Using Symmetric State Transformations for Multi-Target Tracking

Marcus Baum and Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS),
Institute for Anthropomatics,
Karlsruhe Institute of Technology (KIT), Germany.
Email: marcus.baum@kit.edu, uwe.hanebeck@ieee.org

***Abstract*—This paper is about the use of symmetric state transformations for multi-target tracking. First, a novel method for obtaining point estimates for multi-target states is proposed. The basic idea is to apply a symmetric state transformation to the original state in order to compute a minimum mean-square-error (MMSE) estimate in a transformed state. By this means, the known shortcomings of MMSE estimates for multi-target states can be avoided. Second, a new multi-target tracking method based on state transformations is suggested, which entirely performs the time and measurement update in a transformed space and thus, avoids the explicit calculation of data association hypotheses and removes the target identity from the estimation problem. The performance of the new approach is evaluated by means of tracking two crossing targets.**

Keywords: Multi-target tracking, data association, symmetric functions, state transformations, point estimates.

I. Introduction

Tracking multiple targets based on noisy measurements is a fundamental problem arising in many applications [1]–[3]. For instance, in air surveillance, multiple aircraft are to be tracked based on radar devices. A major challenge in multi-target tracking is that the number of feasible association hypotheses grows exponentially with the number of targets. As a consequence, elaborate approximation and reduction techniques are required in order to deal with the complexity of the problem.

A huge variety of multi-target tracking algorithms has been proposed [1]–[3]. For example, a well-known method for data association is the so-called nearest neighbor filter [4] that assigns each observation to the most probable target. A multiple-hypothesis tracker (MHT) [5] manages all feasible hypothesis over time and employs reduction techniques in order to deal with the complexity. The joint probabilistic data association filter (JPDAF) [3] performs a weighted update of all single target estimates according to association probabilities, where the resulting Gaussian mixture is approximated by a single Gaussian density. There exist several JPDAF modifications tailored to closely-spaced targets. For instance, the Coupled JPDAF [3] estimates the complete joint state of closely-spaced targets. JPDAF* [6] is based on hypothesis pruning and Set JPDAF [7], [8] is an adjusted JPDAF algorithm for tracking an unordered set of targets. During recent years, Monte Carlo methods [9] for approximating the exact Bayesian solution have been introduced. Even though a particle filter in general allows for approximating the true probability density arbitrary precise, one has to deal with the so-called mixed labeling problem, which is discussed in [10]–[12]. A further multi-target tracking method is the probabilistic multiple-hypothesis tracker (PMHT), which uses the expectation maximization (EM) algorithm in order to compute a maximum a posteriori (MAP) estimate [13].

Multi-target tracking algorithms that do not explicitly evaluate association hypotheses perform so-called implicit data association. The well-known PHD filter [14]–[16] employs a first-order approximation of the optimal Bayesian multi-target filter based on finite set statistics (FISST). The symmetric measurement equation (SME) filter [17]–[20] uses symmetric transformations in order to remove the data association uncertainty from the measurement equation.

Contributions: This paper investigates the use of symmetric transformations for multi-target tracking. A symmetric transformation can be used for removing the target identity from the estimation problem and thus, avoids several problems resulting from ordered densities for multi-target states.

First, symmetric transformations are used in this work for obtaining a suitable point estimate from an ordered density obtained from an arbitrary multi-target tracking algorithm. The basic idea is to use the minimum mean-square-error (MMSE) estimate in a transformed space in order to calculate optimal (labeled or unlabeled) point estimates for multi-targets.

Second, a new concept for tracking closely-spaced targets is proposed. Now, the basic idea is to employ a symmetric transformation in order to perform the filter and update step completely in the transformed space [21]. The original state space is not considered anymore. In doing so, the data association uncertainty and target labels are removed such that the complexity of the problem is reduced. The performance of the new filter is demonstrated by means of a scenario with two targets moving according to a (nearly) constant velocity model.

Overview: In Section II, we introduce the concept of symmetric transformations for ordered multi-target states and we show how point estimates in the transformed space are obtained. Subsequently, in Section III the idea of the so-called Unique State filter (USF) [21] is explained, which performs inference in a transformed state. In Section IV, an example of tracking two targets with a constant velocity model is described. Finally, the results are evaluated in Section V and the paper is concluded in Section VI.

II. MMSE Estimates Using State Transformations

Typical Bayesian multi-target tracking algorithms such as JPDAF [3], MHT [5], or particle filters methods [9] represent the state as an ordered random vector[1]

$$\underline{\boldsymbol{x}} = \left[\underline{\boldsymbol{x}}_1, \ldots, \underline{\boldsymbol{x}}_N\right]^T \ ,$$

where $\underline{\boldsymbol{x}}_i \in \mathbb{R}^n$ denotes the state vector of target i (with $1 \leq i \leq N$). The tracking algorithm provides a probability density for $\underline{\boldsymbol{x}}$ conditioned on the received data $\mathcal{Y}$

$$p\left(\underline{x}|\mathcal{Y}\right) \ . \tag{1}$$

Usually, only an approximation of the true density is available. However, it is theoretically possible to approximate the true density arbitrary well.

In almost every application, it is required to compute a point estimate based on the posteriori density (1). A point estimate is a single deterministic value summarizing the information of the posterior density (1). A common choice for a point estimate is the well-known minimum mean-squared-error (MMSE) estimate [2]

$$\underline{\hat{x}}^{MMSE} := \arg\min_{\underline{\hat{x}}} \mathrm{E}\left\{||\underline{\hat{x}} - \underline{\boldsymbol{x}}||^2|\mathcal{Y}\right\} \ ,$$

which is given by the conditional mean of $\underline{\boldsymbol{x}}$, i.e.,

$$\underline{\hat{x}}^{MMSE} = \mathrm{E}\left\{\underline{\boldsymbol{x}}|\mathcal{Y}\right\} \ .$$

However, MMSE estimates may be improper for multi-target tracking as pointed in [11], [12]. In particular, for multi-modal probability densities, the MMSE may not give a reasonable estimate as demonstrated in the following example.

Example 1 (MMSE in State Space)**.** Consider two one-dimensional targets with state $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Then, the multi-target state vector is $\underline{\boldsymbol{x}} := \left[\boldsymbol{x}_1, \boldsymbol{x}_2\right]^T$. If the joint density of the multi-target state is the following Gaussian mixture

$$p\left(\underline{x}|\mathcal{Y}\right) = \tfrac{1}{2}\mathcal{N}(\underline{x} - \underline{\mu}_1, \Sigma_1) + \tfrac{1}{2}\mathcal{N}(\underline{x} - \underline{\mu}_2, \Sigma_2)$$

with $\underline{\mu}_1 = \left[1, 2\right]^T$, $\underline{\mu}_2 = \left[2, 1\right]^T$, $\Sigma_1 = 0.1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and $\Sigma_2 = 0.1 \cdot \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix}$, the MMSE estimate becomes $\underline{\hat{x}}^{MMSE} = \mathrm{E}\left\{\underline{\boldsymbol{x}}|\mathcal{Y}\right\} = \left[1.5, 1.5\right]^T$. Actually, this result is quite undesirable, as the two targets are collapsed into one single position between the two modes.

The misbehavior of the MMSE is a problem of representing multi-target states as ordered densities, rather than a problem of the MMSE [10]–[12] itself. Several solutions to this problem have been suggested in literature. In [10]–[12], the maximum a posteriori (MAP) estimate instead of the MMSE estimate is used. In [7], [8], the minimum optimal subpattern assignment (MOSPA) metric [22] for obtaining estimates of unlabeled targets is employed.

In this work, we pursue the following idea: The MMSE is unsuitable due to the representation of a multi-target state as an ordered random vector, so let us change this representation in order to obtain meaningful MMSE estimates.

More formally: Let $\underline{\boldsymbol{z}}$ be the random variable that results from applying a suitable (nonlinear) transformation $T(\cdot)$ to $\underline{\boldsymbol{x}}$, i.e.,

$$\underline{\boldsymbol{z}} = T(\underline{\boldsymbol{x}}) \ .$$

The MMSE estimate in the transformed space is given by

$$\underline{\hat{z}}^{MMSE} = \mathrm{E}\left\{\underline{\boldsymbol{z}}|\mathcal{Y}\right\} \ .$$

A point estimate $\underline{\hat{x}}^{Transf}$ for $\underline{\boldsymbol{x}}$ is then given by each $\underline{\hat{x}}^{Transf} \in T^{-1}\left(\underline{\hat{z}}^{MMSE}\right)$. This point estimate is known to minimize the MSE in the transformed space, i.e.,

$$\underline{\hat{x}}^{Transf} := \arg\min_{\underline{\hat{x}}} \mathrm{E}\left\{||T(\underline{\hat{x}}) - T(\underline{\boldsymbol{x}})||^2|\mathcal{Y}\right\} \ .$$

This can also be interpreted as a special type of a Bayesian risk function [23].

Obviously, the transformation $T(\cdot)$ has to be chosen carefully, as it reflects the properties for which an MMSE estimate is determined. A natural choice for $T(\cdot)$ is a symmetric function that is injective up to permutation.

Definition 1 (Symmetric Transformation)**.** $T(\cdot)$ is said to be symmetric iff the order of its arguments can be changed without affecting the result, i.e., $T(\underline{\boldsymbol{x}}_1, \ldots, \underline{\boldsymbol{x}}_N) = T(\pi_i(\underline{\boldsymbol{x}}_1, \ldots, \underline{\boldsymbol{x}}_N))$ for all permutations $\pi \in \Pi_N$ of the state vectors $\underline{\boldsymbol{x}}_1, \ldots, \underline{\boldsymbol{x}}_N$, where Π_N denotes the set of all permutations for $\left[\underline{\boldsymbol{x}}_1, \ldots, \underline{\boldsymbol{x}}_N\right]^T$.

Definition 2 (Injective Symmetric Transformation)**.** A symmetric transformation $T(\cdot)$ is called injective iff it follows from

$$T(\underline{\boldsymbol{x}}_1, \ldots, \underline{\boldsymbol{x}}_N) = T(\underline{\boldsymbol{x}}_1^*, \ldots, \underline{\boldsymbol{x}}_N^*)$$

that $\left[\underline{\boldsymbol{x}}_1, \ldots, \underline{\boldsymbol{x}}_N\right]^T = \pi(\underline{\boldsymbol{x}}_1^*, \ldots, \underline{\boldsymbol{x}}_N^*)$ for a permutation $\pi \in \Pi_N$.

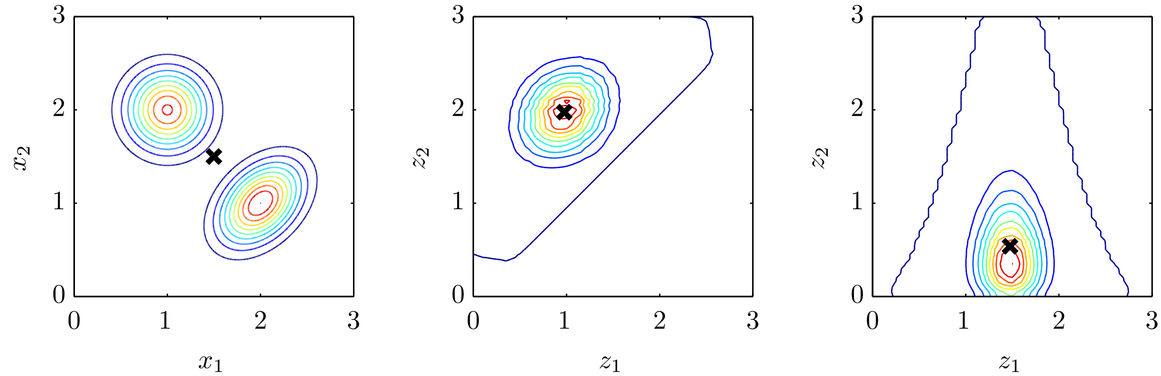
Remark 1. Symmetric functions are also used in the SME filter [17]–[20] to remove data association uncertainty from the measurement equation. Furthermore, the probability generating functional is also a symmetric function [14]–[16], and elementary symmetric polynomials occur in the CPHD filter equations [24]. Point process theory leads to symmetric measures [25] and in [26] symmetric probability densities are used.

A symmetric transformation removes the target identity from the state and hence, the transformed state does not suffer from the mixed labeling problem [10]–[12]. Intuitive examples for suitable symmetric transformations are presented in the following example.

Example 2 (Transformations)**.** This example shows two suitable transformations for the two one-dimensional targets of Example 1. The first transformation $T_1(\cdot)$ [21] is based on extreme values, it simply orders the target positions, i.e.,

$$T_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = \begin{bmatrix} \min\{\boldsymbol{x}_1, \boldsymbol{x}_2\} \\ \max\{\boldsymbol{x}_1, \boldsymbol{x}_2\} \end{bmatrix} \ . \tag{2}$$

1 The time index is omitted here for the sake of simplicity.



(a) Pdf for two targets in original state space. (b) Pdf in the transformed space using $T_1(\cdot)$. (c) Pdf in the transformed space using $T_2(\cdot)$.

Figure 1: Examples of symmetric state transformations.

The point estimate of the transformed state for this example becomes $\hat{\underline{z}}^{MMSE} = \mathrm{E}\{\underline{\boldsymbol{z}}|\mathcal{Y}\} = \left[1,2\right]^T$. The point estimate $\hat{\underline{x}}^{Transf}$ for $\underline{\boldsymbol{x}}$ is then given $\hat{\underline{x}}^{Transf} \in T_1^{-1}(\hat{\underline{z}}^{MMSE}) = \{\left[1,2\right]^T, \left[2,1\right]^T\}$.

The second transformation uses symmetric polynomials, i.e.,

$$T_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = \begin{bmatrix} \frac{\boldsymbol{x}_1+\boldsymbol{x}_2}{2} \\ (\boldsymbol{x}_1 - \frac{\boldsymbol{x}_1+\boldsymbol{x}_2}{2})^2 + (\boldsymbol{x}_2 - \frac{\boldsymbol{x}_1+\boldsymbol{x}_2}{2})^2 \end{bmatrix} \quad (3)$$

maps the target position to the mean of the two targets and their squared distance. The point estimate of the transformed state for this example becomes $\hat{\underline{z}}^{MMSE} = \mathrm{E}\{\underline{\boldsymbol{z}}|\mathcal{Y}\} \approx \left[1.5, 0.7\right]^T$. The point estimate $\hat{\underline{x}}^{Transf}$ for $\underline{\boldsymbol{x}}$ is then given $\hat{\underline{x}}^{Transf} \in T_2^{-1}(\hat{\underline{z}}^{MMSE}) = \{\left[0.7, 2.3\right]^T, \left[2.3, 0.7\right]^T\}$. Note that the point estimates are different for different transformations as they are MMSE estimates according to different representations, i.e., risk functions. Furthermore, if the relativ distance between the targets increases, the MMSE estimate in the transformed space approaches the (unlabeled) MMSE estimate in the original space.

The above example considered one-dimensional target states. In general, multivariate extensions can be obtained in a similar manner. For instance, a natural extension of the polynomial transformation leads to a transformation that computes (normalized/central) sample moments. Sample moments have an intuitive meaning as they represent the shape of a point set.

Example 3 (Multi-dimensional States)**.** A simple transformation for two two-dimensional targets $\underline{\boldsymbol{x}}_1 = \left[\boldsymbol{x}_1, \boldsymbol{y}_1\right]^T$ and $\underline{\boldsymbol{x}}_2 = \left[\boldsymbol{x}_2, \boldsymbol{y}_2\right]$ is given by

$$T_4(\underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2) = \begin{bmatrix} \boldsymbol{x}_1 + \boldsymbol{x}_2 \\ \boldsymbol{x}_1^2 + \boldsymbol{x}_2^2 \\ \boldsymbol{y}_1 + \boldsymbol{y}_2 \\ \boldsymbol{y}_1^2 + \boldsymbol{y}_2^2 \\ \boldsymbol{x}_1\boldsymbol{y}_1 + \boldsymbol{x}_2\boldsymbol{y}_2 \end{bmatrix} \quad .$$

A generalization of transformations based on extreme values (2) has been given in [21] and higher target numbers lead to order statistics [27]. At this point it is important to note that the support of the density (1) may be restricted to a subspace in the transformed space. For instance, the quadratic transformation (3) results in a density with $p\left(\underline{z}_1, \underline{z}_2|\mathcal{Y}\right) = 0$ if $\underline{z}_2 < 0$.

The composition of a metric and a transformation $T(\cdot)$ is a metric on the set of all $S \subset \mathbb{R}^n$ with N elements iff $T(\cdot)$ is a symmetric injective transformation.

Theorem 1. *For $X, Y \in S$, where $X = \{\underline{x}_1, \ldots, \underline{x}_N\}$ and $Y = \{\underline{y}_1, \ldots, \underline{y}_N\}$, a metric on S is given by*

$$d(X, Y) := ||T(\underline{x}) - T(\underline{y})||_2$$

with $\underline{x} = [\underline{x}_1, \ldots, \underline{x}_N]^T$, $\underline{y} = [\underline{y}_{k,1}, \ldots, \underline{y}_N]^T$ and Euclidean norm $||\cdot||_2$ if $T(\cdot)$ is injective.

PROOF. Can be proven by checking the conditions for a metric. □

Symmetric transformations remove the target labeling from the multi-target state. Nevertheless, target labels can be incorporated in the transformation as demonstrated by the following examples.

Example 4 (Incorporating Labels)**.** The transformation (3) can be equipped with target labels as follows

$$T_3(\boldsymbol{x}_1, \boldsymbol{x}_2) = \begin{bmatrix} \min\{\boldsymbol{x}_1, \boldsymbol{x}_2\} \\ \max\{\boldsymbol{x}_1, \boldsymbol{x}_2\} \\ \boldsymbol{x}_1 < \boldsymbol{x}_2 \end{bmatrix} \quad ,$$

where $\boldsymbol{x}_1 < \boldsymbol{x}_2$ is a binary random variable which is $true$ if $\boldsymbol{x}_1 < \boldsymbol{x}_2$. By this means, the target labels are separated from the target position.

There are several interesting connections between the probability density (1) in the transformed state and a random finite set (RFS) used in finite set statistics (FISST) [14]–[16]. According to [7], [8], [14]–[16], the relationship between an ordered density for N targets and an RFS is given by

$$p\left(\{\underline{x}_1, \ldots, \underline{x}_N\}\right) = \frac{1}{N!} \sum_{i=1}^{N!} p\left(\pi_i(\underline{x}_1, \ldots, \underline{x}_N)\right) \quad .$$

In case two different ordered densities specify the same RFS, we say that they belong to the same RFS family [7], [8], [14]–[16].

Theorem 2. *In case of symmetric injective $T(\cdot)$, the probability density $p(\underline{z}|\mathcal{Y})$ is the same for all ordered densities of the same RFS family.*

PROOF. For given $\underline{z}$, let $\underline{x}^*$ be an arbitrary element of $T^{-1}(\underline{z})$, then

$$\begin{aligned} p(\underline{z}|\mathcal{Y}) &= \int p(\underline{z}|\underline{x}) \cdot p(\underline{x}|\mathcal{Y})\, d\underline{x} \\ &= \int \delta(T(\underline{x}) - \underline{z}) \cdot p(\underline{x}|\mathcal{Y})\, d\underline{x} \\ &= \tfrac{1}{N!} \sum_{\pi \in \Pi_N} p(\pi(\underline{x}^*)|\mathcal{Y})\, d\underline{x} \\ &= p(\{\underline{x}_1^*, \dots, \underline{x}_N^*\}|\mathcal{Y}) \ , \end{aligned}$$

where the last term is equal for each ordered density $p(\underline{x}|\mathcal{Y})$. □

Remark 2. In a similar way, it can be shown that the above theorem also holds for non-injective transformations.

A consequence of the above theorem is that the switching algorithm used in Set JPDAF [7] can also be used in case the point estimates are obtained by symmetric state transformations.

A further major insight is that the probability density (1) specifies an RFS in case the transformation is symmetric and injective.

Theorem 3. *For the ordered density $p(\underline{x}_1, \dots, \underline{x}_N)$, the following relationship between the corresponding RFS $p(\{\underline{x}_1, \dots, \underline{x}_N\})$ and $p(\underline{z}|\mathcal{Y})$ holds:*

$$p(\{\underline{x}_1, \dots, \underline{x}_N\}) = p(T(\underline{x}_1, \dots, \underline{x}_N)|\mathcal{Y}) \ .$$

PROOF. For given $\underline{x}_1^*, \dots, \underline{x}_N^*$, the following holds

$$\begin{aligned} p(T(\underline{x}_1^*, \dots, \underline{x}_N^*)|\mathcal{Y}) &= \tfrac{1}{N!} \sum_{\pi \in \Pi_N} p(\pi(\underline{x}_1^*, \dots, \underline{x}_N^*)|\mathcal{Y}) \\ &= p(\{\underline{x}_1^*, \dots, \underline{x}_N^*\}|\mathcal{Y}) \ . \end{aligned}$$

□

Finally, it should be mentioned that in general the point estimate $\hat{\underline{x}}^{Transf}$ is computationally rather easy to obtain. The probability distribution $p(\underline{z}|\mathcal{Y})$ can be computed with the help of a nonlinear filter [28] as it results from a stochastic forward mapping. Based on the mean $\hat{\underline{z}}^{MMSE}$ of $p(\underline{z}|\mathcal{Y})$, the point estimate $\hat{\underline{x}}^{Transf} \in T^{-1}(\hat{\underline{z}}^{MMSE})$ can either be computed algebraically or numerically.

III. THE UNIQUE STATE FILTER (USF)

In the previous section, symmetric transformations have been used to obtain point estimates based on a probability density for an ordered multi-target state vector. Fig. 1 indicates that the bimodal Gaussian mixture density in the original state space becomes a unimodal density in the transformed space. It seems that a Gaussian representation in the transformed space is more suitable than a Gaussian representation in the original space. Because of this observation, the idea pursued in the following is to perform filtering and prediction completely in the transformed space. The original state is not used anymore. Due to the use of injective symmetric transformations, the target identity is removed from the estimation problem and no explicit data association has to be performed. As we have shown that a probability density in the transformed space specifies an RFS, this corresponds to working directly with an RFS. Due to the nonlinear transformation, we introduce new nonlinearities to the problem. However, the benefit lies in the fact that the data association uncertainty and the target labels are removed from the estimation problem.

The basic idea of this approach has already been introduced in [21]. However, in [21] we are restricted to bulk motion, and no subspace measurements were possible. Furthermore, it is worth mentioning that state transformations are a well-known concept used for various problems. For instance, in nonlinear filterering [29], [30] state transformations are used to render nonlinear system functions linear.

In the following, we explain the basic idea for the so-called *Unique State Filter (USF)* [21], which performs inference entirely in a transformed space. For the sake of simplicity, we restrict our discussion to two targets.

A. Problem Formulation

The state vector of the two targets is denoted with $\underline{\boldsymbol{x}}_{k,1} \in \mathbb{R}^n$ and $\underline{\boldsymbol{x}}_{k,2} \in \mathbb{R}^n$. At each time step k, two position measurements $\hat{\underline{y}}_{k,1}$ and $\hat{\underline{y}}_{k,2}$ corrupted with additive Gaussian noise are received, i.e.,

$$\hat{\underline{y}}_{k,\pi(i)} = \mathbf{H}\underline{\boldsymbol{x}}_{k,i} + \underline{\boldsymbol{v}}_{k,i} \ , \quad (4)$$

where $\pi : \{1,2\} \to \{1,2\}$ is the (unknown) target-to-measurement assignment and $i \in \{1,2\}$. The random variables $\underline{\boldsymbol{v}}_{k,i}$ denote Gaussian measurement noise, both with identical statistics.

The target states are assumed to evolve according to a linear system model

$$\underline{\boldsymbol{x}}_{k+1,i} = \mathbf{A}\underline{\boldsymbol{x}}_{k,i} + \underline{\boldsymbol{w}}_{k,i} \quad (5)$$

with system matrix $\mathbf{A}$ and Gaussian system noise $\underline{\boldsymbol{w}}_{k,i}$.

B. Inference in the Transformed Space

The unique state filter performs the measurement and time update with a transformed state $\underline{\boldsymbol{z}}_k$. For this purpose, a suitable symmetric injective transformation

$$\underline{\boldsymbol{z}}_k := T\left(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\right) = T\left(\underline{\boldsymbol{x}}_{k,2}, \underline{\boldsymbol{x}}_{k,1}\right)$$

with

$$T^{-1}(\underline{\boldsymbol{z}}_k) = \{\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\}$$

is used. Note that in general the dimension of $\underline{\boldsymbol{z}}_k$ can be higher than the dimension of the multi-target state.

A linear system equation in the transformed space is then obtained by performing an approximation (see Fig. 2b)

$$\begin{aligned}\underline{\boldsymbol{z}}_{k+1} &= T\left(\mathbf{A}\underline{\boldsymbol{x}}_{k,1} + \underline{\boldsymbol{w}}_{k,1}, \mathbf{A}\underline{\boldsymbol{x}}_{k,2} + \underline{\boldsymbol{w}}_{k,2}\right) \\ &\approx \overline{\mathbf{A}}T\left(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\right) + \overline{\underline{\boldsymbol{w}}}_k \\ &= \overline{\mathbf{A}}\underline{\boldsymbol{z}}_k + \overline{\underline{\boldsymbol{w}}}_k \end{aligned} \tag{6}$$

with system matrix $\overline{\mathbf{A}}$ and Gaussian system noise $\overline{\underline{\boldsymbol{w}}}_k$ that depends on $\underline{\boldsymbol{w}}_{k,1}$ and $\underline{\boldsymbol{w}}_{k,2}$. Note that the precise procedure for obtaining this approximation highly depends on the used transformation. In the following section, we will give a specific example.

Similar, the measurement equation in the transformed space becomes (see Fig. 2a)

$$\begin{aligned}\overline{\underline{y}}_k := T_y\left(\hat{\underline{y}}_{k,1}, \hat{\underline{y}}_{k,1}\right) &= T_y\left(\mathbf{H}\underline{\boldsymbol{x}}_{k,1} + \underline{\boldsymbol{v}}_{k,1}, \mathbf{H}\underline{\boldsymbol{x}}_{k,2} + \underline{\boldsymbol{v}}_{k,2}\right) \\ &\approx \overline{\mathbf{H}}T\left(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\right) + \overline{\underline{\boldsymbol{v}}}_k \\ &= \overline{\mathbf{H}}\underline{\boldsymbol{z}}_k + \overline{\underline{\boldsymbol{v}}}_k \ , \end{aligned} \tag{7}$$

where $T_y(\cdot)$ consists of the components of $T(\cdot)$ that can be computed based on $\hat{\underline{y}}_{1,k}$ and $\hat{\underline{y}}_{2,k}$ and $\overline{\mathbf{H}}$ is measurement matrix. The term $\overline{\underline{\boldsymbol{v}}}_k$ denotes additive Gaussian noise that depends on $\underline{\boldsymbol{v}}_{k,2}$ and $\underline{\boldsymbol{v}}_{k,1}$.

As the derived system and measurement equations are linear, the Kalman filter equations can be used for performing the time and measurement update. The transformed state is assumed to be Gaussian distributed, i.e., $\underline{\boldsymbol{z}}_k \sim \mathcal{N}(\hat{\underline{z}}_k - \mathbf{C}^z_k,)$. The time update then becomes

$$\begin{aligned}\hat{\underline{z}}_{k|k-1} &= \overline{\mathbf{A}}\hat{\underline{z}}_{k-1} \ , \\ \mathbf{C}^z_{k|k-1} &= \overline{\mathbf{A}}\mathbf{C}^z_{k-1}(\overline{\mathbf{A}})^T + \mathbf{C}^{\overline{w}}_{k-1} \ .\end{aligned}$$

The measurement update can be performed with the following equations

$$\begin{aligned}\hat{\underline{z}}_k &= \hat{\underline{z}}_{k|k-1} + \mathbf{K}_k(\overline{\underline{y}}_k - \overline{\mathbf{H}}\hat{\underline{z}}_{k|k-1}) \ , \\ \mathbf{C}^z_k &= \mathbf{C}^z_{k|k-1} - \mathbf{K}_k\overline{\mathbf{H}}\mathbf{C}^z_{k|k-1}\end{aligned}$$

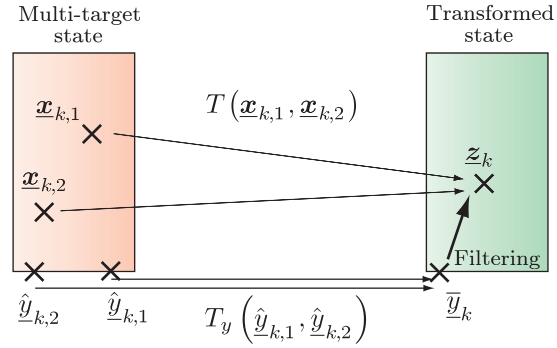
with Kalman gain

$$\mathbf{K}_k = \mathbf{C}^z_{k|k-1}(\overline{\mathbf{H}})^T\left(\mathbf{C}^{\overline{v}}_k + \overline{\mathbf{H}}\mathbf{C}^z_{k|k-1}(\overline{\mathbf{H}})^T\right)^{-1} \ ,$$

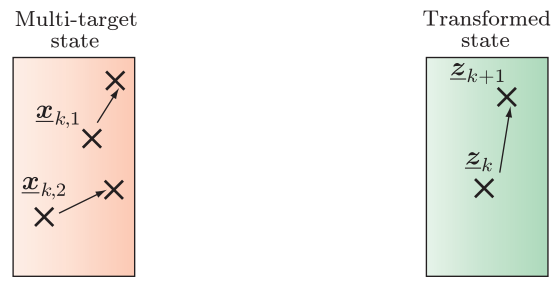
and covariance matrices $\mathbf{C}^{\overline{v}}_k$ and $\mathbf{C}^{\overline{w}}_k$ for the measurement and system noise.

IV. USF Examples: Two Targets / Constant Velocity Model

In this section, we derive particular equations for the time and measurement update for two targets whose state evolves according to a constant velocity model [31]. For this purpose, we employ a symmetric transformation based on symmetric polynomials. Because the derivation of the equations is straightforward, but rather tedious and lengthy, we start with two targets in one-dimensional space for the sake of getting intuition. The equations for two targets in two-dimensional space are then subsequently treated and can be performed in analogy to the one-dimensional case.



(a) Measurement update.



(b) Time update.

Figure 2: Performing inference with a transformed state. In general, the dimension of the transformed state may be higher than the dimension of the original state.

A. One-dimensional Space

In case of two targets in one-dimensional space, the states of the targets are given by $\underline{\boldsymbol{x}}_{k,1} = \left[\boldsymbol{x}_{k,1}, \dot{\boldsymbol{x}}_{k,1}\right]^T$ and $\underline{\boldsymbol{x}}_{k,2} = \left[\boldsymbol{x}_{k,2}, \dot{\boldsymbol{x}}_{k,2}\right]^T$, where $\boldsymbol{x}_{k,i}$ denote the target positions, and $\dot{\boldsymbol{x}}_{k,i}$ the target velocities. We use a constant velocity model with time interval $T = 1$, which leads to the system matrix (5)

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \ .$$

The measurement matrix in (4) is given by $\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$. The symmetric transformation used here is

$$\underline{\boldsymbol{z}}_k := T\left(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\right) = \begin{bmatrix} \boldsymbol{x}_{k,1} + \boldsymbol{x}_{k,2} \\ \boldsymbol{x}^2_{k,1} + \boldsymbol{x}^2_{k,2} \\ \dot{\boldsymbol{x}}_{k,1} + \dot{\boldsymbol{x}}_{k,2} \\ \dot{\boldsymbol{x}}^2_{k,1} + \dot{\boldsymbol{x}}^2_{k,2} \\ \boldsymbol{x}_{k,1}\dot{\boldsymbol{x}}_{k,1} + \boldsymbol{x}_{k,2}\dot{\boldsymbol{x}}_{k,2} \end{bmatrix} \ .$$

As the velocities are not directly measured, the transformation for the measurements is

$$T_y\left(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\right) = \begin{bmatrix} \boldsymbol{x}_{k,1} + \boldsymbol{x}_{k,2} \\ \boldsymbol{x}^2_{k,1} + \boldsymbol{x}^2_{k,2} \end{bmatrix} \ ,$$

which consists of the measured components of $T\left(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\right)$.

Remark 3. For a given point estimate $\hat{\underline{z}}_k$ it is easy to reconstruct point estimates $\{\hat{\underline{x}}_{k,1}, \hat{\underline{x}}_{k,2}\} = T^{-1}(\hat{\underline{z}}_k)$ in the original space. The inverse image $T^{-1}(\hat{\underline{z}}_k)$ can be calculated in closed form, as it only consists of quadratic terms. Apart from the closed form solution, simple numerical optimization algorithms are also suitable in general.

System Equation: The system equation in the transformed space (6) can be derived as follows. First, from

$$T\left(\mathbf{A}\underline{\boldsymbol{x}}_{k,1}, \mathbf{A}\underline{\boldsymbol{x}}_{k,2}\right) = \begin{bmatrix} \boldsymbol{x}_{k,1} + \boldsymbol{x}_{k,2} + \dot{\boldsymbol{x}}_{k,1} + \dot{\boldsymbol{x}}_{k,2} \\ \boldsymbol{x}_{k,1}^2 + \boldsymbol{x}_{k,2}^2 + 2\boldsymbol{x}_{k,1}\dot{\boldsymbol{x}}_{k,1} + \dot{\boldsymbol{x}}_{k,1}^2 + 2\boldsymbol{x}_{k,2}\dot{\boldsymbol{x}}_{k,2} + \dot{\boldsymbol{x}}_{k,2}^2 \\ \dot{\boldsymbol{x}}_{k,1} + \dot{\boldsymbol{x}}_{k,2} \\ \dot{\boldsymbol{x}}_{k,1}^2 + \dot{\boldsymbol{x}}_{k,2}^2 \\ \boldsymbol{x}_{k,1}\dot{\boldsymbol{x}}_{k,1} + \boldsymbol{x}_{k,2}\dot{\boldsymbol{x}}_{k,2} + \dot{\boldsymbol{x}}_{k,1}^2 + \dot{\boldsymbol{x}}_{k,1}^2 \end{bmatrix}$$

it follows that $T\left(\mathbf{A}\underline{\boldsymbol{x}}_{k,1}, \mathbf{A}\underline{\boldsymbol{x}}_{k,2}\right) = \overline{\mathbf{A}}\; T\left(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\right)$ with

$$\overline{\mathbf{A}} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} .$$

Second, the system noise results from

$$T\left(\underline{\boldsymbol{x}}_{k,1} + \underline{\boldsymbol{w}}_{k,1}, \underline{\boldsymbol{x}}_{k,2} + \underline{\boldsymbol{w}}_{k,2}\right) = T\left(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}\right) + \overline{\underline{\boldsymbol{w}}}_k(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}, \underline{\boldsymbol{w}}_{k,1}, \underline{\boldsymbol{w}}_{k,2}) .$$

The noise term $\overline{\underline{\boldsymbol{w}}}_k(\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2}, \boldsymbol{v}_{k,1}, \boldsymbol{v}_{k,2})$ depends on the individual noise $\boldsymbol{w}_{k,1}$ and $\boldsymbol{w}_{k,2}$, but also on the target states $\underline{\boldsymbol{x}}_{k,1}$ and $\underline{\boldsymbol{x}}_{k,2}$. In order to remove this dependency, the current estimates for $\underline{\boldsymbol{x}}_{k,1}$ and $\underline{\boldsymbol{x}}_{k,2}$ can be substituted. Furthermore, the first two moments of $\overline{\underline{\boldsymbol{w}}}_k$ can be calculated analytically in order to obtain additive Gaussian noise [32].

Measurement Equation: Given the measurements $\hat{y}_{k,1}$ and $\hat{y}_{k,2}$, the measurement equation (7) for the transformed space can be derived by simple algebraic reformulations. Because

$$\begin{aligned} T_y\left(\hat{y}_{k,1}, \hat{y}_{k,2}\right) &= T_y\left(\boldsymbol{x}_{k,1} + \boldsymbol{v}_{k,1}, \boldsymbol{x}_{k,2} + \boldsymbol{v}_{k,2}\right) = \\ &\begin{bmatrix} \boldsymbol{x}_{k,1} + \boldsymbol{x}_{k,2} + \boldsymbol{v}_{k,1} + \boldsymbol{v}_{k,2} \\ \boldsymbol{x}_{k,1}^2 + \boldsymbol{x}_{k,2}^2 + 2\boldsymbol{x}_{k,1}\boldsymbol{v}_{k,1} + \boldsymbol{v}_1^2 + 2\boldsymbol{x}_{k,2}\boldsymbol{v}_{k,2} + \boldsymbol{v}_{k,2}^2 \end{bmatrix} \\ &= T_y\left(\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2}\right) + \overline{\underline{\boldsymbol{v}}}(\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2}, \boldsymbol{v}_{k,1}, \boldsymbol{v}_{k,2}) , \end{aligned}$$

the measurement equation (7) is

$$T_y\left(\hat{y}_{k,1}, \hat{y}_{k,2}\right) = \overline{\mathbf{H}}\underline{\boldsymbol{z}}_k + \overline{\underline{\boldsymbol{v}}}_k(\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2}, \boldsymbol{v}_{k,1}, \boldsymbol{v}_{k,2})$$

with

$$\overline{\mathbf{H}} = \left[\underline{e}_1, \underline{e}_2, \underline{e}_5\right]^T ,$$

where $\underline{e}_i$ denotes the i-th unit vector in $\mathbb{R}^5$.

Again, the system noise $\overline{\underline{\boldsymbol{v}}}_k(\underline{\boldsymbol{x}}_{k,1}, \underline{\boldsymbol{x}}_{k,2}, \underline{\boldsymbol{w}}_{k,1}, \underline{\boldsymbol{w}}_{k,2})$ depends on both the individual noise terms $\underline{\boldsymbol{v}}_{k,1}$ and $\underline{\boldsymbol{v}}_{k,2}$, but also on the target states $\underline{\boldsymbol{x}}_{k,1}$ and $\underline{\boldsymbol{x}}_{k,2}$. This dependency can be removed by substituting the current estimates for $\underline{\boldsymbol{x}}_{k,1}$ and $\underline{\boldsymbol{x}}_{k,2}$, and the first two moments of $\overline{\underline{\boldsymbol{v}}}_k$ can be calculated analytically in order to obtain additive Gaussian noise.

Note that the prediction in the transformed space is linear in case of a deterministic system. Furthermore, it can easily be shown that the transformed state is observable by checking the observability condition for linear systems.

B. Two-dimensional Space

In the following, two targets with state vectors $\underline{\boldsymbol{x}}_{k,i} \in \mathbb{R}^4$ with $i \in \{1,2\}$ are given. The state vectors $\underline{\boldsymbol{x}}_{k,i} = \left[\boldsymbol{x}_{k,i}, \boldsymbol{y}_{k,i}, \dot{\boldsymbol{x}}_{k,i}, \dot{\boldsymbol{y}}_{k,i}\right]^T$ consist of the Cartesian positions $\left[\boldsymbol{x}_{k,i}, \boldsymbol{y}_{k,i}\right]^T$ and velocity vectors $\left[\dot{\boldsymbol{x}}_{k,i}, \dot{\boldsymbol{y}}_{k,i}\right]^T$ of the targets. Again, noisy position measurements are available, such that the measurement matrix for a target in (4) is given by

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

and a constant velocity model yields the system matrix (see (5))

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

Transformation: A proper transformation can be constructed in analogy to the one-dimensional case. For this purpose, we need the elementary symmetric functions

$$T_1\left(a, b\right) = \begin{bmatrix} a + b \\ a^2 + b^2 \end{bmatrix}$$

and

$$T_2\left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}\right) = a_1 \cdot a_2 + b_1 \cdot b_2 .$$

Remark 4. We use the following notation:

- $\underline{\boldsymbol{x}}_{k,i}^{(l)}$ denotes the l-th component of the vector $\underline{\boldsymbol{x}}_{k,i}$, e.g., $\underline{\boldsymbol{x}}_{k,i}^{(2)} = \boldsymbol{y}_{k,i}$.
- $\underline{\boldsymbol{x}}_{k,i}^{(l,m)}$ denotes the vector consisting of the l-th and m-th component of $\underline{\boldsymbol{x}}_{k,i}$, e.g., $\underline{\boldsymbol{x}}_{k,i}^{(3,4)} \in \mathbb{R}^2$ are the velocities vectors of the targets.

With this notation, the entire transformation $T(\underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2)$ is given by

$$T(\underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2) = \begin{bmatrix} T_1\left(\underline{\boldsymbol{x}}_1^{(1)}, \underline{\boldsymbol{x}}_2^{(1)}\right) \\ \vdots \\ T_1\left(\underline{\boldsymbol{x}}_1^{(4)}, \underline{\boldsymbol{x}}_2^{(4)}\right) \\ T_2\left(\underline{\boldsymbol{x}}_1^{(1,2)}, \underline{\boldsymbol{x}}_2^{(1,2)}\right) \\ T_2\left(\underline{\boldsymbol{x}}_1^{(1,3)}, \underline{\boldsymbol{x}}_2^{(1,3)}\right) \\ T_2\left(\underline{\boldsymbol{x}}_1^{(2,3)}, \underline{\boldsymbol{x}}_2^{(2,3)}\right) + T_2\left(\underline{\boldsymbol{x}}_1^{(1,4)}, \underline{\boldsymbol{x}}_2^{(1,4)}\right) \\ T_2\left(\underline{\boldsymbol{x}}_1^{(2,4)}, \underline{\boldsymbol{x}}_2^{(2,4)}\right) \\ T_2\left(\underline{\boldsymbol{x}}_1^{(3,4)}, \underline{\boldsymbol{x}}_2^{(3,4)}\right) \end{bmatrix} .$$

1) Measurement and System Equation: The measurement equation can be derived in analogy to the one-dimensional case in Section IV-A. The corresponding measurement matrix is

$$\overline{\mathbf{H}} = \left[\underline{e}_1, \ldots, \underline{e}_4, \underline{e}_9\right]^T ,$$

where $\underline{e}_i$ denotes the i-th unit vector in $\mathbb{R}^{14}$. The probability density of the noise $\underline{\overline{v}}_k$ is also approximated by a Gaussian distribution. For this purpose, $\underline{z}_k$ has to be substituted by its last estimate as $\underline{\overline{v}}_k$ depends on the target state.

The system equation and the system noise can also be derived in analogy to the one-dimensional case.

V. Evaluation of the USF

In this section, the performance of the USF is evaluated. For this purpose, we consider the tracking of two closely-spaced targets in two-dimensional space based on a constant velocity model, which is still a challenging task for many data association algorithms. For instance, the JPDAF is known to suffer from track coalescence, i.e., closely-spaced targets tend to collapse into one single target [6], [7].

The true trajectory of the targets is depicted in Fig. 3a. The target states are assumed to evolve according to a constant velocity model with measurement noise $\mathbf{C}_k^{\overline{v}} = \mathrm{diag}\{0.5, 0.5\}$ and the system noise is

$$\mathbf{C}_k^{\overline{w}} = q_k \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{bmatrix}$$

with $q_k = 0.1$.

The estimation results of the USF are compared to the Coupled JPDAF [3]. Fig. 3a shows the minimum optimal subpattern assignment (MOSPA) [22] of the filtering results. There, it can be seen that USF provides good estimation results, independently on how close the targets are. The Coupled JPDAF, however, shows the well-known track coalescence (see Fig. 3d for an example run) and, hence, yields poor estimation results.

The simulation parameters have been chosen to show that the USF does not suffer from track coalescence. For well-separated targets, the JPDAF may even give slightly better results than the USF, because then JPDAF works similar to the associated Kalman filter (with known association). The USF, however, is still based on nonlinear equations if it uses a polynomial transformation.

VI. Conclusions And Future Work

In this work, we have investigated the use of symmetric transformations for multi-target tracking problems. First, symmetric transformations have been used to obtain optimal point estimates in a transformed space. In doing so, the known problems with MMSE estimates in the original space can be avoided. Furthermore, we have shown that a probability density in the transformed space specifies a random finite set. Based on these findings, a multi-target tracking method has been derived, which completely works in the transformed space. The new method has been evaluated by means of tracking two crossing targets. Simulation results show that the estimation quality is independent of the relative distance of the targets.

Point estimates based on state transformation can be applied to every Bayesian multi-target tracking method using ordered densities for representing multi-target states. Future work consists of further evaluations and investigation of suitable transformations for larger target numbers. Additionally, the underlying multi-target tracking algorithm can be adjusted to the state transformation used for obtaining point estimates.

So far, we have only presented a basic version of the USF for tracking two targets without clutter measurements. Future work consists of evaluating the approach for a larger number of targets and the extension to clutter measurement and detection probabilities. A simple method for dealing with clutter is to explicity evaluate all possible hypothesis. i.e., target generated measurement sets.

References

[1] D. Hall and J. Llinas, *Handbook of Multisensor Data Fusion*. CRC Press, May 2001.

[2] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[3] Y. Bar-Shalom, F. Daum, and J. Huang, "The Probabilistic Data Association Filter," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82 –100, Decembre 2009.

[4] X. Rong Li and Y. Bar-Shalom, "Tracking in Clutter With Nearest Neighbor Filters: Analysis and Performance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 3, pp. 995 –1010, July 1996.

[5] S. Blackman, "Multiple Hypothesis Tracking for Multiple Target Tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5 –18, January 2004.

[6] H. Blom and E. Bloem, "Probabilistic Data Association Avoiding Track Coalescence," *IEEE Transactions on Automatic Control*, vol. 45, no. 2, pp. 247 –259, 2000.

[7] L. Svensson, D. Svensson, and P. Willett, "Set JPDA Algorithm for Tracking Unordered Sets of Targets," *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, vol. s. 1187-1194, 2009.

[8] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting Two Birds with Two Bullets: How to Find Minimum Mean OSPA Estimates," *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, 2010.

[9] J. Vermaak, S. Godsill, and P. Perez, "Monte Carlo Filtering for Multi Target Tracking and Data Association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 309 – 332, January 2005.

[10] H. Blom and E. Bloem, "Permutation Invariance in Bayesian Estimation of Two Targets That Maneuver In and Out Formation Flight," in *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, 6-9 2009, pp. 1296 –1303.

[11] H. Blom, E. Bloem, Y. Boers, and H. Driessen, "Tracking Closely Spaced Targets: Bayes Outperformed by an Approximation?" in *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008)*, June 2008, pp. 1 –8.

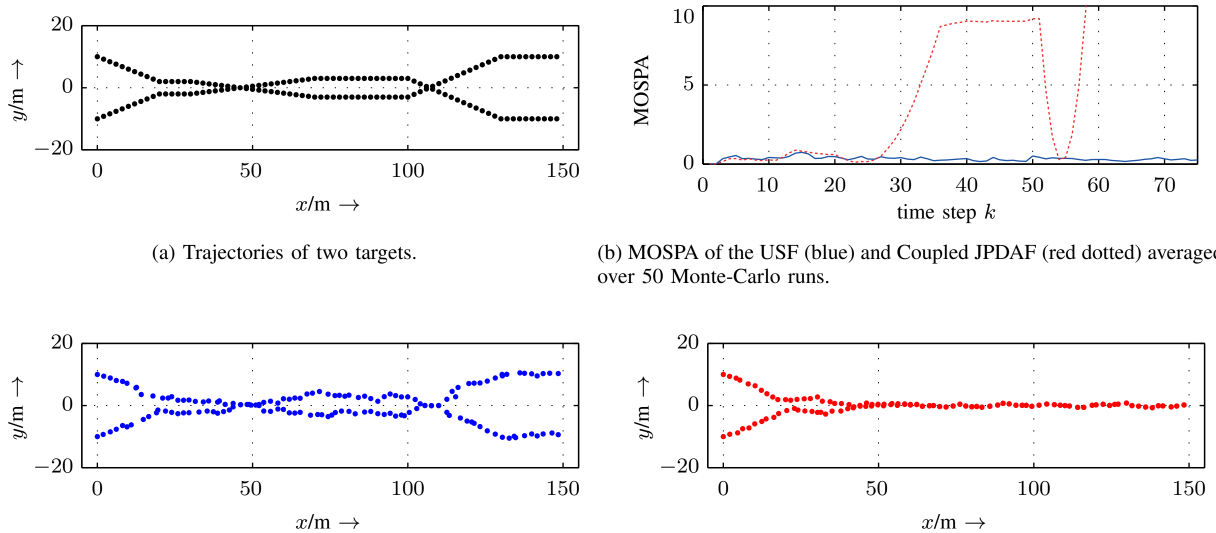
[12] Y. Boers, E. Sviestins, and H. Driessen, "Mixed Labelling in Multitarget Particle Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 792 –802, 2010.

[13] D. Crouse, M. Guerriero, P. Willett, R. Streit, and D. Dunham, "A Look at the PMHT," in *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, July 2009, pp. 332 –339.

[14] R. Mahler, ""Statistics 101" For Multisensor, Multitarget Data fusion," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 53 –64, January 2004.

[15] ——, "Multitarget Bayes Filtering via First-Order Multitarget Moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152 – 1178, Octobre 2003.

[16] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091 –4104, Nov. 2006.



(a) Trajectories of two targets.

(b) MOSPA of the USF (blue) and Coupled JPDAF (red dotted) averaged over 50 Monte-Carlo runs.

(c) USF: Estimated trajectory for an example run.

(d) JPDAF: Estimated trajectory for an example run.

Figure 3: Simulation results for tracking two closely spaced targets with the USF.

[17] E. Kamen, "Multiple Target Tracking based on Symmetric Measurement Equations," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 371 –374, March 1992.

[18] D. J. Muder and S. D. O'Neil, "Multidimensional SME Filter for Multitarget Tracking," *Signal and Data Processing of Small Targets 1993*, vol. 1954, no. 1, pp. 587–599, 1993.

[19] R. L. Bellaire and E. W. Kamen, "New Implementation of the SME Filter Approach to Multiple-Target Tracking," *Signal and Data Processing of Small Targets 1996*, vol. 2759, no. 1, pp. 477–487, 1996.

[20] W. Leven and A. Lanterman, "Unscented Kalman Filters for Multiple Target Tracking With Symmetric Measurement Equations," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 370 –375, February 2009.

[21] M. Baum and U. D. Hanebeck, "Association-free Tracking of Two Closely Spaced Targets," in *Proceedings of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2010)*, Salt Lake City, Utah, Sep. 2010.

[22] 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.

[23] X. Li, "Optimal Bayes Joint Decision and Estimation," in *Proceedings of the 10th International Conference on Information Fusion (Fusion 2007)*, 2007, pp. 1 –8.

[24] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic Implementations of the Cardinalized Probability Hypothesis Density Filters," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3553 –3567, July 2007.

[25] S. S. Singh, B.-N. Vo, A. Baddeley, and S. Zuyev, "Filters for Spatial Point Processes," *SIAM Journal on Control and Optimization*, vol. 48, no. 4, pp. 2275–2295, 2009.

[26] C. Kreucher, K. Kastella, and I. Hero, A.O., "Multitarget Tracking using the Joint Multitarget Probability Density," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1396 – 1414, oct. 2005.

[27] H. David and H. Nagaraja, *Order statistics*. John Wiley & Sons, 2003.

[28] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, 1st ed. Wiley & Sons, August 2006.

[29] T. Lefebvre, H. Bruyninckx, and J. De Schutter, *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*. Springer Berlin / Heidelberg, 2005, ch. The Non-Minimal State Kalman Filter, pp. 77–94.

[30] U. D. Hanebeck, "Optimal Filtering of Nonlinear Systems Based on Pseudo Gaussian Densities," in *Proceedings of the 13th IFAC Symposium on System Identification (SYSID 2003)*, Rotterdam, Netherlands, Aug. 2003, pp. 331–336.

[31] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

[32] M. Baum, B. Noack, F. Beutler, D. Itte, and U. D. Hanebeck, "Optimal Gaussian Filtering for Polynomial Systems Applied to Association-free Multi-Target Tracking," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.