Sciences & Technologie **B** – **N°20**, Décembre (2003), pp. 20-27.

ADAPTIVE TRACKING OF MULTIPLE MANOEUVRING TARGETS IN CLUTTER USING THE IMM AND THE JPDAF

Received 30/06/2002 – Accepted 18/10/2003

Abstract

An algorithm for tracking multiple manoeuvring targets in a cluttered environment is proposed. This algorithm combines the Joint Probability Data Association Filter (JPDAF) and the interacting Multiple Model (IMM) algorithm and uses an adaptive update time. A modified version of the Van Keuk method has been used to adaptively calculate the update time in the resulting algorithm, called the Adaptive IMMJPDAF (AIMMJPDAF). The performance of the proposed algorithm is assessed via Monte Carlo simulation and compared to that of the adaptive IMMJPDAF that uses the original Van Keuk method and the IMMJPDAF that uses a constant update time.

***Keywords**: Radar, Tracking manoeuvring targets, Data association, Variable update time, IMM, JPDAF.*

Résumé

Un algorithme pour la poursuite de plusieurs cibles manœuvrantes dans du fouillis est proposé. Cet algorithme combine le filtre probabiliste conjoint d'association des données (JPDA.F) et l'algorithme à Modèles Multiples Interagissant (IMM) et utilise un temps de mise à jour adaptatif. Une version modifiée de la méthode de Van Keuk est utilisée pour calculer adaptativement le temps de mise à jour dans l'algorithme résultant, appelé Adaptatif IMMJPDAF (AIMMJPDAF). Les performance de cet algorithme ont été comparées via des simulations de Monte Carlo avec celles de l'algorithme IMMJPDAF adaptatif qui utilise la méthode originale de Van Keuk et l'algorithme IMMJPDAF qui utilise un temps de mise à jour fixe.

***Mots clés**: Radar, Poursuite de cibles manoeuvrantes, Association de données, Temps de mise à jour variable, IMM, JPDAF.*

H. BENOUDNINE
M. KECHE
A. OUAMRI
Université d'Oran (USTO)
B.P 1505, El Mnouar
Oran, Algérie

M.S. WOOLFSON
School of Electrical
and Electronic Engineering
University of Nottingham
Nottingham, NG7 2RD, U.K.

ملخص

نقترح في هذه الدراسة خوارزمية لتتبع عدة أهداف مناورة تتحرك في وسط به جلبة. الخوارزمية هذه تدمج المرشح الاحتمالي للربط المشترك للمعطيات (JPDAF) وخوارزمية النماذج المتعددة و المتفاعلة (IMM) مع استعماله لدورة قياس متغيرة تتكيف مع مسار الأهداف.

استعمل في الخوارزمية الناتجة (AIMMJPDAF) شكل معاير لطريقة Van Keuk الأصلية لحساب فترة القياس المتغيرة وقد اختبرت فعالية الخوارزمية المقترحة باستعمال محاكاة Monte Carlo، كما قورنت بفعالية خوارزمية (AIMMJPDAF)

التي تستعمل الطريقة الأصلية ل Van Keuk وخوارزمية (IMMJPDAF) التي تستعمل دورة قياس ثابتة.

***الكلمات المفتاحية**: الرادار، تتبع أهداف مناورة ، ترابط ودمج المعطيات، فترة قياس تكيفية ، IMM، JPDAF .*

Tracking multiple manoeuvring targets in a cluttered environment has been and remains a challenging problem. Two major issues have to be addressed for solving this problem: data association, i.e. determining the origin of each received measurement and track maintenance of manoeuvring targets. A standard solution is to use a Kalman filter with some adaptive technique to track manoeuvring targets and the nearest neighbour technique for data association. However the performance of such a solution is poor in the case of highly manoeuvring targets or dense clutter. Optimal solutions for this problem, such as the Multiple Hypotheses Tracking method, exist [1]. Unfortunately, they are computationally very expensive and sub-optimal solutions with a reasonable complexity and an acceptable performance are preferable in practice. In such solutions, the problem of tracking manoeuvring targets and the problem of data association are usually addressed separately.

Many techniques have been proposed to address the first problem. These, usually, involve the modification of some parameters of the tracking filter, such as the process noise level [2] or the dimension of the tracking filter [3], in response to a change in the target's dynamics. The changes in the target' dynamics are generally detected by using a manoeuvre detection algorithm [4]. However, the methods based upon a manoeuvring detector have the limitation of delayed response to manoeuvres, especially in the presence of clutter, where false detection and no detection of manoeuvre are frequent. The Interacting Multiple Model (IMM) [5] algorithm overcomes this limitation by using more than one model to describe the motion of a target and ensuring a smooth transition from one model to another. In this algorithm, the probabilities of each model being correct are computed at each scan and used to combine the estimated states based on each model.

The ability of a phased array radar to adaptively vary the update time has also been exploited for tracking manoeuvring targets. Basically, a small update time is used to accurately track a target that is manoeuvring, while a larger update time is used to track a target that has a quiescent motion, in order to save the radar resources.

On the other hand, many techniques have been proposed to solve the data association problem that arises when tracking multiple targets in clutter. Amongst these techniques, the Joint Probabilistic Data Association (JPDAF) has been widely used [6]. In this filter, considering that each measurement could have originated from a known target or from clutter, the measurements to targets association probabilities are computed and used to form a weighted average measurement for updating the track of each target.

In Ahmeda *et al.* [7], it has been shown that the use of a variable update time can also be useful for solving the data association problem. In this work, the Van Keuk method [8] has been used to adaptively calculate the update time in the cheap algorithm [9], which is a simplified version of the JPDAF. Ahmeda *et al.* have used a third order Kalman filter which is appropriate for a manoeuvring target but not for a target travelling at a nearly constant velocity.

A significant improvement in the tracking accuracy has been obtained in Benoudnine *et al.* [10], by combining the IMM and the JPDAF (IMMJPDAF). It has been shown that the IMMJPDAF outperforms the JPDAF that uses a second or a third order Kalman filter and that the adaptive IMMJPDAF, where the update time is varied, has a better performance than the constant update time IMMJPDAF.

As in [7], the Van Keuk method has been used to calculate the variable update time in [10]. In this method a track is updated the next time when the variance of the predicted position error crosses a threshold, which is chosen to be proportional to the variance of the measurement position error.

In Benoudnine *et al.* [11], the performance of the AIMMJPDAF has been further improved by using the modified Van Keuk method proposed in [12], to calculate the next update time. In the modified Van Keuk method, the distance between targets is taken into account for the calculation of the update time.

This paper describes in more details the work presented in [10] and [11]. It also presents new simulation results. It is organised as follows: after formulating the problem in section two, the tracking algorithm that combines the IMM and the JPDAF is outlined in section three. This is followed by a description of the use of the original and the modified Van Keuk method for the adaptive selection of the update time in the resulting IMMJPDAF algorithm. The results of simulation are presented in section five and some conclusions are drawn in section six. Finally Appendices A and B are dedicated to the computation of the likelihood in the IMMJPDAF and the update time, respectively.

It should be noted that a related paper [13] has recently been published, where different models are used in the IMM with different update times according to the type of model used. For example, for a constant velocity model a larger update time is used than for a model describing a rapidly accelerating target. For each model in the IMM, the update time is a constant. The various update times are related to each other by a factor of 2. In this method, the wavelet transform is used to further filter the effects of noise. The method is applied to the tracking of a single target in the absence of clutter and the update time is chosen from a discrete set of values. In the work described in the present paper, more than one target in clutter are being tracked and the update time can take a continuous range of values.

1. PROBLEM FORMULATION

The discrete state equation for a target moving in a plane is:

$$\mathbf{x^j}(k+1) = \mathbf{F^j}(k)\mathbf{x^j}(k) + \mathbf{w^j}(k) \tag{1}$$

where $\mathbf{x^j}(k)$ is the state vector of the target, $\mathbf{F^j}(k)$ is the transition matrix and $\mathbf{w^j}(k)$ is the process noise, both at time k and for model j.

In this work, two models are used:

- In the first model (j=1), the state vector is a 4-dimensional vector consisting of the position and the velocity (in each of the 2 Cartesian co-ordinates) :

$$\mathbf{x^1} = [\mathrm{x}\ \ \dot{\mathrm{x}}\ \ \mathrm{y}\ \ \dot{\mathrm{y}}] \tag{2}$$

- In the second model (j=2), the state vector is a 6-dimensional vector consisting of the position, the velocity and the acceleration :

$$\mathbf{x^2} = [\mathrm{x}\ \ \dot{\mathrm{x}}\ \ \ddot{\mathrm{x}}\ \ \mathrm{y}\ \ \dot{\mathrm{y}}\ \ \ddot{\mathrm{y}}] \tag{3}$$

The expression for the model j transition matrix, $\mathbf{F^j}$, given a sampling period equal to T is :

$$\mathbf{F^j} = \begin{bmatrix} f^j & 0 \\ 0 & f^j \end{bmatrix} \tag{4}$$

where

$$f^1 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad f^2 = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The process noise $\mathbf{w^j}(k)$ is assumed to be a zero mean Gaussian process with a known covariance :

$$\mathbf{E}\ [\mathbf{w^j}(k)\mathbf{w^j}(l)] = \mathbf{Q^j}\delta(k,l) \tag{6}$$

where $\delta(k,l)$ is the delta function.

The expression for $\mathbf{Q^j}$ is:

$$\mathbf{Q^j} = \begin{bmatrix} q^j & 0 \\ 0 & q^j \end{bmatrix} \tag{7}$$

where

$$q^1 = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} \\ \frac{T^3}{2} & T^2 \end{bmatrix} q_0^1 \quad (8)$$

and

$$q^2 = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & \frac{T^2}{2} \\ \frac{T^3}{2} & T^2 & T \\ \frac{T^2}{2} & T & 1 \end{bmatrix} q_0^2 \quad (9)$$

In (8) and (9), q_0^1 and q_0^2 are respectively, the variances of the process noises that model the acceleration in model 1 and the acceleration increment over a sampling period in model 2.

The measurement equation is given by :

$$\mathbf{z}(k) = \mathbf{H^j}(k)\mathbf{x^j}(k) + \mathbf{v}(k) \quad (10)$$

where $\mathbf{z}(k)$ is the (m,l) measurement vector at time *k,* due to the return from the target.

$\mathbf{H^j}$ is the (m,n) measurement matrix for model *j,* defined as:

$$\mathbf{H}^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } \mathbf{H}^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (11)$$

and **v** is the measurement noise vector with zero mean and known covariance **R** :

$$\mathbf{E}[\,\mathbf{v}(k)\mathbf{v}(l)^T] = \mathbf{R}\delta(k,l) \quad (12)$$

If, for a given manoeuvring target, the only measurement received is its own return, then the problem would be the choice of the model that fits the best the dynamics of the target at a given time. However, in real situations, in addition to the return from the target, other false measurements are received, originating either from other targets or from clutter. An additional problem arises then, that is which measurement to use for updating the target track. This problem is referred to as the data association problem.

2. DESCRIPTION OF THE IMMJPDAF TRACKING ALGORITHM

In the IMM, several filters are run in parallel, each filter being matched to one of the models used to describe the dynamics of a target. The state estimates from these filters are combined on a probabilistic basis to form the overall state estimate. The switch between different models is assumed to be governed by a Markovian chain. To cope with the uncertainty in the origin of measurements, each filter in the IMM is chosen to be a JPDAF. The resulting algorithm is referred to as the IMMJPDAF, and consists of the following steps :

step 1: *Mixing of state estimates from the previous time.*

For each target, starting with the state estimates $\hat{\mathbf{x}}^j(k-1|k-1)$ matched to the models $M_j(k)$, their covariances $\hat{\mathbf{P}}^j(k-1|k-1)$ and the model probabilities $\mu_{i/j}(k-1|k-1)$, the mixed state estimate $\hat{\mathbf{x}}^{0j}(k-1|k-1)$ and its covariance $\hat{\mathrm{P}}^{0j}(k-1|k-1)$ are computed according to :

$$\hat{\mathrm{x}}^{0j}(k-1|k-1) = \sum_{i=1}^{r} \hat{\mathrm{x}}^i(k-1|k-1)\mu_{i/j}(k-1|k-1), \quad j=1, \ldots r \quad (13)$$

and

$$\hat{\mathrm{P}}^{0j}(k-1|k-1) = \sum_{i=1}^{r} \mu_{i/j}(k-1|k-1)\left[\hat{\mathrm{P}}^i(k-1|k-1) + \hat{\mathrm{P}}_S^{ij}(k-1|k-1)\right] \quad (14)$$

where

$$\hat{\mathrm{P}}_S^{ij}(k-1|k-1) = \left[\hat{\mathrm{x}}^i(k-1|k-1) - \hat{\mathrm{x}}^{0j}(k-1|k-1)\right]\left[\hat{\mathrm{x}}^i(k-1|k-1) - \hat{\mathrm{x}}^{0j}(k-1|k-1)\right]^T, \quad i,j=1,..,r$$

where r denotes the number of interacted models and $\mu_{i/j}(k-1|k-1)$ is the probability that model M_i was in effect at time $(k-1)$ given that M_j is in effect at time k, conditioned on $\mathbf{Z}^{k-1}$, the set of measurements up to $k-1$:

$$\mu_{i/j}(k-1|k-1) = \frac{1}{\bar{c}_j} p_{ij}\mu_i(k-1), \; i,j=1,..,r \quad (15)$$

In the above equation, p_{ij} is the prior probability of transition from model i to model j, $\mu_i(k-1)$ is the probability that model *i* is in effect at time $k-1$ and $\bar{c}_j$ are the normalising constants:

$$\bar{c}_j = \sum_{i=1}^{r} p_{ij}\mu_i(k-1), \quad j=1,\ldots,r \quad (16)$$

step 2: *Prediction of states and measurements.*

The predicted state $\hat{\mathbf{x}}^j(k|k-1)$, its covariance $\hat{\mathbf{P}}^j(k|k-1)$ and the predicted measurement $\hat{\mathbf{z}}^j(k|k-1)$ are computed using the initial state estimate (13) and its covariance (14):

$$\hat{\mathbf{x}}^j(k|k-1) = \mathbf{F^j}\hat{\mathbf{x}}^{\mathbf{0j}}(k-1|k-1) \quad (17)$$

$$\hat{\mathbf{P}}^{\mathbf{j}}(k|k-1) = \mathbf{F^j}\hat{\mathbf{P}}^{\mathbf{0j}}(k-1|k-1)(\mathbf{F^j})^T + \mathbf{Q^j} \quad (18)$$

and

$$\hat{\mathbf{z}}^{\mathbf{j}}(k|k-1) = \mathbf{H^j}\hat{\mathbf{x}}^{\mathbf{j}}(k|k-1) \quad (19)$$

step 3: *Validation of measurements.*

For each target, the set of validated measurements is formed by the measurements falling inside the union of the validation gates corresponding to the models used in the IMM algorithm. A measurement $\mathbf{z}(k)$ is accepted at time *k,* for model *j*, if it satisfies the following criterion:

$$\left(\mathbf{z}(k) - \hat{\mathbf{z}}^{\mathbf{j}}(k|k-1)\right)^T\left[\mathbf{S^j}(k)\right]^{-1}\left(\mathbf{z}(k) - \hat{\mathbf{z}}^{\mathbf{j}}(k|k-1)\right) \le \gamma \quad (20)$$

where γ is a threshold determined from a chi-square distribution with a degree of freedom equal to the dimension of the measurement and $\mathbf{S}^j(k)$ is the innovation covariance matrix.

step 4: *Mode conditioned state estimation.*

Using the predicted state $\hat{\mathbf{x}}^j(k|k-1)$, its covariance $\hat{\mathbf{P}}^j(k|k-1)$ and the validated measurements as inputs, the mode conditioned state estimates and their covariances are calculated via the JPDAF.

step 5: *Computation of the likelihood function.*

The likelihood $\Lambda^j(k)$ of model $M_j(k)$ is :

$$\Lambda^j(k) = P\left[\mathbf{z}(k)\middle|M_j(k), \mathbf{Z}^{k-1}\right] \tag{21}$$

The expression for its computation is derived in Appendix A.

step 6: *Update of the models' probabilities.*

The probability that model $M_j(k)$ is in effect at time k is computed from:

$$\mu_j(k) = P\left[M_j(k)\middle|\mathbf{Z}^k\right],\ j = 1,\cdots,r$$
$$= \frac{1}{c}\Lambda^j(k)\bar{c}_j \tag{22}$$

where $\bar{c}_j$ is defined in (16) and c is the normalisation constant for $\mu_j(k)$ given by:

$$c = \sum_{j=1}^{r} \Lambda^j(k)\bar{c}_j \tag{23}$$

step7: *Combination of the model conditioned estimates.*

For each target, the overall state estimate $\hat{\mathbf{x}}(k|k)$ and its corresponding error covariance $\hat{\mathbf{P}}(k|k)$ are updated as follows :

$$\hat{\mathbf{x}}(k|k) = \sum_{j=1}^{r} \hat{\mathbf{x}}^j(k|k)\mu_j(k) \tag{24}$$

$$\hat{\mathbf{P}}(k|k) = \sum_{j=1}^{r} \mu_j(k)\left(\hat{\mathbf{P}}^j(k|k) + \hat{\mathbf{P}}_s^j(k|k)\right) \tag{25}$$

$$\hat{\mathbf{P}}_s^j(k|k) = \left[\hat{\mathbf{x}}^j(k|k) - \hat{\mathbf{x}}(k|k)\right]\left[\hat{\mathbf{x}}^j(k|k) - \hat{\mathbf{x}}(k|k)\right]^T$$

3. IMMJPDAF WITH VARIABLE UPDATE TIME (AIMMJPDAF)

In [10], a variable update time has been incorporated into the IMMJPDAF using the Van Keuk method [8]. In this method, the next update time is selected so that the predicted error variance in position is kept under a given threshold.

For a target moving in Cartesian co-ordinates, the next update time $T_x(k)$, at the k^{th} scan, in the x direction is determined from :

$$[\hat{\mathbf{P}}(k|k-1)]_{11} = v_0[\mathbf{R}]_{11} \tag{26}$$

where $[\hat{\mathbf{P}}(k|k-1)]_{11}$ is the (1,1) element of the predicted covariance matrix, $[\mathbf{R}]_{11}$ is the measurement variance in the x direction and v_0 is a Track Sharpness Parameter (TSP), that is controlled by the user.

It is shown in Appendix B, that $[\hat{\mathbf{P}}(k|k-1)]_{11}$ is a bi-quadratic polynomial in $T_x(k)$ whose coefficients depend on the elements of the matrices $\hat{\mathbf{P}}^{0j}(k-1|k-1)$, the variances of the process noises used in the models, the components of the initialisation state vectors $\hat{\mathbf{x}}^{0j}(k-1|k-1)$ and the predicted model probabilities $\mu_j(k|k-1)$. The next update time $T_x(k)$ can then be determined by zeroing the bi-quadratic polynomial $\left[\hat{\mathbf{P}}(k|k-1)\right]_{11} - v_0\left[\mathbf{R}\right]_{11}$, and taking the maximum real and positive root.

The same procedure can be used to calculate the update time in the y direction and the global update time in the Cartesian co-ordinate at the k^{th} scan is taken as :

$$T(k) = \min(T_x(k), T_y(k)) \tag{27}$$

3.1- The Modified Van Keuk Method

In the original Van Keuk method, the TSP v_0 is chosen to be constant. In [12], Keche *et al.* have proposed to modify the Van Keuk method by letting the TSP v_0 vary as a function of the distance between targets.

The following simple relation has been used to update v_0 :

$$v_0(d) = \begin{cases} v_{0L} & \text{if} \quad d \le d_L \\ v_{0L} + \dfrac{(v_{0H} - v_{0L})}{(d_H - d_L)}(d - d_L) & \text{if} \quad d_L < d < d_H \\ v_{0H} & \text{if} \quad d \ge d_H \end{cases} \tag{28}$$

where v_{0L}, v_{0H}, d_L and d_H are constants that control the update time and consequently the tracking accuracy.

It has been shown in [11] and [12] that using the modified Van Keuk method to calculate the update time in the JPDAF and the IMMJPDAF may bring about a significant improvement in the performance.

4. SIMULATION RESULTS

A number of Monte Carlo simulations have been carried out to assess and compare the performances of the following algorithms: the IMMJPDAF that uses a constant update time (CIMMJPDAF), the JPDAF that uses a constant update time (CJPDAF), the Adaptive JPDAF and IMMJPDAF based on the original Van Keuk method (OJPDAF and OAIMMJPDAF) and the adaptive IMMJPDAF based on the modified Van Keuk method.

Two scenarios have been used for this purpose: the first scenario (Fig.1) corresponds to two crossing targets that follow, during 150s, straight line paths at a constant speed of 308.67 m/s and cross midway at t =75s, with a given crossing angle. The second scenario (Fig.2) corresponds to two manoeuvring targets which approach each other, manoeuvre, then separate without crossing. Each target trajectory consists of three segments: an initial non manoeuvring segment from 0 to 80s, followed by a

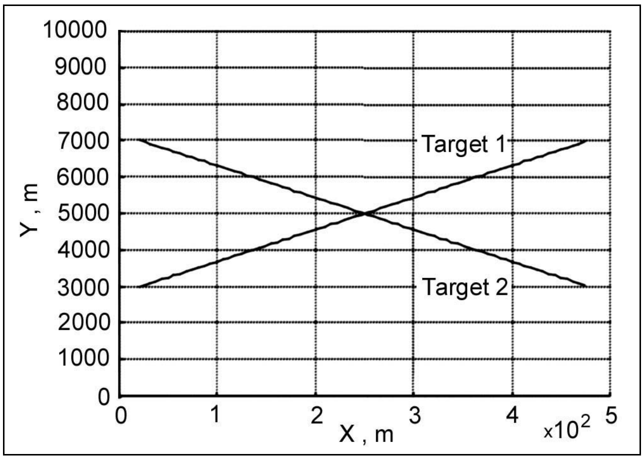

Figure 1: Targets' trajectories used in scenario 1 (two crossing targets).

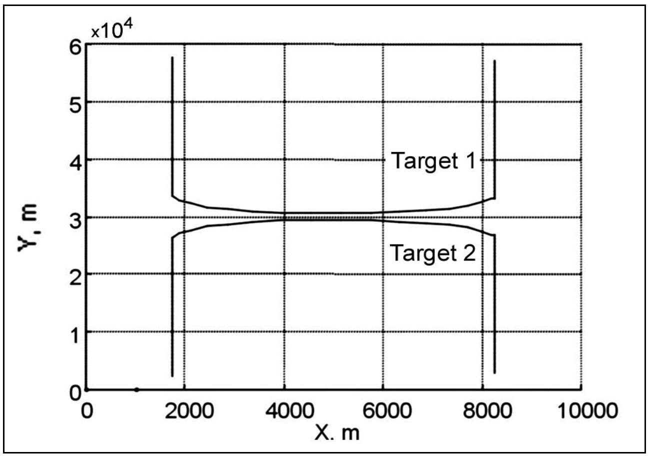

Figure 2: Targets' trajectories used in scenario 2 (two manoeuvring targets).

manoeuvring segment from 80 to 113s, followed by a non manoeuvring segment from 113 to 193s.

In both scenarios the measurement noise is generated in polar co-ordinates with standard deviation of 185.2 m and 2.5×10^{-3} radian in range and bearing, respectively. The range and bearing are then converted to two dimensional Cartesian co-ordinates. The clutter with a density λ, assumed to be known a priori, is introduced in the system at time t=15s. It is generated so that the number of clutter returns observed in the surveillance region, equal to 50km×10km in the first scenario and 10km×60km in the second scenario, is a random number with a Poisson distribution. Two models are used in the IMM with a probability of switching between them equal to 0.05. The first model corresponds to a second order JPDAF filter with a process noise standard deviation equal to 1 m/s^2 and the second to a third order JPDAF filter with a process noise standard deviation equal to 5 m/s^2. The process noise standard deviation used in the JPDAF that uses a second order Kalman filter (JPDAF2) and the JPDAF that uses a third order Kalman filter (JPDAF3) are 20 m/s^2 and 5 m/s^2, respectively.

The probability of detection for each target was assumed to be equal to 1 and the size of the validation gate was chosen so that the probability of validating a true measurement is equal to 0.9995. The figure of merit used to compare different algorithms is the track loss rate defined as the percentage of runs, out of 100, where a track is lost. In figures 3 and 4 the track loss rate is plotted versus the clutter density, for the CJPDAF2, the CJPDAF3 and the CIMMJPDAF that use a 1.5s constant update time. Figure 3 corresponds to the first scenario, with a crossing angle equal to 10 degrees, while figure 4 corresponds to the second scenario with a minimum separation distance between targets equal to 800m. The following observations can be made :

- The CJPDAF2 performs well in the case of crossing targets (non manoeuvring targets), but has a poor performance in the case of manoeuvring targets.
- The performance of the CJPDAF3 is good in the case of manoeuvring targets but is unacceptable in the case of crossing targets.
- The CIMMJPDAF algorithm achieves a compromise between the previous two algorithms and it is thus adaptive to the behaviour of the targets.

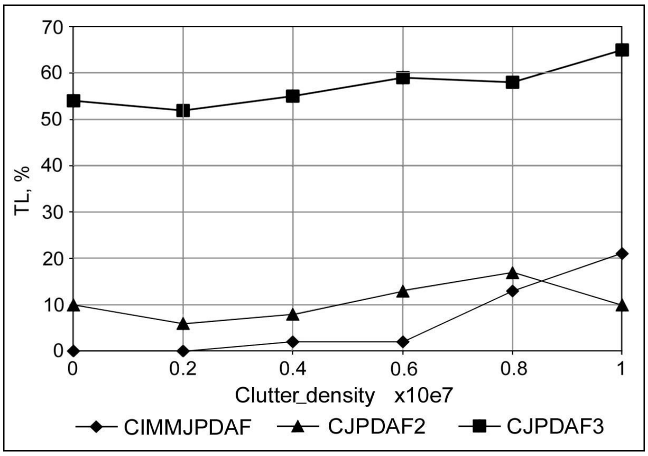

Figure 3: Track loss rate vs. clutter density for crossing targets, with a 10 degrees crossing angle, using CIMMJPDAF,CJPDAF3 and CJPDAF2 with a constant update time equal to 1.5s.

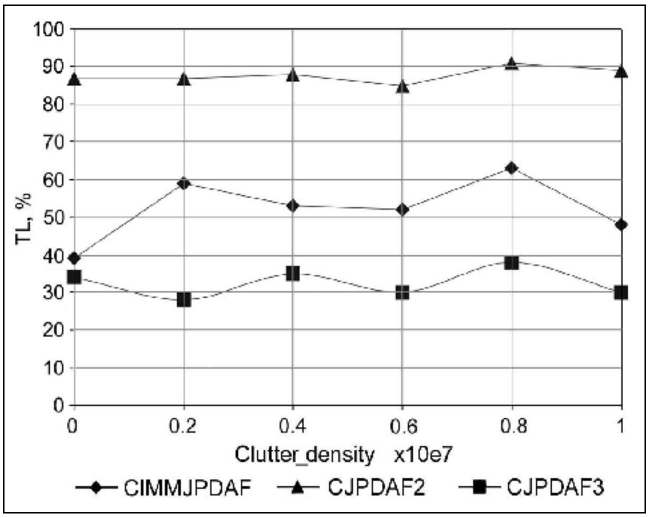

Figure 4: Track loss rate vs. clutter density for manoeuvring targets, with a minimum separation distance equal to 800m, using

CIMMJPDAF, CJPDAF3 and CJPDAF2 with a constant update time equal to 1.5s.

The second series of results, presented in figures 5 and 6, compare in the same condition as previously, the performance of the adaptive JPDAF2, JPDAF3 and IMMJPDAF, that use the original Van Keuk method to calculate the update time. The TSP v_0 was chosen for each algorithm so that the mean update time obtained is approximately equal to 1.5s.

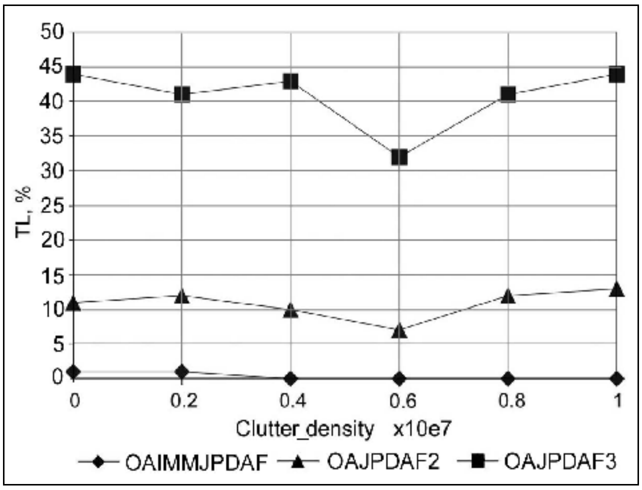

Figure 5: Track loss rate vs. clutter density for crossing targets, with a 10 degree crossing angle, using OAIMMJPDAF, OAJPDAF3 and OAJPDAF2 with a mean update time equal to 1.5s.

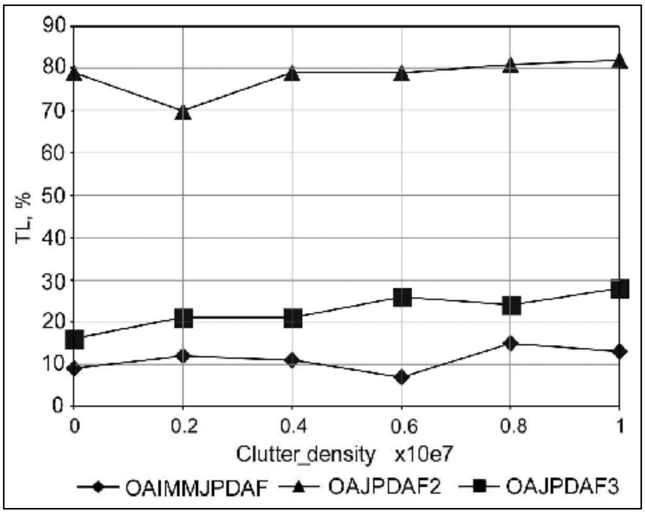

Figure 6: Track loss rate vs. clutter density for manoeuvring targets, with a minimum separation distance equal to 800m, using OAIMMJPDAF, OAJPDAF2 and OAJPDAF3 with a mean update time equal to 1.5s.

It is clear from these figures that the previous results, obtained with a constant update time, are confirmed with a variable update, i.e, the IMMJPDAF has the best performance overall. It can also be observed that the improvement brought about by using a variable update time is more significant in the case of the IMMJPDAF, especially, for manoeuvring targets.

Finally, the last series of results is dedicated to the comparison between the adaptive IMMJPDAF based on the

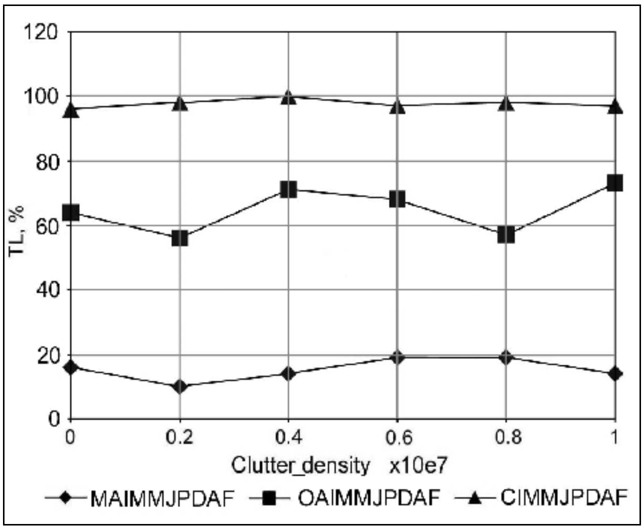

Figure 7: Track loss clutter density for manoeuvring targets, with a minimum separation distance equal to 800 m, using MAIMMJPDAF, OAIMMJPDAF and CIMMJPDAF with a mean update time equal to 2.8s.

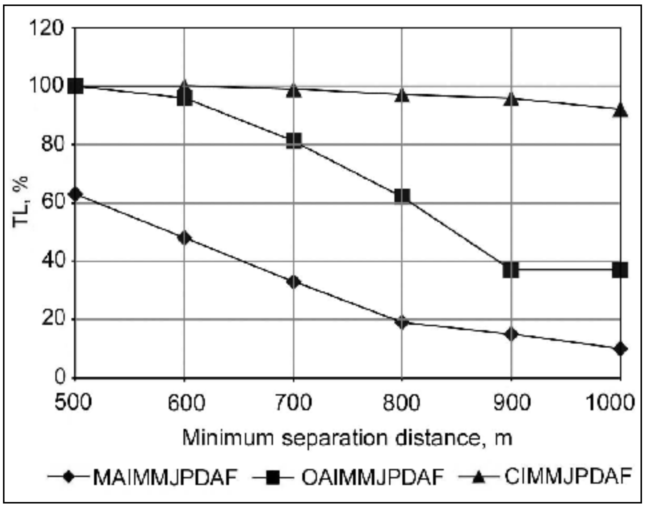

Figure 8: Track loss rate vs. Minimum separation distance for manoeuvring targets, in clutter with a density $\lambda = 0.6\times10^{-6}$, using MAIMMJPDAF, OAIMMJPDAF and CIMMJPDAF with a mean update time equal to 2.8s.

modified Van Keuk method, the adaptive IMMJPDAF based on the original Van Keuk method and the IMMJPDAF that uses a constant update time. For a fair comparison, the value v_0, when kept constant, is adjusted so that the mean update time obtained is approximately the same as that obtained with a variable v_0 using the parameters: $v_{0L} = 0.9$, $v_{0H} = 8$, $d_{0L} = 400m$ and $d_{0H} = 15Km$. This value for the mean update time is used in the simulation with a constant update time.

The track loss rate obtained with a fixed update time, a variable update time with a fixed v_0 and a variable update time with a variable v_0 is displayed in figure 7 as a function of the clutter density, for scenario 2 with a minimum separation distance equal to 800m. Figure 8 presents the track loss rate against the minimum separation distance in the presence of a clutter with a density equal to 0.6x10^{-7}. For scenario 1, it has been observed that the results obtained using the MAIMMJPDAF, the OAIMMJPDAF and the CIMMJPDAF are similar. The track loss rate is equal to 0 if the crossing angle is larger than 6 degrees and it is approximately the same for smaller crossing angles because in this case, the v_0 used in the OAIMMJPDAF algorithm is always almost equal to v_{0L}. Both figure 7 and 8 show clearly that using a variable update time improves the performance in terms of successful tracking. They also show that the AIMMJPDAF that uses the modified Van Keuk method has a better performance than the AIMMJPDAF that uses the original Van Keuk Method, especially for small minimum distances between trajectories. This demonstrates the efficiency of adjusting the update time as a function of the distance between trajectories in the modified Van Keuk method.

CONCLUSION

In this paper a variable update time has been incorporated into an algorithm that combines the IMM and the JPDAF for tracking manoeuvring targets in a cluttered environment. The modified and the original Van Keuk methods have been used to analytically calculate the update time. It has been shown that the IMMJPDAF outperforms the JPDAF that uses a second or a third order Kalman filter and that using a variable update time improves the performance in terms of successful tracking. It has also been shown that a better selection of the update time and thus a better management of the radar resources is obtained by using the modified Van Keuk method.

Appendix A: COMPUTATION OF THE LIKELIHOOD

The definition of the likelihood $\Lambda^j(k)$ of model $M_j(k)$ is :

$$\Lambda^j(k)=P\left[\mathbf{z}(k)\middle|M_j(k),\mathbf{Z}^{k-1}\right]=\frac{P\left[\mathbf{z}(k),M_j,\mathbf{Z}^{k-1}\right]}{P\left[M_j,\mathbf{Z}^{\mathbf{k-1}}\right]} \quad \text{(A.1)}$$

The probability of the feasible association event $\theta(k)$, conditioned upon $\mathbf{Z}^{\mathbf{k}}$ can be written as:

$$P\left[\theta(k)\middle|\mathbf{Z}^{\mathbf{k}},M_j\right]=P\left[\theta(k)\middle|\mathbf{z}(k),M_j,\mathbf{Z}^{\mathbf{k-1}}\right]=$$

$$=\frac{1}{c_j}P\left[\theta(k)\middle|M_j,\mathbf{Z}^{\mathbf{k-1}}\right]P\left[\mathbf{z}(k)\middle|\theta(k),M_j,\mathbf{Z}^{k-1}\right] \quad \text{(A.2)}$$

where

$$c_j=\frac{P\left[\mathbf{z}(k),M_j,\mathbf{Z}^{k-1}\right]}{P\left[M_j,\mathbf{Z}^{\mathbf{k-1}}\right]}=\Lambda^j(k) \quad \text{(A.3)}$$

In [6], it is shown that for false measurements with a Poissonian probability, the mass function $\mu_F(\phi)$ given by :

$$\mu_F(\phi)=\frac{e^{-\lambda V}(\lambda V)^{\phi}}{\phi!}\,. \quad \text{(A.4)}$$

The posterior probability of $\theta(k)$ is given by

$$P\left(\theta(k)\middle|Z^K,M_j\right)=\frac{1}{c_j}\frac{e^{-\lambda V}}{m_k!}\lambda^{\phi}f\left(\theta(k)\right) \quad \text{(A.5)}$$

where

$$f(\theta(k))=\prod_{j=1}^{m_k}\left[N\left(z_j(k)\right)\right]^{\tau_j}\prod_{t=1}^{T}\left(\left(P_D^t\right)^{\delta_t}(1-P_D^t)^{1-\delta_t}\right) \quad \text{(A.6)}$$

ϕ is the number of false measurements, $N(x)$ represents a Gaussian probability distribution function in the variable x, λ is the clutter density, V is the surveillance volume, m_k is the number of received measurements, P_D^t is the probability of detection for target t, T is the number of targets, δ_t is a binary target detection indicator that indicates whether target t is associated with any measurement in the association event $\theta(k)$ and τ_j is a binary variable which indicates if measurement j is associated with any target in event $\theta(k)$.

Since $\sum_{\theta}P\left(\theta(k)\middle|\mathbf{Z}^{\mathbf{k}},M_j\right)=1$, $\Lambda^j(k)$ can be computed from :

$$\Lambda_j(k)=\frac{e^{-\lambda V}}{m_k!}\sum_{\theta(k)}\lambda^{\phi}f\left(\theta(k)\right) \quad \text{(A.7)}$$

Appendix B: COMPUTATION OF THE UPDATE TIME

In this appendix, the analytical expression for the next update time is derived, when Van Keuk criterion is used. It is required for this to express the elements (1,1) and (4,4) of the predicted covariance matrix as a function of $T_x(k)$ and $T_y(k)$, the update times in the x and y direction, respectively.

If we consider for instance the x direction, the expression for the (1,1) element of the predicted covariance matrix is given by :

$$[\hat{\mathbf{P}}(k|k-1)]_{11}=\sum_{j=1}^{r}\mu_j(k|k-1)\left\{\left[\hat{\mathbf{P}}^j(k|k-1)\right]_{11}+\left[\mathbf{P}_S^j(k|k-1)\right]_{11}\right\} \quad \text{(B.1)}$$

where $\mu_j(k|k-1)$ is the predicted probability of model j, equal to $\bar{c}_j$ given by (16),

$$[\hat{\mathbf{P}}^j(k|k-1)]_{11}=[\mathbf{F}^j(T_x(k))\hat{\mathbf{P}}^{0j}(k-1|k-1)(\mathbf{F}^j(T_x(k))^T+\mathbf{Q}^j(T_x(k))]_{11} \quad \text{(B.2)}$$

and

$$\left[\hat{\mathbf{P}}_{\mathbf{S}}^{\mathbf{j}}(k|k-1)\right]_{11}=$$

$$=\left[\left\{\hat{\mathbf{x}}^{\mathbf{j}}(k|k-1)-\hat{\mathbf{x}}(k|k-1)\right\}\left\{\hat{\mathbf{x}}^{\mathbf{j}}(k|k-1)-\hat{\mathbf{x}}(k|k-1)\right\}^T\right]_{11}$$

$$=\left[[\hat{\mathbf{x}}^{\mathbf{j}}(k|k-1)]_{11}-[\hat{\mathbf{x}}(k|k-1)]_{11}\right]^2 \quad \text{(B.3)}$$

In (B.2) $\mathbf{F}^j$ and $\mathbf{Q}^j$ denote, respectively, the transition matrix and the process noise covariance matrix, both matched to model M_j. Considering the case of a third order JPDA Filter, i.e, using expressions (4) and (7), with j=2 for $\mathbf{F}^j$ and $\mathbf{Q}^j$, the following expression is obtained:

$$\left[\hat{\mathbf{P}}^j(k(k-1)\right]_{11}=0.25\left\{\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{33}+\left(q_0^j\right)^2\right\}T_x^4(k)+$$
$$+\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{23}T_x^3(k)$$
$$+\left\{\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{22}+\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{13}\right\}T_x^2(k)+$$
$$+2\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{12}T_x(k)+\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{11} \quad \text{(B.4)}$$

If instead of a third order JPDA Filter, a second order JPDA Filter is used, then equation (B.4) becomes:

$$\left[\hat{\mathbf{P}}^j(k(k-1)\right]_{11}=0.25(q_0^j)^2T_x^4(k)+\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{22}T_x^2(k)+$$
$$+2\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{12}T_x(k)+\left[\hat{\mathbf{P}}^{0j}(k-1(k-1)\right]_{11} \quad \text{(B.5)}$$

where $(q_0^j)^2$ denotes the process noise variance of model j.

Equation (B.4) and (B.5) show that $\left[\hat{\mathbf{P}}^{\mathbf{j}}\left(k|k-1\right)\right]_{11}$ is a bi_quadratic polynomial in $T_x(k)$ whose coefficients depend on the elements of matrix $\hat{\mathbf{P}}^{\mathbf{0j}}\left(k-1|k-1\right)$ and the variances $(q_0^j)^2$ of the process noise.

$\left[\hat{\mathbf{P}}_{\mathbf{S}}^{\mathbf{j}}\left(k|k-1\right)\right]_{11}$ is also a bi-quadratic polynomial in $T_x(k)$. To show this, let us consider the case of a third order JPDAF filter. From the equation of the predicted state (17), one can write:

$$\left[\hat{\mathbf{x}}^{\mathbf{j}}\left(k|k-1\right)\right]_{11}=\left[\mathbf{F}^{\mathbf{j}}\hat{\mathbf{x}}^{\mathbf{0j}}\left(k-1|k-1\right)\right]_{11}=$$
$$\hat{\mathbf{x}}^{\mathbf{0j}}\left(k-1|k-1\right)+\hat{\mathbf{x}}^{\mathbf{0j}}\left(k-1|k-1\right)T_x(k)+\frac{\hat{\hat{\mathbf{x}}}^{\mathbf{0j}}\left(k-1|k-1\right)}{2}T_x{}^2(k) \quad \text{(B.6)}$$

Hence, $\left[\hat{\mathbf{x}}^{\mathbf{j}}\left(k|k-1\right)\right]_{11}$ is a quadratic polynomial in $T_x(k)$. $\left[\hat{\mathbf{x}}\left(k|k-1\right)\right]_{11}$ which is equal to

$$[\hat{\mathbf{x}}\left(k|k-1\right)]_{11}=\sum_{j=1}^{r}\mu_j\left(k|k-1\right)[\hat{\mathbf{x}}^j\left(k|k-1\right)]_{11} \quad \text{(B.7)}$$

is also a quadratic polynomial in $T_x(k)$:

$$\left[\hat{\mathbf{x}}\left(k|k-1\right)\right]_{11}=\sum_{j=1}^{r}\hat{\mu}_j\left(k|k-1\right)\left[\hat{\mathbf{x}}^{\mathbf{j}}\left(k|k-1\right)\right]_{11}$$
$$=\sum_{j=1}^{r}\hat{\mu}_j\left(k|k-1\right)\hat{\mathbf{x}}^{\mathbf{0j}}\left(k-1|k-1\right)+\sum_{j=1}^{r}\hat{\mu}_j\left(k|k-1\right)\hat{\mathbf{x}}^{0j}\left(k-1|k-1\right)+$$
$$+\frac{1}{2}\sum_{j=1}^{r}\hat{\mu}_j\left(k|k-1\right)\hat{\hat{\mathbf{x}}}^{0j}\left(k-1|k-1\right)T_x{}^2(k) \quad \text{(B.8)}$$

Then, from (B.3), (B.6) and (B.8), one can deduce that $\left[\hat{\mathbf{P}}_{\mathbf{S}}^{\mathbf{j}}\left(k|k-1\right)\right]_{11}$ is a bi-quadratic polynomial in $T_x(k)$.

If a second order JPDAF is used, then $\hat{\hat{\mathbf{x}}}^{\mathbf{0j}}$ in equations (B.6) and (B.8) should be set to 0 and $[\hat{\mathbf{P}}(k|k-1)]_{11}$ would be a quadratic polynomial in $T_x(k)$.

Using equation (B.1), (B.3), (B.4) (or B.5), (B.6) and (B.8), $[\hat{\mathbf{P}}(k|k-1)]_{11}$ can be computed. A similar equation can be derived for computing $[\hat{\mathbf{P}}(k|k-1)]_{44}$.

REFERENCES

[1]- Reid D.B., "An algorithm for tracking multiple targets", *IEEE Trans. Automatic Control*, vol. 24, (1979), pp. 843-854.

[2]- Castella F.R., "An adaptive two dimensional Kalman tracking filter", *IEEE Trans. Aerospace and Electronic Systems*, Vol. 16, (1980), pp. 822-829.

[3]- Bar Shalom Y. and Birmiwal K., "Variable dimension filter for manoeuvring targets tracking", *IEEE Trans. Aerospace and Electronic Systems*, Vol. 18, (1982), pp. 621-629.

[4]- Woolfson M.S., "An evaluation of manoeuvre detector algorithms", *GEC Journal of Research* , Vol. 3, (1985), pp. 181-190.

[5]- Bar Shalom Y. and Li X.R., "Estimation and tracking : principles, techniques and software", *Artech House*, Inc., (1993), pp 222-228.

[6]- Y. Bar Shalom Y. and Fortmann T.E., "Tracking and data association", *Academic Press*, (1988).

[7]- Ahmeda S.S., Keche M., Harrison I. and Woolfson M.S., "Adaptive joint probabilistic data association algorithm for tracking multiple targets in cluttered environment", *IEE Proc Radar, Sonar and Navigation*, Vol. 144, (1997), pp. 309-314.

[8]- Van Keuk G., "Adaptive computer controlled target tracking with a phased array radar", *IEEE International Radar conference*, April (1977), pp. 429-434.

[9]- Fitzgerald R.J., "Development of practical PDA logic for multi-target tracking by microprocessor", In Proc. of the American Controls Conference, Seattle, Wash., (1986), pp. 889-898.

[10]- Benoudnine H., Keche M., Ouamri A., Woolfson M.S. and Harrison I., "AIMMJPDAF algorithm for tracking multiple manoeuvring targets in a cluttered environment", IASTED International Conference on Modelling, Identification and Control, Austria, February (1999).

[11]- Benoudnine H., Keche M., Ouamri A., Woolfson M.S. and Harrison I., "Improved Adaptive IMMJPDAF algorithm for tracking multiple manoeuvring targets in a cluttered environment", RADAR99, International Conference on RADAR Systems, Brest, 17-21 Mai (1999).

[12]- Keche M., Woolfson M.S., Harrison I., Ouamri A. and Ahmeda S.S., "Improved adaptive joint probabilistic data association filter for multiple targets angles tracking", *Electronics Letters,* Vol. 33, (1997), pp. 1361-1362.

[13]- Hong L., "Multirate Interacting Multiple Model Filtering for Target Tracking Using Multirate Models", *IEEE Transactions on Automatic Control*, vol. 44, (1999), pp. 1326-1340.

□