An Algorithm Unifying IMM and JPDA Approaches[1]

Ljudmil V. Bojilov[1], Kiril M. Alexiev[1], Pavlina D. Konstantinova[1]
Central Laboratory for Parallel Processing, Bulgarien Academy of Sciences, acad. "G. Bonchev" str.bl. 25A, Sofia, Bulgaria, e-mail: bojilov@bas.bg alexiev@bas.bg pavlina@bas.bg

L. V. Bojilov, K. M. Alexiev, P. D. Konstantinova

Abstract. In this paper, an attempt to unify the IMM and the JPDA approaches in a common algorithm is presented. For tracking manoeuvring target in clutter, IMM-PDA algorithm can be implemented. It solves successfully the problems of tracking single target in clutter. When multiple targets in clutter have to be tracked, the JPDA algorithm is one of the cheapest solutions in terms of computational load. If we assume manoeuvring targets in clutter, however, we will need of an algorithm with features of IMM and JPDA algorithms at the same time. We consider the more simple case for IMM algorithm – two models, one of which is embedded in the other. The main drawback arises when the association probabilities have to be calculated. To avoid combinatorial explosion due to the two state predictions for every track, we accept the next trade-off - we merge these two predictions, calculate association probabilities and the respective super innovation and finally go back to IMM case spliting the super innovation into two innovations.

Keywords: Tracking, manoeuvring, cluttered environment, assignment.

1. Introduction. The theoretically most powerful approach for tracking multiple targets in clutter is known to be MHT method. The MHT method, however, more often leads to combinatorial explosion and computational overload that makes implementation of MHT method questionable. In recent years, there is a strong interest in constructing algorithms capable to compute a ranked set of assignments of measurements to targets [4,5]. Such algorithms make MHT approach for the first time practically implementable. The development of an MHT program, however, is quite complex process. Therefore, an alternative of this approach can be JPDA algorithm with some features of Interacting Multiple Model (IMM) approach. This alternative deals with the most complicated case – manoeuvring targets in heavy clutter.

For tracking a single manoeuvring target in clutter there are many descriptions of IMM-PDA algorithm [1,2,3], unifying the features of IMM and PDA algorithms respectively. The numerical results cited in many papers reveal a very good performance of this algorithm in terms of low probability of target missing and, hence, successful track maintenance. In all IMM-PDA algorithm realisations PDA algorithm is embedded at some step in the IMM algorithm frame. No problems or ambiguity arises. When one tries to combine IMM and JPDA algorithms, however, some difficulties and drawbacks spring up.

[1] This work is partially supported by the Bulgarian Ministry of Education and Science under grants I-1205/2002 and I-1202/2002 and by Center of Excellence BIS21 grant ICA1-2000-70016.

A brief description of the IMM-PDA algorithm is provided in the next section. An attempt to develop the IMM-JPDA algorithm is exposed and the arising problems and drawbacks are discussed. In section 3, an IMM-JPDA algorithm is set out which overcomes discussing drawbacks. Finally, in Section 4, presented simulation results show a good behaviour of the algorithm in the moderate target density case.

2. Problem Formulation. The IMM-PDA algorithm with two imbeded models includes the next steps [2]:

Step 1. Computation of the mixed initial conditions for the filter matched to model t:

a) the mixed state estimate

$$\hat{x}^{0t}(k-1|k-1)=\sum_{s=1}^{2}\hat{x}^{s}(k-1|k-1)\mu_{s|t}(k-1|k-1) \qquad t=1,2$$

where

$$\mu_{s|t}(k-1|k-1)\equiv P\{M(k-1)=M_s|M(k)=M_t,Z^{k-1}\}=\frac{p_{st}\mu_s(k-1)}{\sum_{s=1}^{2}p_{st}\mu_s(k-1)},$$

p_{st} are Markov model-switching probabilities and $\mu_s(k-1)$ are the model probabilities computed at the time $k-1$.

b) computation of the corresponding covariance:

$$P^{0t}(k-1|k-1)=\sum_{s=1}^{2}\mu_{s|t}(k-1|k-1)\left\{P^{s}(k-1|k-1)+\left[\hat{x}^{s}-\hat{x}^{0t}\right]\cdot\left[\hat{x}^{s}-\hat{x}^{0t}\right]'\right\}$$

Here $\hat{x}^{s}$ and $\hat{x}^{0t}$ are referred at time $k-1$.

Step 2. This step is performed in parallel (i.e. independently) for every of the model-matched filters and is referred to as PDA step. For every one of the filters the likelihood function is calculated as a joint probability density function of the innovations:

$$\Lambda_t(k)\equiv p\left[Z(k)|M_t(k),Z^{k-1}\right]=p\left[v_1^t(k),\ldots,v_m^t(k)|m(k),Z^{k-1}\right]=$$

$$=\left[b(k)+\sum_{j=1}^{m(k)}e_j(k)\right]\cdot\frac{P_D P_G}{m(k)}V(k)^{-m(k)+1},$$

where

$$e_j(k)=\frac{1}{P_G}N\left[v_j;0,S(k)\right],$$

$$b(k)=\frac{m(k)(1-P_D P_G)}{P_D P_G V(k)},$$

$V(k)$ is the validation region, $m(k)$ is the number of measurements falling in the validation region and $S(k)$ is innovation covariance.

Step 3. This step is also a part of the PDA algorithm frame. First, the association probabilities are to be calculated

$$\beta_j(k)=\frac{e_j(k)}{b(k)+\sum_{j=1}^{m(k)}e_j(k)} \qquad j=1,\ldots,m(k)$$

$$\beta_0(k)=\frac{b(k)}{b(k)+\sum_{j=1}^{m(k)}e_j(k)}$$

Next, the combined innovation is defined as a weighted sum of the $m(k)$ measurements' innovations

$$\nu(k)=\sum_{j=1}^{m(k)}\beta_j(k)\nu_j(k)$$

At this point, the Kalman filter starts with the only exception – instead the standard form, the modified update equation for covariance P is used.

The results of Kalman filter are the state $\hat{x}^t(k|k)$ and the covariance $P^t(k|k)$ estimations. The last term of the output of model-matched filters is computed in the

Step 4. This is an IMM step. Here the model probability is updated as follows:

$$\mu_t(k)\equiv P\left[M_t(k)\middle|Z^k\right]=\frac{\Lambda_t(k)\sum_{s=1}^{2}p_{st}\mu_s(k-1)}{\sum_{t=1}^{2}\Lambda_t(k)\sum_{s=1}^{2}p_{st}\mu_s(k-1)}$$

Step 5. This step is for output only. It gives the final result of the IMM algorithm – combined model-conditioned state estimate and covariance, according to the following equations:

$$\hat{x}(k|k)=\sum_{t=1}^{2}\hat{x}^t(k|k)\mu_t(k)$$

$$P(k|k)=\sum_{t=1}^{2}\mu_t(k)\left\{P^t(k|k)+\left[\hat{x}^t(k|k)-\hat{x}(k|k)\right]\cdot\left[\hat{x}^t(k|k)-\hat{x}(k|k)\right]'\right\}$$

Now, let us try to extend this algorithm in the next way: instead PDA (steps 2 and 3) to include JPDA algorithm. JPDA algorithm begins with clusterisation. When clusters are formed, hypotheses generation is the next. The number of all feasible hypotheses depends on the number of the targets in a given cluster and on the number of the measurements in their validation regions. Every feasible hypothesis, H_l, represents a particular way of measurements to targets association in the cluster. For every target, however, there exist several models, and hence, so many predicted target positions. We have to run the entire JPDA procedure with every combination of the models of all targets in this cluster. Therefore, another explosion of hypotheses arises. For example, if the cluster contains m targets and we use r models, the number of all possible combinations will be as many as r^m. Even for moderate cases, this number is inadmissible.

To avoid this drawback let us to look at the last step of the IMM-PDA algorithm. Using the probabilities of every one of the models calculated at the end of the scan, the overall estimate is obtained. However, at the step, when target predictions are determined and where we want to include the JPDA algorithm, these model probabilities are not yet calculated. Nonetheless, we can easily estimate the predicted values of these probabilities before their computation [6]. Using the total probability theorem, we obtain:

$$\mu^t(k|k-1)\equiv P\left\{M(k)=M^t\middle|Z^{k-1}\right\}=$$

$$=\sum_{s=1}^{2}P\left\{M(k)=M^t\middle|M(k-1)=M^s,Z^{k-1}\right\}P\left\{M(k-1)=M^s\middle|Z^{k-1}\right\}=\sum_{s=1}^{2}p_{st}\mu^t(k-1)$$

Here $\mu^t(k-1)$ is the probability of the model t computed at scan $(k\text{-}1)$, p_{st} is Markov model-switching probability and the event that model t is in effect at time k is denoted by $M(k)=M^t$. Now, we can use these 'predicted model probabilities' to merge the individual model state predictions. In this way, every single target will remain with one-state prediction and the hypothesis explosion will be avoided.

3. The IMM-JPDA Algorithm Description. This algorithm starts with the same step as IMM-PDA algorithm, but in cycle for every particular target in the cluster.

Step 1. Computation of the mixed initial conditions for every target i and for the filter matched to model t:

a) mixed state estimate

$$\hat{x}_i^{0t}(k-1|k-1)=\sum_{s=1}^{2}\hat{x}_i^s(k-1|k-1)\mu_{s|t}^i(k-1|k-1)$$

Here, it is supposed that mixing probabilities $\mu_{s|t}^i$ are already computed.

b) mixed covariance estimate

$$P_i^{0t}(k-1|k-1)=\sum_{s=1}^{2}\mu_{s|t}{}^i(k-1|k-1)\left\{P_i^s(k-1|k-1)+\left[\hat{x}_i^s-\hat{x}_i^{0t}\right]\cdot\left[\hat{x}_i^s-\hat{x}_i^{0t}\right]'\right\}$$

Next, some JPDA steps follows.

Step 2. State prediction and covariance prediction for every target and for every model are to be calculated. Therefore, we obtain

$\hat{x}_i^{0t}(k|k-1)$ and $P_i^{0t}(k|k\text{ - }1)\cdot$

Step 3. In this step, receiving the set of measurements at scan k, a clusterization is performed. Further on, it is assumed that the algorithm will proceed with every particular cluster.

At this point, in the traditional JPDA algorithm, hypotheses generation have to be performed. However, to avoid combinatorial explosion we include here our innovation, folowing next.

Step 4. Calculation of 'predicted model probabilities' [6]:

$$\mu_i^t(k|k-1)=\sum_{s=1}^{2}p_{st}\mu_i^t(k-1)$$

Now, the individual model state predictions are merged for every particular target:

$$\hat{x}_i^0(k|k-1)=\sum_{i=1}^{2}\mu_i^t(k|k-1)\hat{x}_i^{0t}(k|k-1)$$

Step 5. We are now ready to continue with the hypotheses generation and hypotheses score computation. After generating all feasible hypotheses, hypotheses probability is computed by the expression

$$P'(H_l)=\beta^{[N_M-(N_T-N_{nD})]}(1-P_D)^{N_{nD}}P_D{}^{(N_T-N_{nD})}g_{ij}\ldots g_{mn}\cdot\quad,\qquad\text{where}$$

β - is probability density for false returns,

$g_{ij}=\dfrac{e^{-\frac{d_{ij}}{2}}}{\sqrt{|2\pi S_i|}}$ - is probability density that measurement j originates from target i, and the next additional notations: N_M - total number of measurements in the cluster, N_T - total number of targets, N_F - number of false returns, N_{nD} - number of not detected targets. The step ends with standard normalisation

$P(H_l)=\frac{P'(H_l)}{\sum_{l=1}^{N_H} P'(H_l)}$, where N_H is the total number of hypotheses.

Step 6. At this step, association probabilities are calculated. To compute for a fixed i the probability p_{ij} that observation j originates from track i we have to take a sum over the probabilities of those hypotheses in which this event occurs:

$$p_{ij} = \sum_{l \in L_{ij}} P(H_l), \qquad \text{for } j = 1,\dots,m_i(k) \text{ and } i = 1,\dots,N_t$$

where L_{ij} is a set of indices of all hypotheses, which include the event mentioned above, and N_t is the total number of targets in the cluster.

Step 7. After association probabilities computation, the JPDA algorithm continues as a PDA algorithm for every individual target. For every target the 'merged' combined innovation is computed as follows

$$\nu_i(k) = \sum_{j=1}^{m_i(k)} p_{ij} \nu_{ij}(k).$$

Step 8. This is the last step of our description. At this step, our algorithm returns to the multiple models case by splitting 'merged' combined innovation from the previous equation. For every individual target and for every particular model the combined innovations are computed:

$$\nu_i^t(k) = \nu_i(k) + H_i(k)\hat{x}_i^0(k|k-1) - H_i^t(k)\hat{x}_i^{0t}(k|k-1)$$

The last few steps of our algorithm fully coincide with the well-known IMM-PDA algorithm [1] and will be omitted.

4. Simulation results. For testing the presented algorithm we construct a range of scenarios with increasing complexity in terms of targets number and presence of clutter. The chosen scenarios include 2,3 and 4 targets with closely spaced and intercepted trajectories. The included clutter is modeled as a Poisson process with parameter βV , where β is spatial false alarm density and V is validation volume:

$$P(N = m_k|\beta V) = \frac{(\beta V)^{m_k} e^{-\beta V}}{m_k!}$$

For every scenario two levels of clutter have been tested: with $\beta V = 1$ - moderate clutter, and $\beta V = 2$ - heavy clutter. The received results can be summarized as follows:

A. Scenario with 2 targets.

The presented algorithm successfully tracks two closely spaced manoeuvring targets with crossing trajectories. The times for data processing per scan are much less than the threshold needed for real time implementation.

Table 1. Results for 2 targets scenario

βV	Average time per cluster (*sec.*)	Total time(*sec.*)	Number of scans with clusters
1	0.0088	0.78	89
2	0.012	1.1	89

B. Scenario with 3 targets.

In this case the algorithm continues to track the targets successfully. Second level of clutter ($\beta V = 2$) decreases in some extent the speed of data processing (Table 2) but the needed time per scan remains under the threshold, mentioned above.

Table 2. Results for 3 targets scenario.

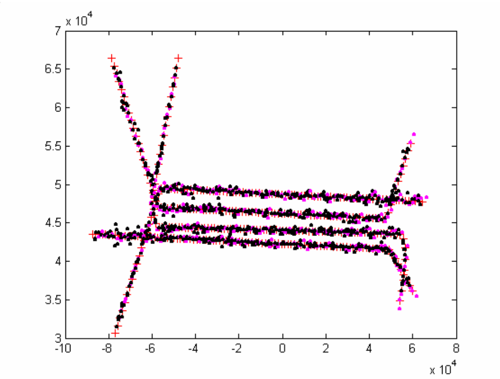
βV	Clusters with 2 targets			Clusters with 3 targets		
	Average time(*sec.*)	Total time(*sec.*)	Number of scans	Average time(*sec.*)	Total time(*sec.*)	Number of scans
1	0.016	0.55	35	0.062	4.27	69
2	0.011	0.38	35	0.136	9.39	69

C. Scenario with 4 targets.

Table 3. Results for 4 targets scenario.

βV	Clusters with 3 targets			Clusters with 4 targets		
	Average time(*sec.*)	Total time(*sec.*)	Number of scans	Average time(*sec.*)	Total time(*sec.*)	Number of scans
1	0.03	0.06	2	3.94	331	84
2	0.22	0.22	1	124.7	10598	85

This case illustrates the limit of real time implementation of the presented algorithm (Figure 1). As it can be seen from the table above (Table 3) when clutter is of moderate level ($\beta V = 1$) the algorithm is yet capable to track these targets.



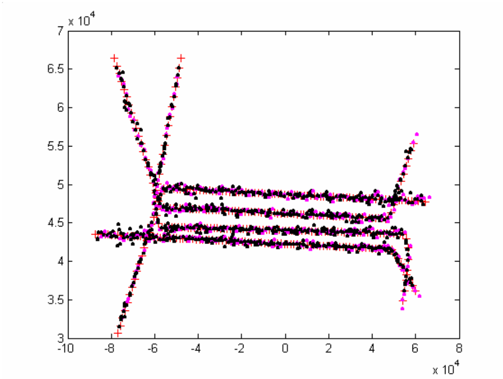


Figure 1. Four targets with crossing trajectories and Poisson parameter $\beta V = 1$ for the left, and $\beta V = 2$ for the right picture.

However, when the clutter arises ($\beta V = 2$), the processing time increases almost exponentially. In both cases of clutter level, when the formed clusters contain no more than 3 targets, the tracking process performs without problems. But when number of targets in a cluster reaches four the processing time literally explodes.

Additional experiments with five targets have been carried out, but even with missing clutter, when number of targets in a particular cluster reaches five, the processing time exceeds the threshold of about 5-10 seconds per scan.

5 Conclusions.

In this paper a new algorithm is presented unifying IMM and JPDA algorithms. The new algorithm is intended to track manoeuvring targets in clutter. For overcoming some drawbacks in putting together these two algorithms we allow an important trade-off – before hypotheses generation at a JPDA stage we merge two model dependent state predictions, avoiding in this way combinatorial explosion. In order to prove the algorithm capability to track manoeuvring targets in moderate and heavy clutter a range of experiments have been performed. The obtaining results confirm our expectations concerning the algorithm's features. In the same time the limits of real time implementation of the algorithm is outlined: with two or three targets even in a heavy clutter algorithm proceed successfully. With four targets in a moderate clutter algorithm is capable to track manoeuvring targets. For more complicated cases, however, the algorithm fails to continue tracking because of real time limitations.

REFERENCES:

[1] BAR-SHALOM, YAAKOV, THOMAS E. FORTHMANN, Tracking and Data Association, San Diego, CA, Academic Press, 1988, PP 163-170; [2] BAR-SHALOM, YAAKOV, (editor), Multitarget-Multisensor Tracking: Advanced Applications, Norwood,MA, 1990; pp 29-31; [3] BAR-SHALOM, Y.,X-R. LI, Multitarget-Multisensor Tracking: Principles and Techniques, Storrs, CT:YBS Publishing, 1995. [4] MILLER, M. L., H.S. STONE, I. J. COX, Optimizing Murty's Ranked Assignment Method, *IEEE Transactions on Aerospace and Electronic Systems,* **33**, 3 (1997), pp. 851-861. [5] COX, I. J.,M. M. MILLER, R. DANCHICK, G.E. NEWNAM, A Comparison of Two Algorithms for Determining Ranked Assignments with Application to Multitarget Tracking and Motion Correspondence, *IEEE Transactions on Aerospace and Electronic Systems,* **33**, 1 (1997), pp. 295-301. [6] JOUAN, A., B. JARRY, H. MIHALSKA, Tracking Closely Maneuvering Targets in Clutter with an IMM-JVC Algorithm, In *Proceedings of 3rd International Conference on Information Fusion FUSION'2000, Paris, France, July 10-13, 2000, pp. MoD2-10 – MoD2-16.*