I.J. Intelligent Systems and Applications, **2011, 2, 45-51**
Published Online March 2011 in MECS(http://www.mecs-press.org/)

A New Joint Possibility Data Association Algorithm Avoiding Track Coalescence

Song-lin Chen
Xi’an Communications Institute, Xi’an, China
Email: chensonglin1024@163.com

Yi-bing Xu
Xi’an Communications Institute, Xi’an, China
Email: cslxycsl@yahoo.cn

Abstract— **For the problem of tracking multiple targets, the Joint Probabilistic Data Association approach has shown to be very effective in handling clutter and missed detections. However, it tends to coalesce neighboring tracks and ignores the coupling between those tracks. To avoid track coalescence，a K Nearest Neighbor Joint Probabilistic Data Association algorithm is proposed in this paper. Like the Joint Probabilistic Data Association algorithm, the association possibilities of target with every measurement will be computed in the new algorithm, but only the first K measurements whose association probabilities with the target are larger than others’ are used to estimate target’s state. Finally, through Monte Carlo simulations, it is shown that the new algorithm is able to avoid track coalescence and keeps good tracking performance in heavy clutter and missed detections.**

Index Terms—**Joint Probabilistic Data Association, track coalescence, K Nearest Neighbor, Scaled Joint Probabilistic Data Association.**

I. Introduction

Joint Probabilistic Data Association (JPDA) algorithm [1] is usually used to solve the problem of tracking multiple targets under the environment of heavy clutter. But the traditional JPDA algorithm will cause track coalescence when the targets are parallel neighboring or small-angle crossing. The adjacent tracks will be attracted by each other and gradually change to be coalescent in above scenarios. The main reason of this phenomenon is that the adjacent targets may share the same measurement as their main measurement in the JPDA algorithm.

The Exact Nearest Neighbor Probabilistic Data Association (EN-NPDA) algorithm has shown that associated events pruning may be an effective way to prevent track coalescence. EN-NPDA [2] involves the enumeration of all peak-to-target association hypotheses as JPDA, but only the most likely hypothesis is retained. While this approach inhibits compound coalescence, it was found to lead to track over commitment and an increased incidence of track divergence in the presence of clutter and absent target detections. Most of the modified JPDA algorithms avoiding track coalescence by means of appropriately pruning associated events or using a scaling factor to produce a new algorithm whose effect ranging between EN-NPDA and JPDA algorithm. The paper [3] proposed a Scaled JPDA (SJPDA) algorithm which introduced a scale factor. When the scale factor value is infinity, the algorithm is equivalent to the EN-NPDA algorithm. And when the scale factor value is one, the algorithm is equivalent to the JPDA algorithm. So when the scale factor value is an appropriate intermediate value, the SPDA algorithm not only can avoid track coalescence, but also can control track divergence in the presence of clutter and absent target detections. But the scale factor comes from the experience.

In this paper, a K Nearest Neighbor Joint Probabilistic Data Association (KNNJPDA) algorithm is proposed. Through the Monte Carlo simulations, track coalescence is avoided and significantly improved efficiency is shown when the KNNJPDA algorithm is used to track targets.

The paper is organized as follows. Target and measurement models are described in Section II, and the steps of the JPDA algorithm are described in Section III, the Principle and Steps of the Modified JPDA Algorithm are described in Section IV. Section V presents simulations that demonstrate the effectiveness of this approach, and conclusions are given in Section VI.

II. Target and Measurement Models

We consider c targets in single sensor system, the target states and measurements in the next time interval can be predicted by

$$x^t(k) = F^t(k-1)x^t(k-1) + v^t(k-1), \tag{1}$$

$$z_j(k) = H(k)x^t(k) + w(k), \tag{2}$$

Where $x^t(k)$ is the n-dimensional state of the target at time k; $F^t(k-1)$ is state transition matrix; $v^t(k-1)$ is a process noise vector which follows the multivariate normal distribution, $N(0;Q)$, where N denotes Gaussian distribution, and Q is the covariance matrix of the process noise vector, $H(k)$ is the measurement matrix, $w(k)$ is the measurement noise vector which is assumed to follow the multivariate normal distribution, $N(0;R)$, where R is

The work of Song-lin Chen was sponsored by Natural Science Foundation of Shanxi Province: S2011JC5766.

the covariance matrix of the measurement noise vector. Let $Z(k)$ be the set of measurement vectors at frame *k*:

$$Z(k)=\{z_1(k),z_2(k),...,z_{m_k}(k)\}, \tag{3}$$

m_k is the number of measurement vectors at time *k.*

III. Steps of the JPDA Algorithm

JPDA is a multi-target extension of Probabilistic Data Association [4-6] which considers each track in isolation. It takes into account the fact that a measurement may fall inside the intersection of two or more validation gates of several different targets and so could have originated from any of these targets or from clutter.

Step 1: *Target state initialization*:

$\hat{x}^t(0|0)$ ， $P^t(0|0)$ are the initial state vector and covariance, which are given by system.

Step 2: *Computing the state and covariance prediction*:

$$\hat{x}^t(k|k-1)=F^t(k-1)\hat{x}^t(k-1|k-1), \tag{4}$$

$$P^t(k|k-1)=F^t(k-1)P^t(k-1|k-1)[F^t(k-1)]^T+Q^t(k-1), \tag{5}$$

Step 3: *Affirming validation measurements*:

Z_k^t is the set of validation measurements which originate from target *t* at time *k* [7]. It follows

$$Z_k^t=\left\{z_j(k)\middle|d_k^2\le\gamma, j=1,...,m_k\right\}, \tag{6}$$

$$d_k^2=v_t^j(k)^T[S^t(k)]^{-1}v_t^j(k), \tag{7}$$

Where γ is the gating size, $v_t^j(k)$ is the measurement residual and $S_t^j(k)$ is the residual covariance.

$$v_t^j(k)=z_j(k)-H(k)\hat{x}^t(k|k-1), \tag{8}$$

$$S^t(k)=H(k)P^t(k|k-1)H(k)^T, \tag{9}$$

Step 4: *Generating the validation matrix and the feasible matrix*:

The validation matrix which is defined as follows:

$$\Omega=[\omega_{jt}], j=1,2,...,m_k; t=1,2,...,c, \tag{10}$$

Where ω_{jt} is binary element, and $\omega_{jt}=1$ stands for the measurement *j* lies in the validation gate of the target *t.* Index $t=0$ stands for “no target.” and the corresponding column of Ω has all units-each measurement could have originated from clutter or false alarm. m_k is the number of the validation measurements and *c* is the number of the targets.

After generating validation matrix, we will split validation matrix to generate feasible matrices, which must follow two principles:

a) There is only one nonzero element in every row of feasible matrix after splitting validation matrix;

b) There is not more than one nonzero element in every column of feasible except the first column.

Each joint event θ_i can be represented by a feasible matrix.

$$\hat{\Omega}(\theta_i)=[\hat{\omega}_{jt}(\theta_i)], \tag{11}$$

It consists of the units in Ω corresponding to the associations assumed in event θ_i.

Step 5: *Computing association probabilities*:

$$\beta_{jt}=\sum_{i=1}^{n_k}P\{\theta_i|Z^k\}\hat{\omega}_{jt}(\theta_i), \tag{12}$$

Where $P\{\theta_i|Z^k\}$ is the probability of the joint feasible event, Z^k is the set of all validation measurements.

$$\hat{\omega}_{jt}(\theta_i)=\begin{cases}1 & \theta_{jt}\subset\theta_i\\ 0 & otherwise\end{cases}, \tag{13}$$

Where θ_{jt} stands for the association event that $z_j(k)$ originate from target *t.*

Step 6: *State estimate*:

The state vectors $\hat{x}^t(k|k)$ and the covariance matrices $P^t(k|k)$ of the targets are updated as

$$\hat{x}^t(k|k)=\hat{x}^t(k|k-1)+K^t(k)\sum_j\beta_{jt}{''}v_t^j(k), \tag{14}$$

$$P^t(k|k)=P^t(k|k-1)-(1-\beta_{0t}{''})K^t(k)S^t(k)K^t(k)^T$$
$$+K^t(k)\{\sum_j\beta_{jt}{''}v_t^j(k)v_t^j(k)^T-\sum_j\beta_{jt}{''}v_t^j(k)[\sum_j\beta_{jt}{''}v_t^j(k)]^T\}K^t(k)^T, \tag{15}$$

$$K^t(k)=P^t(k|k-1)H(k)[S^t(k)]^{-1}, \tag{16}$$

Set *k=k+1*, and go to Step 2.

IV. Principle and Steps of the Modified JPDA Algorithm

A. Principle of the Modified JPDA Algorithm

The association possibilities of the target with every measurement will be computed in the KNNJPDA algorithm, but only the first K measurements whose association probabilities with the target are larger than others’ are used to estimate target’s state. When K=1, the new algorithm is equivalent to the EN-NPDA algorithm. And when K is given a large value, the algorithm is equivalent to the JPDA algorithm. So when K is given an appropriate intermediate value, the new algorithm not only can avoid track coalescence, but also can prevent track divergence in the presence of clutter and absent target detections.

In SJPDA algorithm, an arbitrary positive scaling factor is introduced to multiply the maximum probabilities of every target associated with measurements. The scaling factor ranges from one to infinity. When the scale factor value is infinity, the algorithm is equivalent to the EN-NPDA algorithm. And when the scale factor value is one, the algorithm is equivalent to the JPDA algorithm. When the scale factor value is an appropriate intermediate value, the SJPDA algorithm can avoid track coalescence, and track divergence can be controlled slightly too.

B. Steps of the Modified JPDA Algorithm

Comparing with the traditional JPDA algorithm, the KNNJPDA algorithm adds one step: Association event pruning.

Step: *Association event pruning*

Comparing the association probabilities of a target with every measurement, the first K measurements whose association probabilities with the target are larger than others' are looked as the last validation measurements of the target. For example, assuming that $\beta_{11},\beta_{21},...,\beta_{m_k 1}$ is the association probabilities of target 1 with every measurement and $\beta_{11}\ge\beta_{21}\ge...\ge\beta_{m_k 1}$, $z_1(k),z_2(k),...,z_K(k)$ corresponding $\beta_{11},\beta_{21},...,\beta_{K1}$ will be looked as the last validation measurements of target 1 and $\beta_{(K+1)1},\beta_{(K+2)1},...,\beta_{m_k 1}$ will be set at 0.

After pruning the association events, the association probabilities of each target with measurements are normalized.

$$\beta_{jt}' = \frac{\beta_{jt}}{\sum_{j=1}^{K}\beta_{jt}}, \qquad (17)$$

This step will be added before the sixth step in the JPDA algorithm.

Comparing with the traditional JPDA algorithm, the SJPDA algorithm also adds one step: Multiplying the Scaled factor.

Step: *Multiplying the Scaled factor*

Finding out the maximum probability measurement associated with the target.

Assuming that $\beta_{r_1 1}=\max\limits_{1\le j\le m_k}\beta_{j1}$, $\beta_{r_2 2}=\max\limits_{1\le j\le m_k}\beta_{j2}$,…, $\beta_{r_c c}=\max\limits_{1\le j\le m_k}\beta_{jc}$, so $z_{r_1}(k)$, $z_{r_2}(k)$,…, $z_{r_c}(k)$ are respectively the exclusive measurements of target *1*, target *2*,…,target *c*.

A factor is used to multiply the maximum probabilities of every target associated with measurements. The value of the factor ranges from one to infinity.

$$\beta_{r_i 1} = factor\times\beta_{r_i 1}, i=1,2,...,c\ ,1\le factor\le+\infty\ ,(18)$$

This step will be added before the sixth step in the JPDA algorithm too. In Scaled JPDA algorithm, the scale factor must come from the experience. Through simulation test, the best tracking performance can be won when $300\le factor\le 800$.

V. Simulation Test

A. Simulation Scenario Settings

In case of two targets, we respectively use the JPDA algorithm and the KNNJPDA algorithm to track flight target *a* and target *b* in the cluttered environment. Two targets stay at constant velocity. Sampling cycle *T* is 1 second, and gating size γ is 10, 60 sampling points for each target. There are 6 clutters coming into the targets' associated gates at a sample on the average.

$$F(T)=\begin{bmatrix}1 & T & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & T\\ 0 & 0 & 0 & 1\end{bmatrix},\quad Q(T)=\begin{bmatrix}T^2/2 & 0\\ T & 0\\ 0 & T^2/2\\ 0 & T\end{bmatrix}, \qquad (19)$$

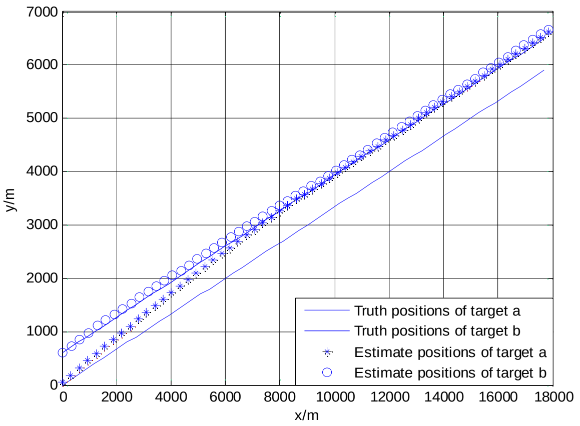
$$H(k)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}, \qquad (20)$$

***Parallel neighboring scenario (scenario 1)*:** Target *a* and *b* flight from southwest to northeast in parallel. The initial positions are (0,0) and (0,600), *x* velocities of two targets are 300 *m/s*, the mean error of measurement is 300 meters, *y* velocities of two targets are 100 *m/s*, the mean error of measurement is 100 meters.

***Small-angle crossing scenario (scenario 2)*:** The initial positions of target *a* and *b* are respectively (0,0) and (0,600), *x* velocity of *a* target is 300 *m/s*, the mean error of measurement is 300 meters, *y* velocity of *a* target is 0 *m/s*, the mean error of measurement is 100 meters. The *x* velocity of *b* target is 300 *m/s*, the mean error of measurement is 300 meters, *y* velocity of *b* target is -50 *m/s*, the mean error of measurement is 100 meters.

B. Simulation Results and Analysis Comparison Results and Discussion

We use the JPDA algorithm to track flight targets for 50 Monte Carlo runs in scenario 1 and 2, the simulation results as follows:



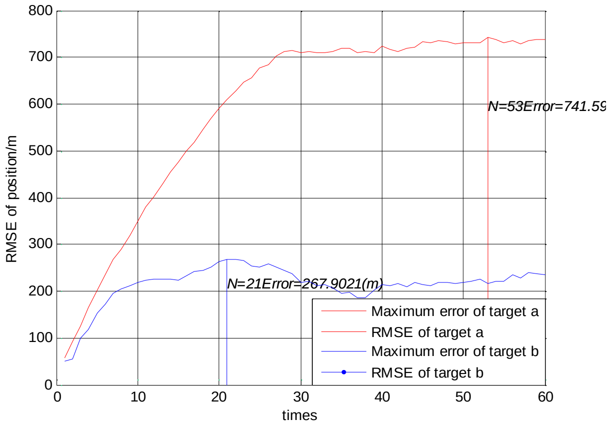
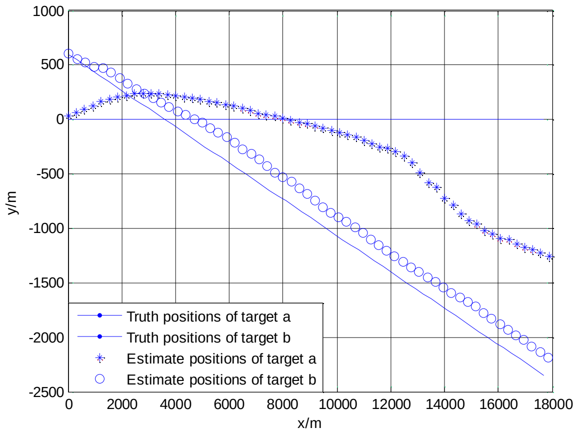


Figure 1.Trajectories of two targets using JPDA in scenario 1

Figure 2.RMSE of position using JPDA in scenario 1



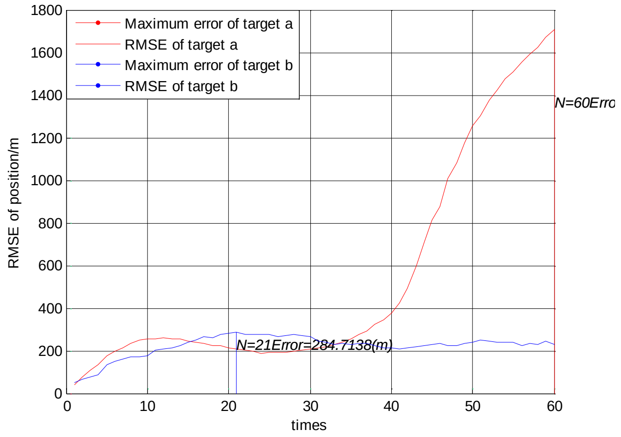


Figure 3.Trajectories of two targets using JPDA in scenario 2

Figure 4.RMSE of position using JPDA in scenario 2

Fig. 1 and Fig. 2 show the trajectories of two targets and the root mean squared error (RMSE) of position using the JPDA algorithm in scenario 1. The tracks of two targets gradually merge together and serious track coalescence happens in Fig. 1. The position RMSE of target *a* is very large in Fig. 2 and the maximum position error of target *a* is close to 800 m. So we can see that track coalescence will be caused using the JPDA algorithm in the parallel neighboring scenario.

Fig. 3 and Fig. 4 show the simulation result using the JPDA algorithm in scenario 2. In Fig. 3, the trajectory of target *a* always tends to close with the trajectory of target *b*. The position RMSE of target *a* gradually changes to be very large after the 30 seconds. It is shown that track coalescence will be caused using the JPDA algorithm in the small-angle crossing scenario.

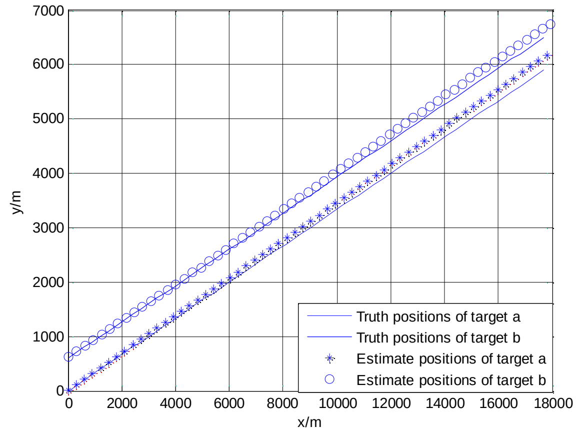


Figure 5.Trajectories of two targets using SJPDA in scenario 1($factor = 300$)

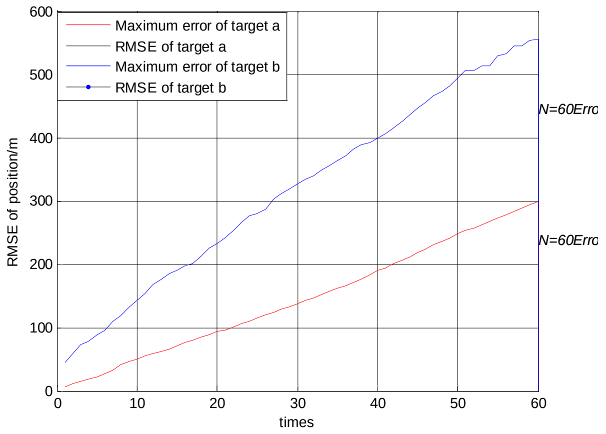


Figure 6.RMSE of position using SJPDA in scenario 1($factor = 300$)

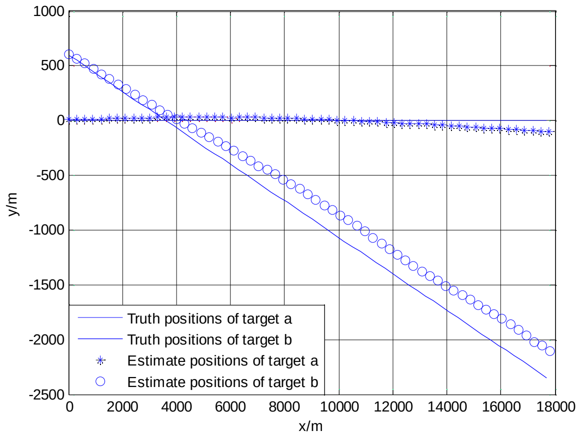


Figure 7.Trajectories of two targets using SJPDA in scenario 2($factor = 300$)

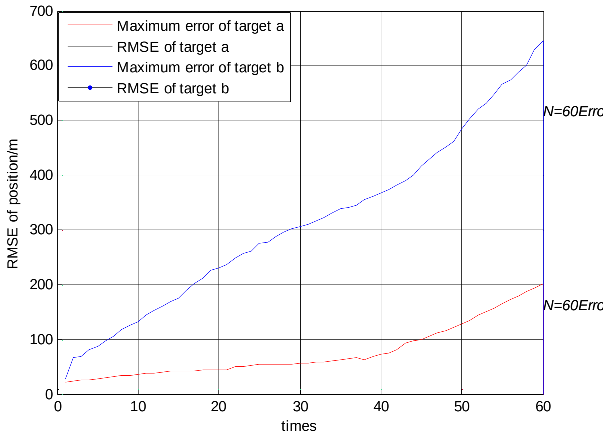


Figure 8.RMSE of position using SJPDA in scenario 2($factor = 300$)

Then, we use the SJPDA algorithm to track flight targets for 50 Monte Carlo runs in scenario 1 and 2. Fig. 5 and Fig. 6 respectively show the trajectories of two targets and the root mean squared error (RMSE) of position using the SJPDA algorithm in scenario 1($factor = 300$). Comparing Fig. 5 with Fig. 1, we can see that the track coalescence is controlled in Fig. 5. At the same time, the position RMSE of target *a* is very small, but the position RMSE of target *b* is very large in Fig. 6. It is shown that the SJPDA algorithm can avoid track coalescence, but track divergence can't be totally prevented using the SJPDA algorithm in the small-angle crossing scenario.

Fig. 7 and Fig. 8 respectively show the trajectories of two targets and the root mean squared error (RMSE) of position using the SJPDA algorithm in scenario 2($factor = 300$). In this case, we can see that the track coalescence can be also avoided by the SJPDA algorithm, but the trajectory of target *b* deviates from the real track and the tracking error is large.

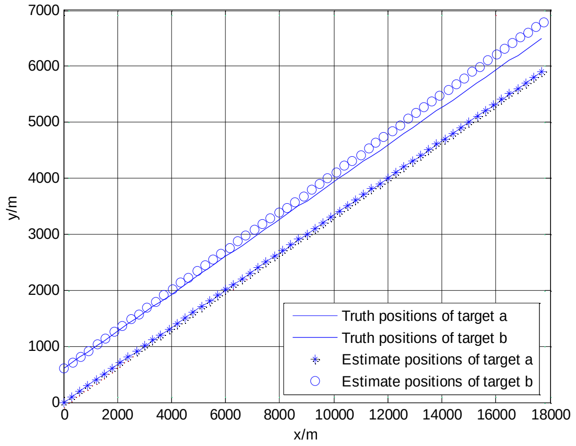


Figure 9.Trajectories of two targets using KNNJPDA in scenario 1(K=1)

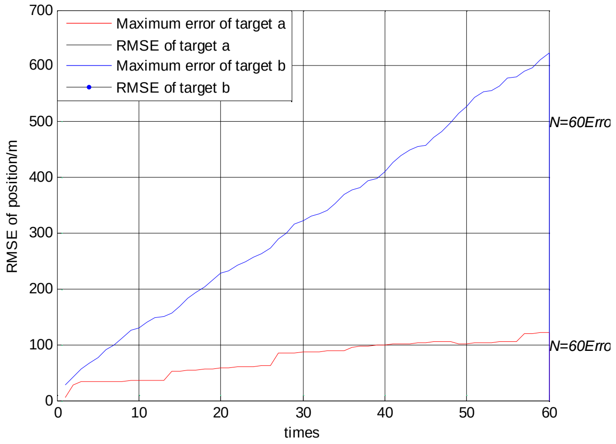


Figure 10.RMSE of position using KNNJPDA in scenario 1(K=1)

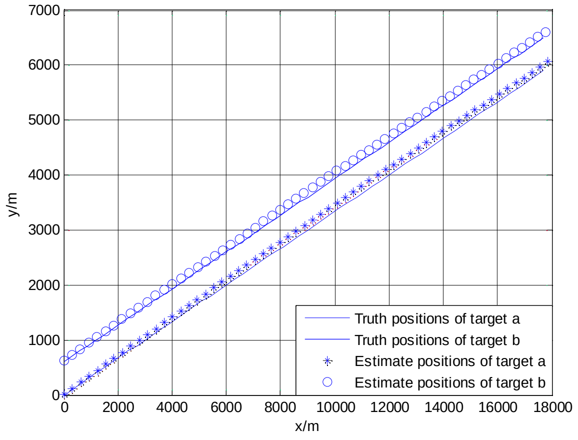


Figure 11.Trajectories of two targets using KNNJPDA in scenario 1(K=2)

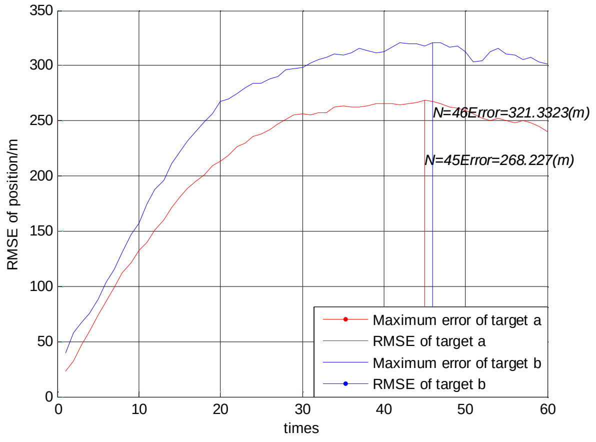


Figure 12.RMSE of position using KNNJPDA in scenario 1(K=2)

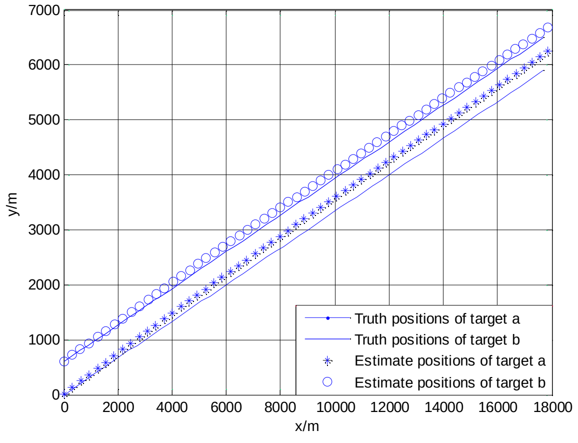


Figure 13.Trajectories of two targets using KNNJPDA in scenario 1(K=3)

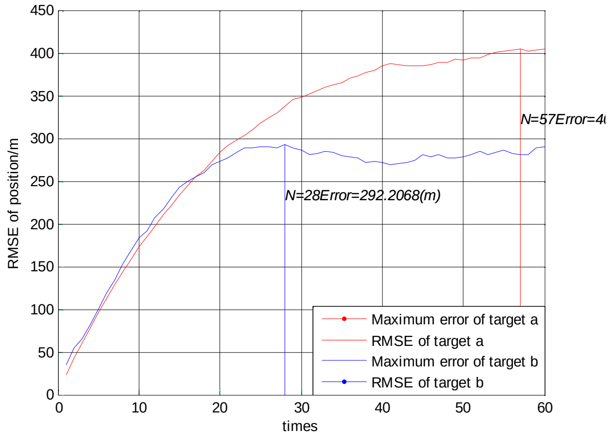


Figure 14.RMSE of position using KNNJPDA in scenario 1(K=3)

Fig. 9 and Fig. 10 show the trajectories of two targets and the root mean squared error (RMSE) of position using the KNNJPDA algorithm (K=1) in scenario 1. It is shown that track coalescence is avoided in this case but the trajectory of target *b* deviates severely from the real track. This simulation result is similar to the simulation result of ENNPDA.

When K=2, the simulation result of the KNNJPDA algorithm is shown in Fig 11 and Fig 12. The tracks don't merge together in Fig. 11. In Fig. 12 the estimated position errors of two targets are small. It can be confirmed that the KNNJPDA algorithm (K=2) can avoid track coalescence, and ensure very good track precision in scenario 1.

When K=3, the simulation result of the KNNJPDA algorithm is shown in Fig. 13 and Fig. 14. Slight track coalescence appears in Fig. 13 and the RMSE of target *a* in Fig. 14 is larger than it's in Fig. 12. In fact, if K is given bigger value, track coalescence will be more serious and the RMSE of target *a* will be larger. When K ≥5, the simulation result of the KNNJPDA algorithm will be similar to JPDA's.

Comparing Fig. 11 and Fig. 12 with Fig. 5 and Fig. 6, we can see that the tracking precision of KNNJPDA is better than SJPDA's in the parallel neighboring scenario.

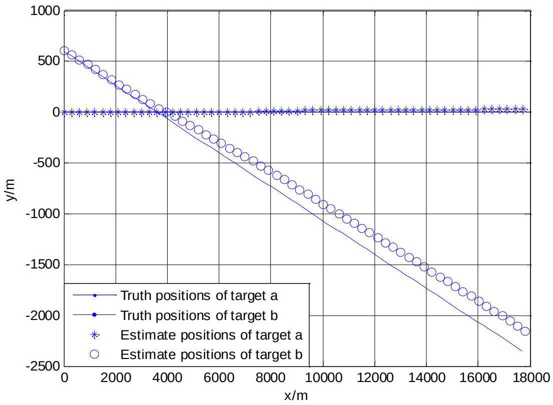


Figure 15.Trajectories of two targets using KNNJPDA in scenario 2(K=1)

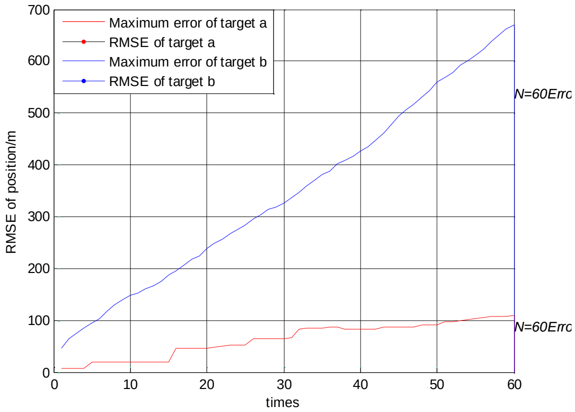


Figure 16.RMSE of position using KNNJPDA in scenario 2(K=1)

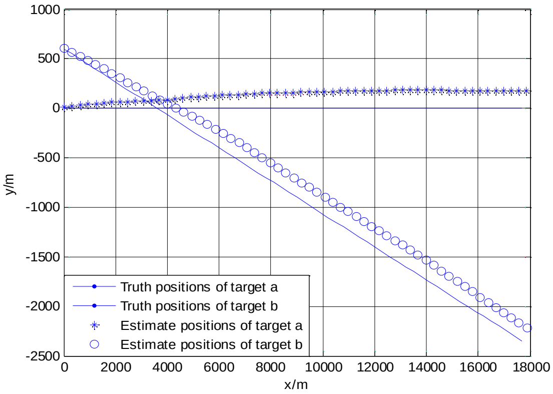


Figure 17.Trajectories of two targets using KNNJPDA in scenario 2(K=2)

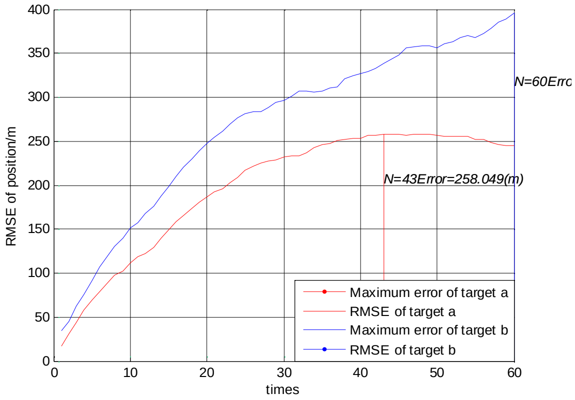


Figure 18.RMSE of position using KNNJPDA in scenario 2(K=2)

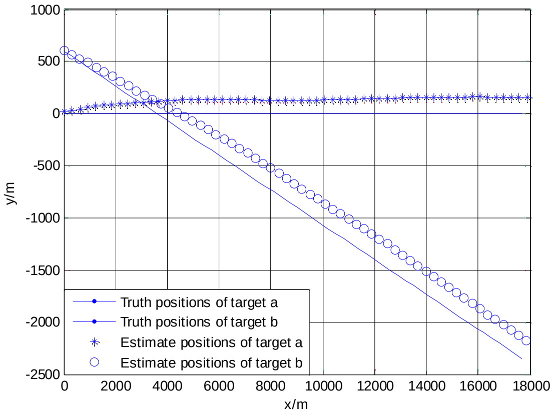


Figure 19.Trajectories of two targets using KNNJPDA in scenario 2(K=3)

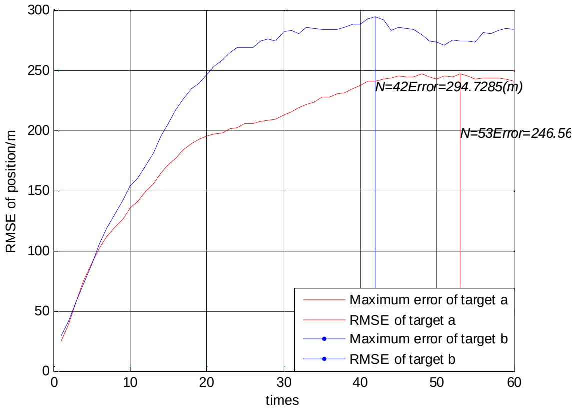


Figure 20.RMSE of position using KNNJPDA in scenario 2(K=3)

The simulation results of the KNNJPDA algorithm in scenario 2 are shown in Fig. 15-20. When K=1, the simulation result of the KNNJPDA algorithm is shown in Fig. 15 and Fig. 16. It can be seen that track coalescence is avoided in this case but the trajectory of target *b* deviates severely from the real track and track divergence is caused.

When K=2, the simulation result of the new algorithm is shown in Fig. 17 and Fig. 18. Comparing Fig. 16 with Fig. 18, the position RMSE of target *b* in Fig. 18 is smaller than the position RMSE of target *b* in Fig. 16. It is confirmed that track divergence is controlled by the KNNJPDA algorithm when K=2.

When K=3, the simulation result in the same case is shown in Fig. 19 and Fig. 20. It's shown that track coalescence is avoided and the new algorithm ensures very good track precision in this case. In fact, when K=4,5,6, the simulation result of the KNNJPDA algorithm is similar to the simulation result when K=3. When K≥7, the simulation result of the KNNJPDA algorithm will be similar to JPDA's.

Comparing Fig. 19 and Fig. 20 with Fig. 7 and Fig. 8, we can see that the tracking precision of KNNJPDA is also better than JPDA's in the small-angle crossing scenario.

In addition, the time of the SJPDA algorithm for 50 Monte Carlo simulation runs is approximately 34.38s in scenario 1, and the time of the KNNJPDA algorithm for 50 MC simulation runs is approximately 27.66s in the same case. Thus we can see that the computation of SJPDA is bigger than KNNJPDA's.

V. Conclusions

In this paper, a KNNJPDA algorithm has been proposed and implemented. The association possibilities of the target with every measurement will be computed in the new algorithm, but only the first K measurements whose association probabilities with the target are larger than others' are used to estimate target's state. The results of simulation test show that, when K=2 or K=3, the new algorithm can avoid track coalescence and ensure good precision in all scenarios. When K=1, the new algorithm is equivalent to the ENNPDA algorithm and track divergence can't be avoided in the presence of clutter. In parallel neighboring scenario, the best track precision can be achieved using the new algorithm when K=2, and the track effect of the new algorithm is similar to JPDA's when K≥5. In small-angle crossing scenario, the best track precision can be achieved using the new algorithm when K=3,4,5,6, and the track effect of the new algorithm is similar to JPDA's when K≥7.

In addition, the simulation results of SJPDA have been predicted in this paper. Through comparing the SJPDA's simulation results with the KNNJPDA's, it has been confirmed that two algorithms all can avoid track coalescence but the performances of the KNNJPDA algorithm in tracking precision and computation are better than SJPDA's.

References

[1] Y. Bar-Shalom and T. E. Fortmann. Tracking and Data Association. *New York: Academic* , 1988.

[2] R. J. Fitzgerald, Development of practical PDA logic for multitarget tracking by microprocessor, in Multitarget-Multisensor Tracking:Advanced Applications, Y. Bar-Shalom, Ed. Reading, MA: Artech House, 1990. pp: 1–23.

[3] Hugh L. Kennedy .Controlling Track Coalescence with Scaled Joint Probabilistic Data Association. *IEEE RADAR* 2008, pp：440-445.

[4] Y. Bar-Shalom and E. Tse. Tracking in a Cluttered Environment with Probabilistic Data Association. Automatica, vol. 11. Sep.1975. pp:451-460.

[5] T. Kirubarajan and Y. Bar-Shalom. Probabilistic data association techniques for target tracking in clutter. *Proceedings of the IEEE*, vol.92, no. 3. Mar. 2004. pp: 536-557.

[6] Y. Bar-Shalom, T. Kirubarajan and X. Lin. Probabilistic data association techniques for target tracking with applications to sonar, radar and EO sensors. *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8. Aug. 2005. pp: 37-56.

[7] Seokwon Yeom, Efficient Multi-target Tracking with Sub-event IMM-JPDA and One-point Prime Initialization. *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* Seoul, Korea, 2008, pp:451-456.

Song-lin Chen was born in Hunan, China in 1983. He graduated in communication engineering from the Air Force Engineering University in 2003. He is the Master of Computer Application Technology and studying at Xi'an Communications Institute in Xi'an. His research areas include Multi-target Tracking and Data Association. He has published several papers which were retrieved by EI, ISTP.

Yi-bing Xu was born in 1965. He received the M.Sc. degree in control theory from National University of Defense Technology and the PhD degree in Information Technologies from the Xi'an Jiaotong University in 1990 and 2006, respectively.

He then joined the Xi'an Communications Institute, where he is currently a vice-professor. His current research interests include information fusion and information security.