Multi-Sensor Multi-Target Passive Locating and Tracking

Mei Liu, Nuo Xu, and Haihao Li

Abstract: The passive direction finding cross localization method is widely adopted in passive tracking, therefore there will exist masses of false intersection points. Eliminating these false intersection points correctly and quickly is a key technique in passive localization. A new method is proposed for passive locating and tracking multi-jammer target in this paper. It not only solves the difficulty of determining the number of targets when masses of false intersection points existing, but also solves the initialization problem of elastic network. Thus this method solves the problem of multi-jammer target correlation and the elimination of static false intersection points. The method which dynamically establishes multiple hypothesis trajectory trees solves the problem of eliminating the remaining false intersection points. Simulation results show that computational burden of the method is lower, the elastic network can more quickly find all or most of the targets and have a more probability of locking the real targets. This method can eliminate more false intersection points.

Keywords: Data clustering, elastic net, false intersection points, passive locating and tracking.

1. INTRODUCTION

Eliminating the false intersection points is a difficult problem in the bearing angles-only tracking system [1]. In the passive tracking system the number of targets is not known. Because of the existence of false intersection points, determining the number of targets exactly becomes a difficult problem. Most of the methods of determining the number of targets are based on the maximum likelihood principle. The likelihood functions of number and state of targets are firstly needed in this method and then are obtained by using numerical method. When using this method, the estimation of the number of targets is biased and the estimation of state is prone to being trapped in local minima. Unbiased estimation can be got by combing this method with other algorithm (such as simulated annealing algorithm).

However, the method above needs a great deal of computation and the accurate algorithm is prone to being trapped in local minima too. There are many kinds of other methods to determine the number of targets such as integer programming approach and genetic algorithm. The integer programming approach involves simultaneously solving a large system of equations [2]. This approach has poor scaling characteristics which lead to impractical computational demands. Although the genetic algorithm can find good solutions, this method involves a coding scheme which makes the solution space increase exponentially. Therefore it is computationally prohibitive with a large number of targets and sensors [3].

Because the bearing angle information is not a complete description of position, the false intersection points appearing in the bearing angle measuring system have great impact on the estimation of state. Adding new sensors to the system can solve the problem of false intersection points. Although using more sensors, with the number of targets increasing, some new false intersection points will appear. Meanwhile when the number of sensors is larger than three, this problem is NP-hard.

Locating multiple targets with passive sensors, in essence, requires finding small clusters of points generated by the intersection of pairs of direction vectors emanating from each pair of sensors toward each target. The size of the solution space to be searched increases exponentially with the number of targets and sensors. A number of attempts have been made at solving the multi-target multi-sensor passive tracking problem. These have included correlation based techniques [4,5], integer programming [6], Lagrangian relaxation [2]. With the development of neural network, such as genetic algorithm [7], Hopfield neural network [8,9] and self-organizing neural network, solving the problem of target tracking and locating became one of the major factors leading to the resurgence of interest in the field of neural

Manuscript received May 12, 2006; revised December 20, 2006; accepted February 2, 2007. Recommended by Editor Jae Weon Choi. This work is supported by Heilongjiang Province Science and Technology Plan Term (No.2005G1472-00) and National Natural Science Foundation of China (No.60172036).

Mei Liu, Nuo Xu, and Haihao Li are with the Department of Electronic and Communication Engineering, Harbin Institute of Technology, Harbin, Heilongjiang Province, China (e-mails: liumei@hit.edu.cn, echski26@hotmail.com, lihaihao @sohu.com)

network in the early 1990's. Self-organizing neural network which was proposed by Kohonen in 1981 is a kind of self-organizing map (SOM). We will introduce the multiple elastic modules (MEM) model which generalizes the self-organizing principles of the SOM, to make the model amenable to a wide range of difficult optimization problems, such as computer vision and DNA sequence. Hopfield neural network is prone to being trapped in local minima with poor solutions. Self-organizing neural network, such as the elastic network, has been successfully applied to geometric combinatorial optimization problems, specifically the traveling salesman problem.

Applying multiple elastic modules model to the passive tracking problem we can construct disjoint fully connected subgraphs in order to solve the combinatorial optimization problem having to identify from a large number of possible selections of intersection points, those that form a small spatial cluster [10-12]. The disadvantage of this method is that the number of targets must be known at first but it is difficult to estimate the number of targets. The selecting of initial value has great impact on the elastic neural network. Proper initial value not only cuts the computational burden down but also conduces to the convergence of the method to correct intersection points. Otherwise, elastic neural network will not lock the correct intersection points and diverge. Up till now, as a result of not having a proper method to select initial value, the probability of locating targets correctly is low. In the multi-target multi-sensor passive tracking system, there are two kinds of intersection points. One maintains large spatial distribution and the other maintains small spatial distribution. Applying multiple elastic modules model for the optimization problem can eliminate the first kind of intersection points but not the second kind. These accidental target-like arrangements of intersection points are referred to as ghost targets. The only method for discriminating between real targets and ghost targets is to monitor the configuration of intersection points for a period of time. The intersection points forming a ghost target will eventually diverge whereas those associate with a real target will maintain their small spatial distribution. Hence the method of dynamic tracking can eliminate the second kind of intersection points.

To cope with the problem encountered above, a new method for solving the multi-target multi-sensor passive tracking problem is proposed in this paper. This method can be used for passive tracking multi-jammer target and eliminate ghost targets dynamically.

The rest of this paper is organized as follows. Section 2 gives problem description and method for solving the problem. Section 3 illustrates data clustering for direction finding cross localization. The multiple elastic modules model is briefly introduced in Section 4. Section 5 describes the approach of eliminating ghost targets dynamically. Simulation results and analysis are given in Section 6. Conclusions are summarized in Section 7.

2. PROBLEM DESCRIPTION AND METHOD FOR SOLVING THE PROBLEM

2.1. Problem description

The general problem can be stated as follows. A number of passive sensors s, s=1,2,3,……S are used to detect the presence of targets in a particular surveillance area. Each sensor makes a number of measurements $\theta_{Si}(i=1,2,3,\cdots,N_s)$, indicating the bearing angles of targets relative to the sensor location. An example of passive tracking scenario with three targets and three sensors is given in Fig. 1. Targets positions are found by identifying those areas which are bearing lines intersecting one another from different sensors in the surveillance region. However, duo to the geometry of this problem, the intersection points can be sorted into two kinds. Even a perfect conglomeration of intersection points does not guarantee a real target, as seen the point K at the center of Fig. 1. The other kind of intersection point is point L.

This optimization problem can be formulated as selecting intersection points of different types which form a small cluster in space. In the particular case with three sensors, there are three different intersection types (Q=3) generated from pairings of sensors {1, 2}, {1, 3} and {2, 3}, as show in Fig. 2.

If we associate a particular "type" to each possible pairing of sensors and take S sensors for example, the

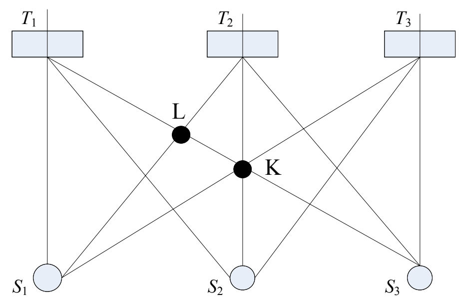


Fig. 1. An example of passive tracking scenario with three targets and three sensors.

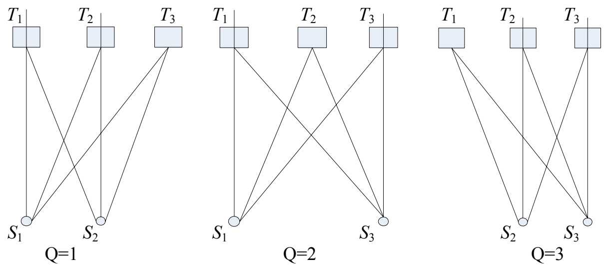


Fig. 2. Types of intersection points.

total number of distinct types will be $Q = C_S^2 = S(S-1)/2$. In this scheme, each pair of bearing angles θ_{qi} and θ_{rj}, from two different sensors, intersect at a specific point $X(x_{qirj}, y_{qirj})$ given by

$$x_{qirj} = \frac{x_q \tan\theta_{qi} - x_r \tan\theta_{rj}}{\tan\theta_{qi} - \tan\theta_{rj}} + \frac{y_r - y_q}{\tan\theta_{qi} - \tan\theta_{rj}}, \quad (1)$$

$$y_{qirj} = y_q + (x_{qirj} - x_q)\tan\theta_{qi}. \quad (2)$$

Thus the coordinate of intersection points of the same type can be given by $X_Q = \{(x_{qirj}, y_{qirj}) \mid i = 1, 2, \cdots, T_q, j = 1, 2, \cdots, T_r\}$. The total number of intersect-tion points generated by bearing lines from two different sensors is T^2.

2.2. Method for solving the problem

Aiming at the problem appearing in multi-target multi-sensor passive tracking system, a scheme is proposed in this paper below.

2.2.1 Clustering with cross localization method

In locating multiple targets algorithm with passive sensors, determining the number of targets in essence is to find small clusters of points generated by the intersection of pairs of direction vectors emanating from each pair of sensors toward each target. The method is a kind of data clustering algorithm [13]. Using the clustering algorithm, the number of targets and the centers of clusters in surveillance area can be obtained. Because of the number of elastic networks having been determined, the searching area gets small and the initial value of the networks can be got. This clustering method not only cuts the computational burden down but also conduces to the convergence to correct intersection points and reduce the probability of locking a false target.

2.2.2 MEM description

After determining the number of targets, we can construct N disjoint fully connected subgraphs, with each subgraph whose initial value is in the neighborhood of the center of cluster representing a expectation templet. With the attraction of the cluster of target, the receptive field of the elastic neural network converges and locks a target at last.

2.2.3 Establishing trajectory trees

There are three kinds of “targets”which are locked by the elastic neural networks: the real targets, the first kind of intersection points and the second kind of intersection points which are referred to as ghost targets. Because of having got the proper initial value, the elastic neural network can not only lock real targets quickly but also increase the locking probability. If the number of real targets is less than 10, the number of the first kind of intersection points is small and the number of the second kind is large relatively. Take 10 targets for example, there is no intersection points of the first kind while there are 3 intersection points of the second kind. These ghost targets will breakup over time. So it is necessary to discriminate between real targets and ghost targets in a dynamic environment. A practicable algorithm is obtained by establishing multiple hypothesis trajectory trees [14] and summing three-continuous-step estimation covariance of every possible trajectory.

3. DATA CLUSTERING FOR DIRECTION FINDING CROSS LOCALIZATION

The aim of this algorithm is to obtain the center $G_i (i = 1, 2, \cdots K)$ of every cluster, the number K of clusters and the results of data clustering. We will analyze the clustering feature of the dataset $d = \{x_1, x_2, \cdots, x_n\} (D \subset R^n)$. Suppose every sample $x_i (i = 1, 2, \cdots, n)$ of D is a particle and its weight is 1. For the sake of simplicity of computation, suppose the coordinate of x_i is integer and there are K clusters whose centers are $G_i (i = 1, 2, \cdots K)$ in D. The clusters can be referred to as $C_i (i = 1, 2, \cdots, K)$. The algorithm involves the following 4 steps: iteration, selecting initial clusters, looking up table and diversity incorporating.

3.1. Iteration

In iteration, the initial value of δ can be set small and $\delta_{\max}$ which is given at first is larger than δ. The value of parameter a_r which adjusts δ during iteration is larger than 1 and smaller than 2.

1) Get the number N_i of samples in the neighborhood $U(x_i, \delta)$ of x_i which belongs to dataset D.
2) The weight $\tilde{m}_j$ of $x_j \in U(x_i, \delta)$ can be set to $\tilde{m}_j = \dfrac{m_j}{N_j}$ and they can form a new system $\tilde{U}(x_i, \delta)$. The weight $\tilde{m}_j$ is used to select initial clusters. Get the center $\tilde{x}$ and weight $\tilde{m}$ of $\tilde{U}(x_i, \delta)$.
3) All the $\tilde{x}$ and $\tilde{m}$ comprise a new dataset $\tilde{D}$, and iterate in $\tilde{D}$: If $\exists \tilde{x}_i = \tilde{x}_j$, they can be integrated into one particle $\tilde{x}$ whose weight is $\tilde{m} = \tilde{m}_i + \tilde{m}_j$. After one iteration, we can get $\tilde{D}$ from D. Set $\delta = \delta \times a_r$, if $\delta > \delta_{\max}$, we can get $\delta = \delta_{\max}$. Set $D = \tilde{D}$ and then go back to step 1).

3.2. Selecting initial clusters

After iteration, it is necessary to find shrinkage center of particle whose weight is large obviously. The method can be illustrated as follows. We can arrange the particles obtained after iteration from large to small according to the weight: $m_1 > m_2 > \cdots > m_{n'}$. Then define $q_i = m_i / m_{i+1} (i = 1, 2, \ldots, n' - 1)$ and get $i' = \min_{q_i > K_G} i$. K_G is a threshold which is given at first. As a result, i' is the number of initial clusters which we want to get and $x_1, x_2, \cdots, x_{i'}$ correspond to $m_1, m_2, \cdots, m_{i'}$ are the positions of centers of the initial clusters.

3.3. Looking up drift table

We can define the center x' of $U(x, \delta)$ as the drift position of x and register the drift position. After iteration, all the particles in D get drift positions which can form a table of drift positions. In the later iteration, it is necessary to update the table. We can look up the table and classify drift particle $x_i (i = 1, 2, \cdots, i')$ into initial cluster i when the iteration algorithm is over.

3.4. Diversity incorporating

Suppose we have obtained $x_1, x_2, \cdots, x_{i'}$ which are the centers of initial clusters correspond to $x_1, x_2, \cdots, x_{i'}$ when the iteration is over. Refer i' initial clusters after looking up the drift table as $C_i' (i = 1, 2, \cdots, i')$. δ_θ and V_θ are thresholds. The diversity incorporating algorithm which aims at judging whether C_i' and C_j' can be incorporated is illustrated below.

1) With all $x \in D$, get $\mathrm{U}(x, \delta_\theta)$ which satisfies boundary conditions $\exists x_i,\ x_j \in U(x, \delta_\theta),\ x_i \in C_i',$ $x_j \in C_j'$.
2) Incorporate all the $\mathrm{U}(x, \delta_\theta)$ which are obtained in step 1) as one boundary set U.
3) We can get the diversity of U, C_i' and C_j' and then compare the diversity of U with the smaller average diversity between C_i' and C_j'. Suppose the average diversity of C_i' is smaller, if $\frac{\text{the diversity of } C_i'}{\text{the diversity of } U} < V_\theta$, we can incorporate C_i' and C_j'. The center $G_i (i = 1, 2, \cdots K)$ of every cluster and the number of clusters can be obtained as a result of this algorithm.

4. THE MULTIPLE ELASTIC MODULES MODEL

The MEM model is presented as a significant extension to self-organizing map (SOM). Applying this representation to the passive tracking problem we can construct disjoint fully connected subgraphs. Take three sensors and N targets for example, we will show the algorithm.

4.1. Configuration of MEM

MEM is a kind of network with inner and outer layers. The neurons of outer layer are initialized to position vectors of sensors. The neurons of inner layer can be separated into N subgraphs which are composed of three fully connected neurons. At the same time, the subgraph is not connected together. Each neuron i in the inner layer responds to a specific feature type f_i. The associated weights connecting neurons of outer layer with neurons of inner layer are position vectors of intersection points while the associated weights connecting neurons of inner layer are distance between them.

The configuration of MEM is illustrated in Fig. 3. A unique feature of the MEM model is the use of a dynamic receptive field r_i (RF) with each neuron i. The size of a neuron's RF is determined according to

$$r_i = p_i h_i + e_i + \xi_i. \tag{3}$$

In (3), h_i and e_i represent locking value and expectation value of the neuron i respectively. The noise parameter ξ_i is a small constant which determines the minimum RF size of all neurons. The p_i term is a measure of the local deformation of the graph about neuron i. This parameter embodies the "goodness" of the current arrangement of neurons (the state of the network) as a solution for the desired objective function. For the passive tracking application, our desired objective is to locate the smallest spatial cluster of intersection points of different types. Therefore, p_i can be defined as

$$p_i = \sum_{j \in L_i} \| m_i - m_j \|, \tag{4}$$

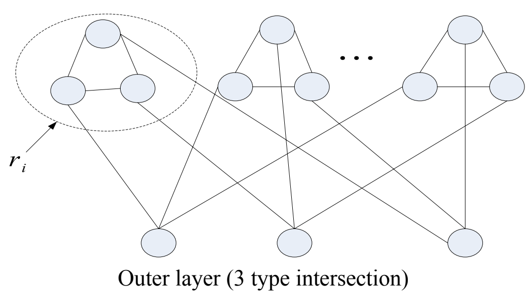


Fig. 3. The configuration of MEM.

where $L_i = \{j \mid g_{ij} = 1\}$. g_{ij}=1 if neuron i is a direct neighbor of neuron j. In other words, L_i is the set of all neurons directly connected to neuron i. In this formulation p_i is the sum of all distances between neuron i and its neighboring neurons. Take three sensors for example, we can make a simplification to this function by setting p_i equal to the perimeter of the triangle formed by the three neuron vertices. All neurons in the same triangle module can thus share the same p_i value. m_i is the position of the neuron i.

4.2. The algorithm of MEM

Formally, the operations performed by the MEM algorithm involve the following steps.

1) Randomly assign neurons to different position vectors.
2) Randomly select an intersection point X_{Q_i} $(Q_i = 1,2,3)$ of type Q as input.
3) Construct a set B according to the neurons of the same type.

   $$B = \{i \mid X_i \in Q \cap \| X_{Q_i} - X_i \| \le r_i\}$$

4) If set B is empty, we repeat this procedure by picking another random intersection point. When a nonempty set B is generated, the neurons within this set are allowed to compete, using a WTA mechanism, to find the closest neuron X_i^* to the input point: $X_i^* = \min_{i \in B} \| X_{Q_i} - X_i \|$.
5) Adjust the locations of winning neuron and its neighbors according to

   $$X_i^* = X_i^* + \alpha(X_{Q_i} - X_i),$$
   $$X_j = X_j + \beta_j (X_{Q_i} - X_j) j \in s_i,$$

   where α and β_j denote the learning rate. The value of α is less than 1. β_j is defined as $\beta_j = \dfrac{\alpha p_i}{p_i + \eta}$. This update rule implements a nonlinear elastic mechanism on the elastic modules. The elasticity parameter η is used to control the degree of elasticity of these modules.
6) Gradually adjust the receptive field R_i and R_j.
7) When $p_i \le 3\xi_i$, we can determine that the subgraph has locked a target. The center of the subgraph can be considered as the position of target: $X_\tau = \frac{1}{3} \sum_{Q_i=1}^{3} X_{Q_i}$, where X_{Q_i} is the position vector of the neuron in the subgraph.
8) Go back to step 2).

5. THE APPROACH OF ELIMINATING GHOST TARGETS DYNAMICALLY

A practicable algorithm is obtained by establishing multiple hypothesis trajectory trees and summing three-continuous-step estimation covariance of every possible trajectory in order to discriminate between real targets and ghost targets in a dynamic environment.

The Nearest Neighbor Algorithm is usually used to implement data association in bearings-only tracking system. In the association gate of the target, the nearest measurement to the predictive value can be taken as the new measurement of the target in order to implement data association. However, the nearest measurement to the predictive value may not be the real measurement of the target. As a result of dense density of targets, the false probability of association is high. It is difficult to achieve the aim of engineering to adopt some optimization algorithm with optimal performance because of complicated computation of likelihood probability. An association algorithm of tracking which is easy to implement in engineering is proposed in this paper.

Suppose $Z(k) = \{z_i(k)\}_{i=1}^{m}$ denotes a group of measurements got in the scanning of time K, m denotes the number of measurements.

1) Utilize Kalman filtering algorithm to achieve the predictions of the next time with the trajectories established in the scanning of time k-1. Then establish the association gate of every target.
2) Seeing about the number m of measurements in the scanning of time k. If m is larger than the number of trajectories established in the scanning of time k-1, we can get the conclusion that there are new targets or false alarms appearing. Then we can associate measurement $z_i(k)$ in every association gate with the trajectory, establish multiple hypothesis trajectory trees and calculate the estimation covariance of every trajectory.
3) Seeing about the number of measurements in the scanning of time k+1 and achieve the predictions of next time with the trajectories having been established. Then we can associate measurement $z_i(k+1)$ in every association gate with the trajectory, establish multiple hypothesis trajectory trees and calculate the estimation covariance of every trajectory.
4) Seeing about the number of measurements in the scanning of time k+2 and repeat the process of step 3).
5) Add up the estimation covariance of every trajectory in the trajectory trees in steps 2), 3) and 4) and arrange the estimation covariance of every trajectory from small to large.
6) Ascertain the trajectory with the smallest

accumulated covariance in trajectory trees. If there are no trajectories sharing the same measurement, we can determine the real trajectories and eliminate the trajectory branches. If there are trajectories sharing the same measurement, we can determine a real trajectory with the smallest accumulated covariance and search other trajectories of the trajectory trees according to the estimation covariance arranged from small to large of every trajectory until find a trajectory sharing no measurement with the trajectory having been established. Thus we can consider this trajectory as a real trajectory. See about whether there are new measurements appearing according to steps 2), 3) and 4). If there are new measurements appearing, we will judge whether they belong to new targets. If the new measurements belong to new targets, we will establish new trajectories of targets. Otherwise we will preserve the redundant measurements of time k+1 and k+2 and then judge whether there are new targets appearing according to the measurements of time k+3. Seeing about the number of measurements in the scanning of time k+2 and repeat the process of step 3).

The above approach is adopted with unity detection probability. Aiming at the scenario with non-unity detection probability, utilize Kalman filtering algorithm to achieve the predictions of time k with the trajectories established in the scanning of time k-1. If there are no measurements appearing in the association gate, replace filtering value with predictive value. Repeating this process on moments k, k+1, k+2, k+3 successively, if there are no measurements in the association gate for 4 moments successively, the trajectory interrupts.

6. SIMULATION RESULTS AND ANALYSIS

6.1. MEM for tracking in static environment

The unit of the coordinate is kilometer in the simulation. There are ten targets in a rectangle area whose coordinates are (0, 0), (0, 200), (140, 200) and (140, 0). Three sensors whose coordinates are (50, 0), (60, 0) and (70, 0) are arranged on a straight line. The ten targets are distributed randomly. Each sensor can track all the targets and the tracking error of angle is 0.01 radian. The intersection points of bearing lines from different sensors can denote targets. The parameters selected in the simulation are given as follows.

The learning rate α of the winning neuron is 0.05. The elasticity parameter η of the neighboring neurons of the winning neuron is 3. The noise parameter ξ_i is 0.01. The covariance matrix of measurement noise is $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ and the covariance matrix of process noise is $\begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}$.

Table 1 gives the results of comparing MEM model with MEM model based on clustering algorithm. The simulation results of MEM model is given in Figs. 4 and 5. The simulation results of MEM model based on clustering algorithm is given in Figs. 6 and 7.

There are ten targets in Fig. 5. After 100 iterations, there are 8 real targets and 2 ghost targets locked respectively using MEM algorithm and the time of computation is 2.672 second. After 50 iterations, there are 10 real targets and 3 ghost targets locked respectively using MEM based on clustering algorithm under the circumstances that the number of targets is not known and the time of computation is 0.009 second. Using the clustering algorithm, we can obtain the centers of clusters, the number of clusters and the information of the clusters. Because of these, the number of targets and elastic networks can be got. At the same time, the networks can be initialized near the centers of clusters. In the MEM based on clustering algorithm, the elastic networks search targets in the appointed clusters. So the computational burden of the method is lower, the elastic network can more quickly find all or most of the targets and the performance of the MEM can be improved. Because of not having eliminated all the ghost targets, we can

Table 1. The contrast between MEM and MEM based on clustering.

	Number of real targets	Number of targets locked	Time of computation
MEM based on clustering	10	10	0.009s
MEM	10	8	2.672s

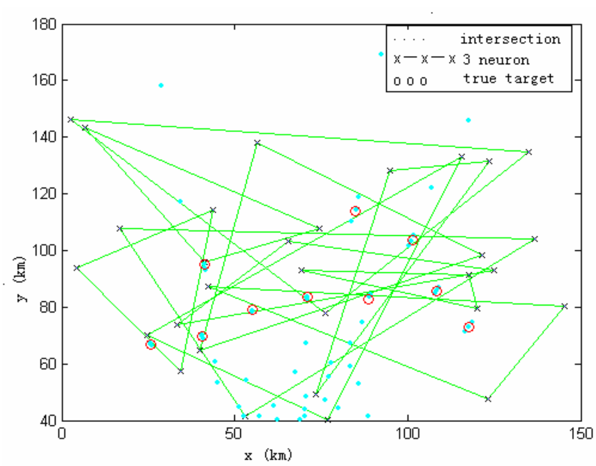


Fig. 4. The initial random state of the modules and the measures of targets in MEM.

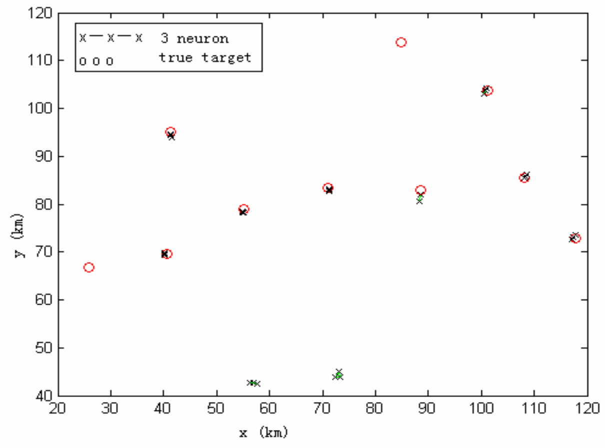


Fig. 5. The network state after 100 iterations.

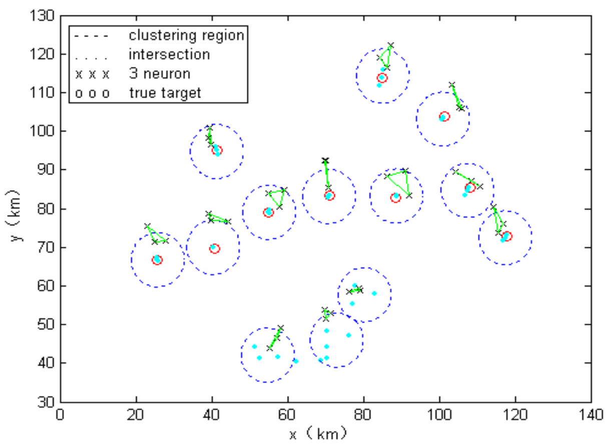


Fig. 6. The initial random state of the modules and the measures of targets in MEM based on clustering algorithm.

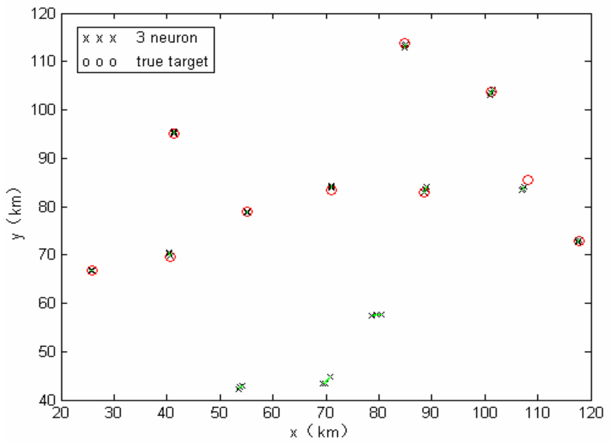


Fig. 7. The network state after 50 iterations.

discriminate between real targets and ghost targets in a dynamic environment in the simulation below.

6.2. Eliminate ghost targets in dynamic environment

The intersection points of ghost targets will not maintain their small spatial distribution when monitored for a period of time. The parameters in the simulation are as above. Ten targets are in uniform linear motion. We can utilize Kalman filtering algorithm to track the targets and calculate the overall error by establishing the trajectory trees. If the error is larger than the threshold, it can be eliminated. By using samplings during 20 scanning, we can get the real trajectories of targets as shown in Fig. 8.

Although there are different ghost targets appearing during every scanning, these ghost targets will breakup over time by using tracking targets in a dynamic environment. So the ghost targets can be eliminated dynamically.

Aiming at the scenario with non-unity detection probability, if there are no measurements in the association gate for 4 moments successively, the trajectory interrupts. 50 Monte Carlo simulations of 10 targets with non-unity detection probability are carried out and simulation results are shown in Table 2. During one Monte Carlo simulation take Target 1 for example, the trajectory interruption probability is 0.02 with detection probability 0.875. The simulation results illustrate that during the process of tracking the trajectory interruption probability will be higher if the detection probability is lower.

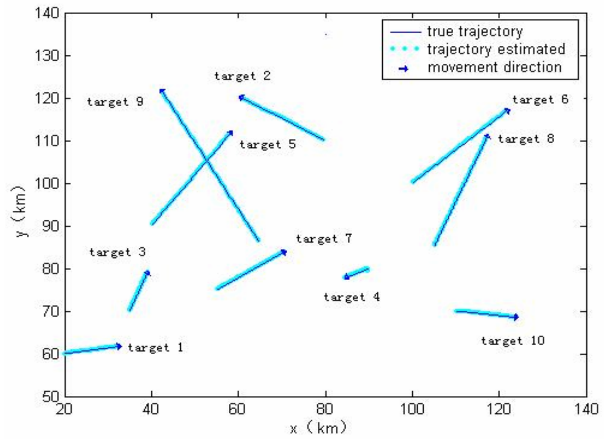


Fig. 8. The trajectories of targets.

Table 2. The probability of trajectory interruption with non-unity detection probability (DP).

DP	0.950	0.900	0.875	0.850	0.825	0.800
Target 1	0	0	0.02	0.02	0.04	0.04
Target 2	0	0	0	0.06	0.04	0.08
Target 3	0	0	0.02	0	0.02	0.06
Target 4	0	0	0	0.02	0	0.08
Target 5	0	0	0	0	0.04	0.04
Target 6	0	0	0	0.02	0.04	0.06
Target 7	0	0	0	0	0.04	0.08
Target 8	0	0	0	0	0.04	0.04
Target 9	0	0.02	0.02	0.04	0	0.06
Target 10	0	0	0	0.02	0.04	0.08

7. CONCLUSIONS

Direction finding cross localization is usually used in passive tracking. But this method will produce a lot of false intersection points. Eliminating these false intersection points correctly and quickly is difficult in passive localization. A new method is proposed for passive locating and tracking multi-jammer target in this paper. It not only solves the difficulty of determining the number of targets when masses of false intersection points existing, but also solves the initialization problem of elastic network. The method which dynamically establishes multiple hypothesis trajectory trees solves the problem of elimination of the remaining false intersection points. Simulation results show that computational burden of this method is lower, the elastic network can more quickly find all or most of the targets and have a more probability of locking the real targets. This method can eliminate more false intersection points. Under the circumstances that the number of targets is 10, all the ghost targets can be eliminated by using this method combining static with dynamic environment. It is meaningful to apply this method for anti-aircraft missile weapon integrated system.

REFERENCES

[1] E. Mazo, A. Averbuch, and Y. Bar-shalom, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103-124, 1998.

[2] K. R. Pattipati, S. Deb, Y. Bar-shlom, and R. B. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Trans. Automatic Control*, vol. 37, no. 2, pp. 198-213, 1992.

[3] J. Angus, *Genetic Algorithm in Passive Tracking*, Claremont Grad School Math Clinic Rep, 1993.

[4] C. R. Sastry, E. W. Kamen, and M. Simaan, "An efficient algorithm for tracking the angles of arrival of moving targets," *IEEE Trans. on Signal Process*, vol. 39, no. 1, pp. 242-246, 1991.

[5] Y. H. Chen and Y. T. Lian, "Multitarget angle tracking algorithm using sensor array," *IEE Proceedings-Radar, Sonar and Navigatioin*, vol. 142, no. 4, pp. 158-161, 1995.

[6] P. R. Williams, *Multiple Target Estimation using Multiple Angle Only Sensors*, Hughes Aircaft, Internal Rep, 1984.

[7] L.-S. Kang, Y. Xie, and S.-Y. You, *Non-Numerical Parallel Algorithm-Simulated Annealing Algorithm* (in Chinese), Science Press, Beijing, 1994.

[8] J. He, "Simulated annealing backfire algorithm for global optimization," *Journal of Wuhan University*, pp. 43-48, 1991.

[9] L.-C. Jiao, *Neural Network Computation*, Xidian University Press, Xi'An, 1993.

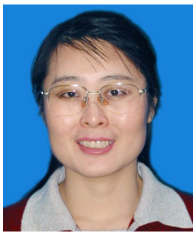
[10] Z.-Z. Shi, *Neural Computation*, Publishing House of Electronic Industry, BeiJing, 1993.

[11] S. Shams, "Neural network optimization for multi-target multi-sensor passive tracking," *Proc. of the IEEE*, vol. 84, no. 10, pp. 1442-1457, 1996.

[12] L.-P. Zhao, "Application of MEM models in locking and tracking multiple random passive targets," *Journal of East China University of Science and Technology*, vol. 27, no. 5, pp. 527-532, 2001.

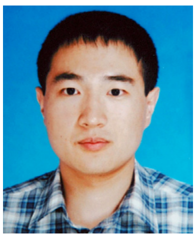
[13] J.-G. He and Q.-Y. Shi, "A new algorithm for clustering analysis," *Journal of Image and Graphics*, vol. 5, no. 5, pp. 401-405, 2000.

[14] Y. Lin, A. Xue, and J. Qian, "Passive tracking and data association algorithm for multiple targets with single sensor," *Proc. of the 5th World Congress on Intelligent Control and Automation*, Hang Zhou, China, June 2004.



Mei Liu received the Ph.D. degree in Information and Communication Engineering from Harbin Institute of Technology in 2005. She has been a Professor with Department of Electronic and Communication Engineering in Harbin Institute of Technology. Her research interests include target tracking, information fusion and sensor network.

Nuo Xu is currently an undergraduate with Department of Electronic and Communication Engineering in Harbin Institute of Technology. Her research interests include target tracking, information fusion and sensor network.



Haihao Li is currently pursuing a M.S. degree with Department of Electronic and Communication Engineering in Harbin Institute of Technology. His research interests include target tracking, information fusion and sensor network.