An Accelerated IMM-JPDA Algorithm for Tracking Multiple Manoeuvring Targets in Clutter

Ljudmil BOJILOV, Kiril ALEXIEV, Pavlina KONSTANTINOVA

1 Introduction

The most complicated case in target tracking is undoubtedly to track multiple manoeuvring targets in heavy clutter. Numerous methods and algorithms have been devoted to this problem and for any one of them *pro* and *cons* can be pointed out. For example, the theoretically most powerful approach for tracking multiple manoeuvring targets in clutter is known to be MHT method. This method, however, more often leads to combinatorial explosion and computational overload that restricts implementation of this method. In recent years, numerous papers have been devoted to algorithms, which are capable to compute a ranked set of assignments of measurements to targets. Such algorithms make MHT approach practically implementable for the first time.

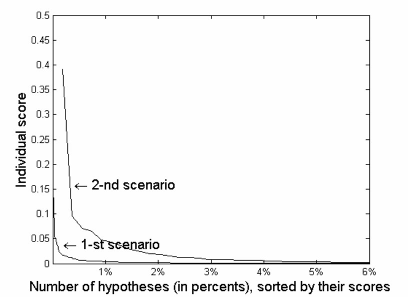
Another, and much less complicated approach, especially for tracking manoeuvring targets is known to be Multiple Models (MM) approach. The most promising algorithm based on this approach is Interacting Multiple Models algorithm. At the price of some suboptimality of its frame, this algorithm reaches the best implementation in terms of speed and stability. When assuming clutter, however, the IMM algorithm most often fails. In the case of cluttered environment, the PDA (and JPDA) approaches can be implemented. When tracking multiple closely spaced targets, the JPDA algorithm can be implemented successfully even in the presence of heavy clutter. In the previous paper[1], we have proposed an algorithm unifying at the same time features of IMM and JPDA algorithms. This algorithm proved to be good alternative of MHT approach for clusters containing up to 4 targets and moderate level of clutter. When the number of targets in the cluster exceeds this limit, however, the total number of all feasible hypotheses increases exponentially. In this paper we propose an extension of the algorithm in [1] where instead of enumeration of all feasible hypotheses we use ranked assignment approach to find the first K-best hypotheses only. The value of K has to ensure that the weight of scores-sum of this K-best hypotheses prevail over the total sum.

This paper is organized as follows. In the next section we expose our motivation for this paper as well as the problem formulation. Here a brief outline of IMM_JPDA algorithm is given and the need of its extension is discussed. In the 3rd section the extended algorithm is described. Here the stress is made over the extension part of the algorithm. In the last, 4th section simulation results are presented. These results reveal that the extended algorithm shows better performance than cited IMM_JPDA algorithm in terms of speed while at the same time keeping its stability of tracking.

2 Motivation and Problem Formulation

When several closely spaced targets form a cluster, the JPDA algorithm starts to generate all feasible hypotheses and to compute their scores. The set of all feasible hypotheses includes such hypotheses as ‘null’ hypothesis and all its ‘derivatives’. Consideration of all possible assignments inclusive the ‘null’ assignments are important for optimal calculation of assignment probabilities[6]. If, for example, the score of every one of these hypotheses differs from any of the others by no more than an order, it should not be possible to truncate some significant part of all hypotheses. If, however, prevailing share of the total score is concentrates in a small percent of the total number of all hypotheses, then temptation to consider only this small percent of all hypotheses becomes very high.

In order to investigate this idea a typical example with five closely spaced targets is used with overlapping validation regions and shared measurements. In the first run (1st scenario) 17 measurements are disposed in the target gates, and in the second run (2nd scenario) 9 measurements are disposed. At every run all feasible hypotheses are generated and their scores are computed and sumarized. The results are given in the figures below (Figure 1 and Figure 2). These two examples are chosen out of numerous experiments as typical one.



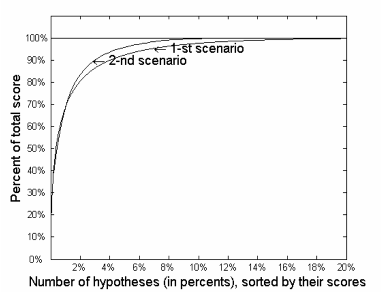


Figure 1. Hypotheses' score distribution.

Figure 2 Cumulative score distribution.

Two plots of Figure 1 show how the individual scores of the sorted feasible hypotheses are distributed. On the figure only first six percents of all hypotheses are depicted for the first end second scenario. It can be seen that the scores of the hypotheses dramatically reduce their values. Even more informative is Figure 2, where the cumulative score's distributions of the two scenarios are given. This figure confirms our expectations that only little number of all hypotheses concentrate the prevailing part of their total sum. And an additional conclusion can be derived. The first scenario is much more complicated with more than 4930 hypotheses generated. In the second scenario the generated hypotheses are approximately 550. It can be seen from the figures that for the more complicated cases the expected effect stands out more definitely.

Next, description of proposed in [1] algorithm follows. For simplicity and without losing generality two models are assumed.

2.1 IMM-JPDA Algorithm Description

The IMM JPDA algorithm starts with the same step as IMM PDA algorithm^5, but in cycle for every particular target in the cluster.

Step 1. Computation of the mixed initial conditions $\hat{x}_i^{0t}$ for every target i and for the filter, matched to model t:

a) mixed state estimate

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

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

b) mixed covariance estimate

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

Here P_i^s is covariance update of model *s* for target *i*.

Next, some JPDA steps follows.

<u>Step 2.</u> State predictions $\hat{x}_i^{0t}(k|k-1)$ and covariance predictions $P_i^{0t}(k|k-1)$ for the next scan *k* for every target and for every model are calculated.

<u>Step 3.</u> In this step, after receiving the set of measurements at scan *k*, a clustering 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.

<u>Step 4.</u> Calculating 'predicted model probabilities':

$$\mu_i^t(k|k-1)=\sum_{s=1}^{2}p_{st}\mu_i^t(k-1) \tag{3}$$

where $\mu_i^t(k-1)$ is probability that of model *t* is correct at scan *(k-1)* and p_{st} are Marcovian switching probabilities.

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

$$\hat{x}_i^0(k|k-1)=\sum_{t=1}^{2}\hat{x}_i^{0t}(k|k-1)\mu_i^t(k|k-1) \tag{4}$$

<u>Step 5.</u> We are now ready to continue with the hypotheses generation and hypotheses score computation. Hypotheses generation is another combinatorial problem, but it will be discussed in the next section.

After generating all feasible hypotheses, hypothesis probability is computed by the expression

$$P'(H_l)=\beta^{[N_M-(N_T-N_{nD})]}(1-P_D)^{N_{nD}}P_D^{(N_r-N_{nD})}g_{ij}\dots g_{mn}\cdot \tag{5}$$

where

β - is probability density for false returns,

$g_{ij} \quad \dfrac{e^{\;\frac{d_{ij}}{2}}}{(2\quad)^{M/2}\sqrt{|S|}}$ - 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, d_{ij} – statistical distance, N_{nD} - number of not detected targets. The step ends with standard normalisation

$$P(H_l) = \frac{P'(H_l)}{\sum_{l=1}^{N_H} P'(H_l)} \ , \tag{6}$$

where N_H is the total number of hypotheses.

Step 6. In 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:

$$p_{ij} = \sum_{l \in L_j} P(H_l), \qquad \text{for} \quad j = 1,\ldots,m_i(k) \text{ and } i = 1,\ldots,N_T \tag{7}$$

where L_j is a set of indices of all hypotheses, which include the event mentioned above, $m_i(k)$ is the number of measurements falling in the gate of target i, 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

$$v_i(k) = \sum_{j=1}^{m_i(k)} p_{ij} v_{ij}(k). \tag{8}$$

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

$$v_i^t(k) = v_i(k) + H_i(k)\hat{x}_i^0(k|k-1) - H_i^t(k)\hat{x}_i^{0t}(k|k-1) \cdot \tag{9}$$

The last few steps of this algorithm fully coincide with the well-known IMM PDA algorithm[5] and will be omitted.

3 Accelerating Extension to IMM JPDA Algorithm

Our extension to IMM JPDA algorithm is directed to the most time consuming part of the algorithm which concerns hypotheses generation and their scores computation. If we take as a simple example a cluster with 4 targets and 10 measurements distributed in their validation regions (Table 1), the total number of all feasible hypotheses for this example approaches 400. When, however, the number of targets in the cluster exceeds 5 or 6 and there is more than 15 measurements in their gates, the number of all hypotheses to be generated reachess thousands. To avoid these overwhelming computations we propose the next trade-off: to take into consideration only little part of all feasible hypotheses with the highest scores and concentrating the prevailing share of the total score sum.

Table 1. Indices of the measurements falling in the gates of corresponding targets.

T1	T2	T3	T4
0	0	0	0
4	6	3	1
8	7	4	2
9	8	5	3
		6	4
		9	

In order to find out the first K-best hypotheses we use an algorithm due to Murty[2] and optimized by Miller[3] *et al*. This algorithm gives a set of assignments to the assignment problem[4], ranked in increasing order of cost. As a first step in solving this problem we have to define the cost matrix of the assignment problem. It can be seen that the score of any particular hypothesis (equation (5)) is an expression of multipliers. The score of every one feasible hypothesis (i.e. the probability of being true) can be calculated using a table similar to this in Table 1, but instead indices in the boxes of the Table 1 we have to put multipliers equal to probability of assigning the given measurement to the corresponding target (Table 2).

Table 2. Multipliers of the corresponding measurements.

T1	T2	T3	T4
$\beta(1-P_D)$	$\beta(1-P_D)$	$\beta(1-P_D)$	$\beta(1-P_D)$
$g_{14}P_D$	$g_{26}P_D$	$g_{33}P_D$	$g_{41}P_D$
$g_{18}P_D$	$g_{27}P_D$	$g_{34}P_D$	$g_{42}P_D$
$g_{19}P_D$	$g_{28}P_D$	$g_{35}P_D$	$g_{43}P_D$
		$g_{36}P_D$	$g_{44}P_D$
		$g_{39}P_D$	

Now, combining indices from Table 1 in an admissible manner, and so, generating every one of feasible hypotheses we can at the same time multiply corresponding elements from Table 2, obtaining the score of the so generated hypothesis (equation (5)). As it is well known feasibility of hypothesis means two important constraints: a) no target can create more than one measurement and b) no measurement can be assigned to more than one target.

On the other side, every solution of the assignment problem represents a sum of elements of the cost matrix. We have to define this cost matrix in such way, that the value of every possible solution of the assignment can be potentially a score of some feasible hypothesis. Let us take logarithm from both sides of (5). From the left-hand side we obtain logarithm of hypothesis probability, and, from the right-hand side, a sum of logarithms of elements from Table 2. This correspondence between multipliers in equation (5) and sum of their logarithms gives a hint of how to construct the cost matrix and to solve the problem mentioned above.

We construct a cost matrix containing instead the elements of Table 2, their negative logarithms. If we find the optimal solution (in this particular case – the minimal) of assignment problem with this cost matrix it will be coincide with the hypothesis with highest probability, i.e., both the optimal solution and the highest probability hypothesis will connect the targets with the same measurements. The cost matrix of a cluster from Table 1 appears in Table 3.

Table 3. The cost matrix of the example

	f1	**f2**	**f3**	**f4**	**z1**	**z2**	**z3**	**z4**	**z5**	**z6**	**z7**	**z8**	**z9**
T1	$\ln^0$	×	×	×	×	×	×	$\ln_{14}$	×	×	×	$\ln_{18}$	$\ln_{19}$
T2	×	$\ln^0$	×	×	×	×	×	×	×	$\ln_{26}$	$\ln_{27}$	$\ln_{28}$	×
T3	×	×	$\ln^0$	×	×	×	$\ln_{33}$	$\ln_{34}$	$\ln_{35}$	$\ln_{36}$	×	×	$\ln_{39}$
T4	×	×	×	$\ln^0$	$\ln_{41}$	$\ln_{42}$	$\ln_{43}$	$\ln_{44}$	×	×	×	×	×

where

$$\ln^0 = -\ln[(1-P_D)\beta]\ ,\quad \ln_{ij} = -\ln\lfloor g_{ij}P_D\rfloor.$$

The symbol × in the matrix represents one and the same value with only requirement to be greater than the greatest element out of the set of elements denoted with $\ln$. In order to use any of the widespread assignment algorithms as well as the algorithm in [3] for finding the K-best hypotheses, the matrix from Table 3 have to be added up to square matrix filling in the remaining rows with the same value $\times$. First four columns of the matrix in Table 3 corresponds to false measurements, i.e., assigning first row to first column (or the second row to the second column, etc.) means that there is not measurement originated from this target. The columns from five to thirteen represent the corresponding measurements falling in the validation regions of the targets.

When algorithm for finding K-best assignments begins his work it will find K solutions of the problem with lowest sums of negative log-likelihood (or with highest probabilities). After receiving these K values their anti-logarithms have to be computed and so to obtain the K-best hypotheses probabilities. Next, these probabilities have to be normalized by equation (6), but now the sum is up to K:

$$P(H_l) = \frac{P'(H_l)}{\sum_{l=1}^{K} P'(H_l)}$$

Henceforth, this algorithm fully coincides with the algorithm described in the previous section, continuing with the step 6.

Here arises a question of practical importance closely connected with the proposed approach: how many hypotheses K to be found out. When deciding the value of K we have to realise that this value, in some sense, has to be optimal. On one hand, the less is the value of K the fast will be the discussed algorithm. On the other hand, however, the small values of K can lead to distortion in assignment probabilities computation (equation (7)). This question will be discussed in the next section.

4 Simulation Results

We compare the algorithm presented in this paper with the same algorithm but without acceleration discussed in previous section (algorithm from [1]) These presented algorithms were tested extensively on variety of scenarios involving different number of manoeuvring and closely spaced targets and in presence of heavy clutter. We construct a set of scenarios with 3,4 and 5 targets in a cluster and in presence of moderate and heavy clutter. The used scenarios are similar to scenarios from [1] where we look for the limit of IMM JPDA algorithm in terms of number of target in the cluster.

The first step in preparing the common frame for testing is to decide how many K-best hypotheses has to be generated. We mentioned in the end of the previous section that the value of K has to be, in some sense, optimal so that: a) to be small enough to ensure acceleration of the algorithm, and, in the same time, b) do not be so small that to give rise to distortion in computing assignment probabilities.

As it can be seen from Figure 1 the scores of feasible hypotheses decrease very rapidly and some 5-10 percents of them (Figure 2) cover more then 95 percents of the total score sum. However, as we know neither the total number, nor the total sum, we try to derive indirect criterion for determining the value of K. One possible expression can be

$$H(n) - H(n+1) < \alpha \cdot H(n)$$

where $\alpha << 1$. Here with *H(n)* the probability density of n^{th} hypothesis to be true is denoted. This criterion however did not give stable results, because very often there are subsets of hypotheses with very close values of their scores, even in the beginning of the sorted hypotheses array. Another expression that proved to be more suitable is

$$H(n) < \alpha \cdot H(1) \tag{10}$$

In order to tune experimentally the value of α a range of experimental runs have been carried out. Every one run is performed with scenario with the same number of 6 targets and 12 measurements but with different reciprocal (relative) location. Averaging over 1000 runs the next results have been received:

The first column of the Table 4 contains the different values of α, the second and third columns contain respectively the mean and the largest number (worst case) of the first N-best hypotheses in accordance with (10). The fourth and fifth columns contain mean and lowest values of ratio of the total score sum of these N-best hypotheses. In opposite to the hypotheses' number the worst case for this ratio is its lowest value.

Table 4. Number of hypotheses and ratio of the total score sum for different values of α as per equation (10)

α	N_{mean}	N_{wst}	R_{mean}	R_{wst}
0.05	32	506	0.7842	0.6185
0.01	179	1510	0.9414	0.8587
0.005	286	2074	0.9690	0.9184
0.001	632	3350	0.9942	0.9784
0.0005	779	3797	0.9973	0.9899
0.0001	1082	4459	0.9995	0.9982

Now, we can choose the most suitable value for α. For example, if we choose the value of $\alpha = 0.005$, after summation of the first 286 hypotheses we ensure, as average, to reach nearly 97% of total score sum. If we take into account that the mean of the total hypotheses number for this experiment is 9780 we can make the conclusion, that choosing the value of $\alpha = 0.005$ we can generate and process the first 3% of all feasible hypotheses ensuring 97% of total score sum. Similar conclusions can be made for α = 0.01. Consequent experiments confirm that the values 0.01 and 0.005 for α are equally appropriate.

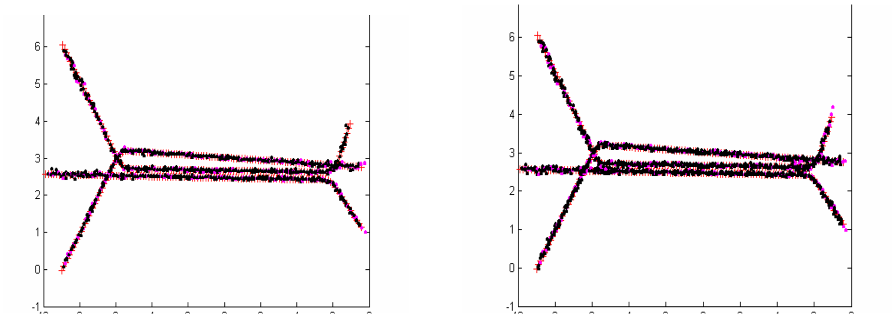


Figure 3. Three targets with crossing trajectories and Poisson parameter $\beta V = 1$ for the left, and $\beta V = 2$ for the right picture.

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 3,4 and 5 targets with closely spaced and crossing trajectories (figure 3 and 4). The included clutter have been modeled as a Poisson process with parameter βV, where B 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 receiving results can be summarized as follows:

A. Scenario with 3 closely spaced targets.

Table 4. Time per cluster in seconds for 3-targets scenario

	All hypotheses computation /targets in a cluster/		First K-best hypotheses only /targets in a cluster/	
	2 targets	3 targets	2 targets	3 targets
βV =1	0.016	0.062	0.02	0.26
βV =2	0.011	0.136	0.09	0.68

Comparison with the algorithm where all feasible hypotheses are computed gives unexpected results – this algorithm spends less processing time. Obviously the program frame for finding out the first K-best hypotheses is heavy and unsuitable for simple cases. Even so, the both approaches give results far below the real time implementation threshold.

B. Scenario with 4 closely spaced targets.

Table 5. Time per cluster in seconds for 4-targets scenario

	All hypotheses computation Targets in a cluster		First K-best hypotheses only Targets in a cluster	
Targets in a cluster	3 targets	4 targets	3 targets	4 targets
βV =1	0.03	3.94	0.79	3.39
βV =2	0.22	124.7	3.42	9.86

It can be seen in this case (Figure 3) that when scenario become denser the results become comparable (especially for clusters with 4-targets) and for the most heavy case (βV =2) the processing time for the first algorithm increases almost exponentially (Table 5). In the same time, the processing time for the new algorithm increases polinomially.

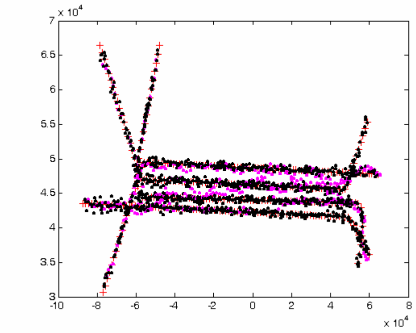


Figure 4. Four-target scenario with $\beta V = 2$

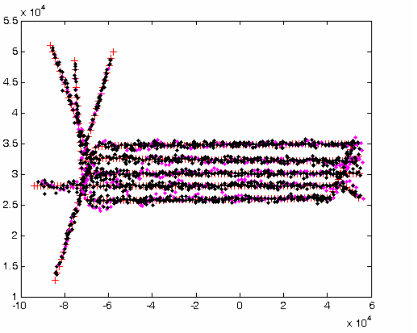


Figure 5. Five-target scenario with $\beta V = 2$

C. Scenario with 5 closely spaced targets.

For this scenario (Figure 4) only the proposed algorithm has been tested. For the most dense case, when five closely spaced targets have to be tracked in heavy clutter we compute average time per scan *t=8.7 sec*. But as it can be seen from the Table 6, when in a given scan all five targets fall into the cluster the processing time become twice the average time. It can be stated that this case is the limit of algorithm implementation.

Table 6. Time per cluster in seconds for 5-targets scenario

	First K-best hypotheses computation Targets in a cluster		
Targets in a cluster	3 targets	4 targets	5 targets
βV =1	0.35	1.16	8.2
βV =2	1.58	6.36	15.4

5 Conclusions

In this paper a new algorithm is presented for tracking closely spaced targets in moderate and heavy clutter. The algorithm is improved version of an algorithm previously presented by the authors. In the new algorithm instead of all feasible hypotheses only part of them are generated. By means of an algorithm for finding the first K-best solutions of the assignment problem we generate the firs K-best feasible hypotheses in terms of their probability of being true. This trade-off do not lead to observable assignment probability degradation and in the same time definitely speedup the algorithm processing.

Acknowlegment

The work on this paper was supported by the Center of Exellence BIS21 under Grant ICA1-2000-70016.

Nortes:

[1] Bojilov, L.V., K.M. Alexiev, P.D. Konstantinova,"An Algorithm Unifying IMM and JPDA Approaches", *Comptes Randue de l'Academie Bulgare des Sciences* (to be published).

[2] Murty, K.G., "An Algorithm for Ranking All the Assignment in Order of Increasing Cost", *Operations Research, Vol. 16, 1968, 682-687*,.

[3] Miller, M.T.,H.S. Stone, I.J. Cox, "Optimizing Murty's Ranked Assignment Method", *IEEE Transactions on AES, Vol. 33, No 3, July 1997, 851-862*,.

[4] Jonker, R., A. Volgenant, "A shortest Augmenting Path Algorithm for Dens and Sparse Assignment Problems", *Computing, Vol. 38, 1987,.325-340*,

[5] Yaakov Bar-Shalom *(editor), Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, Norwood, MA, 1990.

[6] Blackman, S.S., *Multiple-Target Tracking with Radar Applications*, Artech House, Norwood, MA, 1986.

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

Abstract. Theoretically the most powerful approach for tracking multiple targets is known to be Multiple Hypothesis Tracking (MHT) method. The MHT method, however, leads to combinatorial explosion and computational overload. By using an algorithm for finding the K-best assignments, MHT approach can be considerably optimized in terms of computational load. A much simpler alternative of MHT approach can be the Joint Probabilistic Data Association (JPDA) algorithm combined with Interacting Multiple Models (IMM) approach. Even though it is much simpler, this approach can overwhelm computations as well. To overcome this drawback an algorithm due to Murty and optimized by Miller, Stone and Cox is embedded in IMM-JPDA algorithm for determining a ranked set of K-best hypotheses instead of all feasible hypotheses. The presented algorithm assures continuos manoeuvre detection and adequate estimation of manoeuvring targets in heavy clutter. This results in a good overall target tracking performance with limited computational and memory requirements. The corresponding numerical results are presented.

Keywords. Tracking, manoeuvring, cluttered environment, assignment,

Pages: 13.