K)$, in (22) factorizes across time and targets. Consequently, the weights for composite measurements are computed directly from local hypothesis probabilities, $\psi_{t,i}(k_{t,i})$, after normalizing them as probabilities. On the other hand, in the proposed EM algorithm, there are constraints on $k_{t,i}$ across targets at each time as was discussed in Section IV. As a result, calculation of the weights is performed conforming to the constraints, which in this work is implemented using the LBP method presented in Section IV-A. In summary, the proposed EM algorithm with an unconstrained marginal probability computation (but with normalized $\psi_{t,i}$) would be converted into the PMHT algorithm.

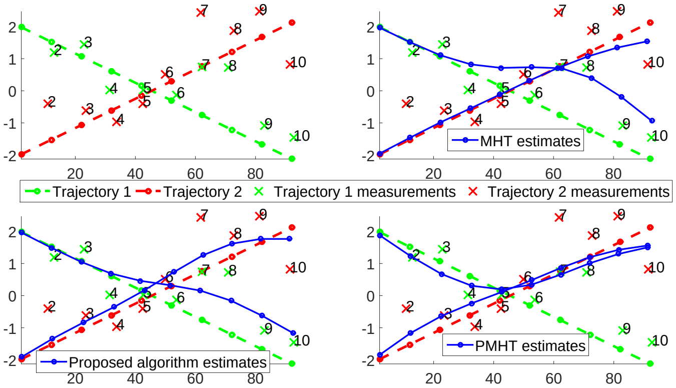

Figure 2: The figure shows slowly moving, crossing target trajectories, along with their corresponding measurements. Numbers adjacent to the measurements indicate time indices of the corresponding measurements. Also shown are the trajectory estimates returned by N-scan pruning based TOMHT with a scan depth of 4, the proposed algorithm and PMHT.

One possible implication of relaxing the point target model constraints in PMHT, when computing marginal probabilities, is track coalescence. We illustrate this using the crossing targets example, depicted in Figure 2, and compare it to the proposed EM algorithm and an N-scan pruning based TOMHT algorithm with a scan depth. The figure shows two crossing trajectories generated with zero process noise; the measurements shown are generated by setting $P_D = 1$ and $\beta_c = 0$. It also shows the trajectory estimates (with process noise standard deviation of $0.2165m$) returned by PMHT, the proposed EM algorithm and TOMHT. From the figure, it is evident that PMHT algorithm coalesces both trajectories after the crossing point. This behaviour can be best explained by observing the marginal probabilities, $\omega_{t,i,j}$, returned by PMHT, when the targets cross. For instance, the marginal probabilities at $t = 6$ are shown in Table I. As it has been emphasized before, PMHT can allow targets to be associated to the same measurement; in this case, both targets get high weights for measurement 2, and get updated with the same measurement, resulting in similar trajectories. On the other hand, weights returned by the proposed algorithm are shown in Table II, wherein, we see that it has ruled out the possibility of assigning the same measurement to both targets and thus, target 1 has high weight for measurement 1 and target 2 for measurement 2. Thus, the targets are kept separated when there are well-spaced measurements as shown in Figure 2. It can also be observed that the estimates from the proposed algorithm are very close to the estimates from TOMHT algorithm, which we use as the benchmark.

B. Comparison with JPDA

The similarities between JPDA [11], [14] and the proposed EM algorithm arise from the fact that both algorithms use composite measurements to update target states at each time instant, which requires computation of the marginal probabilities. One can therefore use similar strategies to compute these probabilities for both algorithms, see Section IV for a discussion on these strategies. However, there is a fundamental difference in the objective of the two algorithms, as pointed out in Section II-C. JPDA is primarily a filtering algorithm that approximates the posterior density $p(X_t|Y_{1:t})$ by moment matching, whereas the proposed EM algorithm is a batch solution that provides MAP estimates of the states. A shortcoming of the recursive moment matching in JPDA is that it can lead to coalescence when there are closely moving targets.

Table I: The table shows values of marginal probabilities $\omega_{t,i,j} \propto \psi_{t,i}(j)$ returned by PMHT at $t = 6$.

$\omega_{t,i,j}$	Measurement 1	Measurement 2
Target 1	0.06	0.93
Target 2	0.08	0.92

Table II: The table shows values of marginal probabilities $\omega_{t,i,j}$ returned by EM algorithm at $t = 6$.

$\omega_{t,i,j}$	Measurement 1	Measurement 2
Target 1	0.987	0.013
Target 2	0.077	0.923

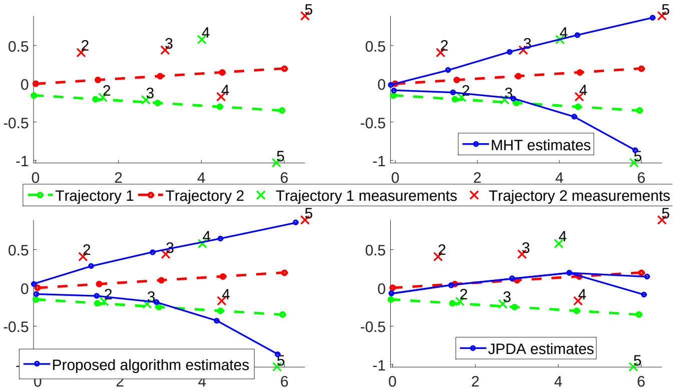

Figure 3: The figure shows two closely moving targets with the corresponding measurements. Also shown are the trajectories estimated by MHT, the proposed EM algorithm and JPDA.

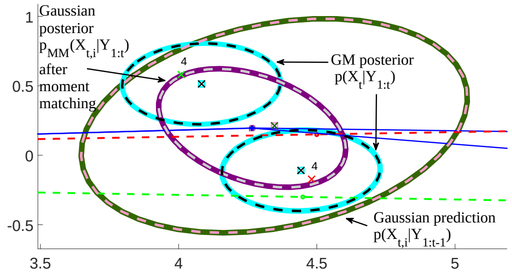

Figure 4: The figure shows the illustration of densities showing the coalescence effect in JPDA due to moment matching at time $t = 4$ for the example in Figure 3. The pink and dark green ellipses represent the Gaussian prediction density for the two targets. The cyan and black pairs of ellipses represent the two Gaussian components in the joint posterior density of the two targets. The purple and the grey ellipses show the posterior density of the two target states after moment matching.

We illustrate the coalescence effect in JPDA using an example shown in Figure 3. The figure shows two targets that are moving slowly in close proximity of each other, along with the measurements that are generated with $P_D = 1$ and $\beta_c = 0$. For this example, JPDA, TOMHT and the proposed algorithm have the same initialization for $X_{0,i}$; the values of $X_{0,i}$ are very close for the two targets $i = 1, 2$. Figure 3 shows the trajectories returned by the three algorithms. It can be observed that MHT and the proposed algorithm return almost identical results whereas JPDA has coalesced the two trajectories. The coalescence in JPDA is due to moment matching, which can be explained by looking closer at the densities, for instance, at time $t = 4$ (Figure 4). Due to close initiations, the prediction densities $p(X_{t,i}|Y_{1:t-1})$ at $t = 4$ for both the targets are almost identical for both targets. The similarity between the densities is illustrated by the overlapping pink and dark green ellipses in Figure 4. Given this prior, there are two possible hypotheses – target 1 is assigned to measurement 1, whereas target 2 assigned to measurement 2 and vice versa – leading to a Gaussian mixture posterior density $p(X_t|Y_{1:t})$ with two components, corresponding to the two hypotheses. However, since the priors are very similar, the two components corresponding to the two hypotheses are almost identical, shown by the overlapping cyan and black ellipses in the figure. Given this posterior density, any MAP estimation algorithm would pick the mean of one of the components, i.e., either the cyan or the black. However, JPDA does a moment matching of the two components and arrives at a Gaussian posterior density $p_{MM}(X_{t,i}|Y_{1:t})$ for the two target states, which are again almost identical, shown by the purple and grey ellipses in the figure. From this illustration, it is clear that the JPDA algorithm tends to coalesce close trajectories even if the measurements are far apart, due to moment matching of the GM.

There are versions of the JPDA algorithm that reduce coalescence effectively. The coupled JPDA filter [34] overcomes coalescence by approximating the joint posterior density across all the targets as a Gaussian, whereas in conventional JPDA filters [35], the target states are assumed to be independent given the measurements until the last time and the marginals are approximated to a Gaussian. However, when there are slowly crossing targets, retaining the correlation between targets can lead to strong coupling leading to coalescence [34, pp. 254], [36]. There are also mixture reduction algorithms [37] where more than one Gaussian component per target is propagated to successive time instants. This could reduce, but does not entirely eliminate the effects of coalescence, especially when the targets stay close for long time [34]. On the other hand, a MAP estimation algorithm like the proposed algorithm or TOMHT, outputs the mode of the posterior density, thus avoiding coalescence, as shown in Figure 3.

VII. Simulations and results

In this section, we evaluate the proposed algorithm for different scenarios and compare its performance with PMHT, JPDA and N-scan pruning based TOMHT. In Section VII-A, we first describe the simulation set-up in terms of model parameter values, comparison metrics and methods of comparison. In Section VII-B, we briefly describe two techniques used for handling the sensitivity to initialization of the proposed algorithm. In Section VII-C, we evaluate the impact of initialization techniques on the proposed algorithm. Computation of marginal data association probabilities using LBP is compared with brute force approach to show the difference in performance of the proposed algorithm. Finally, a comprehensive comparison based on MOSPA and computation time of the four aforementioned algorithms is presented.

A. Simulation scenarios

To evaluate the performance of the proposed EM algorithm, TOMHT, PMHT and JPDA, we have used a four-dimensional constant-velocity model with additive discrete-time noise, with parameters,

$$X_{t,i} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, \; F = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$Q = \sigma_v^2 \begin{bmatrix} T_s^4/4 & 0 & T_s^3/2 & 0 \\ 0 & T_s^4/4 & 0 & T_s^3/2 \\ T_s^3/2 & 0 & T_s^2 & 0 \\ 0 & T_s^3/2 & 0 & T_s^2 \end{bmatrix},$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ and } \; R = \sigma_w^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

In the above equations, T_s is the sampling time. The values of the parameters are $T_s = 0.5$ s, $\sigma_v = 8m/s^2$ and $\sigma_w = \sqrt{5}m$ for all scenarios and the batch length T is 20. The initial velocity for the constant velocity model was set at 20 m/s on both the x and y directions. The x- and y-coordinates of the initial points $X_{0,i}$, $i = 1, \ldots, N_T$ for the targets were chosen uniformly from a region of width 80m. To simulate different scenarios, the following combinations of parameters have been used:

- $N_T = [10, 20, 30]$, $P_D = 0.9$ and $\beta_c = 1.5 \times 10^{-4}$
- $\beta_c = [0.5, 2.5, 4.5] \times 10^{-4}$, $P_D = 0.9$ and $N_T = 20$
- $P_D = [0.8, 0.9, 0.99]$, $\beta_c = 1.5 \times 10^{-4}$ and $N_T = 20$.

For N-scan pruning based TOMHT, we have used a scan depth of 3 in all the scenarios. In JPDA, the LBP method described in Section IV-A has been used to compute the marginal probabilities. We have used Euclidean distance based MOSPA [28] and computation time as the metrics to compare performances. MOPSA calculation was performed using the modified auction algorithm [38]. To compare the computation time, all four algorithms were run on a desktop computer with quad-core processor and 8 gigabyte RAM, running Debian operating system.

B. Handling sensitivity to initialization

The proposed algorithm and PMHT are based on EM, and it is well known that EM algorithm is sensitive to initialization [25], [39] and can converge to a local optimum. To deal with this issue, we use two common strategies in our implementation: 1) Deterministic annealing [25], 2) Homotopy methods [40]. In this section, we describe these methods briefly and show empirically that these methods improve the performance compared to naive initializations.

In the deterministic annealing (DA) [25] strategy, the idea is to compute the E-step with $(q_K^{(n)})^\beta$ instead of $q_K^{(n)}$, where $0 < \beta \leq 1$. The value of β is increased successively, from a small value to begin with, reaching 1 when EM iterations stop.

Table III: The table shows RMSE per time scan for position estimates (in m), and velocity estimates (in m/s) based on the MOSPA assignment. The values for each of the scenarios are averaged over 1000 Monte Carlo trials. The values of β_c shown in the table are scaled by 10^4, i.e., in the range of 0.5×10^{-4} to 4.5×10^{-4}.

Scenario			Position estimates				Velocity estimates			
			TOMHT	Prop.Alg.	PMHT	JPDA	TOMHT	Prop.Alg.	PMHT	JPDA
$P_D = 0.9$ $\beta_c = 1.5$	$N_T =$	10	0.94	1.53	3.27	3.86	0.44	0.59	0.80	0.89
		20	1.23	1.30	2.35	2.61	0.43	0.47	0.78	0.76
		30	1.32	0.98	1.77	1.88	0.42	0.34	0.61	0.68
$P_D = 0.9$ $N_T = 20$	$\beta_c =$	0.5	1.19	0.79	2.76	2.57	0.43	0.34	0.95	0.75
		2.5	1.35	1.25	2.71	2.75	0.48	0.46	0.88	0.78
		4.5	1.47	1.58	2.55	2.97	0.50	0.56	0.81	0.82
$N_T = 20$ $\beta_c = 1.5$	$P_D =$	0.8	1.46	1.51	2.34	2.62	0.51	0.52	0.78	0.79
		0.9	1.08	1.15	2.59	2.79	0.39	0.43	0.76	0.81
		0.99	0.88	0.60	2.56	2.97	0.35	0.31	0.73	0.94

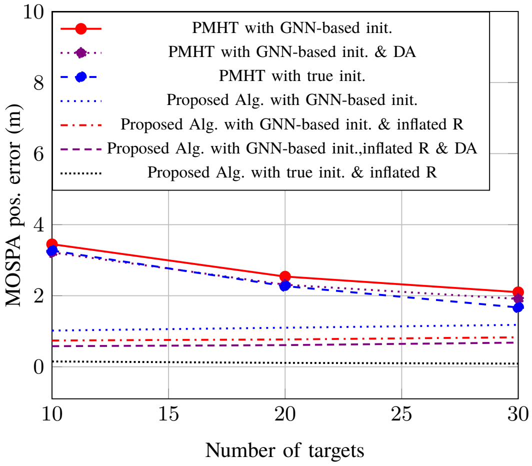

Figure 5: Comparison of MOSPA based position error for the proposed algorithm and PMHT, with GNN based initialization and true state initialization. The impact of deterministic annealing and inflated measurement noise on the MOSPA performance of the proposed algorithm is also shown.

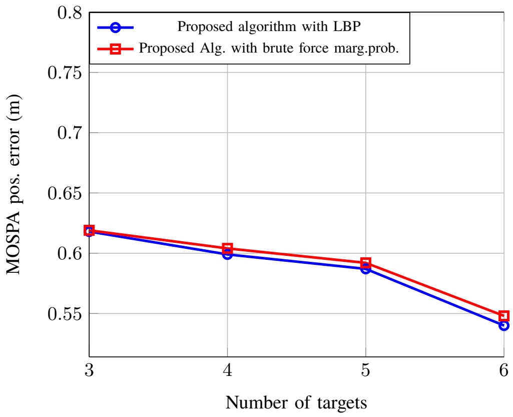

Figure 6: Comparison of MOSPA based position error for the proposed algorithm with brute force computation of marginal data association probabilities and the approximation using LBP.

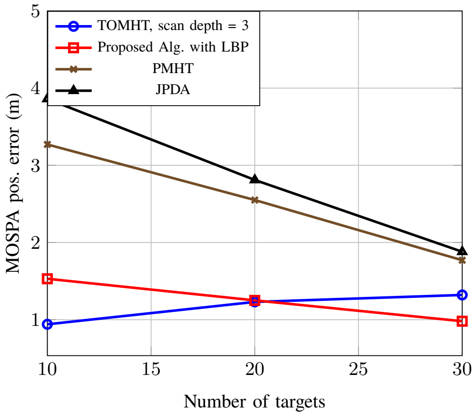

Figure 7: Comparison of MOSPA based position error for the different algorithms considered, for varying number of targets scenario corresponding to the first row in Table III.

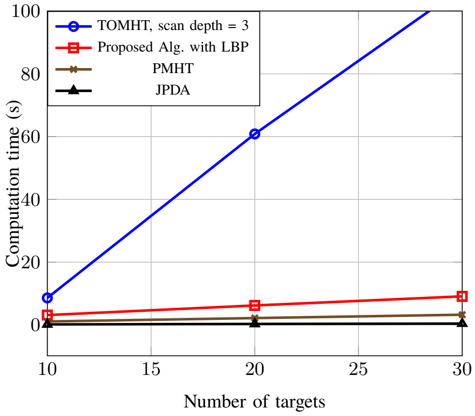

Figure 8: Computation time comparison for the different algorithms considered, for varying number of targets scenario corresponding to the first row in Table III.

In our algorithm 2, introducing the deterministic annealing only affects the Step 5; before passing ψs to the next step, we raise these values by the exponent β in the current iteration.

Inspired from the homotopy methods [40], we use a strategy to run several instances of the algorithm with varying measurement noise covariance R, which we refer to as the inflated R strategy. The value of R is changed only within the algorithm, and not in the generation of the measurements. To begin with, the measurement noise covariance is set to a higher value compared to the true value and the entire EM algorithm is run with this inflated R. After the EM algorithm converges with the current inflated R, the estimates from this run are used as the initialization for the next run of the algorithm. This procedure is repeated until R is reduced to its true value.

C. Results

In this subsection, we present the simulation results highlighting different aspects of the proposed algorithm. First, we present the impact of using homotopy methods and DA on the performance of the proposed algorithm. Second, we compare the quality of the approximation of the marginal probabilities using LBP in the proposed algorithm with the brute force approach. Finally, we compare the MOSPA and computation time performances of the proposed algorithm with TOMHT, PMHT and JPDA for varying set of parameters. All simulation results presented are averaged over 1000 Monte Carlo runs.

1) Impact of homotopy methods and DA: The gains achieved by using homotopy methods and DA for the proposed algorithm are large compared to those achieved by varying PMHT initializations. To highlight these differences, performance of both the algorithms implemented with these methods is compared against two other naive (without homotopy and deterministic annealing) strategies: the naive implementation with the true target states as initialization, and the naive implementation with a global nearest neighbor (GNN) based initializations. The observations are presented in Figure 5 for $N_T = [10, 20, 30]$, $\beta_c = 1.5 \times 10^{-4}$ and $\sigma_v = 6m/s^2$. From Figure 5, it is clearly seen that the proposed algorithm benefits from DA and inflated R strategy. It shows further improved performance when initialized with true target states. On the contrary, we observe that PMHT shows no such improvement even when initialized with true target states. In summary, the proposed algorithm in all its variations considered shows improved performance when compared to PMHT. We also expect it to improve further with better initializations.

2) Comparison of brute force and LBP computation of marginal probabilities: In Section IV, we presented LBP as an efficient method to approximate the marginal association probabilities. We substantiate this approximation by comparing the difference in performance when the proposed algorithm uses LBP and brute force method for the same scenarios. We compared the difference in the position root mean squared error (RMSE) for the two instances of the proposed algorithm, one with LBP and the other with brute force for $N_T = [3, 4, 5, 6]$, limited by the massive computation time of brute force approach. The results are presented in Figure 6, where we observe that there is no significant difference in performance of the proposed algorithm for both methods of computing marginal data association probabilities. Based on this observation, we conclude that the approximation of marginal data association probabilities using LBP does not degrade the performance of the proposed algorithm, despite providing large (from exponential to quadratic) computational advantage.

3) Performance evaluation of the proposed algorithm: In Table III, we present a comprehensive comparison of the MOSPA results for different scenarios for the different algorithms. The table shows the mean values of root mean squared error in position and velocity components of the estimated states. The table captures the performance trend of each of the algorithms in the scenarios considered. For instance, the first row of numeric entries correspond to the scenario when $P_D = 0.9, N_T = 20, \beta_c = 0.5 \times 10^{-4}$, and we notice that the mean position and velocity errors for the proposed algorithm are lower than the other algorithms. Although not reported in the table, we have observed a similar trend of lower standard deviation of the position and velocity errors for the proposed algorithm, by which we infer that the proposed algorithm is more robust than JPDA and PMHT for the scenarios considered. MOSPA based position error for the varying number of targets scenario is presented in Figure 7. From the figure it can be seen that the proposed algorithm performs better than PMHT and JPDA for all scenarios. For dense scenarios, the proposed algorithm even outperforms the TOMHT considered.

In Figure 7, we observe that for the proposed algorithm, PMHT and JPDA, the MOSPA performance decreases with increasing number of targets. This trend is due to two reasons. First, the OSPA computation returns average error per target because of the normalisation in its definition [28]. Therefore, it is possible that the average error decreases with increasing number of targets even though the overall error (without the normalisation) increases. The second reason is that the volume in which the initial positions for the targets are generated is fixed. So, the tracks tend to be dense with increasing number of targets. Due to this reason, while computing OSPA, it becomes easier to find assignments that are close to each other with large number of targets. Despite these two reasons, MOSPA error of TOMHT increases with increasing number of targets in Figure 7. This trend is because as the tracks get denser with more number of targets, the hypotheses tree for each target in TOMHT is also denser. So, to obtain a similar average performance with increasing number of targets, one needs to correspondingly increase the scan depth. But for computational reasons, we have fixed the scan depth to 3 for all the scenarios, leading to the trend in Figure 7.

The computation time comparison is performed based on the run time of the algorithm for varying number of targets, shown in Figure 8. The results presented capture the trend in computation time of the proposed algorithm in comparison to the existing algorithms. It should be noted that the implementation of all the four algorithms can be optimized for run time; for instance, one can parallelize the implementations. In that case, we expect the graph shown in Figure 8 to scale down by an appropriate factor. Nevertheless, it can be seen that the proposed algorithm's run time is less than our implementation of TOMHT.

When the targets are far apart, the proposed algorithm converges within fewer iterations, while giving good MOSPA performance. In our implementation, this happens when there are fewer targets ($N_T = 10$ in Figures 7 and 8), as the volume in which the initial positions are generated is fixed across the varying scenarios. As mentioned before, we use LBP in both the proposed algorithm and JPDA. Therefore, computational cost of the proposed algorithm is roughly same as the computational cost of the JPDA, where as the proposed algorithm clearly outperforms JPDA in terms of MOSPA.

The proposed algorithm delivers improved MOSPA performance in comparison to PMHT. By employing LBP to compute marginal probabilities, the computation time only increases marginally when compared to JPDA (that also uses

LBP) and PMHT. In summary, the proposed algorithm yields performance close to TOMHT, but with a significant computational advantage.

VIII. Conclusions and future work

In conclusion, we have presented a multi-target tracking algorithm based on expectation maximization, and marginal data association probabilities. We have shown that our solution for tracking point targets uses the correct point target model assumptions and does not violate these assumptions in deriving the solution. The solution has been presented for the case of known number of targets and for batch problems, and has been compared comprehensively with JPDA, PMHT and TOMHT, highlighting the prominent differences between them. The performance of the proposed algorithm has been evaluated for varying number of targets, clutter intensities and probability of detection, and we have shown that for the scenarios considered, our algorithm performs considerably better than JPDA and PMHT in MOSPA sense, and is computationally more inexpensive than a basic TOMHT.

We see substantial potential in the proposed algorithm, and envisage many possible enhancements to it. While we have discussed the proposed algorithm as a batch solution, one evident limitation with this approach, as with PMHT, is it not being an online solution, which might be a requirement in many tracking applications. Another possible extension is to accommodate target birth/death into the solution framework, to complete the tracking solution. One of the commonly used strategies to initialize new targets is measurement-driven track initialization. A more coherent approach to track handling is to model target existence uncertainties as a random variable and estimating it. The algorithm has been observed to be sensitive to initialization. So, further improvement to initialization techniques would be desirable as well.

Appendix A
Derivation of the EM algorithm for the Gaussian case

In this section, we will sketch the derivations starting from (18) to (19) in Section III. Starting with (18),

$$X^{(n+1)} = \arg\max_X \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \times \exp\left(\sum_{j=0}^{M_t} w^{(n)}_{t,i,j} \ln p(Y_{t,j}, k_{t,i} = j|X_{t,i})\right). \quad (30)$$

Moving around the $\ln$ and $\exp$, the above equation becomes

$$X^{(n+1)} = \arg\max_X \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \times \left(\prod_{j=0}^{M_t} p(Y_{t,j}, k_{t,i} = j|X_{t,i})^{w^{(n)}_{t,i,j}}\right). \quad (31)$$

Substituting the expressions for $p(Y_{t,j}, k_{t,i} = j|X_{t,i})$ and considering only the terms that depend on X, we get

$$X^{(n+1)} = \arg\max_X \prod_{t,i} p(X_{t,i}|X_{t-1,i}) \times \left(\prod_{j=1}^{M_t} \mathcal{N}(Y_{t,j}; HX_{t,i}, R)^{w^{(n)}_{t,i,j}}\right). \quad (32)$$

Note that $\mathcal{N}(Y_{t,j}; HX_{t,i}, R)^{w^{(n)}_{t,i,j}} \propto \mathcal{N}(Y_{t,j}; HX_{t,i}, \frac{R}{w^{(n)}_{t,i,j}})$. Along with this relation, the following result for the product of Gaussians can be used in (32),

$$\prod_{i=1}^{N} \mathcal{N}(x; \mu_i, P_i) \propto \mathcal{N}(x; \mu, P) \quad (33)$$

where $P = \left(\sum_{i=1}^{N} P_i^{-1}\right)^{-1}$ and $\mu = P\left(\sum_{i=1}^{N} P_i^{-1}\mu_i\right)$. After simplifications, we get

$$X^{(n+1)} = \arg\max_X \prod_{i,t} \mathcal{N}(X_{t,i}; FX_{t-1,i}, Q) \times \mathcal{N}\left(\tilde{Y}^{(n)}_{t,i}; HX_{t,i}, \tilde{R}^{(n)}_{t,i}\right). \quad (34)$$

Equation (34) is same as the equation for the joint density of a single target, where instead of a simple measurement update, the update is performed with the composite measurement $\tilde{Y}^{(n)}_{t,i}$, given in (20), and with uncertainty $\tilde{R}^{(n)}_{t,i}$, given in (21).

Appendix B
Smoothing with the RTS smoother

Rauch-Tung-Striebel (RTS) smoother is the ubiquitously used forward-backward smoothing solution. For linear Gaussian models, the forward filtering is performed using a Kalman filter, with the moments calculated as,

$$\begin{aligned}
\mu_{t+1|t} &= F\mu_t & (35)\\
P_{t+1|t} &= FP_tF^\dagger + Q & (36)\\
S_{t+1} &= HP_{t+1|t}H^\dagger + R & (37)\\
K_{t+1} &= P_{t+1|t}H^\dagger S^{-1}_{t+1} & (38)\\
\mu_{t+1} &= \mu_{t+1|t} + K_t(y_{t+1} - H\mu_{t+1|t}) & (39)\\
P_{t+1} &= P_{t+1|t} - K_{t+1}HP_{t+1|t}. & (40)
\end{aligned}$$

Backward smoothing is performed using the moments computed from the forward filter, with the following smoothing equations:

$$\begin{aligned}
G_t &= P_tF^\dagger P^{-1}_{t+1|t} & (41)\\
\mu^s_t &= \mu_t + G_t(\mu^s_{t+1} - \mu_{t+1|t}) & (42)\\
P^s_t &= P_t + G_t[P^s_{k+1} - P_{t+1|t}]G^\dagger_t. & (43)
\end{aligned}$$

References

[1] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *International Workshop on Intelligent Robots and Systems*. Osaka: IEEE, 1991, pp. 1442–1447.

[2] D. Koller, J. Weber, and J. Malik, *Robust multiple car tracking with occlusion reasoning*. Springer, 1994.

[3] Y. Bar-Shalom, *Multitarget-multisensor tracking: Advanced applications*. Norwood, MA: Artech House, 1990.

[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. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," in *IEEE Conference on Decision and Control*, Nassau, 2004.

Algorithm 3 RTS smoothing

```
Require: Initial estimates $\mu_0, P_0$, and measurement data $y_t$ for $t = 1, \dots, T$
  for all $t = 1, \dots, T$ do
    Compute and store the moments $\mu_{t|t-1}$, $P_{t|t-1}$, $\mu_t, P_t$ according to (35), (36), (39) and (40), respectively
  end for
  for all $t = T-1, \dots, 1$ do
    Compute the smoothed moments $\mu_t^s, P_t^s$ according to (42) and (43), respectively
  end for
```

[6] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[7] J. Kokkala and S. Särkkä, "Combining particle MCMC with Rao-Blackwellized Monte Carlo data association for parameter estimation in multiple target tracking," *Digital Signal Processing*, 2015.

[8] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 618–626, 1978.

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

[10] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: Principles and techniques*. Storrs, CT: YBS Publishing, 1995.

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

[12] T. Kurien, M. Liggins *et al.*, "Report-to-target assignment in multisensor multitarget tracking," in *IEEE Conference on Decision and Control*, Austin, TX, 1988.

[13] G. Pulford, "Taxonomy of multiple target tracking methods," in *IEE Proceedings - Radar, Sonar and Navigation*, vol. 152, no. 5. IET, 2005, pp. 291–304.

[14] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[15] K.-C. Chang and Y. Bar-Shalom, "Joint probabilistic data association for multitarget tracking with possibly unresolved measurements and maneuvers," *IEEE Transactions on Automatic Control*, vol. 29, no. 7, pp. 585–594, 1984.

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

[17] M. R. Morelande, "Joint data association using importance sampling," in *International Conference on Information Fusion*. Seattle, WA: IEEE, 2009, pp. 292–299.

[18] P. Horridge and S. Maskell, "Real-time tracking of hundreds of targets with efficient exact JPDAF implementation," in *International Conference on Information Fusion*. Florence: IEEE, 2006.

[19] J. L. Williams and R. Lau, "Approximate evaluation of marginal association probabilities with belief propagation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2942–2959, 2014.

[20] E. A. Bloem and H. A. P. Blom, "Joint probabilistic data association methods avoiding track coalescence," in *IEEE Conference on Decision and Control*, 1995.

[21] S. Coraluppi, C. Carthel, P. Willett, M. Dingboe, O. O'Neill, and T. Luginbuhl, "The track repulsion effect in automatic tracking," in *International Conference on Information Fusion*. Seattle, WA: IEEE, 2009, pp. 2225–2230.

[22] R. L. Streit and T. E. Luginbuhl, "Probabilistic multi-hypothesis tracking," DTIC Document, Tech. Rep., 1995.

[23] P. Willett, Y. Ruan, and R. Streit, "PMHT: Problems and some solutions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 738–754, 2002.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[25] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," *Neural Networks*, vol. 11, no. 2, pp. 271–282, 1998.

[26] D. F. Crouse, M. Guerriero, and P. Willett, "A critical look at the PMHT," *Journal of Advances in Information Fusion*, vol. 4, no. 2, pp. 93–116, 2009.

[27] H. E. Rauch, C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[28] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[29] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting two birds with two bullets: How to find minimum mean OSPA estimates," in *International Conference on Information Fusion*. Edinburgh: IEEE, 2010, pp. 1–8.

[30] J. L. Williams and R. A. Lau, "Data association by loopy belief propagation," in *International Conference on Information Fusion*. Edinburgh: IEEE, 2010, pp. 1–8.

[31] A. F. Garcia-Fernandez, M. R. Morelande, and J. Grajal, "Bayesian sequential track formation," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6366–6379, 2014.

[32] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University of London, 2003. [Online]. Available: http://www.cse.buffalo.edu/faculty/mbeal/papers/beal03.pdf

[33] 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.

[34] H. A. P. Blom and E. A. Bloem, "Probabilistic data association avoiding track coalescence," *IEEE Transactions on Automatic Control*, vol. 45, no. 2, pp. 247–259, 2000.

[35] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *Control Systems, IEEE*, vol. 29, no. 6, pp. 82–100, 2009.

[36] J. L. Williams, "Gaussian mixture reduction for tracking multiple maneuvering targets in clutter," Master's thesis, Grad. School of Engineering and Management, Air Force Inst. of Tech,. Wright-Patterson Air Force Base, OH, 2003. [Online]. Available: http://ssg.mit.edu/group/jlwil/publications/Wil03.pdf

[37] L. Y. Pao, "Multisensor multitarget mixture reduction algorithms for tracking," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 6, pp. 1205–1211, 1994.

[38] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Transactions on Automatic Control*, vol. 37, no. 2, pp. 198–213, 1992.

[39] B. Wang, D. Titterington *et al.*, "Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model," *Bayesian Analysis*, vol. 1, no. 3, pp. 625–650, 2006.

[40] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

Abu Sajana Rahmathullah Abu Sajana Rahmathullah was born in Chennai, India, in 1986. She received her M.S. in Electrical Communication Engineering from Indian Institute of Science, Bangalore in 2009.

She is currently pursuing her PhD at the Department of Signals and Systems, Chalmers University of Technology, Sweden. Her research interests include Bayesian inference, target tracking and sensor fusion.

Raghavendra Selvan Raghavendra Selvan was born in Bangalore, India, in 1987. He received his MSc degree in electrical engineering in 2015, from Chalmers University, Gothenburg, Sweden.

He is currently pursuing his PhD at the Department of Computer Science, University of Copenhagen. His research interests include medical image analysis and Bayesian tracking theory.

Lennart Svensson Lennart Svensson was born in Älvängen, Sweden in 1976. He received the M.S. degree in electrical engineering in 1999 and the Ph.D. degree in 2004, both from Chalmers University of Technology, Gothenburg, Sweden.

He is currently Associate Professor at the Signal Processing group, again at Chalmers University of Technology. His main research interests include machine learning and Bayesian inference in general, and nonlinear filtering and tracking in particular.