

MULTIPLE TARGET TRACKING USING CHEAP JOINT PROBABILISTIC DATA ASSOCIATION MULTIPLE MODEL PARTICLE FILTER IN SENSORS ARRAY

Messaoudi Zahir^{1,2} Oussalah Mourad¹ and Ouldali Abdelaziz²

¹ EECE, University of Birmingham, Edgbaston, Birmingham B15 2TT, Birmingham, UK.

M.oussalah@bham.ac.uk

² EOSL, Military Polytechnic School, BP 17 Bordj El Bahri, Algiers, Algeria
messaoudi_zahir06@yahoo.fr; aaziz_ouldali@yahoo.fr

ABSTRACT

Joint multiple target tracking and classification is an important issue in many engineering applications. In recent years, multiple sensor data fusion has been extensively investigated by researchers in a variety of disciplines. Indeed, combining results issued from multiple sensors can provide more accurate information than using a single sensor. In the present paper we address the problem of data fusion for joint multiple maneuvering target tracking and classification in cluttered environment where centralized versus decentralized architectures are often opposed. The proposal advocates a hybrid approach combining a Particle Filter (PF) like method to deal with system nonlinearities and Fitzgerald's Cheap Joint Probabilistic Data Association Filter CJPDAF for the purpose of data association and target estimation problems, yielding CJPDA-PF algorithm. While the target maneuverability is tackled using a combination of a Multiple Model Filter (MMF) and CJPDAF, which yields CJPDA-MMPF algorithm. Consequently, at each particle level (of the particle filter), the state of the particle is evaluated using the suggested CJPDA-MMF. In case of several sensors, the centralized fusion architecture and the distributed architecture in the sense of Federated Kalman Filtering are investigated and compared. The feasibility and the performances of the proposal have been demonstrated using a set of Monte Carlo simulations dealing with two maneuvering targets with two distinct operation modes and various clutter densities.

KEYWORDS

Data fusion, Multiple manoeuvring targets tracking; JPDA; MMPF

1. INTRODUCTION

Target tracking is a necessary part of systems that perform functions such as surveillance, guidance or obstacle avoidance. Tracking is the process of filtering noisy measurements from one or more sensors such as radar, sonar or video to achieve the best possible estimate of the state of the target given the state and measurement model with possibly uncertain target-measurement associations. The state model describes the evolution of the target between two consecutive time increments; typically a set of differential equations highlighting the cinematic model of the target was used for the target model. While the measurement model links the target state vector to the measurement vector. Such models allow the system to predict both the state vector of the target and the measurement based on current state of target. The prediction is then combined with sensor measurements to produce an estimate of the target state. On the other hand, the use of noise models allows the system to reduce the influence of uncertainty pervading both the measurement and state models [1].

In order to deal with more than one target, the multiple targets tracking system requires handling the discrete uncertainty of measurement origin. This is known as the data association problem. For this purpose, several algorithms have been put forward. This includes the nearest neighbor (NN) algorithm which associates the measurement to the closest predicted target [2]. The joint probabilistic data association (JPDA) algorithm forms a set of probabilistic hypotheses over every possible target-measurement association [2,3], which constitute the weights of the weighted innovation expression over the set of measurements used for the filter update. Unfortunately the computational complexity of the JPDAF, which increases exponentially with the number of targets, constitutes a practical handicap to its implementation. In order to reduce the burden complexity, Fitzgerald [4] developed the simplified version of the JPDAF, called the cheap JPDAF (CJPDAF) algorithm where the target-measurement association probabilities are computed via a simple and intuitive formula while the main skeleton of the JPDAF algorithm is kept unchanged. For similar reasons, a fuzzy-based approach for JPDAF has also been put forward in [5] where the association probabilities are computed via fuzzy clustering-like approach. The issue of target correlation, which, in the worst case, arises from the common noise pervading all the targets and/or measurements, has been discussed in [3, 6], among others. On the other hand, the presence of high nonlinearities and the change of models when a target executes a maneuver for instance constitute other important challenges to design efficient multiple target tracking algorithms. The former challenge is usually tackled using variants of extended Kalman filters, unscented Kalman filter and/or particle filter (PF) [7,8]. While the model switch problem is usually handled using multiple model filter (MMF) based approach or its variants employing Markov transition matrices [2,3]. The combination of the above issues gives raise to various extension of JPDAF or CJPDAF.

The use of several, possibly asymmetric, sensors raises the issue of optimal fusion of the sensory information. Indeed, the data fusion referred to as a process that combines data and knowledge from different sources with the aim of maximizing the useful information content, for improved reliability or discrimination power [9], is still an open issue for efficient multiple target tracking algorithms. In this course two sub-challenges can be distinguished. The first deals with the issue of sensor management while the second emphasizes the overall fusion architecture highlighting its centralized versus decentralized aspects. Nash [9] suggested handling the sensors management technique as an optimization problem. He used a linear programming like approach to determine sensor-to-target assignment and chooses the trace of the Kalman filter error covariance as the costs coefficients in the objective functions. Kastella and Musick [10] used the method of Nash with discrimination gain as the new cost functions. McIntyre [11] presented a hierarchical sensor management model predicated using information theoretic measure. Dils and al. [12] employed the cross-entropy-based sensors effectiveness metric for sensors assignment. For the architecture aspect, one shall mention the Federated Kalman Filter (FKF) versus the Centralized Kalman Filter (CKF) [13,14]. In this paper, a hybrid approach implementing the multiple-model filter and cheap JPDAF at each particle level, yielding a CJPDA-MMPF for tracking and discriminating several maneuvering targets extending our earlier work in [15]. While for sensors architecture / management issue, the FKF and CKF approaches are compared to the standard the sequential method. More formally, the process of tracking is based on the following ideas: *i*) the estimation process (target kinematics variables and regime variable) is carried out at each particle level; *ii*) the calculus of the joint probabilities is accomplished using CJPDAF; *iii*) the sensors fusion is accomplished using the CKF and the FKF; *iv*) the model switch in case of target maneuver is handled using the MMPF. Section 2 presents the general target/measurement models. Section 3 investigates the data association problem and highlights the proposed solution based on a combination of CJPDA and Multiple Model Particle Filter approach.

Section 4 emphasizes the data fusion architecture where the Federated Kalman filter and centralized Kalman filtering are investigated. In Section 5, some simulations examples are performed, which illustrate the feasibility and quantify performances of the proposal.

2. TARGET/MEASUREMENT MODELS

We shall consider throughout this paper targets that change course during their action with varying velocity and/or acceleration models. For instance, a target starts with a constant acceleration model, then switches to a constant velocity model, next switches to constant turn model, and, finally, switches again to a constant deceleration till it stops. Consequently, given the state vector of the target and the observation acquired by some sensors, the problem of efficient estimation of the state vector of the target becomes crucial. This is referred to as hybrid system. More formally, given a target t , sensor s , a hybrid system is described by the following equations:

$$x_{k+1}^t = f(x_k^t, r^t(k), u^t(k)) \quad t \in \{1, 2, \dots, N\} \quad (1)$$

$$z_k^{t,s} = h^s(x_k^t, w^s(k)) \quad (2)$$

where x_k^t is the state of the target t at time k , N is the number of targets (assumed to be known in advance), $u^t(k)$ is the noise which corrupts the state x_k^t , $f(\cdot, r^t(k), \cdot)$ is the transition function that depends on the mode variable $r^t(k)$. $z_k^{t,s}$ is the received measure from the target t on the s -th sensor, $w^s(k)$ is the measurement noise and finally $h^s(\cdot, \cdot)$ is a measurement transition function. The noise $u^t(k)$ and $w^s(k)$ are assumed zero mean Gaussian with known variance-covariance matrices. The mode variable $r^t(k)$ is commonly modeled by time-homogeneous M -state first order Markov chain with transitional probabilities $\Pi = [\pi_{ij}]$:

$$\pi_{i,j} = P(r^t(k) = j | r^t(k-1) = i) \quad (i, j \in \{1, 2, \dots, M\}) \quad (3)$$

The problem boils down to estimating the state of target state vector x_k^t at time increment k given the observation $z_k^{t,s}$ and evolving model (1).

2.1. Target dynamics

In the present work, we consider two different motion models for target t : the constant velocity and the coordinated turn rate models.

Throughout this paper, for notational simplicity the superscripts t and s refer to target t and sensor s , respectively. We denote by (α^t, ξ^t) the x-y coordinates of the target t . $x^t(k)$ denotes the state vector, which includes both coordinates and velocities components such as

$$x_k^t = [\alpha^t, \dot{\alpha}^t, \xi^t, \dot{\xi}^t]^T \quad (4)$$

The mode variable $r^t(k)$ takes therefore values in $\{1, 2\}$ and allows the system to select either the constant velocity model or the coordinated turn model. For instance, $r^t(k) = 1$ indicates that the target t is evolving according to constant velocity model. While $r^t(k) = 2$ indicates that it

evolves according to constant turn rate model. Also, we shall refer to the notations $(k|k-1)$ or $(k+1|k)$ for prediction and $(k|k)$ or $(k+1|k+1)$ for update.

2.1.1 Constant velocity model

In this model, the target t is assumed to move with a constant velocity yielding a linear trajectory. The evolution of the target state vector between two time increments is given by the following:

$$x_{k+1|k}^t = F^t(r^t(k)=1) x_{k|k}^t + D u^t(k) = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k|k}^t + \begin{bmatrix} \frac{\Delta^2}{2} & 0 \\ \Delta & 0 \\ 0 & \frac{\Delta^2}{2} \\ 0 & \Delta \end{bmatrix} u^t(k) \quad (5)$$

where Δ is the sampling period and $u^t(k) \sim N(0, R^t)$ (Gaussian noise with zero mean and variance-covariance matrix R^t). F^t and D are transition matrices given on the right hand side of expression (5).

2.1.2 Constant rate coordinated turn model

In this model, the target t executes a curvature trajectory with a constant turn angle rate ω^t and a constant speed in terms of the norm of the velocity vector. The associated dynamic model is generated from (5) by changing the transition matrix $F^t(r^t(k)=1)$ in (5) into $F^t(r^t(k)=2)$ as follows

$$F^t(r^t(k)=2) = \begin{bmatrix} 1 & \sin(\omega^t \Delta) & 0 & \frac{(1-\cos(\omega^t \Delta))}{\omega^t} \\ 0 & \cos(\omega^t \Delta) & 0 & -\sin(\omega^t \Delta) \\ 0 & \frac{(1-\cos(\omega^t \Delta))}{\omega^t} & 1 & \frac{\sin(\omega^t \Delta)}{\omega^t} \\ 0 & \sin(\omega^t \Delta) & 0 & \cos(\omega^t \Delta) \end{bmatrix} \quad (6)$$

While the transition matrix D and noise vector $u^t(k)$ remains unchanged as in (5).

2.2. Sensing model

Each sensor s provides two distinct sets of observations: the first one consists of the measurements generated by the detected targets and the second one consists of false detections from the clutter.

The measurements correspond to the range and bearing measurements obtained using the Euclidean distance and trigonometric law, which yield the following model:

$$z_k^{t,s} = h^s(x_{k|k}^t) + w^s(k) = \begin{bmatrix} \sqrt{(\alpha^t)^2(k) + (\xi^t)^2(k)} \\ \arctan\left(\frac{\xi^t(k)}{\alpha^t(k)}\right) \end{bmatrix} + w^s(k) \quad (7)$$

With $w^s(k) \sim N(0, Q^s)$.

Noise realizations $w^t(k)$ and $u^t(k)$ are assumed independent.

3. DATA ASSOCIATION

3.1 Cheap Joint Probabilistic Data Association Filter

When several targets are tracked, the measurement-target association is often uncertain. Namely, given an observation from a sensor s , it is unclear, for instance, whether this observation is originated from target t_1 or target t_2 , yielding either $z^{t_1,s}$ or $z^{t_2,s}$, creating a confusion regarding which target model to use to update the estimation. In order to deal with such situation, a rational is to quantify the probability that a given measurement is originated from a specific target, which will then be included in the estimation process. One of the most popular methods for this purpose is the Joint Probabilistic Data Association Filter (JPDAF). A detailed derivation of JPDAF can be found in [2]. The Kalman update expression reads as

$$\hat{x}_{k|k}^{t,s} = x_{k|k-1}^{t,s} + \mathbf{K}^{t,s}(k) v^{t,s}(k) \quad (8)$$

where $x_{k|k-1}^{t,s}$ is the predicted state given by $F^t(k-1) \cdot x_{k-1|k-1}^{t,s}$, $\mathbf{K}^{t,s}(k)$ is the Kalman gain and $v^{t,s}(k)$ is the combined innovation given by

$$v^{t,s}(k) = \sum_{i=1}^{N^s(k)} \beta_i^{t,s}(k) v_i^{t,s}(k) = \sum_{i=1}^{N^s(k)} \beta_i^{t,s}(k) \left[z^{i,s}(k) - h^s(x_{k|k-1}^{t,s}) \right] \quad (9)$$

where $\beta_i^{t,s}(k)$ is the probability of associating track t with measurement i received on the sensor s . $N^s(k)$ is the number of validated measurements for all tracks on the sensor s . In the standard JPDA, the association probabilities are calculated by considering every possible hypothesis as to the association of the new measurements with existing tracks, which increases substantially the computational cost of the algorithm. A *ad hoc* JPDA formulation, commonly known as the cheap JPDA (CJPDA), providing a good approximation of the association probabilities $\beta_i^{t,s}(k)$, was pointed out by Fitzgerald as [4]:

$$\beta_i^{t,s}(k) = \frac{G_i^{t,s}(k)}{\sum_i G_i^{t,s}(k) + \sum_t G_t^{t,s}(k) - G_i^{t,s}(k) + B} \quad (10)$$

with B is a constant which depends on clutter density (usually with $B = 0$, the algorithm works well) and $G_i^{t,s}(k)$ is proportional to the Gaussian likelihood function indicating the closeness of fit of target t with i th observation, and given by:

$$G_i^{t,s}(k) = \frac{1}{\sqrt{|S_i^{t,s}(k)|}} \exp \left[-\frac{1}{2} \left(z_k^{i,s} - h^s(x_{k|k-1}^{t,s}) \right)^T [S_i^{t,s}(k)]^{-1} \left(z_k^{i,s} - h^s(x_{k|k-1}^{t,s}) \right) \right] \quad (11)$$

The gain $\mathbf{K}^{t,s}(k)$ is given by

$$\mathbf{K}^{t,s}(k) = P^t(k|k-1)H(k)^T \left[H(k)P^t(k|k-1)H(k)^T \right]^{-1} \quad (12)$$

Where H denotes the Jacobian of measurement gain matrix $h^s(.,.)$ and the error covariance is

$$P^t(k|k) = \beta_0^t(k)P^t(k|k-1) + H(k)^T \left[1 - \beta_0^t(k) \right] P_G^t(k|k) + P_O^t(k|k) \quad (13)$$

$$P_G^t(k|k) = \left[I - P^t(k|k-1)H(k) \right] P^t(k|k-1) \quad (14)$$

$$P_O^t(k|k) = W^t(k) \left[\sum_{i=1}^{N^t} \beta_i^t(k) v_i^t(k) (v_i^t(k))^T - v_i^t(k) (v_i^t(k))^T \right] (W^t(k))^T \quad (15)$$

$$P^{t,s}(k|k-1) = A^t(k-1) P^{t,s}(k-1|k-1) (A^t(k-1))^T + Q^t \quad (16)$$

Where A^t stands for the Jacobian of the non-linear state transition functional F^T .

3.2. Particle filter

Unlike other nonlinear estimation methods like extended Kalman filtering, particle filtering (PF) allows us to solve the online estimation problems without any assumption about the dynamics and shape of the conditional density (Gaussian assumption, unimodality). The core idea in particle filter is to represent the required conditional density of the states as a set of random samples called particles, rather than as a function over state space, and, to each particle is associated a weight. More formally, given the cumulative sets of states and measurements up to time k , denoted by $Z_k = \{z_1, z_2, \dots, z_k\}$, the key is to estimate the posterior probability $p(X_k | Z_k)$. The latter is represented by a set of N weighted samples[5]:

$$p(X_k | Z_k) \approx \sum_{i=1}^N w_k^i \delta(X_k - X_k^i) \quad (17)$$

Where X_k^m denotes the m -th particle and w_k^m is the associated importance weight. For normalization purpose, the weights sum up to one. $\delta(\cdot)$ denotes the standard delta function. Especially, it has been shown that as N tends towards infinity, (17) approaches the true posterior distribution $p(X_k | Z_k)$.

Particle filtering is a method of approximately solving the prediction and update equations given in expressions (18) and (19), respectively.

$$p(X_{k+1} | Z_k) = \int p(X_{k+1} | X_k) p(X_k | Z_k) dX_k \quad (18)$$

And

$$p(X_{k+1} | Z_{k+1}) = \frac{p(z_{k+1} | X_{k+1}) p(X_{k+1} | Z_k)}{p(z_{k+1} | Z_{k+1})} \quad (19)$$

Where $p(z_{k+1} | Z_{k+1})$ is the likelihood function indicating the model of the sensor.

Samples are used to represent the density and to propagate it through time. The prediction equation (18) is implemented by proposing new particles from the existing particles using a model of state dynamics and the measurements. The update equation (19) is implemented by assigning a weight to each of the particles that have been proposed using the measurements and the model of state dynamics.

The particle filter provides an estimation of the posterior in (19) in three main steps: sampling, weighting and re-sampling. The sampling step consists of taking samples (particles) from the so-called dynamic prior distribution $p(X_k | Z_k)$. Next, in the weighting step, the resulting particles are weighted by the likelihood term $p(Z_k | X_k)$. Finally a re-sampling step is usually applied to avoid the degeneracy phenomenon. For this purpose, many proposals have been reported in the literature, e.g., importance resampling, residual resampling and minimum variance sampling [8]. In this paper, the residual resampling is used in all our experiments. Detailed computations are reported in Table 1 and Table 2.

The recursive estimate for the importance weights of particle i (as in (17)) is given by:

$$w^i(k) = w^i(k-1) \frac{p(z_k | X_k^i) p(X_k^i | X_{k-1}^i)}{q(X_k^i | X_{k-1}^i, z_k)} \quad (20)$$

Where $q(X_k^i | X_{k-1}^i, z_k)$ is the importance density used to generate the particles. A common choice of such distribution is the prediction distribution $p(x_k | x_{k-1})$. In the latter case, a commonly known *bootstrap* filter or Sequential Importance Resampling SIR [8], where the re-sampling stage is applied at every index time, can be employed. In this respect, samples are generated as

$$X_k^i \sim p(X_k | X_{k-1}^i) \quad (21)$$

With associated weight:

$$w^i(k) = p(z_k | X_k^i) \quad (22)$$

These weights are normalized and then a resampling stage is carried out in order to discard particles with small weights that cause, what is referred to as, a degeneracy problem. The resampling is performed by drawing a new set of particles $\{X_k^i\}$, with replacement from old particles $\{X_{k-1}^i\}$.

Finally, the estimated state is computed by averaging over the set of all particle estimates such as

$$\hat{X}_k \approx \sum_{i=1}^N w^i(k) X_k^i \quad (23)$$

$$P_k \approx \sum_{i=1}^N w^i(k) (X_k^i - \hat{X}_k)(X_k^i - \hat{X}_k)^T \quad (24)$$

3.3 Multiple Model Particle Filter (MMPF)

As real targets rarely obey a single kinematic model, a set of discrete number of models were often used instead. This is particularly true when the target changes its maneuver to deal with some new constraints. In order to accommodate this situation, the estimation process should account for the possibility of occurrence of all possible modes (models). Multiple-model based approach provides an example of solution for this purpose. In this paper, we advocate the use of multiple model particle filter (MMPF) highlighted in [7] (see also [16]) due to its simplicity and its proven performances. The key idea of the MMPF is to augment the state vector of target t to include both a continuous-valued part of target kinematic variables (position, velocity) and a discrete-valued part referring to the dynamic mode variable that determines the dynamic model in effect from time increment $k-1$ to time increment k . More formally, the augmented state vector reads as

$$y^t(k) = \begin{bmatrix} x_k^t \\ r^t(k) \end{bmatrix} \quad (25)$$

With $r^t(k) \in \{1, 2, \dots, l\}$ where l stands for the total number of model switches employed. In this study, we mainly concentrate on two-model switches; namely, the constant velocity and the coordinated turn models examined in our simulation setting.

Strictly speaking, the MMPF provides a sequential Monte Carlo approximation of the expressions (23) and (24) representing the prediction and update states, respectively [7].

$$p(x_k^t, r^t(k)=j | Z_{k-1}) = \sum_i \pi_{ij} \int p(x_k^i | x_{k-1}^i, r^t(k)=j) p(x_{k-1}^i, r^t(k-1)=i | Z_k) dx_{k-1}^i \quad (26)$$

$$p(x_k^t, r^t(k)=j | Z_k) = \frac{p(z_k | x_{k-1}^t, r^t(k)=j) p(x_k^t, r^t(k)=j | Z_{k-1})}{\sum_j \int p(z_k | x_k^t, r^t(k)=j) p(x_k^t, r^t(k)=j | Z_{k-1}) dx_k^t} \quad (27)$$

The first stage of the MMPF algorithm is to generate the random set of particles pertaining to the mode variable $\{r^t(k)\}_{j=1}^N$ based on previous estimates $\{r^t(k-1)\}_{j=1}^N$ and transition matrix $\Pi = [\pi_{ij}]$. This is illustrated in algorithm shown in Table 2 of model transition. The latter is implemented using the rule that if $(r^t(k-1))^i = m$, then $(r^t(k))^i = n$ with probability π_{mn} . More formally, this entails [7]:

If $(r^t(k-1))^i = m$ and $u_n \sim U[0,1]$, then $(r^t(k))^i$ is set to ν where $\nu \in S$ such that $\sum_{j=1}^{\nu-1} \pi_{mj} < u_n < \sum_{j=1}^{\nu} \pi_{mj}$.

Initially ($k=0$), the generation of samples $\{r^t(k)\}_{j=1}^N$ is accomplished using uniform probability distribution. More formally, let $u_n \sim U[0,1]$, then assign $(r^t(k-1))^i = \zeta$ if $\frac{\zeta-1}{l} < u_n \leq \frac{\zeta}{l}$.

[By abuse, one shall denote that $(r^t(k))^i \sim U[0,1]$].

The second stage performs a regime conditioned sequential importance sampling filtering where the optimal regime conditioned importance density was used as the sampling distribution given by

$$q(x_k^t | x_{k-1}^i, r^t(k), z_k)_{opt} = p(x_k^t | x_{k-1}^t, r^t(k), z_k) \quad (28)$$

The detailed implementation is again reported in the pseudo code description of Table 1 and Table 2 and extensively detailed in next section as the weight and particle estimates are ultimately linked to the data association approach employed.

3.4. Cheap Joint Probabilistic Data Association Multiple Model Particle Filter CJPDA-MMPF

This section summarizes the combination of the CJPDA and MMPF for target state estimation. Typically, the CJPDA recursively updates the marginal filtering distribution for each target, while the MMPF is used to account for possible changes in kinematic model of the target. On the other hand, the use of the particle filter allows us to generate a large number of samples that approximate each probability distribution. Therefore, the estimation process will be carried out at each sample and then averaged over all samples to get the final target state estimate. Namely, one notices the following:

- An augmented state vector is formed by joining the target state vector to the mode variable required in MMPF and whose estimation is accomplished at each particle level.
- The CJPDA is next used in order to estimate the joint probability association target/measurement as well as the state estimation for each particle.

- The state vector and variance-covariance matrix pertaining to each target are obtained by averaging over all samples.
- The number of particles N_p is taken constant. In our simulations, setting $N_p = 500$ was found to work well.

Table 1 summarizes the CJPDA-MMPF to joint multiple target tracking and classification using the s^{th} sensor.

Table 1. Cheap JPDA Multiple Model Particule Filtre pseudo code

<p>1. Initialization: Set $k = 0$</p> <ul style="list-style-type: none"> • Generate N_p samples $\{\hat{x}_n^{t,s}(k)\}_{n=1}^{N_p}$ for all targets from $p(x^{t,s}(k))$ with covariance matrix $\hat{P}_n^{t,s}(k) = \hat{P}^{t,s}(k)$ • Generate N_p samples $\{r_n^{t,s}(k)\}_{n=1}^{N_p}$ for all targets from $p(r^{t,s}(k))$. <p>2. Increase k by one increment</p> <p>3. Local filter (see Table 3)</p> <p>For each sensor s ($s = 1, 2, \dots, S$) do</p> <p>$[\hat{x}_n^{t,s}(k), r_n^{t,s}(k), \hat{P}_n^{t,s}(k), x_p^{t,s}(k), P_p^{t,s}(k)] =$</p> <p>Local filter$\left(\left[\left\{\hat{x}_n^{t,s}(k-1)\right\}_{n=1}^{N_p}, \left\{r_n^{t,s}(k-1)\right\}_{n=1}^{N_p}, \left\{\hat{P}_n^{t,s}(k-1)\right\}_{n=1}^{N_p}, R, Q, D\right]\right)$</p>
--

Table 2. Model transition

<p>For $i = 1 : M$</p> <ul style="list-style-type: none"> ○ $c_i(0) = 0$ ○ For $j = 1 : M$ <li style="padding-left: 20px;">$c_i(j) = c_i(j-1) + \Pi_{i,j}$ ○ End <p>End</p> <p>For $n = 1 : N_p$</p> <ul style="list-style-type: none"> ○ Draw $u_n \sim U[0,1]$ ○ Set $i = r_{k-1}^n$ ○ $m = 1$ ○ While $c_i(m) < u_n$ <li style="padding-left: 20px;">$m = m + 1$ ○ End while ○ Set $r_k^n = m$ <p>End</p>

Table 3. Local filter

$\left[\hat{x}^{t,\varepsilon}(k), r^{t,\varepsilon}(k), \hat{P}^{t,\varepsilon}(k), x_p^{t,\varepsilon}(k), P_p^{t,\varepsilon}(k) \right] =$ <p>Local filter $\left(\left[\left\{ \hat{x}_n^{t,\varepsilon}(k-1) \right\}_{n=1}^{N_p}, \left\{ r_n^{t,\varepsilon}(k-1) \right\}_{n=1}^{N_p}, \left\{ \hat{P}_n^{t,\varepsilon}(k-1) \right\}_{n=1}^{N_p}, R, Q, D \right] \right)$</p> <p>For each target t ($t = 1, 2, \dots, N$)</p> <ul style="list-style-type: none"> • Generate a random set $\{r_n^{t,\varepsilon}(k)\}_{n=1}^{N_p}$ based on $\{r_n^{t,\varepsilon}(k-1)\}_{n=1}^{N_p}$ and the transition probabilities matrix $\mathbf{\Pi}$ according to Table 2. • For each particle n ($n = 1, 2, \dots, N_p$) <ul style="list-style-type: none"> ◦ Compute $\{\beta_i^{t,\varepsilon}(k)\}_{i=1}^{N^s(k)}$ from $r_n^{t,\varepsilon}(k)$, $\hat{x}_n^{t,\varepsilon}(k-1)$ and validated measurements $\{z^{i,\varepsilon}(k)\}_{i=1}^{N^s(k)}$ according to (10) ◦ Evaluate $x_{p,n}^{t,\varepsilon}(k) = F^t(r_n^{t,\varepsilon}(k))\hat{x}_n^{t,\varepsilon}(k-1)$ $P_{p,n}^{t,\varepsilon}(k) = F^t(r_n^{t,\varepsilon}(k))\hat{P}_n^{t,\varepsilon}(k-1)\left[F^t(r_n^{t,\varepsilon}(k))\right]^T + DQ^sD^T$ $H(x_{p,n}^{t,\varepsilon}(k))$ is the Jacobian of h^s, given in (6), evaluated at $x_{p,n}^{t,\varepsilon}(k)$ $S_n^{t,\varepsilon}(k) = R^s + H(x_{p,n}^{t,\varepsilon}(k))P_{p,n}^{t,\varepsilon}(k)\left[H(x_{p,n}^{t,\varepsilon}(k))\right]^T$ $K_n^{t,\varepsilon}(k) = P_{p,n}^{t,\varepsilon}(k)\left[H(x_{p,n}^{t,\varepsilon}(k))\right]^T\left[S_n^{t,\varepsilon}(k)\right]^{-1}$ $v_i^{t,\varepsilon}(k) = z^{i,\varepsilon}(k) - h^s(x_{p,n}^{t,\varepsilon}(k)) \quad \forall i \in \{1, 2, \dots, N^s(k)\}$ $W_n^{t,\varepsilon}(k) = \sum_{i=1}^{N^s(k)} \beta_i^{t,\varepsilon}(k) N(v_i^{t,\varepsilon}(k), S_n^{t,\varepsilon}(k))$ $v^{t,\varepsilon}(k) = \sum_{i=1}^{N^s(k)} \beta_i^{t,\varepsilon}(k) v_i^{t,\varepsilon}(k)$ $\hat{x}_n^{t,\varepsilon}(k) = \hat{x}_{p,n}^{t,\varepsilon}(k) + K_n^{t,\varepsilon}(k)v^{t,\varepsilon}(k)$ $\hat{P}_n^{t,\varepsilon}(k) = \beta_0^{t,\varepsilon}(k)P_{p,n}^{t,\varepsilon}(k) + (1 - \beta_0^{t,\varepsilon}(k))\left[I - K_n^{t,\varepsilon}(k)H(x_n^{t,\varepsilon}(k))\right]P_{p,n}^{t,\varepsilon}(k)$
$+ K_n^{t,\varepsilon}(k) \left[\sum_{i=1}^{N^s(k)} \beta_i^{t,\varepsilon}(k) v_i^{t,\varepsilon}(k) [v_i^{t,\varepsilon}(k)]^T - v^{t,\varepsilon}(k) [v^{t,\varepsilon}(k)]^T \right] [K_n^{t,\varepsilon}(k)]^T$ <ul style="list-style-type: none"> • Normalize $\{W_n^{t,\varepsilon}(k)\}_{n=1}^{N_p}$ and obtain $\{\hat{W}_n^{t,\varepsilon}(k)\}_{n=1}^{N_p}$ • Using the weights $\{\hat{W}_n^{t,\varepsilon}(k)\}_{n=1}^{N_p}$, resample the set $\{\hat{x}_n^{t,\varepsilon}(k), \hat{r}_n^{t,\varepsilon}(k), \hat{P}_n^{t,\varepsilon}(k), \hat{x}_{p,n}^{t,\varepsilon}(k), \hat{P}_{p,n}^{t,\varepsilon}(k)\}_{n=1}^{N_p}$ • Obtain: $\hat{r}^{t,\varepsilon}(k) = \frac{1}{N_p} \sum_{n=1}^{N_p} \hat{r}_n^{t,\varepsilon}(k)$, $\hat{x}^{t,\varepsilon}(k) = \frac{1}{N_p} \sum_{n=1}^{N_p} \hat{x}_n^{t,\varepsilon}(k)$, $\hat{P}^{t,\varepsilon}(k) = \frac{1}{N_p} \sum_{n=1}^{N_p} \hat{P}_n^{t,\varepsilon}(k)$ $x_p^{t,\varepsilon}(k) = \frac{1}{N_p} \sum_{n=1}^{N_p} x_{p,n}^{t,\varepsilon}(k) \text{ and } P_p^{t,\varepsilon}(k) = \frac{1}{N_p} \sum_{n=1}^{N_p} P_{p,n}^{t,\varepsilon}(k)$ <p>$\beta_0^{t,\varepsilon}(k)$ is the probability that all measurements in the validated gate of track t are false.</p>

4. DATA FUSION

The data fusion approach combines data from multiple sensors (and associated databases if appropriate) to achieve improved accuracies and more specific inferences that could not be achieved by the use of only a single sensor. Such data can be combined in various ways, for instance [17,18] by

- Linear combiner
- Combination of posteriors (weights, model significance)
- Product of posteriors (independent information)

Many data fusion architectures have been proposed in the literature depending on the nature of inputs, the expected outcome and general framework for representing uncertainty pervading the data and models [13-14]. In the present paper, we consider two different methods of data fusion

which accommodate the general framework employed in this analysis:

- Federated Kalman filter
- Centralized Kalman Filter

4.1. Federated Kalman filter (FKF)

The FKF is one of the architectures that has received considerable attention as a practical means of decentralization. The FKF differs from the conventional Kalman filter because each measurement is processed in local filters, and the results are combined in a master filter. The local filters run completely independent of each other, providing isolation between filters in the instance of sensor failure. The local filter is represented by the pseudo code given in Table 3, which is applied for each sensor s and all the results are sent to master filter. The pseudo code of FKF fusion method is given in Table 4 below.

Table 4. The pseudo code of CJPDA_MMPF_FKF fusion method.

<p>1. Initialization Set $k = 0$</p> <ul style="list-style-type: none"> • Generate N_p samples $\{\hat{x}_n^{t,s}(0)\}_{n=1}^{N_p}$ for all targets from $p(x^{t,s}(0))$ with covariance matrix $\hat{\mathbf{P}}_n^{t,s}(0) = \hat{\mathbf{P}}^{t,s}(0)$ • Generate N_p samples $\{r_n^{t,s}(0)\}_{n=1}^{N_p}$ for all targets from $p(r^{t,s}(0))$ <p>2. Increase k</p> <p>3. Local Filter (see Table 3)</p> <p>For each sensor s ($s = 1, 2, \dots, S$) do</p> $\left[\hat{x}^{t,s}(k), r^{t,s}(k), \hat{\mathbf{P}}^{t,s}(k), x_p^{t,s}(k), \mathbf{P}_p^{t,s}(k) \right] =$ $\text{Local filter} \left(\left[\left[\{\hat{x}_n^{t,s}(k-1)\}_{n=1}^{N_p}, \{r_n^{t,s}(k)\}_{n=1}^{N_p}, \{\hat{\mathbf{P}}_n^{t,s}(k-1)\}_{n=1}^{N_p}, R, Q, D \right] \right] \right)$ <p>4. Master Filter (Fusion)</p> $\left[\hat{\mathbf{P}}_f(k) \right]^{-1} \hat{x}_f(k) = \sum_{s=1}^S \left[\hat{\mathbf{P}}^{t,s}(k) \right]^{-1} \hat{x}^{t,s}(k) \text{ and } \left[\hat{\mathbf{P}}_f(k) \right]^{-1} = \sum_{s=1}^S \left[\hat{\mathbf{P}}^{t,s}(k) \right]^{-1}$ <p>5. Go to step 2</p>
--

4.2. Centralized Kalman filter (CKF)

This method is based on centralized Kalman filter equation in order to fuse the estimates obtained from each local filter. As demonstrated in [19] and provided there is no feedback loop, the performance of the distributed track fusion for the estimate and its variance covariance matrix are the same as that of the corresponding centralized track fusion. So, the centralized equations used are the highlighted in Table 5.

Table 5. The pseudo code of CJPDA_MMPF_CKF fusion method.

<p>1. Initialization: Set $k = 0$</p> <ul style="list-style-type: none"> • Generate N_p samples $\{\hat{x}_n^{t,s}(0)\}_{n=1}^{N_p}$ for all targets from $p(x^{t,s}(0))$ with covariance matrix $\hat{\mathbf{P}}_n^{t,s}(0) = \hat{\mathbf{P}}^{t,s}(0)$ • Generate N_p samples $\{r_n^{t,s}(0)\}_{n=1}^{N_p}$ for all targets from $p(r^{t,s}(0))$ <p>2. Increase k</p> <p>3. Local filter (Table 3)</p> <p>For each sensor s ($s = 1, 2, \dots, S$) do</p> $[\hat{x}^{t,s}(k), r^{t,s}(k), \hat{\mathbf{P}}^{t,s}(k), x_p^{t,s}(k), \mathbf{P}_p^{t,s}(k)] =$ $\text{Local filter} \left(\left[\left\{ \hat{x}_n^{t,s}(k-1) \right\}_{n=1}^{N_p}, \left\{ r_n^{t,s}(k) \right\}_{n=1}^{N_p}, \left\{ \hat{\mathbf{P}}_n^{t,s}(k-1) \right\}_{n=1}^{N_p}, R, Q, D \right] \right)$ <p>4. Fusion step using centralized Kalman equation</p> $x_{f,p}(k) = \mathbf{F}^t(r^t(k))x_f(k-1)$ $\hat{\mathbf{P}}_{f,p}(k) = \mathbf{F}^t(r^t(k))\hat{\mathbf{P}}_f(k-1)[\mathbf{F}^t(r^t(k))]^T + \mathbf{D}Q^s\mathbf{D}^T$ $[\hat{\mathbf{P}}_f(k)]^{-1} = [\hat{\mathbf{P}}_{f,p}(k)]^{-1} + \sum_{s=1}^S [\hat{\mathbf{P}}^{t,s}(k) - \hat{\mathbf{P}}_p^{t,s}(k)]^{-1}$ $[\hat{\mathbf{P}}_f(k)]^{-1} \hat{x}_f(k) = [\hat{\mathbf{P}}_{f,p}(k)]^{-1} \hat{x}_{f,p}(k) + \sum_{s=1}^S \left([\hat{\mathbf{P}}^{t,s}(k)]^{-1} \hat{x}^{t,s}(k) - [\mathbf{P}_p^{t,s}(k)]^{-1} x_p^{t,s}(k) \right)$ <p>5. Go to step 2</p>

5. SIMULATION

The above mentioned algorithm is applied to jointly tracking and classifying two crossing targets. The targets trajectories are assumed as follows (the sampling period $\Delta = 6s$):

- Target 1: It starts from $[1000m \ 5m/s \ 1500m \ 5m/s]^T$ at time $k = 0s$, the target runs for 50Δ with constant velocity model. Next, the target executes a coordinated turn rate model with $0.15^\circ/s$ coordinated for 50Δ .
- Target 2: It starts from $[8171.5m \ -10m/s \ 1398.5m \ 2m/s]^T$ at time $k = 0s$, the target runs for 50Δ with constant velocity model. Next, it executes a coordinated turn rate model with $-0.15^\circ/s$ coordinated for 50Δ .
- The crossing time is $k = 75\Delta$.

The initial state vectors, of the MMPF, are 0.9 of the true initial state with covariance matrix $diag(100 \ 1 \ 100 \ 1)$. The number of particles is $N_p = 500$ and the number of Monte Carlo runs is 100.

The measurement and the state covariance matrices are:

$$R = \begin{bmatrix} 150^2 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } Q = \begin{bmatrix} 0.05^2 & \\ & 0.05^2 \end{bmatrix}$$

The mode switching probability matrix is:
$$= \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}$$

On the other hand, we also considered the possibility of presence of clutter, exemplified by a fixed number of false alarms, randomly generated within the ellipsoid region pertaining to the true observation. The number of the randomly generated false alarms provides a rough indication on the density of the clutter in the sense that the larger the number of false alarms, the higher the clutter density.

In the present simulation both the federated and centralized fusion methods are compared to the sequential method. In the latter, the local filter uses measurements of the first sensor to update the state and its covariance. Then the output of the filtering of the first sensor will be used as an input of the second filter which uses the measurements received from the second sensor, and so on.

Figure 1 presents the true tracks of the two targets as well as the estimated targets in the absence of false alarms. While Figure 2 highlights the performances of the tracking of Target 1 and Target 2 with respect to root mean squared error (RMSE) metric calculated as the squared root of the sum of the squared difference between the true position and the estimated position of the target at each time increment. Similarly, Figure 3 and 4 provide an assessment of the quality of the tracking when four additional alarms have been randomly generated within the ellipsoid region of each true observation and added to the set of measurements, which would add an extra difficulty to the data association problem handled by the CJPDAF. The FKF and CKF fusion schemes are compared to the standard sequential Kalman filtering. In both cases the results correspond to an average of 500 Monte Carlo simulations. Figure 5 and 6 provide an evaluation of the performance of the tracking when the clutter is made of six false alarms at each time increment, which increases the density of the clutter. In Figures 7 and 8, the quality of the tracking when using 12 false alarms is highlighted.

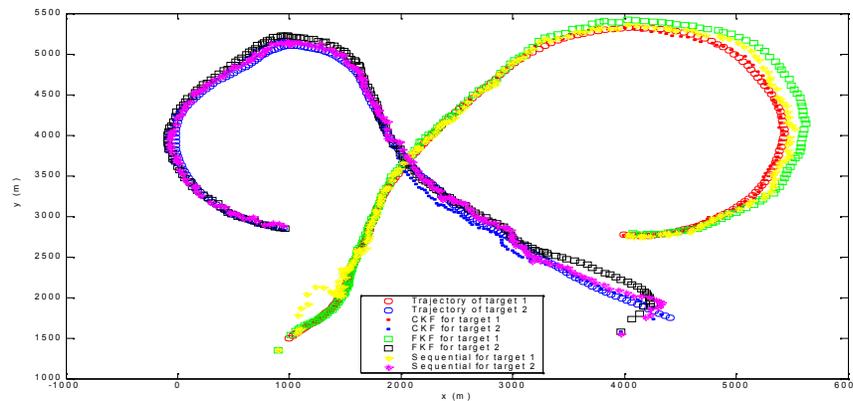


Figure 1. True and estimated target positioning using different data fusion methods in the absence of false alarms.

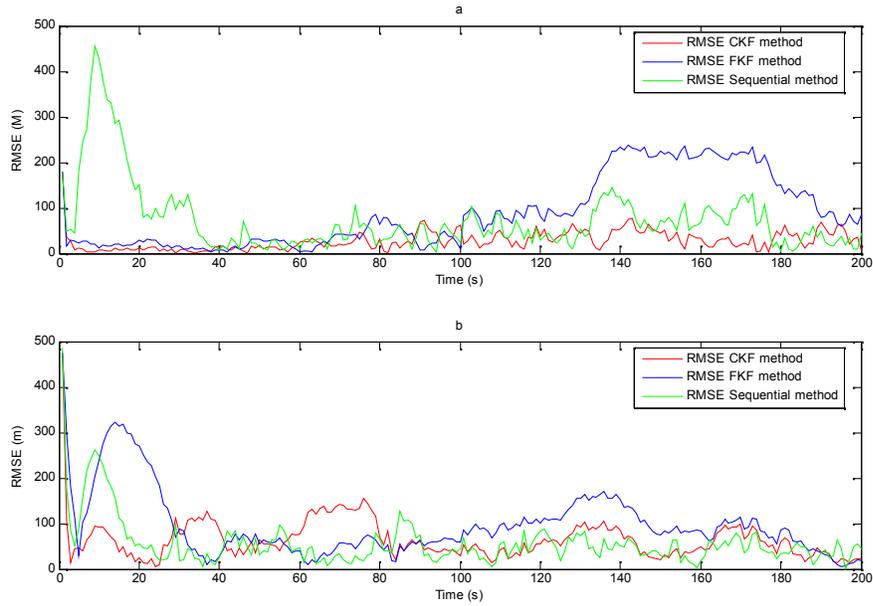


Figure 2. (a) RMSE of different tracks of target 1 using different data fusion methods. (b) RMSE of different tracks of target 2 using different data fusion methods without false alarms.

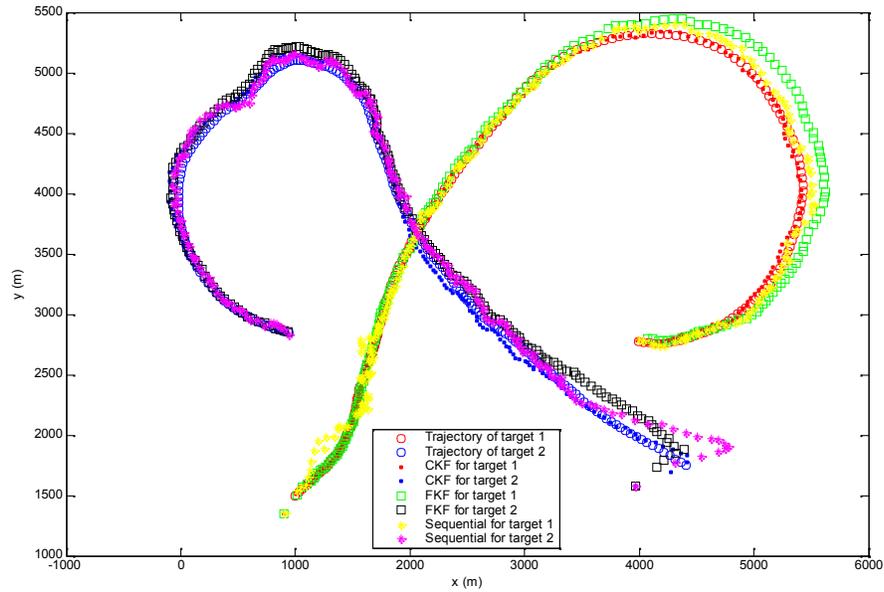


Figure 3. True and estimated target positioning using different data fusion methods when 4 false alarms were added at each time increment.

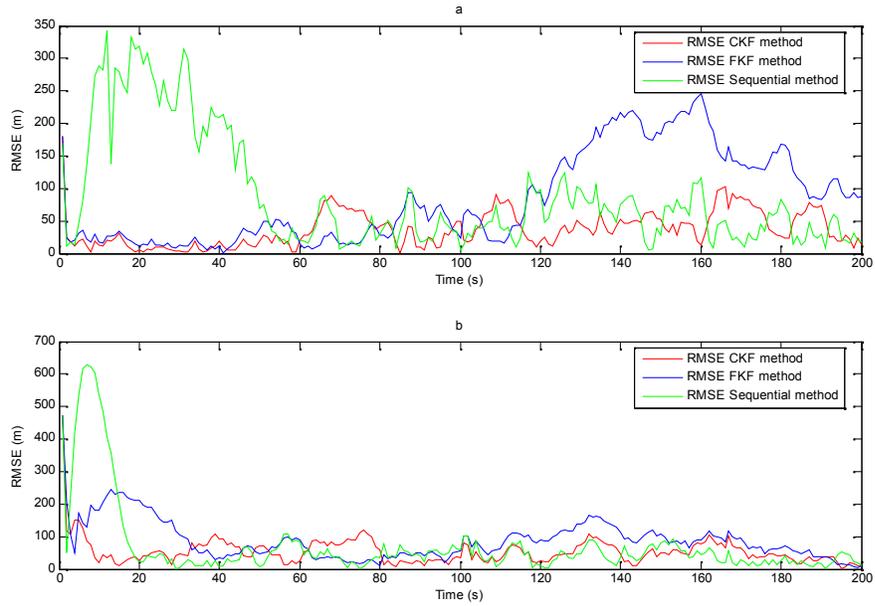


Figure 4. RMSE of different tracks of (a) Target 1 and (b) Target 2, using different data fusion methods when four false alarms were added at each time increment.

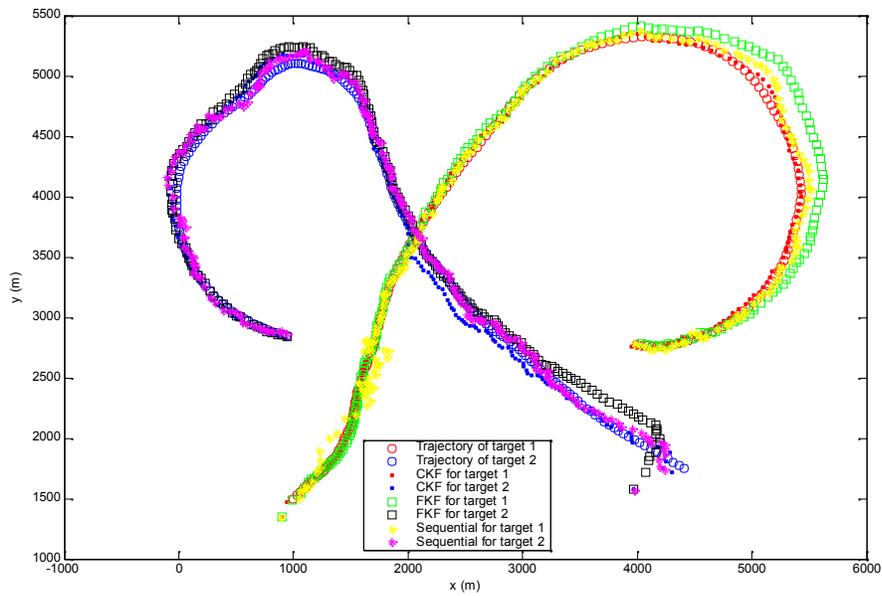


Figure 5. True and estimated target positioning using different data fusion methods when 8 false alarms were added at each time increment.

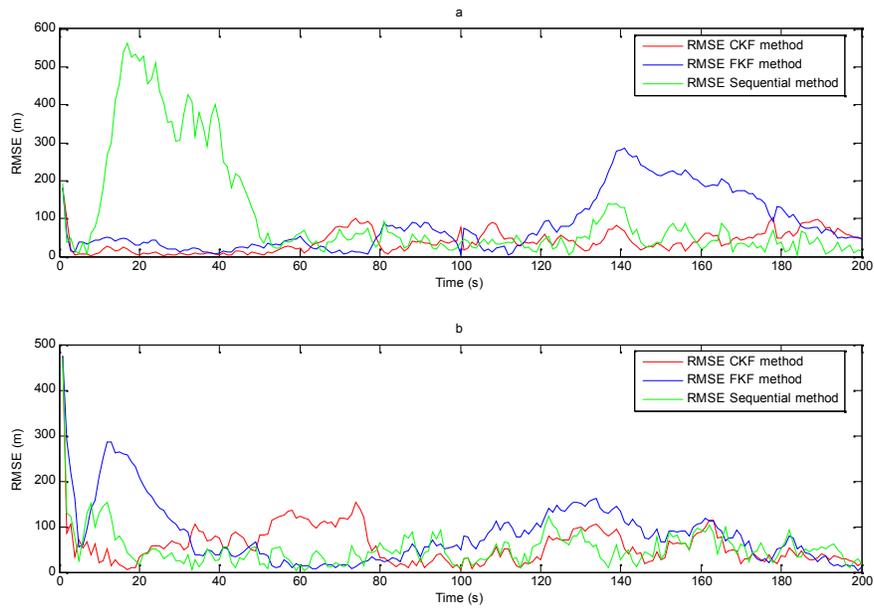


Figure 6. RMSE of different tracks of (a) Target 1 and (b) Target 2, using different data fusion methods when eight false alarms were added at each time increment.

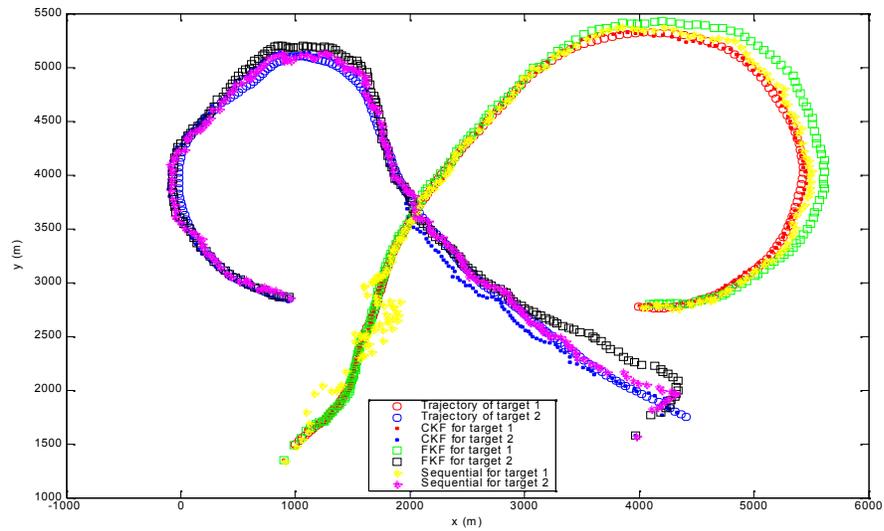


Figure 7. True and estimated target positioning using different data fusion methods when 12 false alarms were added at each time increment.

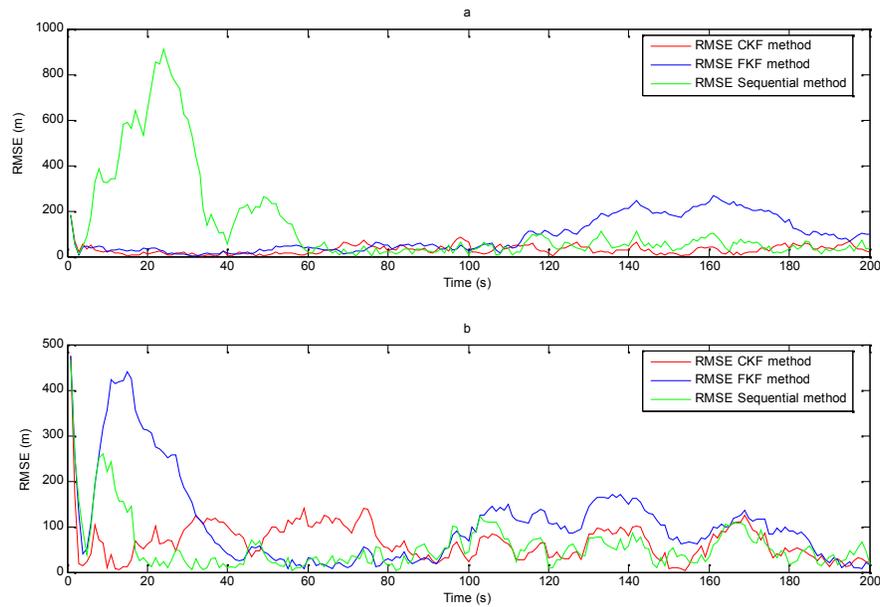


Figure 8. RMSE of different tracks of (a) Target 1 and (b) Target 2, using different data fusion methods when 12 false alarms were added at each time increment.

As it can be seen from figure 1, 3, 5 and 7, the monitoring of the target positioning is in overall well satisfactory as no target was lost and the algorithms discriminate well among the various tracks. On the other hand, as the clutter density increases, it is rational that the tracking performance with respect to RMSE metric degrades due to the effect of false alarms made close enough to other target measurement, which makes the data association difficult. Also, comparing the three employed fusion methods (CKF, FKF and SKF), the central Kalman filter appears to be providing better results in overall while the sequential Kalman filtering exhibits the worse tracking performance. Moreover in order to provide a more quantitative overall evaluation Table 6 summarizes the quality of the tracking by evaluating the RMSE across the whole trajectory while distinguishing the results in the first half of trajectory where model M1 (constant velocity model) is performed and the other one when model M2 (coordinated turn model) is in action.

Table 6. Comparison of RMSE of tracking algorithms over whole trajectory.

		No false alarms		4 false alarms		8 false alarms		12 false alarms	
		M1	M2	M1	M2	M1	M2	M1	M2
CKF	Target 1	19.42	35.47	26.50	46.99	27.68	49.47	26.78	31.82
	Target 2	75.77	52.38	59.42	47.97	68.62	49.13	78.11	54.85
FKF	Target 1	28.19	149.01	30.80	134.92	37.59	128.58	32.62	145.46
	Target 2	100.42	91.15	90.30	84.63	82.78	83.99	113.29	99.64
SEQ	Target 1	84.17	61.58	121.48	52.87	165.06	45.36	216.02	50.05
	Target 2	67.17	44.76	99.63	39.93	50.53	51.93	55.83	54.80

As it can be noticed from the above illustrations, both algorithms CJPDA-MMPF-CKF and CJPDA-MMPF-FKF provide reasonably fair track estimations as no track divergence was observed through the set of Monte Carlo simulations performed in this experiment. A slightly better performance is highlighted by the CKF-based fusion scheme on average as compared to Federated Kalman filtering based fusion. Strictly speaking, this can easily be explained by the

intuition that the CKF handles better the correlation among the various inputs as it has been noticed elsewhere [14, 18].

6. CONCLUSION

In this paper, the problem of tracking several targets in cluttered environment is addressed using a network of sensors and an algorithm combining cheap joint probabilistic data association (CJPDA) with multiple model particle filter (MMPF). It is assumed that the motion model of any target belongs to one of several classes. The MMPF is used to perform nonlinear filtering with switching dynamic models and the CJDAF is used to estimate the joint measurement-target probability association. Two fusion schemes consisting of federated Kalman filtering and centralized Kalman filtering are compared to standard sequential Kalman filtering. A simulation platform of two crossing targets employing both constant velocity model and coordinated turn model was constructed. The developed CJPDA-MMPF has been tested using a set of Monte Carlo simulations with different clutter densities. Standard root mean square error was used as a metric for evaluating the performance of the different fusion schemes. The results shown intuitively appealing outcomes in overall and no target estimation divergence has been noticed. Furthermore, a comparison of the three fusion schemes shows a slight outperformance of the CKF over the FKF, which can partially be explained by the accounting of the different interdependence among the inputs in case of CKF while FKF implicitly assumes independence in this case.

REFERENCES

- [1] Karlsson, E. (2000) Decentralized tracking with feedback: adaptive sample rate and IMM. Master thesis, the Division of Automatic Control, Linkping University, Sweden.
- [2] Bar-Shalom, Y.; Fortmann, T. (1988) Tracking and data association. In Mathematics in Science and Engineering; Academic Press: London, UK.
- [3] Bar-Shalom, Y.; Xiao-Rong, L. (1993) Estimation and Tracking: Principles Techniques and software. Artech Hous: Boston, USA.
- [4] Fitzgerald, R.J. (1999) Development of practical PDA logic for multitarget tracking by microprocessor. In Multitarget Multisensor tracking: Advanced Application; Artech House: Boston, USA; pp. 1-23.
- [5] Oussalah M., J. De Schutter (2002), Hybrid Fuzzy Probabilistic Data Association Filter and Joint Probabilistic Data Association Filter, Information Science, Vol 14(2), pp. 195-226
- [6] Oussalah M. Z. Mesaoudi, A. Ouldali, (2010) Track-To-Track Measurement Fusion Architectures and Correlation Analysis, Journal of Universal Computer Science, Vol. 16(1), 37-61
- [7] Branko, R.; Sanjeev, A.; Neil, G. (2004) Beyond the Kalman filter: Particle filter for tracking application. Artech Housse: Boston.
- [8] Doucet, A.; De Freitas, N.; Gordon, N. (2001) Sequential Monte Carlo Methods in Practice. Springer, Heidelberg.
- [9] Nash, J.M. Optimal Allocation of Tracking Resources. (1977) IEEE Conference on Decision and Control, Vol. 1, 177-1180.
- [10] Kastella, K.; Musick, S. (1996) Comparison of Sensor Management Strategies for Detection and Classification. National Symposium on Sensor Fusion. Naval Postgraduate School. Monterey. CA.
- [11] McIntyre, G.A. (1998) A Comprehensive Approach to Sensor Management and Scheduling. PhD Dissertation, George Mason University, Fairfax, VA, 1998.
- [12] Dils, L.; Zeng, Y.Y. (2003) Sensor Management Based on Cross-entropy. In: Proceedings of Instrumentation and Measurement Technology Conference Vail, USA, 1555-1558.
- [13] Lee, T. (2003) Centralized Kalman Filter with Adaptive Measurement Fusion: its Application to a GPS/SDINS Integration System with an Additional Sensor. Journal of Control, Automation, and Systems, Vol. 1(4), 444-452.
- [14] Yukun, C.; Xicai, S.; Zhigang, L. (2007), Research on Kalman filter based multisensor data fusion. Journal of Systems engineering and Electronics, Vol. 18, 497-502.

- [15] Messaoudi, Z.; Oussalah, M.; Ouldali, A. (2007) Data association for maneuvering targets using controlled based JPDAF particle filter. In Proceedings of IEEE SMC Ireland and United Kingdom Chapter, Dublin, Ireland.
- [16] Wang, X.; Wang, S. (2007); Ma, J.J. An Improved Particle Filter for Target Tracking in Sensor Systems, Sensors, Vol. 7, 144-156
- [17] Starr, A., and Desforges, M. (1998) Strategies in data fusion - sorting through the tool box. In Proceedings, EuroFusion98 International Conference on Data Fusion (editors Bedworth and O'Brien), pp. 85-90.
- [18] Steinberg A. N. and C. L. Boman (2001), Revision to the JDL data fusion model, In: Handbook of Multisensor Data Fusion, D. L. Hall and Llinas (eds), CRC, London, 45-67
- [19] Zhu, Y.; You, Z.; Zhao, J.; Zhang, K.; Rong Li, X. (2001) The optimality for the distributed Kalman filtering fusion with feedback. Automatica, 37, 1789-1493.

Authors

Zahir Messaoudi was born in Bejaia, Algeria, in 1976. He received the diplôme d'ingénieur in electrical engineering in 2002, from the Military Polytechnic School. He also received the diplôme de magister in electronics systems from the Military Polytechnic School. He is currently pursuing the Ph.D degree in signal processing at the same school and a visiting PhD student at university of Birmingham. His main research interests are in statistical signal and array signal processing and Kalman filtering.

Mourad Oussalah holds a PhD in Robotics and Information Fusion from University of Evry Val d'Essonne in France in 1998, and he holds several academic/postdoctoral positions at University Val d'Essonne, KU Leuven Belgium and City University of London. Since 2003 he is with university of Birmingham. He is actively working in information fusion, data mining and wireless systems where he published more than 130 international publications between conferences and journal papers and supervised more than 10 PhD students in the area. He is a Senior member IEEE and Fellow of Royal Statistical Society.

Abdelaziz Ouldali was born in Mostaganem, Algeria, in 1969. He received the diplôme d'ingénieur in electrical engineering in 1992, from the Military Polytechnic School, and the Ph.D. degree in 1998 from the University of Paris XI Orsay (Paris, France). He joined the Military Polytechnic School, where he became an Associate Professor in October 1998. His current research interests are in the areas of parameter estimation, time-frequency analysis and statistical array processing.