Joint IMM and Coupled PDA to track closely spaced targets and to avoid track coalescence

Henk A.P. Blom and Edwin A. Bloem
National Aerospace Laboratory NLR
Amsterdam, The Netherlands
e-mail: blom@nlr.nl, bloem@nlr.nl

Abstract – *For the problem of tracking closely spaced targets from possibly false and missing observations, the paper studies combinations of IMM and PDA where both the IMM step and the PDA step is performed jointly over all targets. The resulting filter algorithms are referred to as Joint IMM Coupled PDA (JIMMCPDA) and track-coalescence-avoiding Joint IMM Coupled PDA (JIMMCPDA*). Through Monte Carlo simulations these novel algorithms are compared to IMMPDA, IMMJPDA, IMMJPDA* and particle filtering implementation of the exact Bayesian filter equation.*

Keywords: Multitarget tracking, Sudden maneuvers, False measurements, Missing measurements, Particle Filtering.

1 Introduction

In a series of papers [1–4]we have studied the problem of tracking multiple closely spaced maneuvering targets. These studies resulted in six types of results that go beyond the IMMJPDA filter algorithm derivation by [5]:

1. Jump-linear descriptor system embedding of the multi target tracking problem [1, 2]
2. Exact Bayesian filter characterization [2, 3]
3. Development of a track-coalescence-avoiding version of IMMJPDA, i.e. IMMJPDA* [1, 2]
4. Development of a combination of IMM and PDA where both steps are performed jointly over all targets [4]
5. Monte Carlo simulations for 1-D scenarios including comparison of results to Particle filter (PF) implementation of the exact Bayesian filter [3, 4]

Based on the Monte Carlo simulations for 1-D scenarios it appeared that the IMMJPDA* filter outperforms the other filter algorithms. Moreover the IMMJPDA* performs on average almost as well as the particle filter implementation of the exact Bayesian filter does, at a 10 to 100 times lower computational load though. Since this is a very good finding for IMMJPDA*, the aim of this paper is to extend the track-coalescence-avoiding approach to the development mentioned under point 5 and also to extend the Monte Carlo simulations mentioned under point 6 to 2-D scenarios.

The paper is organized as follows. Section 2 defines the multi-target tracking problem considered. Section 3 presents the Joint IMM Coupled PDA (JIMMPDA) filter algorithm. Section 4 develops the track-coalescence-avoiding version of JIMMCPDA. Section 5 compares the algorithms through simulation for 1-D and 2-D scenarios; an overview of characteristics of these filter algorithms is given in table 1. Section 6 draws conclusions.

Table 1: Characteristics of filter algorithms considered

	Joint measurements	Joint manoeuvre modes	Hypotheses merging	Hypotheses pruning	Particle filter
IMMPDA [6]	-	-	yes	-	-
IMMJPDA [5]	yes	-	yes	-	-
IMMJPDA* [1, 2]	yes	-	yes	yes	-
JIMMCPDA [4]	yes	yes	yes	-	-
JIMMCPDA*	yes	yes	yes	yes	-
PF [3, 4]	yes	yes	-	-	yes

2 The multi-target tracking problem

Consider M targets and assume that the state of the i-th target is modelled as a jump linear system:

$$x_{t+1}^i = a^i(\theta_{t+1}^i)x_t^i + b^i(\theta_{t+1}^i)w_t^i, \qquad i = 1, ..., M, \quad (1)$$

where x_t^i is the n-vectorial state of the i-th target, θ_t^i is the Markovian switching mode of the i-th target and assumes values from $\{1, .., N\}$ according to a transition probability matrix Π^i, $a^i(\theta_t^i)$ and $b^i(\theta_t^i)$ are $(n \times n)$- and $(n \times n')$-matrices and w_t^i is a sequence of i.i.d. standard Gaussian variables of dimension n' with w_t^i , w_t^j independent for all $i \neq j$ and w_t^i ,x_0^i, x_0^j independent for all $i \neq j$.

A set of measurements consists of measurements originating from targets and measurements originating from clutter. We assume that a potential measurement originating from target i is also modelled as a jump linear system:

$$z_t^i = h^i(\theta_t^i)x_t^i + g^i(\theta_t^i)v_t^i \qquad , i = 1, ..., M \qquad (2)$$

where z_t^i is an m-vector, $h^i(\theta_t^i)$ is an $(m \times n)$-matrix and $g^i(\theta_t^i)$ is an $(m \times m')$-matrix, and v_t^i is a sequence of i.i.d. standard Gaussian variables of dimension m' with v_t^i and v_t^j independent for all $i \neq j$. Moreover v_t^i is independent of x_0^j and w_t^j for all i,j.

Let $x_t \triangleq \text{Col}\{x_t^1, ..., x_t^M\}$, $\theta_t \triangleq \text{Col}\{\theta_t^1, ..., \theta_t^M\}$, $A(\theta_t) \triangleq \text{Diag}\{a^1(\theta_t^1), ..., a^M(\theta_t^M)\}$, $B(\theta_t) \triangleq \text{Diag}\{b^1(\theta_t^1), ..., b^M(\theta_t^M)\}$, and $w_t \triangleq \text{Col}\{w_t^1, ..., w_t^M\}$. Then we can model the state of our M targets as follows:

$$x_{t+1} = A(\theta_{t+1})x_t + B(\theta_{t+1})w_t \tag{3}$$

with A and B of size $Mn \times Mn$ and $Mn \times Mn'$ respectively, with $\{\theta_t\}$ assuming values from $\{1, ..., N\}^M$ according to transition probability matrix $\Pi = [\Pi_{\eta,\theta}]$. If the M targets switch mode independently of each other, then:

$$\Pi_{\eta,\theta} = \prod_{i=1}^{M} \Pi^i_{\eta^i,\theta^i} \tag{4}$$

for every $(\eta, \theta) \in \{1, ..., N\}^{2M}$.

Next with $z_t \triangleq \text{Col}\{z_t^1, ..., z_t^M\}$, $H(\theta_t) \triangleq \text{Diag}\{h^1(\theta_t^1), ..., h^M(\theta_t^M)\}$, $G(\theta_t) \triangleq \text{Diag}\{g^1(\theta_t^1), ..., g^M(\theta_t^M)\}$, and $v_t \triangleq \text{Col}\{v_t^1, ..., v_t^M\}$, we obtain:

$$z_t = H(\theta_t)x_t + G(\theta_t)v_t \tag{5}$$

with H and G of size $Mm \times Mn$ and $Mm \times Mm'$ respectively.

We next assume that with a non-zero detection probability, P_{d}^i, target i is indeed observed at moment t. In addition to this there may be false measurements. We assume that the number of false measurements at moment t, F_t, has Poisson distribution:

$$\begin{aligned} p_{F_t}(F) &= \tfrac{(\lambda V)^F}{F!}\exp\bigl(-\lambda V\bigr), \qquad F = 0, 1, 2, \ldots \\ &= 0, \qquad\qquad\qquad\quad \text{else} \end{aligned} \tag{6.a}$$

where λ is the spatial density of false measurements and V is the volume of the observed region. Thus, λV is the expected number of false measurements in the observed region. We assume that the false measurements are uniformly distributed in the observed region, which means that a column-vector v_t^* of F_t i.i.d. false measurements has the following density:

$$p_{v_t^*|F_t}(v^*|F) = V^{-F} \tag{6.b}$$

Furthermore we assume that the process $\{v_t^*\}$ is a sequence of independent vectors, which are independent of $\{x_t\}, \{w_t\}, \{v_t\}$ and $\{\phi_t\}$.

At moment t a vector observation y_t is made, the components of which consist of the potential observations z_t^i of the detected targets plus the false measurements $\{v_t^*\}$. The multi-target tracking problem considered is to estimate (x_t, θ_t) given observations $Y_t \triangleq \{\text{y}_s,\ 0 \leq s \leq t\}$.

3 Joint IMM Coupled PDA filter

In [1–3] the problem formulated in section 2 has been embedded into one of filtering for a jump-linear descriptor system and the exact Bayesian filter equations have been derived. In [4] these have been used to develop a recursive algorithm by assuming that, for each $\theta \in \{1, ..., N\}^M$, the conditional density $p_{x_t|\theta_t,Y_{t-1}}(x \mid \theta)$ is approximated by a single Gaussian density on $\mathbb{R}^{Mn}$. The resulting algorithm performs both the IMM and the PDA steps jointly over all targets. We refer to the resulting algorithm as the JIMMCPDA (Joint IMM Coupled PDA) filter[1]. It consists of the following six subsequent steps.

JIMMCPDA Step 1: Interaction of the estimates from the previous filter cycle:
For all $\theta \in \{1, ..., N\}^M$, starting with

$$\begin{aligned} \hat{\gamma}_{t-1}(\theta) &\triangleq p_{\theta_{t-1}|Y_{t-1}}(\theta), \\ \hat{x}_{t-1}(\theta) &\triangleq E\{x_{t-1}|\theta_{t-1} = \theta, Y_{t-1}\}, \\ \hat{P}_{t-1}(\theta) &\triangleq E\{[x_{t-1} - \hat{x}_{t-1}(\theta)][x_{t-1} - \hat{x}_{t-1}(\theta)]^T \mid \\ &\qquad\qquad \mid \theta_{t-1} = \theta, Y_{t-1}\} \end{aligned}$$

one evaluates the mixed initial condition for the filter matched to $\theta_t = \theta$ as follows [7]:

$$\begin{aligned} \bar{\gamma}_t(\theta) &= \sum_{\eta \in \{1,...,N\}^M} \Pi_{\eta,\theta} \cdot \hat{\gamma}_{t-1}(\eta) \\ \hat{x}_{t-1|\theta_t}(\theta) &= \sum_{\eta \in \{1,...,N\}^M} \Pi_{\eta,\theta} \cdot \hat{\gamma}_{t-1}(\eta) \cdot \hat{x}_{t-1}(\eta)/\bar{\gamma}_t(\theta) \\ \hat{P}_{t-1|\theta_t}(\theta) &= \sum_{\eta \in \{1,...,N\}^M} \Pi_{\eta,\theta} \cdot \hat{\gamma}_{t-1}(\eta) \cdot \\ &\quad \cdot\Bigl(\hat{P}_{t-1}(\eta) + [\hat{x}_{t-1}(\eta) - \hat{x}_{t-1|\theta_t}(\theta)] \cdot \\ &\qquad \cdot[\hat{x}_{t-1}(\eta) - \hat{x}_{t-1|\theta_t}(\theta)]^T\Bigr)/\bar{\gamma}_t(\theta) \end{aligned}$$

JIMMCPDA Step 2: Prediction for all $\theta \in \{1,...,N\}^M$:

$$\bar{x}_t(\theta) = A(\theta)\hat{x}_{t-1|\theta_t}(\theta) \tag{7.a}$$

$$\bar{P}_t(\theta) = A(\theta)\hat{P}_{t-1|\theta_t}(\theta)A(\theta)^T + B(\theta)B(\theta)^T \tag{7.b}$$

$$\bar{Q}_t(\theta) = H(\theta)\bar{P}_t(\theta)H(\theta)^T + G(\theta)G(\theta)^T \tag{7.c}$$

Let $\bar{Q}_t^i(\theta)$ be the i-th $m \times m$ diagonal block matrix of $\bar{Q}_t(\theta)$.

JIMMCPDA Step 3: Gating, which is based on [6].
Identify for each target the mode for which Det $\bar{Q}_t^i(\theta)$ is largest:

$$\theta_t^{*i} = \text{Arg}\max_{\theta}\{\text{Det } \bar{Q}_t^i(\theta)\}$$

and use this to define for each target i a gate $G_t^i \in \mathbb{R}^m$ as follows:

$$\begin{aligned} G_t^i \triangleq \{z^i \in \mathbb{R}^m; &[z^i - h^i(\theta_t^{*i})\bar{x}_t^i(\theta_t^{*i})]^T \cdot \\ &\cdot\bar{Q}_t^i(\theta_t^{*i})^{-1}[z^i - h^i(\theta_t^{*i})\bar{x}_t^i(\theta_t^{*i})] \leq \nu\} \end{aligned}$$

[1] In [4] this algorithm was referred to as JIMMPDA, but JIMMCPDA is more appropriate.

with ν the gate size. If the j-th measurement y_t^j falls outside gate G_t^i; i.e. $\mathrm{y}_t^j \notin G_t^i$, then the j-th component of the i-th row of $[\Phi(\phi)^T\tilde{\chi}]$ is assumed to equal zero at moment t. This reduces the set of possible detection/permutation hypotheses to be evaluated at moment t for various ϕ to $\tilde{\mathcal{X}}_t(\phi)$.

JIMMCPDA Step 4: Evaluation of the detection/permutation hypotheses taking into account the reduced detection probability due to the limited gate size ν:

$$\begin{aligned}\beta_t(\phi,\tilde{\chi},\theta) &= \tfrac{\bar{\gamma}_t(\theta)}{c_t}\cdot F_t(\phi,\tilde{\chi},\theta)\lambda^{(L_t-D(\phi))}\cdot\\ &\cdot\left[\prod_{i=1}^{M}\left(1-P_{\mathrm{d}}^i\cdot\mathrm{Chi}^2_m(\nu)\right)^{(1-\phi_i)}\cdot\left(P_{\mathrm{d}}^i\cdot\mathrm{Chi}^2_m(\nu)\right)^{\phi_i}\right]\\ &\qquad\qquad \text{for } \tilde{\chi}\in\tilde{\mathcal{X}}_t(\phi),\\ &= 0 \qquad\qquad \text{else}\end{aligned} \tag{8.a}$$

$$F_t(\phi,\tilde{\chi},\theta)\cong[(2\pi)^{mD(\phi)}\mathrm{Det}\{Q_t(\phi,\theta)\}]^{-\frac{1}{2}}\cdot \cdot\exp\{-\frac{1}{2}\mu_t^T(\phi,\tilde{\chi},\theta)Q_t(\phi,\theta)^{-1}\mu_t(\phi,\tilde{\chi},\theta)\} \tag{8.b}$$

where

$$\mu_t(\phi,\tilde{\chi},\theta)\stackrel{\triangle}{=}\underline{\tilde{\chi}}\mathrm{y}_t-\underline{\Phi}(\phi)H(\theta)\bar{x}_t(\theta) \tag{8.c}$$

$$Q_t(\phi,\theta)\stackrel{\triangle}{=}\underline{\Phi}(\phi)\big(H(\theta)\bar{P}_t(\theta)H(\theta)^T+G(\theta)G(\theta)^T\big)\underline{\Phi}(\phi)^T \tag{8.d}$$

with c_t normalizing $\beta_t(\phi,\tilde{\chi},\theta)$ and $\mathrm{Chi}^2_m(\cdot)$ the Chi-squared cumulative distribution function with m degrees of freedom.

JIMMCPDA Step 5: Measurement-based update equations:

$$\hat{\gamma}_t(\theta)=\sum_{\phi,\tilde{\chi}}\beta_t(\phi,\tilde{\chi},\theta) \tag{9}$$

$$\hat{x}_t(\theta)\cong\bar{x}_t(\theta)+\sum_{\substack{\phi\\ \phi\neq 0}}K_t(\phi,\theta)\Big(\sum_{\tilde{\chi}}\beta_{t|\theta}(\phi,\tilde{\chi})\mu_t(\phi,\tilde{\chi},\theta)\Big) \tag{10}$$

$$\begin{aligned}\hat{P}_t(\theta)\cong\bar{P}_t(\theta)&+\\ &-\sum_{\substack{\phi\\ \phi\neq 0}}K_t(\phi,\theta)\underline{\Phi}(\phi)H(\theta)\bar{P}_t(\theta)\Big(\sum_{\tilde{\chi}}\beta_{t|\theta}(\phi,\tilde{\chi})\Big)+\\ &+\sum_{\substack{\phi\\ \phi\neq 0}}K_t(\phi,\theta)\Big(\sum_{\tilde{\chi}}\beta_{t|\theta}(\phi,\tilde{\chi})\mu_t(\phi,\tilde{\chi},\theta)\cdot\\ &\qquad\cdot\mu_t^T(\phi,\tilde{\chi},\theta)\Big)\cdot K_t^T(\phi,\theta)+\\ &-\left(\sum_{\substack{\phi\\ \phi\neq 0}}K_t(\phi,\theta)\Big(\sum_{\tilde{\chi}}\beta_{t|\theta}(\phi,\tilde{\chi})\mu_t(\phi,\tilde{\chi},\theta)\Big)\right)\cdot\\ &\cdot\left(\sum_{\substack{\phi'\\ \phi'\neq 0}}K_t(\phi',\theta)\Big(\sum_{\tilde{\chi}'}\beta_{t|\theta}(\phi',\tilde{\chi}')\mu_t(\phi',\tilde{\chi}',\theta)\Big)\right)^T\end{aligned} \tag{11}$$

with:

$$\begin{aligned}K_t(\phi,\theta) &= \bar{P}_t(\theta)H(\theta)^T\underline{\Phi}(\phi)^TQ_t(\phi,\theta)^{-1} \quad \text{if } \phi\neq 0,\\ &= 0 \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{else}\end{aligned}$$

$$\beta_{t|\theta}(\phi,\tilde{\chi})=\beta_t(\phi,\tilde{\chi},\theta)/\hat{\gamma}_t(\theta)$$

JIMMCPDA Step 6: Output equations:

$$\hat{x}_t=\sum_{\theta\in\{1,\ldots,N\}^M}\hat{\gamma}_t(\theta)\cdot\hat{x}_t(\theta) \tag{12}$$

$$\hat{P}_t=\sum_{\theta\in\{1,\ldots,N\}^M}\hat{\gamma}_t(\theta)\big(\hat{P}_t(\theta)+[\hat{x}_t(\theta)-\hat{x}_t]\cdot[\hat{x}_t(\theta)-\hat{x}_t]^T\big) \tag{13}$$

4 Track-coalescence-avoiding JIMMCPDA filter

A shortcoming of CPDA is its sensitivity to track coalescence. With the CPDA* approach, [8] has shown that this is due to CPDA's merging over permutation hypotheses, and that a suitable hypothesis pruning may provide an effective countermeasure. The CPDA* filter equations can be obtained from the CPDA algorithm by pruning per (ϕ_t,ψ_t)-hypothesis all less likely χ_t-hypotheses prior to measurement updating. The physical explanation for why this is working for two targets has been explained by Koch and Van Keuk [9]: "If targets move closely spaced for a longer period of time, it seems to be reasonable to represent the pdf by a symmetric form invariant against a permutation of the objects." In order to apply this approach to JIMMCPDA the CPDA* hypothesis pruning strategy is now extended: evaluate all (ϕ_t,ψ_t,θ_t) hypotheses and prune per (ϕ_t,ψ_t,θ_t)-hypothesis all less-likely χ_t-hypotheses. To do so, define for every ϕ, ψ and θ, satisfying $D(\psi)=D(\phi)\leq \mathrm{Min}\{M,L_t\}$, a mapping $\hat{\chi}_t(\phi,\psi,\theta)$:

$$\hat{\chi}_t(\phi,\psi,\theta)\stackrel{\triangle}{=}\underset{\chi}{\mathrm{Argmax}}\,\beta_t(\phi,\chi^T\Phi(\psi),\theta)$$

where the maximization is over all permutation matrices χ of size $D(\phi)\times D(\phi)$.
The pruning strategy of evaluating all (ϕ,ψ,θ)-hypotheses and only one χ-hypothesis per (ϕ,ψ,θ)-hypothesis implies that for $D(\phi)>0$ we adopt the following pruned hypothesis weights $\hat{\beta}_t(\phi,\psi,\theta)$:

$$\begin{aligned}\hat{\beta}_t(\phi,\psi,\theta) &= \beta_t(\phi,\hat{\chi}(\phi,\psi,\theta)^T\Phi(\psi),\theta)/\hat{c}_t\\ &\qquad\qquad \text{if } D(\phi)=D(\psi)\leq\mathrm{Min}\{M,L_t\}\\ &= 0 \qquad\qquad \text{else}\end{aligned}$$

with $\hat{c}_t$ a normalization constant for $\hat{\beta}_t$; i.e. such that

$$\sum_{\substack{\phi,\psi,\theta\\ D(\psi)=D(\phi)}}\hat{\beta}_t(\phi,\psi,\theta)=1$$

By inserting these particular weights within JIMMCPDA, we get JIMMCPDA*. One cycle of the JIMMCPDA* filter algorithm consists of 7 steps, the first four of which are equivalent to the JIMMCPDA steps:

JIMMCPDA Step 1:* Mixing
Equivalent to JIMMCPDA Step 1 in section 3.

JIMMCPDA Step 2:* Prediction
Equivalent to JIMMCPDA Step 2 in section 3.

JIMMCPDA Step 3:* Gating
Equivalent to JIMMCPDA Step 3 in section 3.

JIMMCPDA Step 4:* Evaluation of the detection/permutation hypotheses
Equivalent to JIMMCPDA Step 4 in section 3.

JIMMCPDA Step 5:* Track-coalescence hypothesis pruning.
First evaluate for every (ϕ, ψ, θ) such that $0 < D(\psi) = D(\phi) \leq \text{Min}\{M, L_t\}$:

$$\hat{\chi}_t(\phi, \psi, \theta) \triangleq \underset{\chi}{\text{Argmax}}\, \beta_t(\phi, \chi^T \Phi(\psi), \theta)$$

Next evaluate all $\hat{\chi}_t(\phi, \psi, \theta)$ hypothesis weights:

$$\begin{aligned}\hat{\beta}_t(\phi, \psi, \theta) &= \beta_t(\phi, \hat{\chi}_t(\phi, \psi, \theta)^T \Phi(\psi), \theta)/\hat{c}_t \\ &\qquad \text{if } 0 < D(\psi) = D(\phi) \leq \text{Min}\{M, L_t\} \\ &= \beta_t(\{0\}^M, \{\}^{L_t}, \theta)/\hat{c}_t \\ &\qquad \text{if } D(\psi) = D(\phi) = 0 \\ &= 0 \qquad \text{else}\end{aligned}$$

where $\hat{c}_t$ is a normalizing constant for $\hat{\beta}_t$.

JIMMCPDA Step 6:* Measurement update equations
For all $i \in \{1, ..., M\}$, $\theta^i \in \{1, ..., N\}$:

$$\hat{\gamma}_t(\theta) \cong \sum_{\phi, \psi} \hat{\beta}_t(\phi, \psi, \theta) \tag{14.a}$$

$$\hat{x}_t(\theta) \cong \bar{x}_t(\theta) + \sum_{\substack{\phi \\ \phi \neq 0}} K_t(\phi, \theta) \Bigg(\sum_{\psi} \hat{\beta}_{t|\theta}(\phi, \psi) \hat{\mu}_t(\phi, \psi, \theta) \Bigg) \tag{14.b}$$

$$\begin{aligned}\hat{P}_t(\theta) \cong\ & \bar{P}_t(\theta) + \\ & - \sum_{\substack{\phi \\ \phi \neq 0}} K_t(\phi, \theta) \underline{\Phi}(\phi) H(\theta) \bar{P}_t(\theta) \Bigg(\sum_{\psi} \hat{\beta}_{t|\theta}(\phi, \psi) \Bigg) + \\ & + \sum_{\substack{\phi \\ \phi \neq 0}} K_t(\phi, \theta) \Bigg(\sum_{\psi} \hat{\beta}_{t|\theta}(\phi, \psi) \hat{\mu}_t(\phi, \psi, \theta) \cdot \\ & \qquad \cdot \hat{\mu}_t(\phi, \psi, \theta)^T \Bigg) \cdot K_t^T(\phi, \theta) + \\ & - \Bigg(\sum_{\substack{\phi \\ \phi \neq 0}} K_t(\phi, \theta) \Bigg(\sum_{\psi} \hat{\beta}_{t|\theta}(\phi, \psi) \hat{\mu}_t(\phi, \psi, \theta) \Bigg) \Bigg) \cdot \\ & \cdot \Bigg(\sum_{\substack{\phi' \\ \phi' \neq 0}} K_t(\phi', \theta) \Bigg(\sum_{\psi'} \hat{\beta}_{t|\theta}(\phi', \psi') \hat{\mu}_t(\phi', \psi', \theta) \Bigg) \Bigg)^T\end{aligned} \tag{14.c}$$

with

$$\begin{aligned}K_t(\phi, \theta) &= \bar{P}_t(\theta) H(\theta)^T \underline{\Phi}(\phi)^T Q_t(\phi, \theta)^{-1} && \text{if } \phi \neq 0, \\ &= 0 && \text{else} \\ \hat{\mu}_t(\phi, \psi, \theta) &= \mu_t(\phi, \hat{\chi}(\phi, \psi, \theta)^T \Phi(\psi), \theta) \\ \hat{\beta}_{t|\theta}(\phi, \psi) &= \hat{\beta}_t(\phi, \psi, \theta)/\hat{\gamma}(\theta)\end{aligned}$$

JIMMCPDA Step 7:* Output equations
Equivalent to JIMMCPDA Step 6 in section 3.

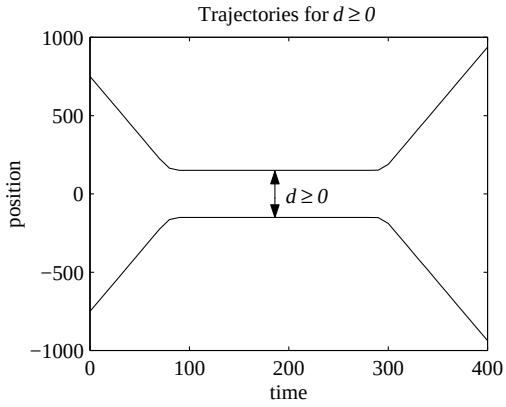
5 Monte Carlo simulations

In this section some Monte Carlo simulation results are given for the IMMJPDA, JIMMCPDA, IMMJPDA*, and JIMMCPDA* filter algorithms, and for an IMMPDA which updates an individual track using PDA by assuming the measurements from the adjacent targets as false. The simulations primarily aim at gaining insight into the behavior and performance of the filters when objects move in and out close approach situations, while giving the filters enough time to converge after a manoeuvre has taken place. In the example scenarios there are two targets. First we simulate a 1-D position example for 100 different parameterizations. Next we simulate a 2-D position example for 8 different parameterizations. For the 1-D position scenarios we also provide the exact Bayesian filter based particle filtering (PF) results of [3, 4], though the computational load increases a factor 10 to 100 over the others.

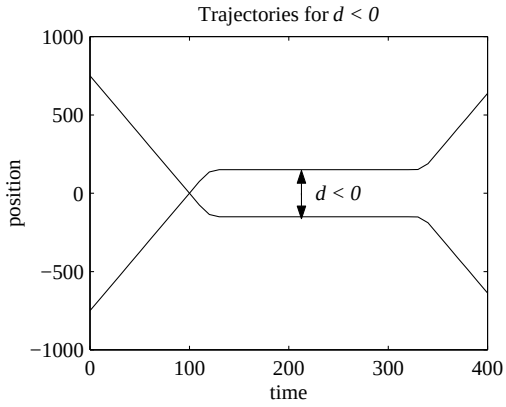
5.1 1-D position scenarios

In the simple example scenarios (see figure 1), two objects start moving in 1-D position towards each other, each with constant initial velocity V_{initial} (i.e. the initial relative velocity $V_{\text{rel, initial}} = -2V$). At a certain moment in time both objects start decelerating with -50 m/s^2 until they both have zero velocity. The moment at which the deceleration starts is such that when the objects both have zero velocity, the distance between the two objects equals d. After spending a significant number of scans with zero velocity, both objects start accelerating with 50 m/s^2 away from each other without crossing until their velocity equals the opposite of their initial velocity. From that moment on the velocity of both objects remains constant again (thus the final relative velocity $V_{\text{rel, final}} = 2V$). Note that $d < 0$ implies that the objects have crossed each other before they have reached zero velocity. Each simulation runs for 40 scans and the filters start with perfect estimates. Examples of the trajectories for $d > 0$ and $d < 0$ are depicted in figures 1a and 1b respectively.
For each target, the underlying model of the potential target measurements is given by (1) and (2)

$$\begin{aligned}x_{t+1}^i &= a^i(\theta_{t+1}^i) x_t^i + b^i(\theta_{t+1}^i) w_t^i && (1) \\ z_t^i &= h^i(\theta_t^i) x_t^i + g^i(\theta_t^i) v_t^i && (2)\end{aligned}$$



1a. Trajectories examples for $d \geq 0$



1b. Trajectories examples for $d < 0$

Fig. 1: Trajectories examples for $d \geq 0$ and for $d < 0$

with for $i = 1, 2$ and $\theta_t^i \in \{1,2\}$:

$$a^i(1) = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad a^i(2) = \begin{bmatrix} 1 & T_s & \frac{1}{2}T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$$

$$b^i(1) = \sigma_a^i \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad b^i(2) = \sigma_a^i \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$h^i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad g^i = \sigma_m^i$$

$$\Pi = \begin{bmatrix} 1 - T_s/\tau_1 & T_s/\tau_1 \\ T_s/\tau_2 & 1 - T_s/\tau_2 \end{bmatrix}$$

where σ_a^i represents the standard deviation of acceleration noise and σ_m^i represents the standard deviation of the measurement error. For simplicity we consider the situation of similar targets only; i.e. $\sigma_a^i = \sigma_a$, $\sigma_m^i = \sigma_m$, $P_{\mathrm{d}}^i = P_{\mathrm{d}}$. With this, the scenario parameters are P_{d}, λ, d, V, T_s, σ_m, σ_a, τ_1, τ_2, and the gate size ν. We used fixed parameters $\sigma_m = 30$, $\sigma_a = 50$, $\tau_1 = 50$, $\tau_2 = 5$, and $\nu = 25$. Table 2 gives the other scenario parameter values that are being used for the Monte Carlo simulations.

During our simulations we counted track i "O.K.", if

$$\mid h^i \hat{x}_T^i - h^i x_T^i \mid \leq 9\sigma_m$$

Table 2: Scenario parameter values;
$d \in \{-12, ..., -1, 0, 1, ..., 12\}$
IMMPDA's $\lambda = 0.00001$ for scenarios A1 and A3

Scenario	P_{d}	λ	d	V	T_s
A1	1	0	Variable	75	1
A2	1	0.001	Variable	75	1
A3	0.9	0	Variable	75	1
A4	0.9	0.001	Variable	75	1

where $\mid \cdot \mid$ denotes the l_2-norm. We counted track $i \neq j$ "Swapped", if

$$\mid h^i \hat{x}_T^i - h^j x_T^j \mid \leq 9\sigma_m$$

and we counted track i and j "Coalescing" at scan t, if

$$\mid h^i x_t^i - h^j x_t^j \mid > 9\sigma_m \quad \wedge \quad \mid h^i \hat{x}_t^i - h^j \hat{x}_t^j \mid \leq \sigma_m$$

For each of the scenarios Monte Carlo simulations containing 100 runs have been performed for each of the tracking filters. To make the comparisons more meaningful, for all tracking algorithms the same random number streams were used. The Monte Carlo simulation results for the four scenarios are presented in Table 3.

Table 3: Monte Carlo simulation results.

	Average % Both Tracks O.K.				Average % Both Tracks O.K. or Swapped			
	A1	A2	A3	A4	A1	A2	A3	A4
IMMPDA	19	10	6	4	28.3	18.9	8.5	5.6
IMMJPDA	66	56	63	41	99.96	92.5	99.8	76.6
IMMJPDA*	73	68	69	50	100	96.8	100	81.0
JIMMCPDA	54	47	52	35	79.6	77.3	80.1	65.6
JIMMCPDA*	70	66	68	49	99.8	97.3	99.9	76.8
PF	75	70	72	57	96.2	94.6	95.8	82.3
	Average number of Coalescing scans				Average CPU time per scan (in milliseconds)			
	A1	A2	A3	A4	A1	A2	A3	A4
IMMPDA	9.7	11.0	18.9	14.5	16	38	14	38
IMMJPDA	1.5	2.1	1.7	2.6	22	54	20	61
IMMJPDA*	0.4	0.3	0.5	0.5	23	48	20	56
JIMMCPDA	3.3	3.7	3.4	3.8	42	70	37	85
JIMMCPDA*	0.4	0.4	0.6	0.5	40	63	36	78
PF	1.3	1.4	1.3	1.5	440	7960	440	7810

For the 1-D position example considered, the simulation results show that JIMMCPDA, JIMMCPDA*, IMMJPDA and IMMJPDA* perform much better than IMMPDA. The results in table 3 also show that JIMMCPDA* and IMMJPDA* perform on average better than JIMMCPDA and IMMJPDA. Both JIMMCPDA* and IMMJPDA* avoid track coalescence and are less sensitive to track loss than JIMMCPDA and IMMJPDA are. As a result of this, the

IMMJPDA and IMMJPDA* perform on average better than JIMMCPDA and JIMMCPDA* respectively. As expected, a particle filter implementation of the exact Bayesian filter [3, 4] provides the best performance when the tracking scenario is most demanding (scenario A4). However, even then, the improvement over IMMJPDA* and JIMMCPDA* is rather limited. Previous results [3, 4] also show that each of the algorithms may outperform the others at some particular d value. Thus if the comparison would be for one d-value only, as is common practice in literature, each of the algorithms might be a best performing tracking algorithm.

5.2 2-D position scenarios

So far all simulation results apply to a 1-dimensional position example. Now we verify if similar results also apply to a 2-dimensional example. To do so, we perform Monte Carlo simulations for two targets flying the 2-D trajectory patterns as pictured in Figure 2 and in Figure 3.

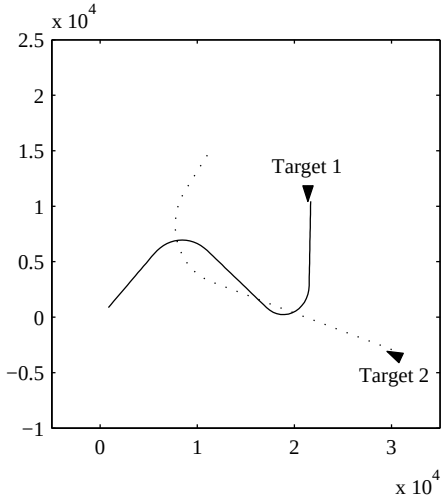


Fig. 2: Trajectory patterns of scenario R0

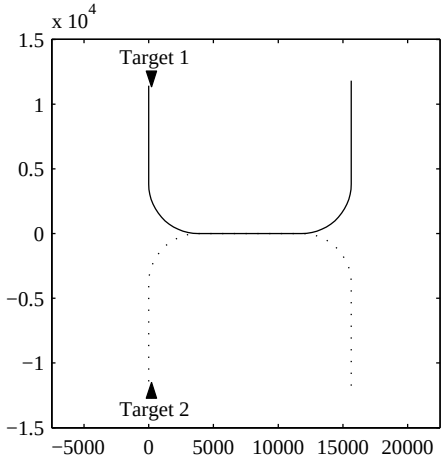


Fig. 3: Trajectory patterns of scenario R1

The target trajectory patterns in Figure 2 are of [5]. We refer to this as **scenario R0**. In addition to this we simulate trajectory patterns that are kind of 2-D position versions of the 1-D position jointly manoeuvering target scenarios of section 5.1. These are depicted in Figure 3. From 0 to 20s, targets 1 and 2 fly at a speed of 400 m/s in a straight line in south and north direction respectively. From 20 to 35s, both targets make a coordinated turn to the east. From 35 s to 55s, both targets fly in a straight line to the east. From 55s to 70s, targets 1 and 2 make a coordinated turn to the north and to the south respectively. From 70s to 90s, targets 1 and 2 fly in a straight line to the north and to the south respectively. Of the jointly manoeuvering target trajectories we consider seven scenarios, which differ in the initial position of Target 1 only:

Scenario R1: Target 1 starts at (0,11820m) and target 2 starts at (0,-11820m).

Scenario R2/R2′: Same as R1 but initial position of target 1 is shifted 200/100m to the south.

Scenario R3/R3′: same as R1 but initial position of target 1 is shifted 200/100m to the north.

Scenario R4/R4′: Same as R1 but initial position of target 1 is shifted 200/100m to the east.

Similar as in [5], the target motion models for the two targets are identical. In each mode the target dynamics are modelled in Cartesian coordinates as given by (1):

$$x_{t+1}^i = a^i(\theta_{t+1}^i)x_t^i + b^i(\theta_{t+1}^i)w_t^i \tag{1}$$

where the state of the target is position, velocity and acceleration in each of the the two Cartesian coordinates (x, y). Thus x_t^i is of dimension 6 $(n = 6)$. Three modes for $\{\theta_t^i\}$ are adopted. The corresponding system matrices $a^i(\theta^i)$ and $b^i(\theta^i)$, for $\theta^i \in \{1, 2, 3\}$ and $i \in \{1, 2\}$, are defined as:

$$a^i(\theta^i) = \begin{bmatrix} a_1^i(\theta^i) & 0 \\ 0 & a_2^i(\theta^i) \end{bmatrix},\ a_j^i(1) = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$a_j^i(2) = a_j^i(3) = \begin{bmatrix} 1 & T_s & \frac{1}{2}T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$$

$$b^i(\theta^i) = \begin{bmatrix} b_1^i(\theta^i) & 0 \\ 0 & b_2^i(\theta^i) \end{bmatrix},\ b_j^i(1) = \sigma_a^i(1) \cdot \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \\ 0 \end{bmatrix}$$

$$b_j^i(2) = \sigma_a^i(2) \cdot \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \\ 1 \end{bmatrix}, b_j^i(3) = \sigma_a^i(3) \cdot \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \\ 1 \end{bmatrix}$$

where T_s is the sampling period.

- *Model* 1: nearly constant velocity model with zero mean perturbation in acceleration with $a^i(\theta) = a^i(1)$ and $b^i(\theta) = b^i(1)$. The standard deviation $\sigma_a^i(1)$ of the process noise is $\sigma_a^i(1) = 5m/s^2$.

- *Model* 2: Wiener process acceleration (nearly constant acceleration motion) with $a^i(\theta) = a^i(1)$ and $b^i(\theta) = b^i(2)$. The standard deviation $\sigma_a^i(1)$ of the process noise is $\sigma_a^i(2) = 7.5m/s^2$.

- *Model* 3: Wiener process acceleration (model with large acceleration increments, for the onset and termination of maneuvers) with $a^i(\theta) = a^i(3)$ and $b^i(\theta) = b^i(3)$. The standard deviation $\sigma_a^i(3)$ of the process noise is $\sigma_a^i(2) = 40m/s^2$.

The initial model probabilities for the two targets are identical: $\hat{\gamma}_0^i(1) = 0.8$, $\hat{\gamma}_0^i(2) = 0.1$, $\hat{\gamma}_0^i(3) = 0.1$. The mode switching probability matrix for each of the two targets is also identical and is given by:

$$\Pi^i = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

The potential sensor measurements for target i are as given by (2):

$$z_t^i = h^i(\theta_t^i)x_t^i + g^i(\theta_t^i)v_t^i \tag{2}$$

with

$$h^i(\theta^i) = \begin{bmatrix} h_1^i(\theta^i) & 0 \\ 0 & h_2^i(\theta^i) \end{bmatrix}, \; g^i(\theta^i) = \begin{bmatrix} g_1^i(\theta^i) & 0 \\ 0 & g_2^i(\theta^i) \end{bmatrix}$$

$$h_j^i(\theta^i) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad g_j^i(\theta^i) = \sigma_m, \qquad j \in \{1, 2\}$$

The standard deviation σ_m of the measurement error is $\sigma_m = 20$m. The sensor is assumed to be located at the coordinate system origin. The sampling interval $T_s = 1$s and it was assumed that the probability of detection $P_{\mathrm{d}} = 0.997$. For generating false measurements in simulations the clutter was assumed to be Poisson distributed with expected number of $\lambda = 1 \times 10^{-6}/\mathrm{m}^2$. The gates for setting up the validation regions for the measurements were based on the threshold $\nu = 25$.

For each of the scenarios Monte Carlo simulations containing 100 runs have been performed for each of the tracking filters. To make the comparisons more meaningful, for all tracking algorithms the same random number streams were used. Using the same criteria as for the 1-D position example, the results of the Monte Carlo simulations for the scenarios are depicted in four Tables:

- The percentage of Both tracks "O.K.", in Table 4.
- The percentage of Both tracks "O.K." or "Swapped", in Table 5.
- The percentage of "Coalescing" tracks, in Table 6.
- The average CPU time per scenario in Table 7.

Most remarkable is the dramatic decrease in performance by JIMMCPDA for scenarios where the two targets come closer than 200m to each other, i.e. R1 (0m), R2′ (100m), R3′ (100m) and R4′ (100m). These scenarios have in common that they cause JIMMCPDA to be caught in a situation where it has strongly coupled uncertainty about which target is gone in which direction. As a result of this JIMMCPDA increases its covariance and then diverges. For these scenarios, the permutation hypothesis pruning of JIMMCPDA* appears to mitigate this problem effectively.

Table 4: % Both tracks "O.K.".

	R0	R1	R2	R2′	R3	R3′	R4	R4′
IMMPDA	94	0	1	0	17	0	8	0
IMMJPDA	97	0	71	42	96	59	98	39
IMMJPDA*	97	51	73	57	95	59	98	71
JIMMCPDA	97	0	83	4	91	5	97	10
JIMMCPDA*	97	46	80	39	95	32	98	78

Table 5: % Both tracks "O.K." or "swapped".

	R0	R1	R2	R2′	R3	R3′	R4	R4′
IMMPDA	94	1	15	0	23	1	18	2
IMMJPDA	97	0	92	91	98	94	98	90
IMMJPDA*	97	97	96	96	98	97	98	98
JIMMCPDA	97	0	97	15	93	16	97	15
JIMMCPDA*	97	97	96	96	98	97	98	98

Table 6: % Coalescing tracks. (i.e. tracks with three or more subsequently coalescing scans)

	R0	R1	R2	R2′	R3	R3′	R4	R4′
IMMPDA	2	99	82	98	77	98	71	99
IMMJPDA	0	74	0	0	0	0	0	1
IMMJPDA*	0	0	0	0	0	0	0	0
JIMMCPDA	0	99	0	51	5	43	1	59
JIMMCPDA*	0	0	0	0	0	0	0	0

Table 7: Average CPU time per scan (in milliseconds).

	R0	R1	R2	R2′	R3	R3′	R4	R4′
IMMPDA	18	17	17	17	16	16	16	17
IMMJPDA	20	74	20	21	19	19	18	21
IMMJPDA*	20	18	18	18	18	18	18	18
JIMMCPDA	33	138	33	123	36	130	30	127
JIMMCPDA*	31	29	31	29	28	28	27	27

Although to a far less degree, the same negative phenomenon for JIMMCPDA is working for scenario R3, in which target 1 stays 200m north of target 2. This for JIMMCPDA negative phenomenon is gone for scenario R4, in which target 1 stays 200m behind target 2. JIMMCPDA works best of all for scenario R2, in which target 1 crosses target 2 and then stays 200m south of it.
In contrast with JIMMCPDA, IMMJPDA keeps performing quite well for all scenarios, except for R1, i.e. when the two targets come at zero distance of each other. IMMJPDA*, IMMJPDA* and IMMJPDA perform similarly well for 4 scenarios (R0, R2, R3, R4), and IMMJPDA* performs dramaticly to significantly better than IMMJPDA for 3 scenarios (R1, R2′, R4′). If track swap is preferred above track loss, then IMMJPDA* even performs significantly better than IMMJPDA for 5 scenarios (R1, R2, R2′, R3′, R4′)

The results can be summarized as follows:

Scenario R0: Significant improvement of IMMJPDA, JIMMCPDA, JIMMCPDA* and IMMJPDA* over IMMPDA, and similar performance by these four.

Scenario R1: Dramatic improvement of IMMJPDA* and JIMMCPDA* over IMMPDA, IMMJPDA and JIMMCPDA.

Scenarios R2, R3 and R4: Dramatic improvement of IMMJPDA, JIMMCPDA, JIMMCPDA* and IMMJPDA* over IMMPDA. For R2, JIMMCPDA and JIMMCPDA* perform significantly better than IMMJPDA and IMMJPDA*. For R3, IMMJPDA, JIMMCPDA and IMMJPDA* perform significantly better than JIMMCPDA. For R4 similar performance by all four.

Scenarios R2′, R3′, R4′: Dramatic improvement of IMMJPDA, JIMMCPDA* and IMMJPDA* over IMMPDA and JIMMCPDA. Moreover, significant improvement of IMMJPDA* over IMMJPDA for R2′ and R4′, and similar performance for R3′.

If two targets fly for a while very close to each other (i.e. R1), then IMMJPDA* and JIMMCPDA* perform far better than the others. If the targets stay 5 times σ_m from each other (i.e. R2′, R3′, R4′), then IMMJPDA* performs similar or significantly better than IMMJPDA. If the targets stay 10 times σ_m from each other (i.e. R2, R3, R4), then IMMJPDA, JIMMCPDA, IMMJPDA* and JIMMCPDA* perform almost equally well. Similar as for the 1-D position example, each of the four may perform best in some special situations, i.e. IMMJPDA for R3, JIMMCPDA for R2, JIMMCPDA* for R4′, IMMJPDA* for R1 and R2′. Situations of significant track coalescence apply to IMMPDA for R1-R4; to IMMJPDA for R1 and to JIMMCPDA for R1, R2′, R3, R3′ and R4′. It is also nice to see that IMMJPDA* clearly outperforms IMMPDA at a 10%-20% higher computational load only.

6 Concluding remarks

For the problem of tracking closely spaced maneuvering targets from possibly false and missing observations, the paper has developed a track-coalescence-avoiding version (JIMMCPDA*) of the JIMMCPDA in [4]. Through Monte Carlo simulations for 1-D and 2-D scenarios, JIMMCPDA* is compared to IMMJPDA, IMMJPDA* and JIMMCPDA. For closely spaced maneuvering targets, JIMMCPDA* and IMMJPDA* perform best in terms of the chance that both tracks are ”OK” or ”swapped”, whereas JIMMCPDA* has about a 50% higher computational load than IMMJPDA*. The paper has also shown that in case of significant clutter density there remains some room for improvement by using a better approximation of the exact Bayesian equations (e.g. by a good particle filter). However the computational load is then much higher.
Interesting follow up work is to extend IMMJPDA*, JIMMCPDA* and particle filtering implementations of the exact Bayesian filter into other directions such as finite sensor resolution (e.g. [9]), Variable Structure IMM (e.g. [10]). and Integrated track initiation (e.g. [11, 12]),

References

[1] Blom, H. A. P., and E. A. Bloem, “Combining IMM and JPDA for tracking multiple maneuvering targets in clutter,” *Proc. 5th Int. Conf. on Information Fusion*, July 8-11, 2002a, Vol. 1, pp. 705-712.

[2] Blom, H. A. P., and E. A. Bloem, “Interacting Multiple Model Joint Probabilistic Data Asscoiation avoiding track coalescence,” *Proc. IEEE Conf. on Decision and Control*, December 2002b, pp. 3408-3415.

[3] Blom, H. A. P., and E. A. Bloem, “Joint IMMPDA Particle filter,” *Proc. 6th Int. Conf. on Information Fusion*, July 8-11, 2003a, Vol. 1, pp. 785-792.

[4] Blom, H. A. P., and E. A. Bloem, “Tracking multiple maneuvering targets by Joint combinations of IMM and PDA,” *Proc. 42nd IEEE Conf. on Decision and Control*, December 9-12, 2003b, pp. 2965-2970.

[5] Chen, B., and J. K. Tugnait, “Tracking of multiple maneuvering targets in clutter using IMM/JPDA filtering and fixed-lag smoothing,” *Automatica*, vol. 37, pp. 239-249, Feb. 2001.

[6] Bar-Shalom, Y., and X. R. Li, “Multitarget-Multisensor Tracking: Principles and Techniques,” *YBS Publishing*, Storrs, CT, 1995.

[7] Blom, H. A. P., and Y. Bar-Shalom, “The Interacting Multiple Model algorithm for systems with Markovian switching coefficients,” *IEEE Tr. on Automatic Control*, Vol. 33 (1988), pp. 780-783.

[8] Blom, H. A. P., and E. A. Bloem, “Probabilistic Data Association Avoiding Track Coalescence,” *IEEE Tr. on Automatic Control*, Vol. 45 (2000), pp. 247-259.

[9] Koch, W. and G. Van Keuk, “Multiple Hypothesis Track Maintenance with Possibly Unresolved Measurements,” *IEEE Trans. Aerospace and Electronic Systems*, Vol. 33, No. 3, July 1997, pp. 883-892.

[10] Li, X. R., and Y. Bar-Shalom,“Multiple-Model estimation with Variable Structure,” *IEEE Transactions on Automatic Control*, Vol. 41, No. 4, April 1996, pp. 478-493.

[11] Mušicki, D., and R.J. Evans, “Joint Integrated Probabilistic Data Association - JIPDA,” *Proc. Int. Conf on Information Fusion 2002*, pp. 1120-1125.

[12] Moreland, M.R., and S. Challa, “A multitarget tracking algorithm based on random sets,” *Proc. Int. Conf on Information Fusion 2003*, pp. 807-814.