Set JPDA Filter for Multitarget Tracking

Lennart Svensson, *Senior Member, IEEE*, Daniel Svensson, Marco Guerriero, and Peter Willett, *Fellow, IEEE*

***Abstract*—In this article, we show that when targets are closely spaced, traditional tracking algorithms can be adjusted to perform better under a performance measure that disregards identity. More specifically, we propose an adjusted version of the joint probabilistic data association (JPDA) filter, which we call set JPDA (SJPDA). Through examples and theory we motivate the new approach, and show its possibilities. To decrease the computational requirements, we further show that the SJPDA filter can be formulated as a continuous optimization problem which is fairly easy to handle. Optimal approximations are also discussed, and an algorithm, Kullback–Leibler SJPDA (KLSJPDA), which provides optimal Gaussian approximations in the Kullback–Leibler sense is derived. Finally, we evaluate the SJPDA filter on two scenarios with closely spaced targets, and compare the performance in terms of the mean optimal subpattern assignment (MOSPA) measure with the JPDA filter, and also with the Gaussian-mixture cardinalized probability hypothesis density (GM-CPHD) filter. The results show that the SJPDA filter performs substantially better than the JPDA filter, and almost as well as the more complex GM-CPHD filter.**

***Index Terms*—Bayes methods, filtering theory, random finite set theory, recursive estimation, target tracking.**

I. Introduction

TRADITIONAL target tracking algorithms are designed to track targets over time, and discriminate between them by labeling them at each time instant. Examples of such algorithms are the probabilistic data association (PDA) and joint PDA (JPDA) filters [1], [2], and the multiple hypothesis tracking (MHT) algorithm [3]–[6]. In many applications, such as those in which evidence of target type is being accumulated, the identity (labeling) of the targets is of great importance; but in other cases where one might say that "a threat is a threat", it is not. In this article we show that by not considering target identity, the traditional algorithms can be significantly improved both in terms of density approximations and of estimation accuracy, when evaluated with a metric that disregards target identity.

For the traditional algorithms, the aim is to minimize the mean square error (MSE) between target states and corresponding track estimates. When target identity is not of interest, minimization under such a measure subjects itself to an unnecessary constraint. Instead we need a measure which only describes how good an estimate is in determining where there are targets, regardless of which is which. In this article, we propose the use of the optimal subpattern assignment (OSPA) metric [7]. More specifically, we study the mean OSPA (MOSPA). The tracking problem can thus be formulated as the scan-by-scan minimization of the MOSPA which, as we will see, is significantly different from the problem of minimizing the MSE.

A framework suitable for the description of tracking without target identity is finite set statistics (FISST) [8]. Within that framework, the aim is to track an unordered, or unlabeled, set of targets, described as a random finite set (RFS). Two popular algorithms have been derived within FISST, namely, the probability hypothesis density (PHD) and the cardinalized PHD (CPHD) filters [9]–[12]. We notice two drawbacks with FISST-based techniques. First, it is difficult to find an analytical expression for the posterior density of the RFS. Therefore, the standard technique of calculating and approximating the posterior density is not applicable. Instead, the filters operate on a first-order approximation of the density of unordered targets, referred to as the intensity function. Second, the first-order approximation has the effect that all targets be assumed independent and identically distributed. Because of these drawbacks, we believe that it is relevant to search for alternative approaches. In the current article, one such alternative is suggested, where advantages of FISST methods and those of classic recursive filtering are combined, and where it is shown how a traditional algorithm can be improved for the problem of minimizing MOSPA.

The approach that we propose relies on the fact that there is a relation between an ordered density and the density of a random finite set, here called the unordered density, or set density. In fact, there is an infinite number of ordered densities that correspond to the same set density. We refer to a set of such ordered densities as an RFS family. Since optimal MOSPA estimates can be derived from the set density, any ordered density in the corresponding RFS family can be used without losing optimality. We thus have a previously unrecognized possibility of switching ordered densities within traditional tracking filters, to obtain better approximations and performance, when target identity is irrelevant. This is useful since the different densities in the RFS family are more or less suitable to use in practice. Many tracking algorithms, such as the JPDA and MHT, rely on Gaussian approximations. That is, the posterior density is often described as a Gaussian mixture, where the number of

Manuscript received October 03, 2010; revised February 08, 2011 and June 10, 2011; accepted June 13, 2011. Date of publication July 07, 2011; date of current version September 14, 2011. The work of L. Svensson and D. Svensson was supported by National Aeronautic Research Program (NFFP), which is funded by VINNOVA (The Swedish Governmental Agency for Innovation Systems). The work of P. Willett and M. Guerriero was partially supported by the ONR under Contracts N00014-09-10613 and N00014-10-10412. The associate editor coordinating the review of this manuscript and approving it for publication Prof. Maria S. Greco.

L. Svensson is with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden (e-mail: lennart.svensson@chalmers.se; daniel.svensson@chalmers.se).

D. Svensson was with Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden (e-mail: daniel.svensson@chalmers.se).

M. Guerriero is with Far-Systems S.p.A., I-37069 Villafranca di Verona, Italy.

P. Willett is with University of Connecticut, Storrs, CT 06269 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2161294

mixture components is controlled. In the JPDA filter, which is the filter that we specifically consider in this article, the posterior density is approximated by a single Gaussian in each iteration. By combining traditional methods and FISST techniques, we show that this Gaussian approximation can be made more accurate, when target identity is not of interest. The approach is to utilize the possibility of switching from the original Gaussian mixture, to another density in the same RFS family that can be better approximated as Gaussian. We further show that there is an optimal way, in the Kullback–Leibler sense, of making that density switch. From this, we develop a refined JPDA algorithm for optimal approximations, which we call the Kullback–Leibler set JPDA (KLSJPDA). The drawback with the algorithm is its computational demand. We thus also develop a more computationally efficient algorithm, called the set JPDA (SJPDA). The SJPDA algorithm, which is also based on density switches, optimizes MOSPA performance under the constraint of remaining within the family of Gaussian mixtures, and this generally also leads to a density which is better approximated as Gaussian. Through examples and theory we discuss the reason for that. The details about the optimization criterion for the SJPDA algorithm, and its relation to optimal MOSPA estimates, are found in [13]. Throughout the article, we make the assumption that the number of targets is known. To avoid confusion, we note that "set" has been used in conjunction with various target tracking approaches in other contexts, such as with belief or Dempster–Shafer ideas [14]. Our use of "set" derives from random finite sets.

The article is outlined as follows. In Section II, the problem formulation is stated. Section III concerns two conceptual solutions to the formulated problem, based on traditional approaches and FISST, respectively. In Section IV, a conceptual solution based on the new approach is presented. We also discuss how and why traditional tracking methods can be improved when MOSPA is the target cost function. Section V considers optimal Gaussian approximations in the Kullback–Leibler sense, and the derivation of the KLSJPDA algorithm. In Section VI, the SJPDA filter is derived, and the optimization step is formulated as a continuous optimization problem. Section VII regards evaluations of the proposed KLSJPDA and SJPDA algorithms. For the KLSJPDA algorithm we observe similar performance as the SJPDA filter, but the computational complexity is much larger. For the SJDPA algorithm, we see that the tracking performance in terms of MOSPA is better than that of the JPDA filter, and almost as good as that of the more complex Gaussian mixture CPHD filter [12]. We also see that the track loss probability is dramatically decreased, which leads to a much longer track life. In Section VIII, the article is concluded.

II. Problem Formulation

A. Target Modeling Assumptions

In this article, we study the problem of tracking an arbitrary, but known and fixed, number of targets. Further, we are not interested in the identities of the targets. To reflect this, we are using a different cost function from the standard formulation in which MSE is used.

To formulate the problem, we introduce the vector of ordered (labeled) target states

$$\mathbf{X}_k = \left[\left(\mathbf{x}_k^{(1)}\right)^T \left(\mathbf{x}_k^{(2)}\right)^T \dots \left(\mathbf{x}_k^{(n)}\right)^T \right]^T \tag{1}$$

where $\mathbf{x}_k^{(i)}$ is the state vector of target number i at time k, and where n is the number of targets in the scene. All targets follow the same linear process model, given by

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{2}$$

where $\mathbf{v}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$. As seen, we sometimes omit the superscript on the state vectors, and write $\mathbf{x}_k$ instead of $\mathbf{x}_k^{(i)}$. The prior distribution of the target states, $p(\mathbf{X}_0)$, is an ordinary joint Gaussian density with known mean and covariance matrix.

We further introduce the collection of measurements $\mathbf{Z}^k$

$$\mathbf{Z}^k = \{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_k\} \tag{3}$$

up to the current time step k, where $\mathbf{Z}_k$ is a matrix of measurement vectors at time k. Each measurement matrix comprises target-generated measurements as well as spurious measurements due to false alarms and clutter. At each measurement scan, each target can give rise to at most one detection, with detection probability P_D. The relation between a target $\mathbf{x}_k$ and its measurement $\mathbf{z}_k$ is given by the measurement model

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k \tag{4}$$

where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is measurement noise. The clutter measurements are assumed Poisson distributed in number, with rate parameter $\lambda \cdot |\mathrm{FoV}|$, where λ is the clutter intensity and FoV is the field-of-view of the sensor.

B. MOSPA Measure and Optimal Estimates

The ubiquitous measure in the literature is the squared error (SE)

$$\mathrm{SE}(\hat{\mathbf{X}}_k, \mathbf{X}_k) = \sum_{i=1}^{n} \left(\hat{\mathbf{x}}_k^{(i)} - \mathbf{x}_k^{(i)}\right)^T \left(\hat{\mathbf{x}}_k^{(i)} - \mathbf{x}_k^{(i)}\right). \tag{5}$$

To evaluate the measure, we need estimates of the states of target 1, 2, and so on, i.e., of the labeled targets. Since this paper is about describing where there are targets, rather than where a target with a certain label is, the squared error is not a good measure. We instead seek a measure that can capture the quality of an algorithm to estimate the set of targets.

Several multitarget performance measures have been proposed in the literature. An ad-hoc optimal assignment-based approach, with arbitrary cost function, was given in [15], while the first rigorous theory of multi-object distances for tracking was given in [16]. The approach in [16] is based on the Wasserstein distance [17], [18], which is a widely used metric in theoretical statistics [16]. The paper introduces the so-called optimal mass transfer (OMAT) metric. When the number of targets is known, the OMAT metric is identical to the optimal assignment procedure. OMAT has a number of weaknesses, discussed in [7]. Apart from not being a metric, the major weaknesses appear when the number of targets is not

known and the estimated number is not always equal to the true number. As a remedy, a new measure called the optimal subpattern assignment (OSPA) metric was proposed in [7]. Since OSPA both is a true metric and an intuitively appealing measure which has received much attention of late, we use it as the basis measure for the problem.

Let $\mathbf{X}$ be the set of true target states and $\hat{\mathbf{X}}$ be the set of target estimates, in our case both with n elements. The OSPA measure $\tilde{d}_p^{(c)}$ is then defined as

$$\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\mathbf{X}) = \left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{n} d^{(c)}\left(\mathbf{X}^{(i)},\hat{\mathbf{X}}^{\pi(i)}\right)^p\right)\right)^{1/p}. \tag{6}$$

Here, $d^{(c)}(\mathbf{x},\hat{\mathbf{x}}) \triangleq \min(c, d(\mathbf{x},\hat{\mathbf{x}}))$ is the distance d between $\mathbf{x}$ and $\hat{\mathbf{x}}$, cut-off at c. Further, Π_n is the set of all possible permutations of $\hat{\mathbf{X}}$. The notation $\hat{\mathbf{X}}^{\pi(i)}$ describes the ith vector in permutation π of $\hat{\mathbf{X}}$. In this article, we use $p=1$ and we let d be the squared Euclidean distance, identical to $\mathrm{SE}(\cdot,\cdot)$ for single-target vectors. In practice, the measure performs an optimal assignment of target estimates to true target states, possibly clamped at c.[1]

To describe the performance of an estimator, and to have a measure for which we can define an optimal algorithm, we average over all state vectors, which gives us a definition of the *mean* OSPA (MOSPA) measure

$$\mathrm{MOSPA}_p^{(c)}(\hat{\mathbf{X}}) \triangleq \mathbb{E}_{p(\mathbf{X}\,|\,\mathbf{Z}^k)}\left\{\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\mathbf{X})\right\}. \tag{7}$$

An optimal estimator, in the MOSPA sense, is an estimator which minimizes the MOSPA measure. Such an estimator is referred to as a minimum MOSPA (MMOSPA) estimator[2].

C. Motivation of the Problem and the MOSPA Measure

In this paper we study the problem of estimating the unordered set of targets, using the MOSPA measure as performance metric. We here give three examples of when minimizing MOSPA provides a solution that is more reasonable than the one obtained when minimizing the Mean Square Error (MSE) using an ordered density.

The first application is in radar cueing, i.e., for the problem of steering a radar sensor to areas of high target existence probability. In such cases, there is no interest in which target is which, but the question of where there are targets is very important. The risk with using an ordered density for this application is that if there is an uncertainty in the labeling of the targets, the posterior density will be multimodal. Then, there is a high probability that the mean value is in an area of low target existence probability. Steering the main lobe of the sensor to such an area is thus prone to low probability of true-target returns. By disregarding ordering, the posterior density can be made less multimodal, and the main lobe can then with higher probability be directed to an area where targets are likely to be. A second example is in the automotive industry: In collision avoidance systems, it is not of interest to know which car is which—the only interest is to avoid all cars. Finally, when tracking extended objects using radar measurements, those objects are often described by a set of reflectors. The problem of interest is then to track that set of reflectors, and not try to distinguish which reflector is which.

[1]For $c=\infty$, the above measure $\tilde{d}_p^{(c)}$ is equal to the OMAT metric. A consequence of this relation, using results from [16], is that the OSPA measure for known target numbers, and $c=\infty$, reduces to the optimal assignment approach, presented earlier.

[2]Note that the relation between OSPA, MOSPA and MMOSPA is analogous to the relation between the common acronyms SE, MSE and MMSE.

TABLE I
CONCEPTUAL SOLUTION I—ORDERED DENSITIES

1)	Compute the ordered density $p(\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(n)}\,\vert\,\mathbf{Z}^k)$ recursively.
2)	Derive MMOSPA state estimates from the ordered density.

III. CONCEPTUAL SOLUTIONS TO THE MMOSPA ESTIMATION PROBLEM

For the case of target tracking, there are two well-studied optimal, or conceptual, solutions. In this section, these solutions are discussed for the MMOSPA problem. By doing so, we believe that it is easier to understand the new approach of this paper, which is based on a third conceptual solution introduced in Section IV.

The basis of both conventional solutions is first to calculate an optimal description of the joint target density, and then to derive MMOSPA estimates from the optimal description. In the description of the first conceptual solution, we also describe the data association problem, and how it is solved in the conventional frameworks.

A. Conceptual Solution I—Ordered Densities

The first conceptual solution is to use the traditional approach of first calculating the ordered posterior density $p(\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(n_k)}\,|\,\mathbf{Z}^k)$, and then to derive MMOSPA state estimates from it. For the conventional problem of minimizing MSE, the traditional approach is the foundation for methods such as JPDA and MHT. The first conceptual solution to the considered problem is presented in Table I.

There are two difficulties with this conceptual solution. First, the computation of the exact posterior density is computationally infeasible, since the problem of data association makes the complexity of the problem grow exponentially with time. One therefore has to resort to approximations. Second, it is not obvious how estimates with low MOSPA should be derived from the ordered density. As we show in Example 1 in Section IV-C, the posterior mean, which is normally the estimate used in the traditional problem of minimizing MSE, might not at all be a suitable estimate in the MOSPA sense.

B. Conceptual Solution II—Unordered Densities

The second conceptual solution relies on the fact that to compute an MMOSPA estimate, it is sufficient to know the density of unordered targets $p(\{\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(n)}\}\,|\,\mathbf{Z}^k)$, called the unordered density, or RFS density. This is true since the labeling of the targets has no influence on the MOSPA measure. An optimal solution utilizing the sufficiency of the unordered density is given in Table II.

The second conceptual solution has the same two difficulties as the first one, viz. that approximations are necessary to keep complexity at a constant level and that derivation of MMOSPA estimates from an unordered density is a research topic in itself.

TABLE II
CONCEPTUAL SOLUTION II—UNORDERED DENSITIES

1)	Compute the unordered density $p(\{\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(n)}\}\vert\mathbf{Z}^k)$ recursively.
2)	Derive MMOSPA state estimates from the unordered density.

Step one of the second conceptual solution (cf. Table II) is the foundation of the family of PHD/CPHD algorithms [9]–[12] and tracking using the joint multitarget probability density (JMPD) [19].

IV. CONCEPTUAL SOLUTION III—ORDERED DENSITIES WITH SWITCHING

In the previous section, two possible approaches to the problem of estimating the set of targets, with MOSPA as goal function, were presented. In this section, we introduce a third approach to the problem that utilizes the relation between ordered and unordered densities. This third approach is the foundation of the current paper.

The section has three subsections. First, the relationship between ordered and unordered densities is described, and the effects and possibilities of that relationship is discussed. Second, a third conceptual solution to the considered problem is presented. Finally, a simple example is used to illustrate the potential benefits of using methods based on that solution.

A. The RFS Family

The first two conceptual solutions included the calculation of the ordered, and unordered, posterior densities of the joint target states, respectively. A key insight, upon which we capitalize, is that there is a relation between these densities. For n targets, the relation is the following:

$$p\left(\left\{\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(n)}\right\}=\left\{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(n)}\right\}\right)=\sum_{\pi\in\Pi_n}p(\mathbf{X}_k=\boldsymbol{\alpha}^{\pi}) \quad (8)$$

where $\boldsymbol{\alpha}=[\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(n)}]$ is a point in the joint target state space. That is, to go from an ordered density to a set density, we sum the ordered density over all possible permutations of the state vector. To distinguish between the two densities, we denote RFS densities by $p(\{\cdot\})$, and ordered densities by $p(\,\cdot\,)$. Note that the set density is a density (see, e.g., page 362 in [8]), which integrates to one using the set integral. One important consequence of this relation is described in the following proposition.

Proposition 1: For $n>1$, the mapping from densities of ordered state vectors, $p(\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(n)})$, to RFS densities, $p(\{\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(n)}\})$, is many-to-one.

Proof of Proposition 1: Select any permutation $\pi_i\in\Pi_n$, any constant $\lambda\in(0,1)$, and any volume V for which $\int_V p(\mathbf{X})d\mathbf{X}>0$. Then

$$p_1(\mathbf{X})=\begin{cases}\lambda p(\mathbf{X}), & \text{if } \mathbf{X}\in V\\ p(\mathbf{X})+(1-\lambda)p(\mathbf{X}^{\pi_i}), & \text{if } \mathbf{X}^{\pi_i}\in V\\ p(\mathbf{X}), & \text{otherwise}\end{cases} \quad (9)$$

is such that $p(\mathbf{X})\neq p_1(\mathbf{X})$, and $p(\{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}\})=\sum_{\pi\in\Pi_n}p(\mathbf{X}=\alpha^{\pi})=\sum_{\pi\in\Pi_n}p_1(\mathbf{X}=\alpha^{\pi})$. This shows that there are several ordered densities for each set density. It is clear from (8) that an ordered density defines the corresponding set density uniquely. This concludes the proof.

Since many ordered densities correspond to the same unordered density, and since the RFS density is sufficient to derive optimal estimates, it is fair to say that the ordered density contains more information than necessary. For all densities which correspond to the same unordered density, we make the following definition:

Definition 1: The *RFS family* $\mathcal{R}_{p(\mathbf{X})}$ of an ordered density $p(\mathbf{X})$ is

$$\mathcal{R}_{p(\mathbf{X})}=\left\{p_1(\mathbf{X}):\sum_{\pi\in\Pi_n}p(\mathbf{X}=\boldsymbol{\alpha}^{\pi})=\sum_{\pi\in\Pi_n}p_1(\mathbf{X}=\boldsymbol{\alpha}^{\pi})\right\}. \quad (10)$$

So, using (8), we obtain the same RFS density regardless of whether $p(\mathbf{X})$ or $p_1(\mathbf{X})\in\mathcal{R}_{p(\mathbf{X})}$ is used. In Fig. 1, two labeled densities in the same RFS family are shown, together with the RFS density. Obviously, even though the ordered densities belong to the same RFS family, their shape, expected values and covariance matrices can be very different.

The importance of the RFS family concept is due to the following two propositions.

Proposition 2: For any two ordered densities $p(\mathbf{X})$ and $p_1(\mathbf{X})\in\mathcal{R}_{p(\mathbf{X})}$ it holds that

$$\mathbb{E}_{p(\mathbf{X})}\left\{\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\mathbf{X})\right\}=\mathbb{E}_{p_1(\mathbf{X})}\left\{\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\mathbf{X})\right\}=\mathbb{E}_{p(\{\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(n)}\})}\left\{\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\mathbf{X})\right\}. \quad (11)$$

Proof of Proposition 2: A brief proof of this result was given in [13]. Here we give a more detailed proof.
Introducing

$$V=\left\{\boldsymbol{\alpha}:\boldsymbol{\alpha}^{(1)}\leq\boldsymbol{\alpha}^{(2)}\leq\cdots\leq\boldsymbol{\alpha}^{(n)}\right\} \quad (12)$$

the MOSPA of $p(\mathbf{X})$ can be expressed as

$$\begin{aligned}\mathbb{E}_{p(\mathbf{X})}&\left\{\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\mathbf{X})\right\}\\&=\int\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\boldsymbol{\alpha})p(\mathbf{X}=\boldsymbol{\alpha})d\boldsymbol{\alpha}\\&=\sum_{\pi\in\Pi_n}\int_{\boldsymbol{\alpha}\in V}\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\boldsymbol{\alpha}^{\pi})p(\mathbf{X}=\boldsymbol{\alpha}^{\pi})d\boldsymbol{\alpha}\\&=\int_{\boldsymbol{\alpha}\in V}\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\boldsymbol{\alpha})\sum_{\pi\in\Pi_n}p(\mathbf{X}=\boldsymbol{\alpha}^{\pi})d\boldsymbol{\alpha}\\&=\int_{\boldsymbol{\alpha}\in V}\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\boldsymbol{\alpha})\sum_{\pi\in\Pi_n}p_1(\mathbf{X}=\boldsymbol{\alpha}^{\pi})d\boldsymbol{\alpha}\\&=\mathbb{E}_{p_1(\mathbf{X})}\left\{\tilde{d}_p^{(c)}(\hat{\mathbf{X}},\mathbf{X})\right\}\end{aligned} \quad (13)$$

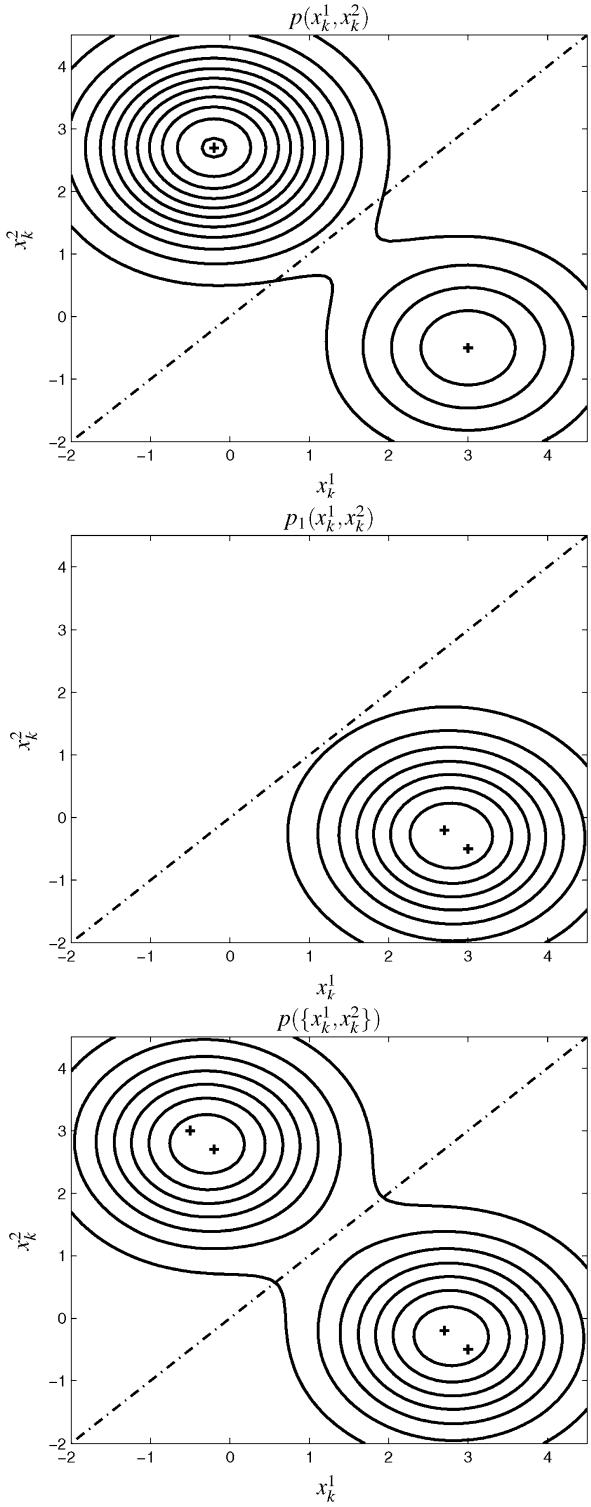

Fig. 1. Posterior densities before [left] and after [middle] switching the indexes under $\mathcal{H}_2$, and the RFS density [right]. In our opinion, a Gaussian approximation is much more appropriate in the middle figure than in the left one. The symmetry line is dash-dotted and the pluses indicate expected values of the Gaussian mixture components. The RFS density is inherently symmetric and is not affected by label switches in the ordered densities.

for any $p_1(\mathbf{X}) \in \mathcal{R}_{p(\mathbf{X})}$. In the derivation, we used that $\tilde{d}_p^{(c)}(\hat{\mathbf{X}}, \boldsymbol{\alpha}) = \tilde{d}_p^{(c)}(\hat{\mathbf{X}}, \boldsymbol{\alpha}^\pi)$ for any $\pi \in \Pi_n$. The MOSPA for the set density is given by

$$\begin{aligned}
&\mathbb{E}_{p(\{\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(n)}\})}\left\{\tilde{d}_p^{(c)}(\hat{\mathbf{X}}, \mathbf{X})\right\} \\
&\quad = \int \tilde{d}_p^{(c)}\left(\hat{\mathbf{X}}, \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\}\right) \\
&\qquad \times p\left(\left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\}\right) \delta\left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\} \\
&\quad = \frac{1}{n!} \int \tilde{d}_p^{(c)}(\hat{\mathbf{X}}, \mathbf{X}) p\left(\left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\}\right) d\mathbf{x}^{(1)} \cdots d\mathbf{x}^{(n)} \\
&\quad = \frac{1}{n!} \int \tilde{d}_p^{(c)}(\hat{\mathbf{X}}, \boldsymbol{\alpha}) \sum_{\pi \in \Pi_n} p(\mathbf{X} = \boldsymbol{\alpha}^\pi) d\boldsymbol{\alpha} \\
&\quad = \int_{\boldsymbol{\alpha} \in V} \tilde{d}_p^{(c)}(\hat{\mathbf{X}}, \boldsymbol{\alpha}) \sum_{\pi \in \Pi_n} p(\mathbf{X} \\
&\quad = \boldsymbol{\alpha}^\pi) d\boldsymbol{\alpha} = \mathbb{E}_{p(\mathbf{X})}\left\{\tilde{d}_p^{(c)}(\hat{\mathbf{X}}, \mathbf{X})\right\}. \qquad (14)
\end{aligned}$$

This concludes the proof.

Thus, all densities within an RFS family yield the same MOSPA distance. Further, that distance is equal to the MOSPA distance given by the corresponding set density.

Proposition 3: Suppose $p_1(\mathbf{X}_k \,|\, \mathbf{Z}^k) \in \mathcal{R}_{p(\mathbf{X}_k \,|\, \mathbf{Z}^k)}$. Then, it holds that $p_1(\mathbf{X}_{k+1} \,|\, \mathbf{Z}^k) \in \mathcal{R}_{p(\mathbf{X}_{k+1} \,|\, \mathbf{Z}^k)}$, and that $p_1(\mathbf{X}_{k+1} \,|\, \mathbf{Z}^{k+1}) \in \mathcal{R}_{p(\mathbf{X}_{k+1} \,|\, \mathbf{Z}^{k+1})}$.

Proof of Proposition 3: Starting with $p_1(\mathbf{X}_k \,|\, \mathbf{Z}^k)$, the RFS density after prediction is given by

$$\begin{aligned}
&p_1\left(\left\{\mathbf{x}_{k+1}^{(1)}, \ldots, \mathbf{x}_{k+1}^{(n)}\right\} \middle| \mathbf{Z}^k\right) \\
&\quad = \sum_{\pi \in \Pi_n} p_1\left(\mathbf{X}_{k+1}^\pi \middle| \mathbf{Z}^k\right) \\
&\quad = \sum_{\pi \in \Pi_n} \int p\left(\mathbf{X}_{k+1}^\pi \middle| \mathbf{X}_k^\pi = \boldsymbol{\alpha}\right) p_1\left(\mathbf{X}_k^\pi = \boldsymbol{\alpha} \middle| \mathbf{Z}^k\right) d\boldsymbol{\alpha} \\
&\quad = \sum_{\pi \in \Pi_n} \int p(\mathbf{X}_{k+1} \,|\, \mathbf{X}_k = \boldsymbol{\alpha}) p_1\left(\mathbf{X}_k^\pi = \boldsymbol{\alpha} \middle| \mathbf{Z}^k\right) d\boldsymbol{\alpha} \\
&\quad = \int p(\mathbf{X}_{k+1} \,|\, \mathbf{X}_k = \boldsymbol{\alpha}) \sum_{\pi \in \Pi_n} p_1\left(\mathbf{X}_k^\pi = \boldsymbol{\alpha} \middle| \mathbf{Z}^k\right) d\boldsymbol{\alpha} \\
&\quad = p\left(\left\{\mathbf{x}_{k+1}^{(1)}, \ldots, \mathbf{x}_{k+1}^{(n)}\right\} \middle| \mathbf{Z}^k\right). \qquad (15)
\end{aligned}$$

Thus, $p_1(\mathbf{X}_{k+1} \,|\, \mathbf{Z}^k) \in \mathcal{R}_{p(\mathbf{X}_{k+1} \,|\, \mathbf{Z}^k)}$. The RFS density of $p_1(\mathbf{X}_{k+1} \,|\, \mathbf{Z}^{k+1})$ after Bayesian data update is

$$\begin{aligned}
&p_1\left(\left\{\mathbf{x}_{k+1}^{(1)}, \ldots, \mathbf{x}_{k+1}^{(n)}\right\} \middle| \mathbf{Z}^{k+1}\right) \\
&\quad = \sum_{\pi \in \Pi_n} p_1\left(\mathbf{X}_{k+1}^\pi \middle| \mathbf{Z}^{k+1}\right) \\
&\quad = \sum_{\pi \in \Pi_n} \frac{p\left(\mathbf{Z}_{k+1} \middle| \mathbf{X}_{k+1}^\pi\right) p_1\left(\mathbf{X}_{k+1}^\pi \middle| \mathbf{Z}^k\right)}{p(\mathbf{Z}_{k+1} \,|\, \mathbf{Z}^k)} \\
&\quad = \frac{p(\mathbf{Z}_{k+1} \,|\, \mathbf{X}_{k+1})}{p(\mathbf{Z}_{k+1} \,|\, \mathbf{Z}^k)} \sum_{\pi \in \Pi_n} p_1\left(\mathbf{X}_{k+1}^\pi \middle| \mathbf{Z}^k\right) \\
&\quad = \frac{p(\mathbf{Z}_{k+1} \,|\, \mathbf{X}_{k+1})}{p\left(\mathbf{Z}_{k+1} \,|\, \mathbf{Z}^k\right)} \sum_{\pi \in \Pi_n} p\left(\mathbf{X}_{k+1}^\pi \middle| \mathbf{Z}^k\right) \qquad (16)
\end{aligned}$$

where the final result follows from the first part of the proposition. This concludes the proof.

As we can see, all labeled densities that belong to the same RFS

TABLE III
CONCEPTUAL SOLUTION III—ORDERED DENSITIES WITH SWITCHING

1)	Compute the ordered density $p(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n_k)} \mid \mathbf{Z}^k)$ recursively.
2)	Replace $p(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n_k)} \mid \mathbf{Z}^k)$ with another ordered density $p_1(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n_k)} \mid \mathbf{Z}^k)$ which corresponds to the same RFS density.
3)	Derive MMOSPA state estimates from the ordered density $p_1(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(n_k)} \mid \mathbf{Z}^k)$.

family $\mathcal{R}_{p(\mathbf{X}_k \mid \mathbf{Z}^k)}$ at time step k, still belong to the same (but updated) RFS family $\mathcal{R}_{p(\mathbf{X}_{k+1} \mid \mathbf{Z}^{k+1})}$ at the next time step. To phrase this differently, all densities that belong to the same RFS family should result in the same MMOSPA estimates, both now and for all future times, i.e., also when new data is available. We can therefore replace our density at hand with any other density within the RFS family, without influencing the estimates. We conclude our findings in a corollary.

Corollary 1: In a recursive filtering framework, we may replace the posterior density $p(\mathbf{X}_k \mid \mathbf{Z}^k)$ with any density $p_1(\mathbf{X}_k \mid \mathbf{Z}^k) \in \mathcal{R}_{p(\mathbf{X}_k \mid \mathbf{Z}^k)}$ without affecting any current or future MMOSPA estimates.

B. A Third Conceptual Solution

From Corollary 1, we see that there is a previously unrecognized possibility of switching densities at any time in a filtering framework, without affecting the optimal performance. Based on this, we can describe a third conceptual solution (see Table III) to the problem of this paper—a conceptual solution that utilizes the possibility of switching densities within the RFS family.

This conceptual solution provides us with new possibilities and tools for designing novel suboptimal algorithms with better performance. As a switch within the RFS family does not affect the estimates, we could for instance make a change to a density which is more accurately approximated by a Gaussian density. The relevance of that is obvious, considering for example the JPDA filter, which relies heavily on Gaussian approximations. An example of such a switch of densities is seen in Fig. 1. By switching to the density p_1, a more accurate Gaussian approximation is possible.

In practice, the posterior mean is often used as estimate even when the cost function is not the squared error. With the same reasoning as above, there may thus also exist densities within the RFS family whose expected value is closer to the optimal estimate than the expected value of the original density. In fact, in [13] it is shown that there exists a density in the RFS family whose mean value is the MMOSPA estimate. In Section IV.C, we study an example which shows that a density switch can lead to both improved approximations and to improved state estimates when the posterior mean is the estimator.

C. An Example of the Use of an RFS Family

By considering an example, we here illustrate and discuss the potential benefits of density switches, both in terms of accuracy of the Gaussian approximations and the accuracy of the state estimates, when the posterior mean is used as estimate.

Example 1: Consider a scenario where we have two Gaussian distributed targets. The example is illustrated in Fig. 1. The probability of detection is one for both targets, and we have received two detections. We represent the two possible data associations by the hypotheses $\mathcal{H}_1$ and $\mathcal{H}_2$. Now, suppose that calculations yield the numbers,[3]

$$\Pr\{\mathcal{H}_1\} = 0.3, \quad \mathbf{X}_k \mid \mathcal{H}_1 \sim \mathcal{N}\left(\begin{bmatrix} 3 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \tag{17}$$

$$\Pr\{\mathcal{H}_2\} = 0.7, \quad \mathbf{X}_k \mid \mathcal{H}_2 \sim \mathcal{N}\left(\begin{bmatrix} -0.2 \\ 2.7 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right). \tag{18}$$

We realize that the two sets $\{x_k^{(1)} = \beta_1, x_k^{(2)} = \beta_2\}$ and $\{x_k^{(1)} = \beta_2, x_k^{(2)} = \beta_1\}$ represent the same set of targets. We may therefore move density from one such labeled point to the other, without changing the RFS density. One way to do this is by switching the indexes under $\mathcal{H}_2$,

$$\Pr\{\mathcal{H}_1\} = 0.3, \quad \mathbf{X}_k \mid \mathcal{H}_1 \sim \mathcal{N}\left(\begin{bmatrix} 3 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \tag{19}$$

$$\Pr\{\mathcal{H}_2\} = 0.7, \quad \mathbf{X}_k \mid \mathcal{H}_2 \sim \mathcal{N}\left(\begin{bmatrix} 2.7 \\ -0.2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right). \tag{20}$$

The densities $p(\mathbf{X}_k)$, given by (17)–(18), and $p_1(\mathbf{X}_k)$, given by (19)–(20), thus correspond to the same RFS density

$$p\left(\left\{x_k^1, x_k^2\right\}\right) = p(\mathbf{X}_k) + p(\pi\mathbf{X}_k) = p_1(\mathbf{X}_k) + p_1(\pi\mathbf{X}_k) \tag{21}$$

where the matrix π is a permutation matrix, defined for this simple two-scalar-object case as

$$\pi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{22}$$

Since the ordered densities p and p_1 belong to the same RFS family, they should, ideally, render the same estimates. As illustrated in Fig. 1, the suggested switch leads to a simpler problem, since *the marginalized densities of $x_k^{(1)}$ and $x_k^{(2)}$ can be approximated by a Gaussian density more accurately*.

The dash-dotted line in Fig. 1 is a symmetry line. When probability mass is moved from one labeled point to another, the movement is through this line to the mirror point. Of course, the example is selected to highlight the advantages with a switch, and one can easily construct situations when it is better not to switch indices. Still, the example illustrates a general technique that can be employed by most tracking algorithms that use merging.

[3]All probabilities and densities are conditioned on data, but this is omitted for notational convenience.

To further improve the understanding of the concepts in Example 1, we stress the relation to the RFS densities. Let $\tilde{p}(x_k^{(1)}, x_k^{(2)})$ and $\tilde{p}_1(x_k^{(1)}, x_k^{(2)})$ denote the Gaussian approximations of $p(x_k^{(1)}, x_k^{(2)})$ and $p_1(x_k^{(1)}, x_k^{(2)})$, respectively. Both $p(x_k^{(1)}, x_k^{(2)})$ and $p_1(x_k^{(1)}, x_k^{(2)})$ correspond to the same RFS density, i.e., $p(\{x_k^{(1)}, x_k^{(2)}\}) = p_1(\{x_k^{(1)}, x_k^{(2)}\})$ (cf. (21)). As the approximation $\tilde{p}_1(x_k^{(1)}, x_k^{(2)}) \approx p_1(x_k^{(1)}, x_k^{(2)})$ is fairly accurate, it follows that $\tilde{p}_1(\{x_k^{(1)}, x_k^{(2)}\}) \approx p_1(\{x_k^{(1)}, x_k^{(2)}\}) = p(\{x_k^{(1)}, x_k^{(2)}\})$. However, it is likely that the approximation $\tilde{p}(\{x_k^{(1)}, x_k^{(2)}\}) \approx p(\{x_k^{(1)}, x_k^{(2)}\})$ is less accurate. Hence, by switching densities we will make approximations that better preserve the information about the desired RFS density, $p_1(\{x_k^{(1)}, x_k^{(2)}\})$.

It is not obvious how to compute MMOSPA estimates from a given RFS density. Instead, the proposed algorithms in Sections V and VI use the MMSE estimate, i.e., the posterior mean, of a density of ordered targets. The idea is to select a density within the RFS family such that the MMSE estimates are close to the MMOSPA estimates. For the considered example, we illustrate the importance that the choice of density has on the posterior means and the MOSPA performance, by studying the MMSE state vector estimates before and after the index switch. (For additional information on the relation between the MMSE and the MMOSPA estimates, see [13].) In the original indexation, the posterior means are

$$\hat{x}_k^{(1)} = 0.76, \quad \hat{x}_k^{(2)} = 1.74 \tag{23}$$

whereas the posterior means after the switch are

$$\hat{x}_k^{(1)} = 2.79, \quad \hat{x}_k^{(2)} = -0.29. \tag{24}$$

The latter posterior means are probably close to the optimal estimates since, under both hypotheses, one target is fairly close to 2.79 whereas the other target is reasonably close to -0.29. Hence, although the initial objective was to improve the Gaussian approximation, the density switch also seems to yield MMSE estimates which are closer to the MMOSPA estimates.

In the following two sections, we utilize the possibility of performing density switches to derive two tracking algorithms. In Section V, an algorithm for optimal Gaussian approximations is derived, while Section VI presents a computationally more efficient algorithm that both enables accurate Gaussian approximations and improved state estimates (compared to the JPDA filter).

V. Optimal Approximations in the Kullback–Leibler Sense

In this section, we present a way of switching densities within the RFS family such that the new density can be most accurately approximated with a Gaussian density, in the Kullback–Leibler sense. Note that the description is only made for a two-target case, but the results can be generalized.

In what follows, we assume that the posterior density is a Gaussian mixture, denoted

$$p(\mathbf{X}) = \sum_{h=1}^{N_{\mathcal{H}}} \beta_h \mathcal{N}(\mathbf{X}; \boldsymbol{\mu}_h, \mathbf{P}_h) \tag{25}$$

where β_h is given by the JPDA filter (see (57)). For the two-target case, the RFS family $\mathcal{R}_{p(\mathbf{X})}$ of $p(\mathbf{X})$ is expressed as (cf. (10))

$$\mathcal{R}_{p(\mathbf{X})} = \{f : f(\mathbf{X}) + f(\pi\mathbf{X}) = p(\mathbf{X}) + p(\pi\mathbf{X})\}. \tag{26}$$

Here, $\mathbf{X}$ is the stacked vector of the state vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ of the two targets, and π is a permutation matrix (cf. (22)).

Depending on the choice of density $f(\mathbf{X})$, the Gaussian approximation may be more or less accurate. Our objective here is to

1) find a density $f(\mathbf{X})$ that enables the most accurate Gaussian approximations;
2) and then to find the Gaussian density $\mathcal{N}(\mathbf{X}; \bar{\mathbf{X}}, \mathbf{R})$ that best approximates $f(\mathbf{X})$.

As a measure for how accurate a density approximation is, we will use the Kullback–Leibler divergence [20]. Of course, our interest in $f(\mathbf{X})$ is only as a means to obtain $\mathcal{N}(\mathbf{X}; \bar{\mathbf{X}}, \mathbf{R})$, which means that one could imagine a technique that does not involve $f(\mathbf{X})$. The mathematical definition of the problem that we would like to solve is

$$\{\bar{\mathbf{X}}, \mathbf{R}\} = \arg\min_{\bar{\mathbf{X}}, \mathbf{R}} \left\{ \min_{f(\mathbf{X}) \in \mathcal{R}_{p(\mathbf{X})}} \mathrm{KL}[f(\mathbf{X}) \,\|\, \mathcal{N}(\mathbf{X}; \bar{\mathbf{X}}, \mathbf{R})] \right\} \tag{27}$$

where

$$\mathrm{KL}[f(\mathbf{X}) \,\|\, \mathcal{N}(\mathbf{X}; \bar{\mathbf{X}}, \mathbf{R})] \triangleq \int f(\mathbf{X}) \log \frac{f(\mathbf{X})}{\mathcal{N}(\mathbf{X}; \bar{\mathbf{X}}, \mathbf{R})} \, d\mathbf{X} \tag{28}$$

and where KL denotes Kullback–Leibler divergence. In the standard procedure (used in for instance PDA and JPDA) one would use $f(\mathbf{X}) = p(\mathbf{X})$ and only optimize over $\bar{\mathbf{X}}$ and $\mathbf{R}$, for which the optimal solution is given by moment matching. By also optimizing over $f(\mathbf{X})$ we will show that the approximation errors will decrease significantly compared to JPDA, and slightly compared to the algorithm that we present in Section VI.

To design an algorithm that can search for $\bar{\mathbf{X}}$ and $\mathbf{R}$, the following results are very useful.

Theorem 1: The solution to

$$\{\bar{\mathbf{X}}, \mathbf{R}\} = \arg\min_{\bar{\mathbf{X}}, \mathbf{R}} \mathrm{KL}[f(\mathbf{X}) \,\|\, \mathcal{N}(\mathbf{X}; \bar{\mathbf{X}}, \mathbf{R})] \tag{29}$$

is given by moment matching,

$$\bar{\mathbf{X}} = \mathbb{E}_{f(\mathbf{X})}\{\mathbf{X}\} \tag{30}$$

$$\mathbf{R} = \mathrm{Cov}_{f(\mathbf{X})}\{\mathbf{X}\}. \tag{31}$$

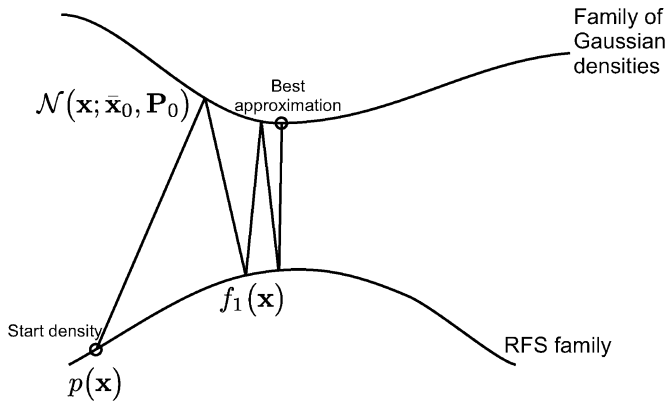

Fig. 2. Illustration of the iterative optimization algorithm which finds the best Gaussian approximation of the RFS family, in the Kullback–Leibler sense. The curves represent the functional spaces of densities within the RFS family (below), and the Gaussian densities (above).

Furthermore, for the two-target case, the density $f(\mathbf{X}) \in \mathcal{R}_{p(\mathbf{X})}$ that minimizes $\mathrm{KL}[f(\mathbf{X}) \,||\, \mathcal{N}(\mathbf{X};\bar{\mathbf{X}},\mathbf{R})]$ is

$$f(\mathbf{X}) = (p(\mathbf{X}) + p(\pi\mathbf{X})) \cdot \frac{\mathcal{N}(\mathbf{X};\bar{\mathbf{X}},\mathbf{R})}{\mathcal{N}(\mathbf{X};\bar{\mathbf{X}},\mathbf{R}) + \mathcal{N}(\pi\mathbf{X};\bar{\mathbf{X}},\mathbf{R})}. \quad (32)$$

Proof of Theorem 1: The results in (30) and (31) are well known, see, e.g., [21]. For (32), see [22]. Further, it is possible to show (see [22]) that $f(\mathbf{X})$ is the unique global minimizer of the function $\mathrm{KL}[f(\mathbf{X}) \,||\, \mathcal{N}(\mathbf{X};\bar{\mathbf{X}},\mathbf{R})]$.

Based on the above theorem, we propose an iterative optimization algorithm. The algorithm is illustrated in Fig. 2, where the initial density is $p(\mathbf{X})$.

1) Initiate with $i = 1$ and let $\bar{\mathbf{X}}_0$ and $\mathbf{R}_0$ be the first two moments of $p(\mathbf{X})$.
2) Set

$$f_i(\mathbf{X}) = (p(\mathbf{X}) + p(\pi\mathbf{X})) \times \frac{\mathcal{N}(\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1})}{\mathcal{N}(\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1}) + \mathcal{N}(\pi\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1})}. \quad (33)$$

3) Compute

$$\bar{\mathbf{X}}_i = \mathbb{E}_{f_i(\mathbf{X})}\{\mathbf{X}\} \quad (34)$$
$$\mathbf{R}_i = \mathrm{Cov}_{f_i(\mathbf{X})}\{\mathbf{X}\}. \quad (35)$$

4) If $\bar{\mathbf{X}}_i \approx \bar{\mathbf{X}}_{i-1}$ and $\mathbf{R}_i \approx \mathbf{R}_{i-1}$ we stop. Otherwise, set $i = i + 1$ and go back to 2).

The algorithm is essentially straightforward. However, to implement it we need the ability to calculate expected values with respect to the densities $f_i(\mathbf{X})$, $i = 1, 2, \ldots$. Unfortunately, the algorithm is not guaranteed to converge to an optimal point. However, the Kullback–Leibler divergence can be shown to be monotonically decreasing. Say that we have $\bar{\mathbf{X}}_{i-1}$, $\mathbf{R}_{i-1}$ and the function f_{i-1} from iteration $i-1$. At iteration i one first finds the function f_i that minimizes $\mathrm{KL}[f(\mathbf{X}) \,||\, \mathcal{N}(\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1})]$. Second, $\bar{\mathbf{X}}_i$ and $\mathbf{R}_i$ are the minimum values of the divergence between f_i and a Gaussian density. Thus, it follows that

$$\begin{aligned}&\mathrm{KL}[f_i(\mathbf{X}) \,|\, \mathcal{N}(\mathbf{X};\bar{\mathbf{X}}_i,\mathbf{R}_i)] \\ &\quad \le \mathrm{KL}[f_i(\mathbf{X}) \,||\, \mathcal{N}(\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1})] \\ &\quad \le \mathrm{KL}[f_{i-1}(\mathbf{X}) \,|\, \mathcal{N}(\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1})] \end{aligned} \quad (36)$$

which shows the monotonicity. Additionally, due to the Kullback–Leibler divergence being bounded from below, the monotonicity implies that the algorithm will converge, albeit not necessarily to the minimum point.

It appears complicated to find analytical expressions for the expected values in (34) and (35) and we therefore suggest a numerical method based on importance sampling. As importance density we use

$$q(\mathbf{X}) = \frac{1}{2}(p(\mathbf{X}) + p(\pi\mathbf{X})). \quad (37)$$

The expected values can be reformulated as

$$\bar{\mathbf{X}}_i = \mathbb{E}_{f_i(\mathbf{X})}\{\mathbf{X}\} = \mathbb{E}_{q(\mathbf{X})}\left\{\mathbf{X}\frac{f_i(\mathbf{X})}{q(\mathbf{X})}\right\} \quad (38)$$

$$\begin{aligned}\mathbf{R}_i &= \mathbb{E}_{f_i(\mathbf{X})}\left\{(\mathbf{X}-\bar{\mathbf{X}}_i)(\mathbf{X}-\bar{\mathbf{X}}_i)^T\right\} \\ &= \mathbb{E}_{q(\mathbf{X})}\left\{(\mathbf{X}-\bar{\mathbf{X}}_i)(\mathbf{X}-\bar{\mathbf{X}}_i)^T\frac{f_i(\mathbf{X})}{q(\mathbf{X})}\right\}\end{aligned} \quad (39)$$

and for the selected importance density it is easy to evaluate the ratio

$$\frac{f_i(\mathbf{X})}{q(\mathbf{X})} = 2\frac{\mathcal{N}(\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1})}{\mathcal{N}(\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1}) + \mathcal{N}(\pi\mathbf{X};\bar{\mathbf{X}}_{i-1},\mathbf{R}_{i-1})}. \quad (40)$$

Based on i.i.d. samples $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N \sim q(\mathbf{X})$, we can approximate the desired entities as

$$\bar{\mathbf{X}}_i \approx \sum_{n=1}^{N} \mathbf{X}_n w_n \quad (41)$$

$$\mathbf{R}_i \approx \sum_{n=1}^{N} (\mathbf{X}_n - \bar{\mathbf{X}}_i)(\mathbf{X}_n - \bar{\mathbf{X}}_i)^T w_n \quad (42)$$

where

$$w_n = \frac{f_i(\mathbf{X}_n)/q(\mathbf{X}_n)}{\sum_{r=1}^{N} f_i(\mathbf{X}_r)/q(\mathbf{X}_r)}. \quad (43)$$

In (42), we replace $\bar{\mathbf{X}}_i$ with the approximated value from (41). A beneficial property with the suggested importance function is that since the RFS family is preserved we can use the same samples $\mathbf{X}_1, \ldots, \mathbf{X}_N$ for all iterations $i = 1, 2, \ldots$. After the optimization algorithm has been run, the resulting ordered density is approximated by a single Gaussian density by moment matching. We call the complete algorithm the Kullback–Leibler Set JPDA (KLSJPDA) algorithm.

In this section, we have presented the optimal approach of performing density switches within the RFS family, such that the final density can most accurately be approximated with a Gaussian density. We have also presented a numerical approach of computing the required expected values. Since the presented algorithm is computationally demanding, we are interested in a

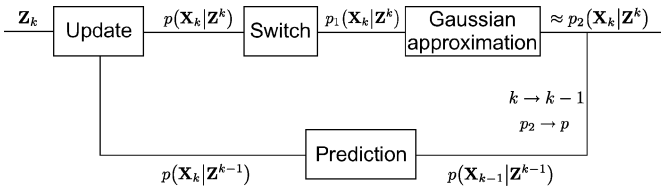

Fig. 3. SJPDA block diagram.

sub-optimal approach, which still has the properties of enabling better Gaussian approximations than the JPDA algorithm, and which also presents better estimates than JPDA in the MOSPA sense. The development of such an approach is the topic of the following section.

VI. Set JPDA Algorithm

In this section, we derive a suboptimal approach to the third conceptual solution, called the Set JPDA (SJPDA) algorithm. The filter utilizes the fact that several ordered densities correspond to the same set density, which provides the option of switching between those ordered densities.

The Set JPDA algorithm is a modification of the classic JPDA filter. The difference is that once the posterior density is described as a weighted sum of Gaussian densities, we allow ourselves to switch that density for another ordered density in the same RFS family. On a high level, the SJDA algorithm works as follows:

1) Formulate the set of global measurement hypotheses, $\mathcal{H}$, and calculate conditional densities of all targets as well as the probabilities of all hypotheses. Approximate the targets as independent.
2) Reorder the target indexes under the different hypotheses with the objective to make the marginalized densities resemble Gaussian densities (by minimization in one's favorite sense, see Sections VI-A and VI-B for details).
3) Approximate the marginalized posterior densities of all targets as independent Gaussian. Then go back to 1).

Note that steps 1) and 3) are identical to the JPDA filter.

A block diagram description of the SJPDA filter is shown in Fig. 3. As seen in the figure, the difference between SJPDA and JPDA lies in the switching block. If that block is removed, we obtain the JPDA filter.

The key aspect of the SJPDA filter is the switching of densities. To obtain a filter with good performance, and one that is accurately approximated as Gaussian, the switching criterion is very important. In the following two sections, we propose a goal function, and formulate the problem of minimizing that function (while remaining within the family of Gaussian mixtures) as a continuous optimization problem.

A. Goal Function Proposal and Motivation

For the problem considered, the labeled density is a Gaussian mixture. In this section, we propose and motivate a goal function for the optimization problem of finding the best switching of such densities in the SJPDA filter.

Generally speaking, a Gaussian mixture can be accurately approximated by a single Gaussian density as long as the Gaussian mixture is not too distinctly multimodal. In the SJPDA filter, we wish to adjust the indexation within each global hypothesis, in order to find a labeled density which is less multimodal. The goal function that we propose is hence a function that measures the multimodality of the density. The proposed goal function is

$$\sum_{i=1}^{n} \operatorname{tr}\left\{\mathbf{P}_k^i\right\} \tag{44}$$

for which $\mathbf{P}_k^i$ is the posterior covariance of target i after moment matching. That minimization of this function leads to a less multimodal density is seen in (51), where the minimum point is given by the re-indexing the global hypotheses such that the expected value under each hypothesis is as close as possible to the joint expected value $\bar{\mathbf{x}}$. Indeed, this gives a more unimodal density. In the following, we motivate that goal function, basically by arguing that through minimization of the function, we will obtain both better approximations and better estimates than what we would obtain using the JPDA filter.

In [13], the problem of minimizing MOSPA is considered in more detail. An important result from the paper is that *the problem of minimizing the sum of the trace of the posterior covariances is equivalent to the problem of minimizing MOSPA (assuming $c = \infty$)*, given that the expected value is used as estimate and that we search in the entire RFS family (not restricting the search to Gaussian mixtures in the RFS family). Thus, the switching criterion that we use is optimal for the estimation problem.

The second advantage with the cost function is that it enables better Gaussian approximations. The reason for this is that by minimizing the trace of the covariance matrices, we make the posterior density less multimodal. We, therefore, argue that the Gaussian approximations in the SJPDA filter are at least as accurate as the JPDA filter approximations, where equality holds if no switches are made. However, for the density approximations, there are no optimality results.

B. SJPDA as a Continuous Optimization Problem

In [23], the switching problem of SJPDA was formulated as a discrete optimization, for which a brute force solution can be implemented easily. In this section, we instead show how the problem can be reformulated as a continuous optimization problem, for which fast solvers can be applied.

We start by assuming that the posterior density is described by a Gaussian mixture, as in (25), i.e., one which is given by enumerating all data association hypotheses, and calculating their weights using the JPDA algorithm. To give the problem a continuous formulation, we introduce the variable $\phi_{i,h}$ which represents the weight of the Gaussian component after permutation. In this way, we can have a linear combination of the original component and the permuted component. The posterior after switching is given by

$$q(\mathbf{x}) = \sum_{i=1}^{n!} \sum_{h=1}^{N_{\mathcal{H}}} \phi_{i,h} \mathcal{N}\left(\mathbf{x}; \pi_i \boldsymbol{\mu}_h, \pi_i \mathbf{P}_h \pi_i^T\right) \tag{45}$$

where π_i is permutation matrix number i, and where $n!$ is the total number of possible permutations. The new weights have to fulfill the following constraints:

$$\sum_{i=1}^{n!} \phi_{i,h} = \beta_h \quad \forall h \tag{46}$$

$$\phi_{i,h} \geq 0 \quad \forall \quad i,h. \tag{47}$$

Note that the densities $\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_h,\mathbf{P}_h)$ and $\mathcal{N}(\mathbf{x};\pi_i\boldsymbol{\mu}_h,\pi_i\mathbf{P}_h\pi_i^T)$ lie in the same RFS family for all permutation matrices π_i.

To conveniently express the goal function in the above parameters, we introduce the notations shown in (48) and (49) at the bottom of the page such that

$$\bar{\mathbf{x}} \triangleq \sum_{i=1}^{n!}\sum_{h=1}^{N_{\mathcal{H}}} \pi_i\boldsymbol{\mu}_h\phi_{i,h} = \mathbf{V}\boldsymbol{\phi}. \tag{50}$$

We wish to minimize

$$\begin{aligned}
&\mathrm{tr}\left\{\mathrm{Cov}_{q(\mathbf{x})}\{\mathbf{x}\}\right\} \\
&\quad= \sum_{i=1}^{n!}\sum_{h=1}^{N_{\mathcal{H}}} \phi_{i,h}\left\{(\pi_i\boldsymbol{\mu}_h - \bar{\mathbf{x}})^T(\pi_i\boldsymbol{\mu}_h - \bar{\mathbf{x}}) + \mathrm{tr}\left\{\pi_i\mathbf{P}_h\pi_i^T\right\}\right\} && (51)\\
&\quad= \left\{\sum_{i=1}^{n!}\sum_{h=1}^{N_{\mathcal{H}}} \phi_{i,h}\boldsymbol{\mu}_h^T\boldsymbol{\mu}_h\right\} - \bar{\mathbf{x}}^T\bar{\mathbf{x}} + \sum_{h=1}^{N_{\mathcal{H}}}\mathrm{tr}\{\mathbf{P}_h\}\sum_{i=1}^{n!}\phi_{i,h} && (52)\\
&\quad= \left\{\sum_{h=1}^{N_{\mathcal{H}}} \beta_h\boldsymbol{\mu}_h^T\boldsymbol{\mu}_h\right\} - \boldsymbol{\phi}^T\mathbf{V}^T\mathbf{V}\boldsymbol{\phi} + \sum_{h=1}^{N_{\mathcal{H}}}\mathrm{tr}\{\mathbf{P}_h\}\beta_h && (53)
\end{aligned}$$

where $-\boldsymbol{\phi}^T\mathbf{V}^T\mathbf{V}\boldsymbol{\phi}$ is the only part that depends on $\boldsymbol{\phi}$, subject to the constraints presented in (46) and (47). To summarize, we have the optimization problem

$$\min_{\boldsymbol{\phi}} \quad -\boldsymbol{\phi}^T\mathbf{V}^T\mathbf{V}\boldsymbol{\phi}. \tag{54}$$
$$\text{s.t.} \begin{cases} \phi_{i,h} \geq 0 & \forall i,h \\ \sum_{i=1}^{n!}\phi_{i,h} = \beta_h & \forall h \end{cases}$$

The problem is non-convex. The goal function is concave, and it is to be minimized over a convex region, which implies that the optimal point is along the border of the constraints. We also note that since the optimal solution is on the border of the constraints, the permutations either occur or not, i.e., there will be no partial permutations in the optimal point. A benefit with the continuous formulation is that it enables the use of commercial optimization solvers, which scale well with increasing number of targets and hypotheses. To use an optimization solver, we need knowledge regarding existence of local minima, and how they can be avoided.

TABLE IV
SJPDA ALGORITHM DESCRIPTION

I: Data hypothesis extraction
  formulate all $N_{\mathcal{H}}$ global data association hypotheses
II: Data update
  for each data association hypothesis $h = 1, 2, \ldots, N_{\mathcal{H}}$, do
    for each target $i = 1, \ldots, n$, do
      update the expected value and covariance matrix of the predicted state vector $\hat{\mathbf{x}}_k^i$ according to (55)–(56)
  calculate the mixture weight β_h according to (57)
III: Optimization
  solve the problem defined in (54) using an optimization solver with a suitable selection of initiation point(s)
IV: Permutation and update
  compute the expected value and covariance matrix of the joint state vector $\mathbf{X}_k$ after data update, optimal permutation and Gaussian approximation, according to (65)–(66)
V: Prediction
  for each target $i = 1, 2, \ldots, n$, do
    predict the state vector and covariance matrix of target i, according to (67)–(68).

It can be analytically shown that there are no local minima for the case of scalar state vectors and two targets. But when the dimensionality of the state vector increases, there will in some situations exist local minimum points in which an optimization algorithm can be trapped. It is thus important to select a suitable starting point for the algorithm. Through empirical studies, we have seen that it is only in a small region around the minimum that the negative gradient points in the direction of the local minimum. Therefore, only starting points in a small region around potential local minima should be avoided. For the problems considered in this article, we have found a suitable set of starting points which almost always lead the selected optimization algorithm to the global minimum. For more details on the characteristics of the optimization problem, see [22].

C. Algorithm Description

The SJPDA algorithm has five steps which are executed for each time index k. In the following, we describe each of those five steps in more detail, and summarize the algorithm in Table IV.

The algorithm description starts at time k, at which we have available a set of measurements, $\mathbf{Z}_k$, and predicted states $\mathbf{x}_{k\,|\,k-1}^i$ and covariance matrices $\mathbf{P}_{k\,|\,k-1}^i$ for each target i. Note that steps I, II, and V are identical to the JPDA algorithm (one version of it). Note also that we assume linear and Gaussian process and measurement models in the description, although the SJPDA algorithm can be easily extended to handle nonlinear models using, for example, an extended Kalman filter (EKF).

$$\boldsymbol{\phi} = \begin{bmatrix}\phi_{1,1} & \ldots & \phi_{n!,1} & \phi_{1,2} & \ldots & \phi_{n!,2} & \ldots & \phi_{n!,N_{\mathcal{H}}}\end{bmatrix}^T \tag{48}$$

$$\mathbf{V} = \begin{bmatrix}\pi_1\boldsymbol{\mu}_1 & \ldots & \pi_{n!}\boldsymbol{\mu}_1 & \pi_1\boldsymbol{\mu}_2 & \ldots & \pi_{n!}\boldsymbol{\mu}_2 & \ldots & \pi_{n!}\boldsymbol{\mu}_{N_{\mathcal{H}}}\end{bmatrix} \tag{49}$$

Step I: Data Hypothesis Extraction: The first step of the SJPDA algorithm is to formulate all possible global data association hypotheses, $\mathcal{H}$. A hypothesis is possible if it describes the origin of each measurement in $\mathbf{Z}_k$ (target-generated or false), and if the total number of target-originated measurements is at most n.

Step II: Data Update: The second step of the algorithm is to update the predicted state vectors and covariance matrices, and to calculate the mixture weights, β_h. For a certain target i, the updated state and covariance matrix under hypothesis h are given by

$$\mathbf{x}_{k\,|\,k}^{i,h} = \mathbf{x}_{k\,|\,k-1}^{i} + \mathbf{K}_k^i \left(\mathbf{H}_k \mathbf{x}_{k\,|\,k-1}^{i} - \mathbf{z}_k^{i,h} \right) \tag{55}$$

$$\mathbf{P}_{k\,|\,k}^{i,h} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k\,|\,k-1}^{i} \tag{56}$$

where $\mathbf{z}_k^{i,h}$ is the measurement associated to target i under hypothesis h. If no measurement is associated, we use $\mathbf{z}_k^{i,h} = \mathbf{0}$. Further, the mixture weight is given by the JPDA filter equations

$$\begin{aligned} \beta_h &= \frac{\bar{\beta}_h}{\sum_{h=1}^{N_{\mathcal{H}}} \bar{\beta}_h} \\ \bar{\beta}_h &= \prod_{\mathcal{S}_0^h} \lambda \cdot \prod_{\mathcal{S}_u^h} (1 - P_D) \cdot \prod_{\{i,j\} \in \mathcal{S}_a^h} P_D \, g_{ij} \end{aligned} \tag{57}$$

where a constant detection probability is assumed, $\mathcal{S}_0^h$ is the set of unassigned measurements, $\mathcal{S}_u^h$ is the set of unassigned targets, and $\mathcal{S}_a^h$ is a set including the pairs of detected targets, i, and their assigned measurements, j. Further,

$$\begin{aligned} g_{ij} &= \mathcal{N}(\boldsymbol{\nu}_{ij}; \mathbf{0}, \mathbf{S}_k) = \frac{1}{|2\pi \mathbf{S}_k|^{1/2}} e^{-\frac{d_{ij}^2}{2}} \\ d_{ij}^2 &= \boldsymbol{\nu}_{ij}^T \mathbf{S}_k^{-1} \boldsymbol{\nu}_{ij}, \quad \boldsymbol{\nu}_{ij} = \mathbf{z}_k^j - \mathbf{H}_k \mathbf{x}_{k\,|\,k-1}^{(i)}. \end{aligned} \tag{58}$$

Step III: Optimization: Step three of the SJPDA algorithm is the main step, namely to find the optimal permutation of state vectors under the data association hypotheses. We find the optimum solution, $\boldsymbol{\phi}^*$, to (54) by applying an optimization solver with different initiation points. A suitable pair of starting points for the two-target problem in Section VII is shown in (59) and (60) at the bottom of the page and a useful third initiation point is to perform a 70% permutation under the N (e.g., 10) most probable hypotheses. In real time implementations, the optimization algorithm may not always converge in time for the next recursion. In such situations we use the best ϕ vector found so far. However, for the scenarios considered thus far, convergence of the SJPDA algorithm has been fast, which have enabled real-time operation.

Step IV: Permutation and Update: After the optimal solution $\boldsymbol{\phi}^*$ has been found, we permute the state vectors according to that solution. The permutation does not have to be done explicitly, since we are only interested in the Gaussian approximation of the permuted posterior density. To express the mean and covariance after permutation, we describe the calculation in the joint state vector $\mathbf{X}_k$. Before permutation, the expected value and covariance matrix of that vector is

$$\mathbf{X}_{k\,|\,k}^{h} = \left[\left(\mathbf{x}_{k\,|\,k}^{1,h} \right)^T \left(\mathbf{x}_{k\,|\,k}^{2,h} \right)^T \dots \left(\mathbf{x}_{k\,|\,k}^{n,h} \right)^T \right]^T \tag{61}$$

$$\mathbf{P}_{k\,|\,k}^{h} = \operatorname{diag} \left\{ \mathbf{P}_{k\,|\,k}^{1,h}, \mathbf{P}_{k\,|\,k}^{2,h}, \dots, \mathbf{P}_{k\,|\,k}^{n,h} \right\}. \tag{62}$$

After permutation, the expected value and covariance matrix are given by

$$\mathbf{X}_{k\,|\,k} = \mathbf{V} \boldsymbol{\phi}^* \tag{63}$$

$$\begin{aligned} \mathbf{P}_{k\,|\,k} = \sum_{h=1}^{N_h} \sum_{j=1}^{n!} \boldsymbol{\phi}^*(i(h,j)) \Big[& \boldsymbol{\Pi}_j \mathbf{P}_{k\,|\,k}^{h} \boldsymbol{\Pi}_j \\ & + \left(\boldsymbol{\Pi}_j \mathbf{X}_{k\,|\,k}^{h} - \mathbf{X}_{k\,|\,k} \right) \\ & \times \left(\boldsymbol{\Pi}_j \mathbf{X}_{k\,|\,k}^{h} - \mathbf{X}_{k\,|\,k} \right)^T \Big], \end{aligned} \tag{64}$$

where the function $i(h,j)$ gives the index in $\boldsymbol{\phi}^*$ which corresponds to permutation i and hypothesis h. We approximate the targets as independent, which implies that the posterior covariance matrix $\mathbf{P}_k^i$ for target i (the indices are irrelevant) is the corresponding block in the matrix $\mathbf{P}_{k\,|\,k}$.

Step V: Prediction: The final step of the SJPDA algorithm is to let $k \to k-1$ (see Fig. 3), and predict the state and covariance matrices of the set of targets to the next time step k. For linear models, that prediction is done using the ordinary Kalman prediction equations

$$\mathbf{x}_{k\,|\,k-1}^{i} = \mathbf{F}_k \mathbf{x}_{k-1\,|\,k-1}^{i} \tag{65}$$

$$\mathbf{P}_{k\,|\,k-1}^{i} = \mathbf{F}_k \mathbf{P}_{k-1\,|\,k-1}^{i} \mathbf{F}_k^T + \mathbf{Q}_k. \tag{66}$$

In Fig. 4, the densities of the joint state vector (with two targets and scalar states) is illustrated for different steps in the JPDA and SJPDA algorithms. In the example, the predicted density is a Gaussian, given by the top figure. The second row of densities show the density after measurement update and optimal permutations (for SJPDA). From the figures, we see that the SJPDA density can be described much more accurately by a Gaussian density. The last row shows the Gaussian approximations of the JPDA and SJPDA posterior densities. It is clear that the covariance matrix of the SJPDA approximation is smaller.

VII. Evaluations

Two different tracking scenarios are considered for evaluation of the SJPDA algorithm, and comparison with JPDA and a Gaussian-mixture CPHD filter with known target number. The first scenario, for which also the KLSJPDA algorithm is evaluated, is the tracking of two targets, whose trajectories are as illustrated in Fig. 5. The parameters used in the evaluation are

$$\boldsymbol{\phi}_{\text{start},1} = [\, 0.7\beta_1 \quad 0.3\beta_1 \quad \dots \quad 0.7\beta_{N_h} \quad 0.3\beta_{N_h} \,]^T \tag{59}$$

$$\boldsymbol{\phi}_{\text{start},2} = [\, 0.7\beta_1 \quad 0.3\beta_1 \quad 0.3\beta_2 \quad 0.7\beta_2 \quad \dots \quad 0.7\beta_{N_h} \quad 0.3\beta_{N_h} \,]^T \tag{60}$$

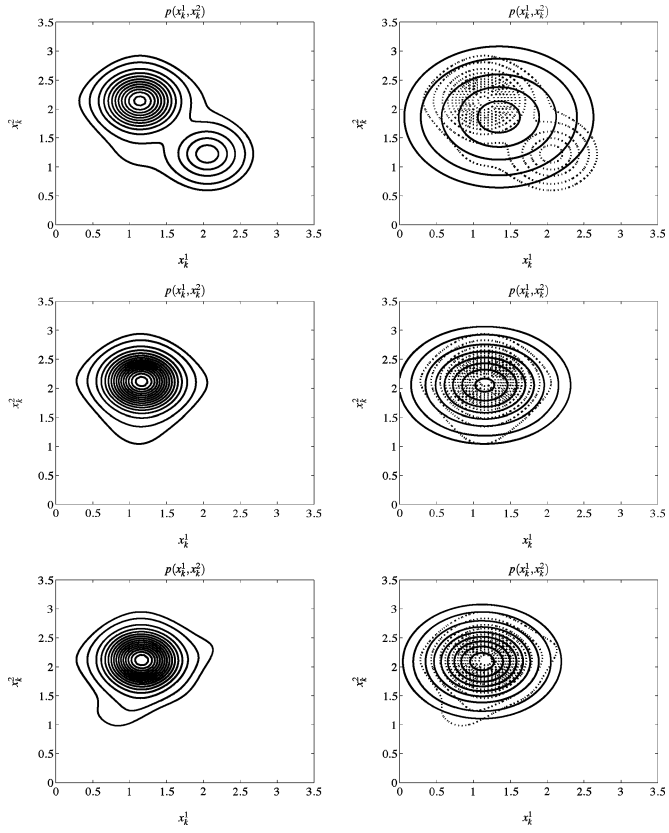

Fig. 4. One-step iteration of the JPDA, SJPDA and KLSJPDA filters. The prior density is a Gaussian, there are two targets, the detection probability is 0.7, and two target-originated measurements are received. Posterior density [top left], posterior density after SJPDA optimization [top middle], posterior density after KLSJPDA optimization [top right], JPDA approximation [bottom left], SJPDA approximation [bottom middle], KLSJPDA approximation [bottom right]. The dashed contours in the bottom-row figures represent the densities that are approximated. Clearly, both the SJPDA and KLSJPDA posterior densities are better approximated by a Gaussian density than the original posterior density.

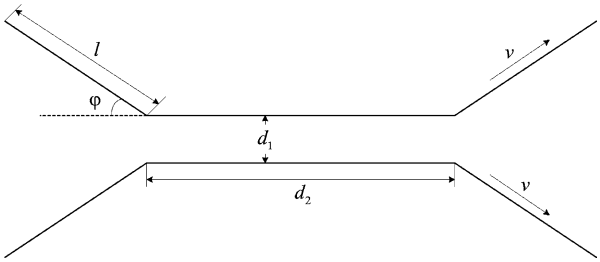

Fig. 5. Illustration of the scenario for the first evaluation.

$l = 10$ m, $d_1 = 0.5$ m, $d_2 = 30$ m, $\varphi = \pi/3$, and $v = 1$ m/s. The second scenario, illustrated hereafter in Fig. 8, is also a two-target example, but in that scenario the two targets move around a hexagonal shape. The aim of that scenario is to evaluate how fast the JPDA and SJPDA algorithms lose track in a challenging scenario. In both scenarios, a sensor collects measurements at even time intervals of $T_0 = 1$ second. Although the simulated trajectories are not typical outcomes of the employed motion model, they are still useful for a performance comparison.

In the filtering algorithms, a nearly constant velocity model is assumed (see (2) for a general description), which is governed by the system matrix

$$\mathbf{F}_{k-1} = \begin{bmatrix} \mathbf{I}_{2\times 2} & T_0\mathbf{I}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{I}_{2\times 2} \end{bmatrix} \tag{67}$$

where $\mathbf{I}_{2\times 2}$ is a 2×2 identity matrix, and $\mathbf{0}_{2\times 2}$ is a 2×2 matrix of zeroes, and by the process noise $\mathbf{v}_{k-1}$ which is zero-mean Gaussian with covariance matrix

$$\mathbf{Q} = q_0 \begin{bmatrix} T_0^3\mathbf{I}_{2\times 2}/3 & T_0^2\mathbf{I}_{2\times 2}/2 \\ T_0^2\mathbf{I}_{2\times 2}/2 & T_0\mathbf{I}_{2\times 2} \end{bmatrix} \tag{68}$$

where q_0 is a tuning parameter. In the simulations, the best tuning parameter is selected for each filter, where the parameter which yields the lowest average MOSPA is used. Further, the measurement model is assumed linear and Gaussian (cf. (4)) with observation matrix

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{69}$$

which means that the sensor delivers position measurements. The Gaussian measurement noise $\mathbf{w}_k$ is zero-mean, with covariance matrix

$$\mathbf{R}_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{70}$$

where we have used $\sigma_x = \sigma_y = 0.2$ m.

For the MOSPA results we use a quadratic distance measure, a cut-off of $c = 1$ m, and $p = 1$. For known cardinality, the impact of the cutoff distance is marginal; it merely puts a threshold on the maximum error. See, e.g., the results for the JPDA filter in Fig. 7.

Fig. 6 shows an example of the JPDA and SJPDA output for $d_1 = 0.5$ m, $\lambda = 0.01$ m^{-2}, $|\mathrm{FoV}| = 1.4\cdot 10^3$ m^2, and $P_D = 1$. In the figure we see the track coalescence tendency of the JPDA filter, which makes it hard for the filter to detect the separation of the tracks, since the two estimates are on top of each other before separation, making the targets indistinguishable after separation. This leads to high MOSPA after the track separation, and it also leads to a high risk of losing tracks. In Fig. 7, the MOSPA performance over 100 Monte Carlo runs is shown for detection probabilities of 1 and 0.85, respectively.

The figures show that the average OSPA performance of the SJPDA filter is better than for the JPDA filter for almost the entire scenario, and the figures also show that the differences between the filters are very large at the time when the tracks separate. This is due to the track coalescence of the JPDA filter. Note that the filters are run with different process noises, where the noise level has been selected to yield the best average MOSPA. When the filters are run with the same process noise, the SJPDA filter always gives an average OSPA which is lower than, or equal to, that of the JPDA filter. For the KLSJPDA algorithm, the performance is equivalent to the SJPDA filter. For the lower detection probability, the estimates are calculated by the SJPDA optimization, whereas the KLSJPDA optimization is used for the Gaussian approximations. As a consequence of the much higher computational complexity of the KLSJPDA filter, and

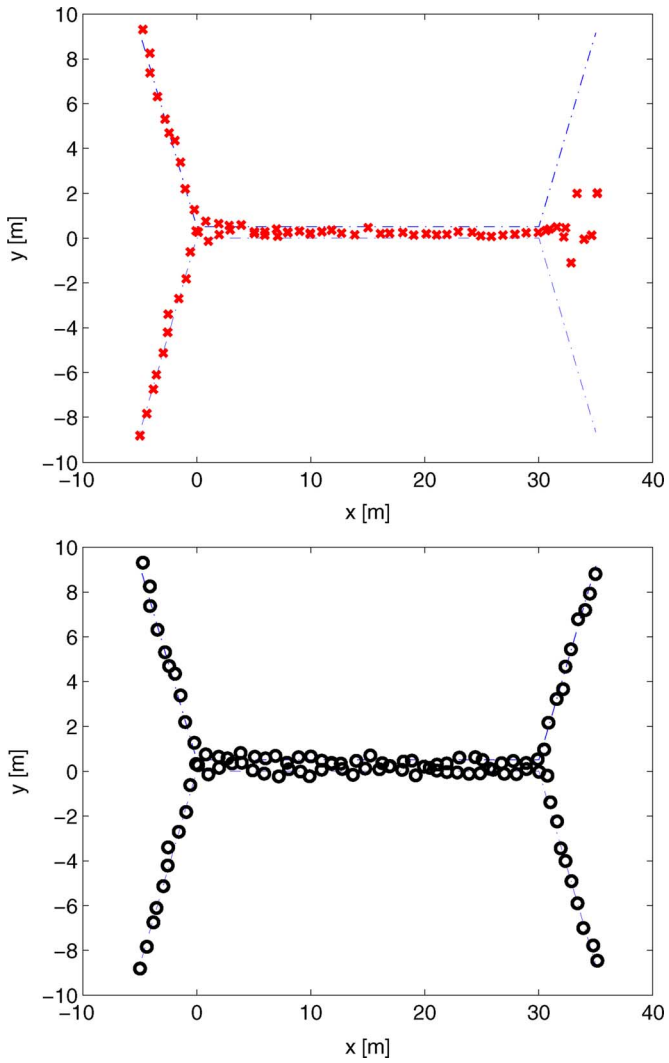

Fig. 6. Output of JPDA [left] and SJPDA [right] for one sequence of measurements.

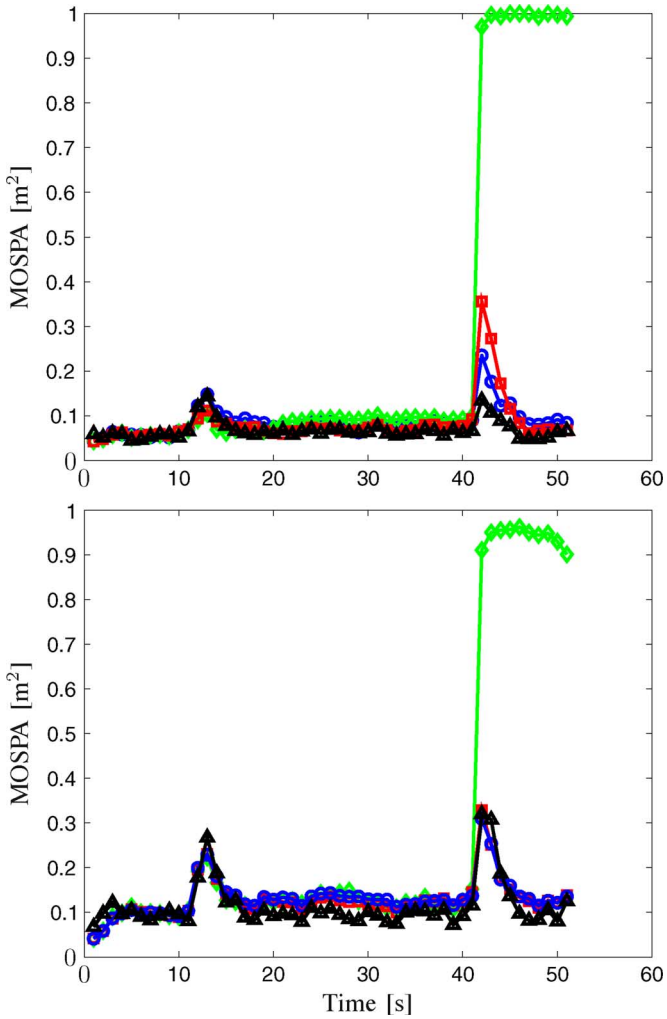

Fig. 7. Evaluation of JPDA (green, diamonds), SJPDA (red, squares), KL-SJPDA with 10 000 samples (blue, circles), and GM-CPHD (black, triangles) on the two-target scenario for detection probabilities $P_d = 1$ [left] and $P_d = 0.85$ [right].

the similarity in performance to the SJPDA filter, we suggest that the SJPDA filter be used.

When compared to GM-CPHD, we see that both the SJPDA and the KLSJPDA filters for large detection probabilities perform on average equally well as GM-CPHD, and almost as well for lower detection probabilities. The results are rather surprising since the GM-CPHD is a more complex algorithm. For the GM-CPHD algorithm, the best process noise parameter q_0 in the aforementioned set is used, and the filter also uses pruning (threshold 0.0001) and merging (threshold $U = 1$, cf. [12]). We note that the somewhat similar JPDA* [24] was compared to the SJPDA in [25]; in fact the JPDA* performs well, but the SJPDA is notably superior.

On the workstation that the simulations were carried out, the average computation time for one iteration of the JPDA, SJPDA[4] and GM-CPHD filters were 3.3 ms, 24.5, and 67.5, respectively for $P_d = 1$, and 3.2, 25.4, and 103 ms, respectively, for $P_d = 0.85$. For this scenario, the SJPDA is hence 7–8 times slower than the JPDA, and the GM-CPHD is 3–4 times slower than the SJPDA. Since the measurement interval is 1 s, real-time operation is possible with all filters for this scenario.

For the second scenario, the average track life and average track loss for JPDA and SJPDA are summarized in Table V, for four different values of detection probability. In the evaluation, the best process noise parameter in the set $q_0 = \{0.05, 0.1, 0.2, 0.3, 0.4\}$ has been used, and otherwise the same parameter values as before. A track is considered lost if the covariance matrix elements corresponding to the uncertainty in the x or y dimensions have surpassed 225 m^2, the estimate of a track is further away than 50 m from the true value, or if there has been only two measurements in the gate of a track for the last five scans. The selected process noise is such that the average track life is maximized. The trajectories that the two targets travel along are illustrated in Fig. 8, where example outputs of the JPDA and SJPDA filters are included. The angle φ (cf. Fig. 5) is $\pi/3$, and the targets travel at a maximum 10 laps around the circuits, which correspond to 1010 s, or almost 17 min. From the table we see that the SJPDA filter has much longer average track life than the JPDA filter,

[4]Note that the results are for three initiation points; a single run with the SJPDA is about one third of the stated values.

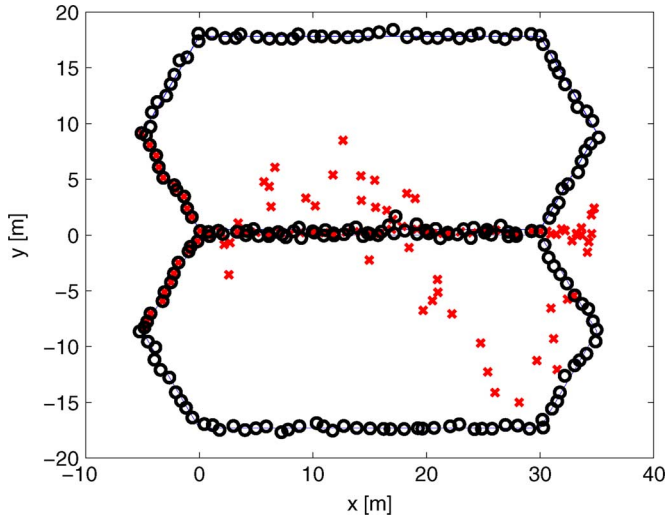

Fig. 8. Example of filter output of the JPDA (crosses) and SJPDA (circles) algorithms for the track life evaluation scenario (for $P_D = 1$). In the plot, the output for the first 101 time steps is presented, which is the time it takes for the two targets to make one loop around the path. In the evaluations, the full trajectories make ten loops around the hexagonal path.

TABLE V
AVERAGE TRACK LIVES AND AVERAGE TRACK LOSS PROBABILITIES FOR THE JPDA AND SJPDA ALGORITHMS, FOR DIFFERENT DETECTION PROBABILITIES, P_D

		JPDA	SJPDA	Improvement SJPDA vs. JPDA
$P_D = 1$	Length	97s	993s	924%
	Loss	100%	3%	97%
$P_D = 0.95$	Length	128s	734s	473%
	Loss	100%	53%	47%
$P_D = 0.9$	Length	152s	456s	200%
	Loss	100%	88%	12%
$P_D = 0.85$	Length	142s	251s	77%
	Loss	100%	100%	0%

due to its avoidance of track coalescence and better description of where there are targets. The lower track-loss probabilities of the SJPDA filter are evident for all four detection probabilities, but the improvement compared to the JPDA filter is better the higher the detection probability.

VIII. CONCLUSION

In this article, we have shown how a traditional tracking algorithm can be improved when target identity is not of interest. The presented approach uses the relation between the density of ordered (labeled) targets, and the density of unordered (the set of) targets. More specifically, there is an infinite number of ordered densities which correspond to the same unordered density, and by switching between those densities we can obtain densities which have better characteristics; most notably that they are more accurately approximated by simpler models such as are often at the core of a working tracking algorithm. In order to reflect the fact that target identity is irrelevant, we use the mean optimal subpattern assignment (MOSPA) metric instead of the mean squared error (MSE).

In the first part of the article, we discuss how the new problem of minimizing MOSPA differs from the classic tracking problem, and show that there is substantial room for improvements of the traditional algorithms when evaluated according to MOSPA.

In the second part of the article we study the joint probabilistic data association (JPDA) filter and how it can be adjusted to the considered problem. First, we describe how optimal Gaussian approximations in the Kullback–Leibler sense can be found. From this, we develop a new filter called the Kullback–Leibler set JPDA (KLSJPDA) filter. Since the filter is computationally intensive, we propose another approach that operates on a goal function that leads to both good Gaussian approximations and low MOSPA. The minimization of that goal function can be formulated as a continuous optimization problem. From this, we develop another version of the JPDA filter, called (with less adornment) the set JPDA (SJPDA) filter.

In the final part of the paper, the SJPDA and KLSJPDA filters are evaluated on two simulation examples and compared to the JPDA and the Gaussian-mixture cardinalized probability hypothesis density (GM-CPHD) filters. The results show that the SJPDA and KLSJPDA filters have similar performance and that they perform substantially better than the JPDA in terms of MOSPA. Hence, because of its lower complexity, we prefer the SJPDA filter. The results also show that the SJPDA filter has a much longer average track length than JPDA for the considered scenario. Further, it is seen that the SJPDA filter performs almost as well as the more complex GM-CPHD filter, and that it even performs slightly better for a unity detection probability.

REFERENCES

[1] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, Sep. 1975.
[2] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, no. 3, pp. 173–183, Jul 1983.
[3] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.
[4] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A generalized s-d assignment algorithm for multisensor multitarget state estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 2, pp. 523–538, Apr. 1997.
[5] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.
[6] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.
[7] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.
[8] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2007.
[9] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
[10] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.
[11] R. Mahler, "PHD filters of higher order in target number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, Oct. 2007.
[12] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3553–3567, Jul. 2007.
[13] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting two birds with two bullets: How to find minimum mean OSPA estimates," in *Proc. 13th Int. Conf. Inf. Fusion*, 2010.
[14] E. Blasch and H. Lang, "Data association through fusion of target track and identification sets," in *Proc. 3rd Int. Conf. Inf. Fusion*, Jul. 2000.

[15] O. Drummond and B. Fridling, "Ambiguities in evaluating performance of multiple target tracking algorithms," in *Proc. SPIE Signal and Data Processing of Small Targets*, 1992, vol. 1698, pp. 326–337.
[16] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans*, vol. 34, no. 3, pp. 327–336, May 2004.
[17] C. Givens and R. Shortt, "A class of Wasserstein metrics for probability distributions," *Michigan Math. J.*, vol. 31, pp. 231–240, 1884.
[18] A. Szulga, "On the Wasserstein metric," in *Proc. 8th Prague Conf. Inf. Theory, Stat. Dec. Functions, Random Processes*, 1978, pp. 267–273.
[19] C. Kreucher, K. Kastella, and A. Hero, "Multitarget tracking using the joint multitarget probability density," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, pp. 1396–1414, 2005.
[20] S. Kullback, *Information Theory and Statistics*. New York, NY: Wiley, 1959.
[21] A. Runnalls, "Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 989–999, Jul. 2007.
[22] L. Svensson, D. Svensson, and M. Guerriero, Set JPDA Filter for Multi-Target Tracking Department of Signals and Systems, Chalmers University of Technology, Tech. Rep. R012/2010, 2010.
[23] L. Svensson, D. Svensson, and P. Willett, "Set JPDA algorithm for tracking unordered sets of targets," in *Proc. 12th Int. Conf. Inf. Fusion*, 2009.
[24] H. Blom and E. Bloem, "Probabilistic data association avoiding track coalescence," *IEEE Trans. Autom. Control*, vol. 45, no. 2, pp. 247–259, Feb. 2000.
[25] D. Crouse, Y. Bar-Shalom, P. Willett, and L. Svensson, "The JPDAF in practical systems: Computation and snake oil," in *Proc. SPIE Signal and Data Processing of Small Targets*, 2010.

Lennart Svensson (SM'10) 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 an Associate Professor at the Signal Processing group, again at Chalmers University of Technology. His research interests include Bayesian inference in general, and nonlinear filtering and tracking, in particular.

Daniel Svensson was born in Borås, Sweden, in 1979. He received the M.Sc. degree in electrical engineering in 2004 and the Ph.D. degree in 2010, both from Chalmers University of Technology, Göteborg, Sweden.

Between 2004 and 2005, he was a Systems Engineer at Ericsson Microwave Systems, Göteborg, Sweden. Since November 2010, he has held a Postdoctoral position at Chalmers University of Technology, and since April 2011 he has been a Systems Engineer at Electronic Defence Systems, Saab AB, Göteborg, Sweden. His research interests include nonlinear filtering and target tracking.

Marco Guerriero was born in Salerno, Italy, on June 18th, 1981. He received the B.A.Sc. and M.Sc. degrees in electrical engineering from the University of Salerno, Italy in 2002 and 2005, respectively, and received the Ph.D. degree from the University of Connecticut, Storrs, in 2009.

In fall 2009, he was a Visiting Scientist at the NATO Undersea Research Centre (NURC) La Spezia, Italy. He was a System Analyst at the Research and Advanced System Design Department at ELETTRONICA S.p.A., Rome, Italy. His research interests lie in the areas of signal processing, with particular focus on distributed detection and estimation in sensor networks, target tracking and data fusion and radar signal processing, and electronic warfare. Currently, he is a System Engineer at Far-Systems S.p.A.

Peter Willett (S'82–M'86–SM'97–F'03) received the B.A.Sc. degree in engineering science from the University of Toronto, Toronto, ON, Canada, in 1982 and the Ph.D. degree from Princeton University, Princeton, NJ, in 1986.

Since then, he has been a faculty member at the University of Connecticut, Storrs, and since 1998, he has been a Professor. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion and tracking. He also has interests in (and has published in the areas of) change/abnormality detection, optical pattern recognition, communications and industrial/security condition monitoring.

Dr. Willett is Editor-in-Chief for IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, and until recently was an Associate Editor for three active journals IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS (for Data Fusion and Target Tracking) and IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PARTS A AND B. He is also an Associate Editor for the *IEEE Aerospace and Electronic Systems (AES) Magazine*, Associate Editor for ISIFs electronic *Journal of Advances in Information Fusion*, is a member of the editorial board of *IEEE Signal Processing Magazine*, and was first editor of the IEEE AES Magazine's periodic Tutorial issues. He was a member of the IEEE AESS Board of Governors 2003–2009. He was General Co-Chair (with Stefano Coraluppi) for the 2006 ISIF/IEEE Fusion Conference in Florence, Italy, and for the 2008 ISIF/IEEE Fusion Conference in Cologne, Germany, Program Co-Chair (with Eugene Santos) for the 2003 IEEE Conference on Systems, Man & Cybernetics, Washington, DC, and Program Co-Chair (with Pramod Varshney) for the 1999 Fusion Conference, Sunnyvale.