Progress In Electromagnetics Research, PIER 65, 169–185, 2006

TABU SEARCH TRACKER WITH ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM FOR MULTIPLE TARGET TRACKING

I. Turkmen

Civil Aviation School
Department of Aircraft Electrical and Electronics
Erciyes University
38039, Kayseri, Turkey

K. Guney

Faculty of Engineering
Department of Electronic Engineering
Erciyes University
38039, Kayseri, Turkey

Abstract—In this paper, a tabu search tracker with adaptive neuro-fuzzy inference system (TST-ANFIS) is presented for multiple target tracking (MTT). First, the data association problem, formulated as an N-dimensional assignment problem, is solved using the tabu search algorithm (TSA), and then the inaccuracies in the estimation are corrected by the adaptive neuro-fuzzy inference system (ANFIS). The performances of the TST-ANFIS, the joint probabilistic data association filter (JPDAF), the tabu search tracker (TST), Lagrangian relaxation algorithm (LRA), and cheap joint probabilistic data association with adaptive neuro-fuzzy inference system state filter (CJPDA-ANFISSF) are compared with each other for six different tracking scenarios. It was shown that the tracks estimated by using proposed TST-ANFIS agree better with the true tracks than the tracks predicted by the JPDAF, the TST, the LRA, and the CJPDA-ANFISSF.

1. INTRODUCTION

Target tracking [1] is important to military radars as well as to most civilian radars. In military radars, tracking is responsible for fire control and missile guidance; in fact, missile guidance is almost impossible without proper target tracking. Commercial radar systems, such as civilian airport traffic control radars, may utilize tracking as a means of controlling incoming and departing airplanes. The main objective of MTT, is to partition the radar data into sets of observations (tracks) produced by the same source. Once the tracks are formed and confirmed, the dynamics of each target (position, velocity, and acceleration) can be computed. The accuracy of data association is critically important since mis-association of data to the targets will lead to tracking failure.

Several methods [1–10], varying in accuracy and computational effort, have been presented and used to solve the data association problem. The JPDAF [1] is one of method commonly used for the MTT. In the JPDAF algorithm, the association probabilities are determined from the joint likelihood functions corresponding to the joint hypotheses associating all the returns to different permutations of the targets and clutter points. The real time computation requirements of the JPDAF are very high and the computational complexity increases exponentially as the number of targets increases. To reduce this computational complexity, Fitzgerald [2] has proposed a simplified version of the JPDAF, called the cheap JPDAF (CJPDAF) algorithm. The CJPDAF method is very fast and easy to implement; however, the tracking performance of the CJPDAF also decreases in either dense target or cluttered environment. In previous work [3], we proposed the CJPDA-ANFISSF for MTT. In [3], the state update step of the CJPDAF was realised with the use of the ANFIS state filter instead of Kalman filter to improve the accuracy of the CJPDAF. We also used neural networks for computing the association probabilities [4]. Recently, assignment algorithms [5–10] have been shown to be effective in data association for target tracking in the presence of clutter. In assignment, the data association is formulated as a constrained optimisation problem, where the cost function, usually, a combined likelihood function evaluated using the results from the state estimator, is minimised. The N-dimensional assignment problem for associating data from three or more scans of measurements ($N \geq 3$) is known to be NP-hard. It is well known that solving such a constrained optimisation problem is intractable. The TSA can be used successfully to solve NP-hard problems [11]. It is a metaheuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality. In our previous work [12], an alternative method to LRA [9, 10] based on the TSA was proposed for MTT. As far as we know, except our work [12], there is no any application of the TSA to the target tracking problems, however, we expect that it will find a wide application area such as the other heuristic optimisation techniques. In the TST, the target states are updated by using Kalman filter [13]. The Kalman filtering is effective for simple scenarios such as in a clutterless environment or a single sensor tracking a single target. However, under dense target environment, extraneous sensor reports may be incorrectly used by the Kalman filter for track update, thus resulting in degraded performance, possibly loss of track may occur. In this paper, the inaccuracies in the estimation of Kalman filter are corrected by using ANFIS [14, 15].

The ANFIS is a fuzzy inference system (FIS) implemented in the framework of an adaptive fuzzy neural network, and is a very powerful approach for building complex and non-linear relationship between a set of input and output data. It combines the explicit knowledge representation of FIS with the learning power of neural networks.

In the following sections, the TSA, the ANFIS, and the TST-ANFIS technique proposed in this paper are described briefly, and the simulation studies are then presented.

2. TABU SEARCH ALGORITHM (TSA)

The TSA [11] is a heuristic approach for solving optimisation problems by using a guided, local search procedure to explore the entire solution space without becoming easily trapped in local optima. One characteristic of tabu search is that it finds good near-optimal solutions early in the optimisation run. It does not require initial guesses, does not use derivatives, and is also independent of the complexity of the cost function considered. Its flexible control framework and several spectacular successes in solving NP-hard problems caused rapid growth in its application [11]. Unlike genetic algorithm and other heuristic techniques, TSA uses a flexible memory of search history to prevent cycling and to avoid entrapment in local optima.

Tabu search starts with a present solution x_{now} which has a set of neighbors, Q. A neighbor, x^*, can be produced by applying a simple modification to the present solution x_{now}. This modification is called a move. In order to get rid of a local minima in the search space, the move to x^* can be applied even if x^* is worse than x_{now}. However, this can cause the cycling of the search. To prevent the search from facing cycling problem as much as possible, a table which is called tabu list is introduced. All moves that are not permitted, which are called tabu moves, are stored in the tabu list. The moves existing in the tabu list are those carried out most frequently and recently. The use of the tabu list decreases the possibility of cycling because it prevents the return within a certain number of iterations to a solution visited recently. After a subset of feasible solutions, Q^*, are produced from the present one according to the tabu list and evaluated for the problem, the next solution is selected from Q^* and then the tabu list is modified depending on the selected next solution. The solution evaluated as best is selected as the next solution x_{next}. After accepting the next solution to be the present one, the second loop of the search starts. The loop is repeated until a predetermined stopping criteria is satisfied.

In order to represent a solution, an integer-valued vector is used in this paper. The neighbors are generated by adding small integer values into the parameter values of the current solution. In order to classify a move to be tabu or not, criterion called tabu restrictions are employed. The tabu search used in this work has two tabu restrictions, which were based on recency and frequency memories:

$$\text{Tabu Restrictions} = \begin{cases} \text{recency } (k) > \text{recency limit} \\ \text{or} \\ \text{frequency } (k) < \text{frequency limit} \end{cases} \tag{1}$$

The recency of a move is the difference between the present iteration count and the last iteration count at which that move was created. The frequency of a move is the count of changes of that move. The use of tabu restrictions might prevent the search from finding a solution that has not been visited yet, or the restrictions might even sometimes cause all available moves to be classified as tabu. Therefore, the tabu restrictions might be ignored when a freedom is needed. An aspiration criterion is employed to determine which move should be freed in such cases. In this work, the following aspiration criterion was employed when all available moves are classified tabu: a tabu move that loses its tabu status by the least increase in the value of current iteration is freed from the tabu list.

3. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

The ANFIS can simulate and analysis the mapping relation between the input and output data through a learning algorithm to optimise the parameters of a given FIS [14, 15]. It combines the powerful features of FISs with those of artificial neural networks.

The ANFIS architecture consists of fuzzy layer, product layer, normalised layer, de-fuzzy layer, and summation layer. A typical

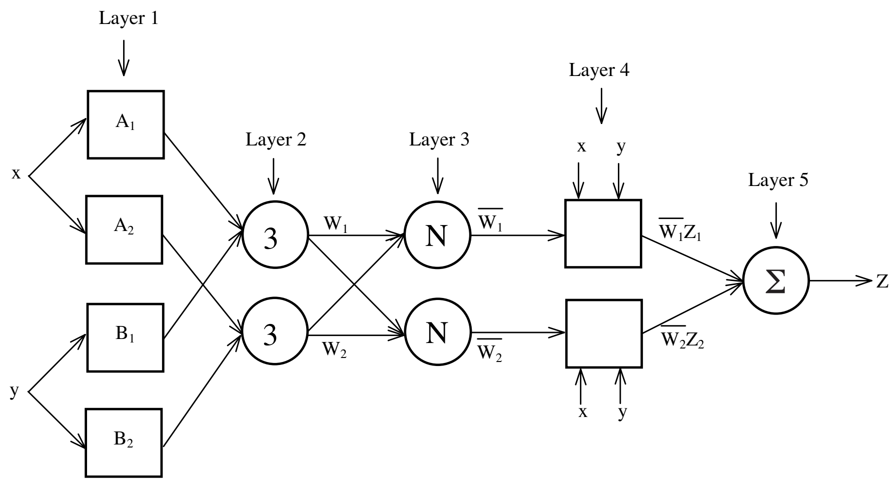


Figure 1. Architecture of ANFIS.

architecture of ANFIS is shown in Fig. 1, in which a circle indicates a fixed node, whereas a square indicates an adaptive node. For simplicity, we consider two inputs x, y and one output z in the FIS. The ANFIS used in this paper implements a first-order Sugeno fuzzy model [15]. For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules can be expressed as

$$\text{Rule 1: If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{then } z_1 = p_1x + q_1y + r_1 \tag{2a}$$

$$\text{Rule 2: If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \text{then } z_2 = p_2x + q_2y + r_2 \tag{2b}$$

where A_i and B_i are the fuzzy sets in the antecedent, and p_i, q_i, and r_i are the design parameters that are determined during the training process. As in Fig. 1, the ANFIS consists of five layers. Every ith node in the first layer is an adaptive node with a node output defined by

$$O_i^1 = \mu_{A_i}(x), \qquad i = 1, 2 \tag{3a}$$

$$O_i^1 = \mu_{B_{i-2}}(y), \qquad i = 3, 4 \tag{3b}$$

where $\mu_{A_i}(x)$ and $\mu_{B_{i-2}}(y)$ can adopt any fuzzy membership function (MF). In this paper, the following triangular MF is used.

$$triangle(x; a, b, c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases} \tag{4}$$

where $\{a_i, b_i, c_i\}$ is the parameter set that changes the shapes of the MF. Parameters in this layer are termed *the premise parameters*.

Every node in the second layer is a fixed node labeled Π, whose output is the product of all the incoming signals:

$$O_i^2 = \omega_i = \mu_{A_i}(x)\mu_{B_i}(y), \qquad i = 1, 2 \tag{5}$$

Each node output represents the firing strength of a rule.

Every node in the third layer is a fixed node labeled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rules' firing strengths:

$$O_i^3 = \overline{\omega}_i = \frac{\omega_i}{\omega_1 + \omega_2}, \qquad i = 1, 2 \tag{6}$$

where $\overline{\omega}_i$ is referred to as *the normalised firing strengths*.

Every ith node in the fourth layer is an adaptive node with the following node function:

$$O_i^4 = \overline{\omega}_i z_i = \overline{\omega}_i \left(p_i x + q_i y + r_i\right), \qquad i = 1, 2 \tag{7}$$

where $\overline{\omega}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer are termed the *consequent parameters*.

The single node in the fifth layer is a fixed node labeled Σ that computes the overall output as the summation of all incoming signals:

$$O_1^5 = \sum_{i=1}^{2} \overline{\omega}_i z_i = \frac{\omega_1 z_1 + \omega_2 z_2}{\omega_1 + \omega_2} \tag{8}$$

It is seen from the ANFIS architecture that when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters:

$$z = (\overline{\omega}_1 x)\, p_1 + (\overline{\omega}_1 y)\, q_1 + (\overline{\omega}_1)\, r_1 + (\overline{\omega}_2 x)\, p_2 + (\overline{\omega}_2 y)\, q_2 + (\overline{\omega}_2)\, r_2 \tag{9}$$

The optimal values of the consequent parameters can be found by using the least-square method (LSM). When the premise parameters are not fixed, the search space becomes larger and the convergence of training becomes slower. The hybrid learning algorithm [14] combining the LSM and the backpropagation (BP) algorithm can be used to solve this problem. This algorithm converges much faster since it reduces the dimension of the search space of the BP algorithm. During the learning process, the premise parameters in the layer 1 and the consequent parameters in the layer 4 are tuned until the desired response of the FIS is achieved.

4. TABU SEARCH TRACKER WITH ANFIS (TST-ANFIS)

The data association problem can be written as the following N-dimensional assignment problem:

$$\begin{aligned}
&\text{minimise:}\\
&\sum_{i_1=0}^{M_1}\cdots\sum_{i_N=0}^{M_N} c_{i_1,\ldots,i_N}\chi_{i_1,\ldots,i_N},\\
&\text{subject to:}\\
&\sum_{i_2=0}^{M_2}\cdots\sum_{i_N=0}^{M_N}\chi_{i_1,\ldots,i_N}=1, \qquad i_1=1,\ldots,M_1,\\
&\sum_{i_1=0}^{M_1}\cdots\sum_{i_{k-1}=0}^{M_{k-1}}\sum_{i_{k+1}=0}^{M_{k+1}}\cdots\sum_{i_N=0}^{M_N}\chi_{i_1,\ldots,i_N}=1,\\
&\text{for } i_k=1,\ldots,M_k \text{ and } k=2,\ldots,N-1,\\
&\sum_{i_1=0}^{M_1}\cdots\sum_{i_{N-1}=0}^{M_{N-1}}\chi_{i_1,\ldots,i_N}=1, \qquad i_N=1,\ldots,M_N\\
&\chi_{i_1,\ldots,i_N}\subset\{0,1\},\ \text{for all } i_1,\ldots,I_N,
\end{aligned} \tag{10}$$

where $c_{i_1,\ldots,i_N}$ is the cost of associating the measurement sequence to track t, $\chi_{i_1,\ldots,i_N}$ is the binary variable, taking values 0 or 1, and M is the number of the measurements in each scan. The TSA is employed to find the most likely set of N-tuples such that each measurement is assigned to one and only one target, or declared false and each target receives at most, one measurement from each scan.

The cost $c_{i_1,\ldots,i_N}$ can be expressed as the cumulative negative log-likelihood ratio given by

$$c_{i_1,\ldots,i_N}=-\sum_{m=1}^{N}\ln L_{i_m}(i_1,\ldots,i_m) \tag{11}$$

with the negative log-likelihood ratio of the track-measurement sequence $i_1,\ldots,i_m$ is

$$\begin{aligned}
-\ln L_{i_m}(i_1,\ldots,i_m) \;=\; & \frac{1}{2}\left[z_{i_m}-\hat{z}_{i_1,\ldots,i_m}\right]'\cdot S^{-1}_{i_1,\ldots,i_m}\cdot\left[z_{i_m}-\hat{z}_{i_1,\ldots,i_m}\right]\\
&+\ln\frac{\lambda_e|2\pi S_{i_1,\ldots,i_m}|^{1/2}}{P_D}
\end{aligned} \tag{12}$$

where z is the measurement associated with track t at scan m, $\hat{z}$ is the predicted measurement from target t with covariance S, λ_e is the spatial density of the false alarms, and P_D is the detection probability.

The cost function for the TSA is defined as:

$$E = \text{minimise}\left(\sum_{i_1=0}^{M_1} \cdots \sum_{i_N=0}^{M_N} c_{i_1,\ldots,i_N} \chi_{i_1,\ldots,i_N}\right) \tag{13}$$

To find the best suitable measurement sequence, the cost function given in Eqn. (13) will be minimised by the TSA.

It is evident from the literature [1] that the estimation error is determined primarily by the state space model and the measurement model. Since the estimation and the prediction vectors are closely related to the tracker, and hence to the estimation error, therefore they are suitable to be the inputs of the ANFIS. Thus, the inputs of the ANFIS are the position difference (δ_1) between the measurement and the estimation vectors, the position difference (δ_2) between the estimation and the prediction vectors, and the velocity difference (δ_3) between the estimation and the prediction vectors. The output of the ANFIS is the estimation correction. Fig. 2 shows a block diagram of the method proposed in this paper. The proposed method can be called as TST-ANFIS.

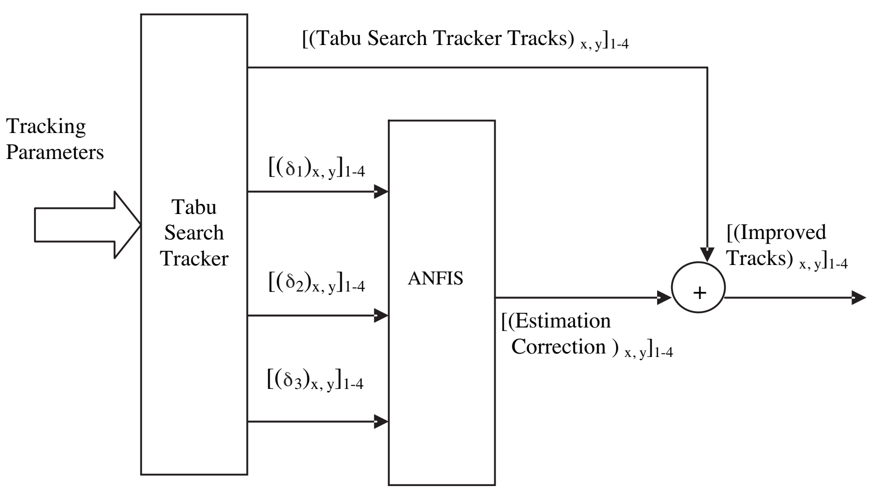


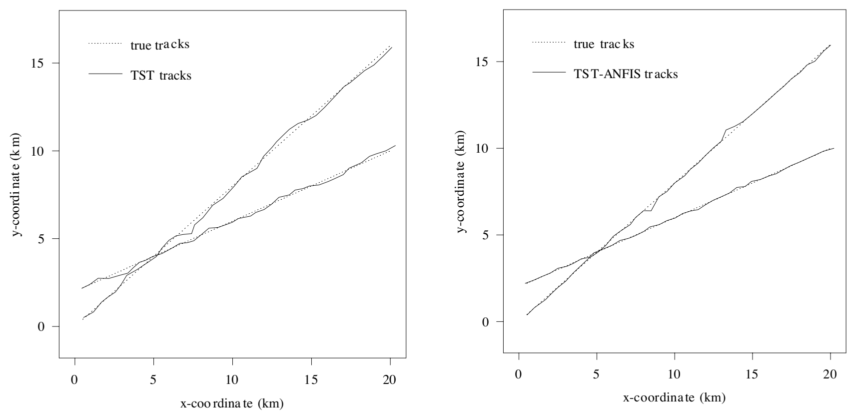
Figure 2. Block diagram of TST-ANFIS.

Table 1. Initial positions and velocities for targets in scenarios 1–6.

Scenarios		Targets	x (km)	y (km)	$\dot{x}$ (km/s)	$\dot{y}$ (km/s)
1	Two crossing targets	1	0	0	0.50	0.4
		2	0	2	0.50	0.2
2	Two parallel targets	1	0.5	3.2	0.50	0.001
		2	0.5	4.8	0.47	0.001
3	Two targets (one manoeuvring)	1	20.45	11.25	-0.45	-0.25
		2	20.45	6.8	-0.45	0.001
4	Two manoeuvring targets	1	20.45	11.25	-0.45	-0.25
		2	20.45	2.75	-0.45	0.25
5	Four crossing targets	1	0	0	0.50	0.4
		2	0	2	0.50	0.2
		3	0	16	0.50	-0.4
		4	0	10	0.50	-0.2
6	Four parallel targets	1	0.5	3.2	0.50	0.001
		2	0.5	4.8	0.47	0.001
		3	0.5	6.7	0.52	0.001
		4	0.5	8.3	0.48	0.001

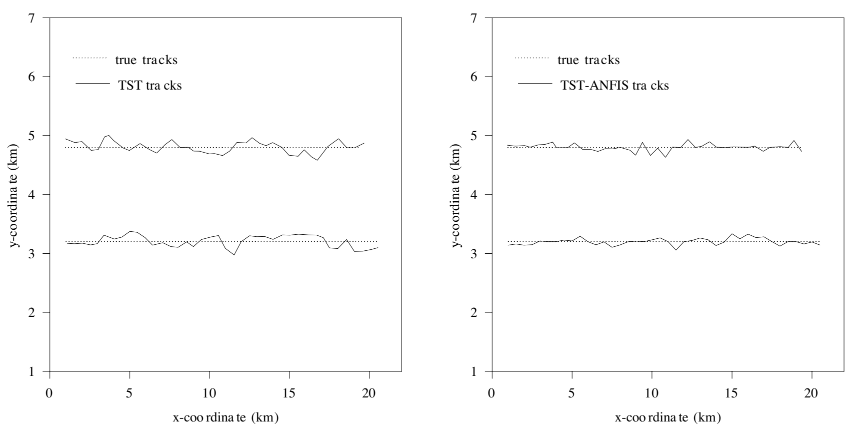
5. SIMULATIONS

In this section, several simulation examples are presented to illustrate the implementation of the proposed tracker that incorporates an ANFIS into the TST, and to compare its performance with those of the JPDAF, TST, LRA, and CJPDA-ANFISSF methods. Six different test scenarios are considered for this purpose. The trajectories of two crossing targets in scenario 1, two parallel targets in scenario 2, two targets (one maneuvering) in scenario 3, two maneuvering targets in scenario 4, four crossing targets in scenario 5, and four parallel targets in scenario 6 are shown in Figs. 3–8, respectively. The initial positions and velocities of targets in scenarios 1-6 are listed in the Table 1. The initial states are the true states with prescribed covariance. In Fig. 5, the first target is maneuvered in the y direction with acceleration of $50\,\mathrm{m/s^2}$ from $t = 13\,\mathrm{s}$ to $t = 23\,\mathrm{s}$ while the second target is not maneuvered. In Fig. 6, the first target is maneuvered in the y direction with acceleration of $50\,\mathrm{m/s^2}$ from $t = 13\,\mathrm{s}$ to $t = 23\,\mathrm{s}$ and the second target is maneuvered in the y direction with acceleration of $-50\,\mathrm{m/s^2}$ from $t = 13\,\mathrm{s}$ to $t = 23\,\mathrm{s}$. The maximum number of targets that can be tracked by the proposed method is the one that is usually encountered in practice. The trajectories in scenarios 1-6 are also similar to the



(a) (b)

Figure 3. Tracking two crossing targets in scenario 1 using (a) TST and (b) TST-ANFIS.



(a) (b)

Figure 4. Tracking two parallel targets in scenario 2 using (a) TST and (b) TST-ANFIS.

ones widely used in the literature.

The discretized state equation for each target can be written as

$$X_t(k+1) = F_t(k)X_t(k) + G_t(k)w_t(k) \tag{14}$$

where X is the target state vector, F and G are known matrices

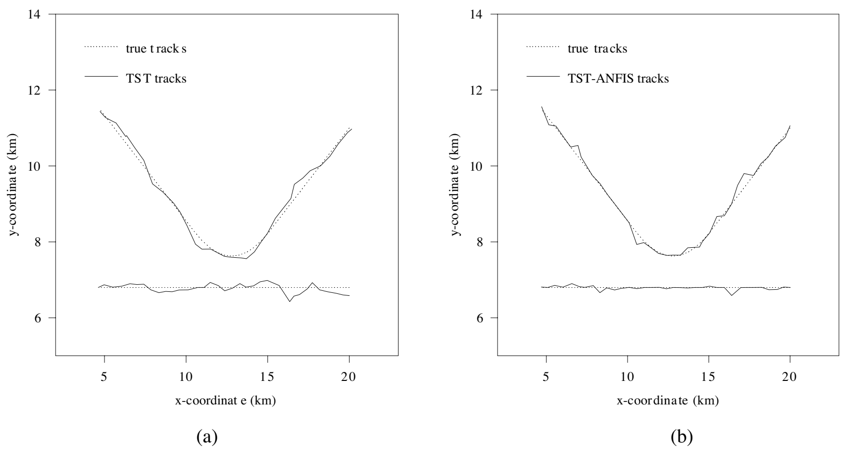


Figure 5. Tracking two targets (one maneuvering) in scenario 3 using (a) TST and (b) TST-ANFIS.

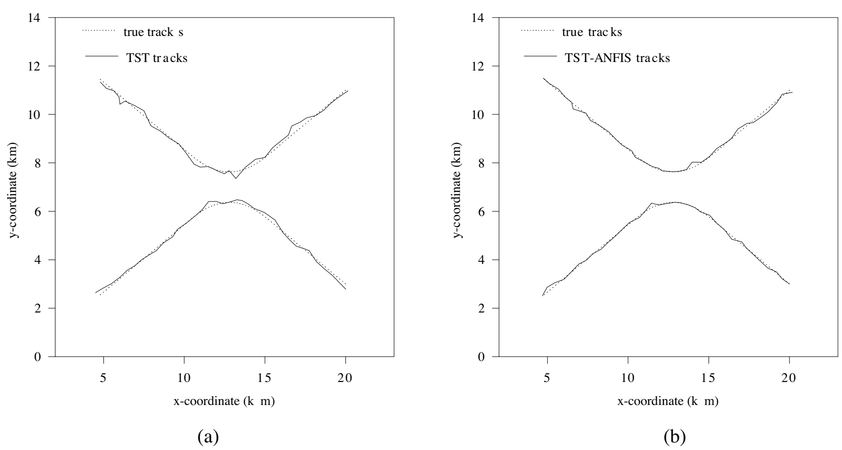


Figure 6. Tracking two maneuvering targets in scenario 4 using (a) TST and (b) TST-ANFIS.

describing the dynamics of the target, and w is a vector of zero-mean Gaussian noise uncorrelated with any such noise vector at a different instant of time. The state vector is given by

$$X_t(k) = [x\ y\ \dot{x}\ \dot{y}]^T \tag{15}$$

where x and y are the position components, and $\dot{x}$ and $\dot{y}$ are the velocity

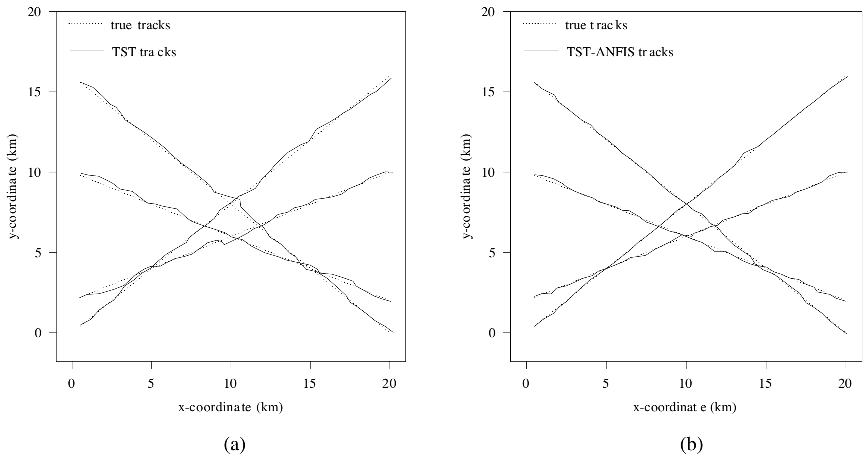


Figure 7. Tracking four crossing targets in scenario 5 using (a) TST and (b) TST-ANFIS.

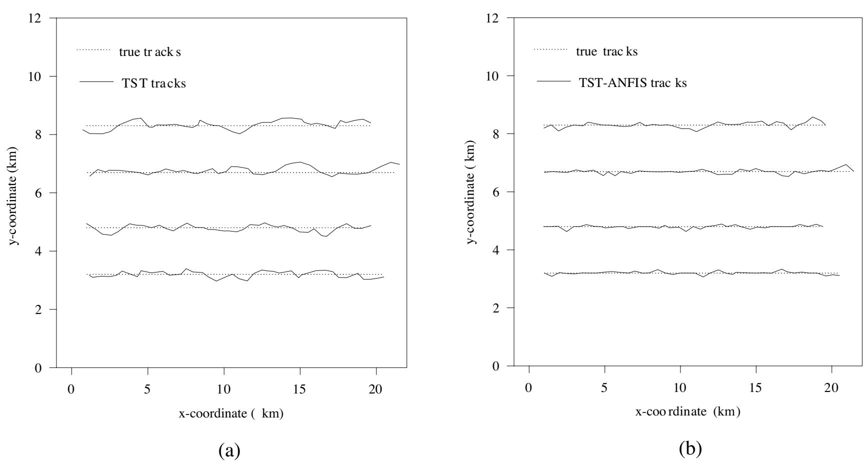


Figure 8. Tracking four parallel targets in scenario 6 using (a) TST and (b) TST-ANFIS.

components of the target t at time k. $F_t(k)$ and $G_t(k)$ are defined by

$$F_t(k) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad G_t(k) = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix} \tag{16}$$

where T is sampling interval. In the simulation the sampling interval was assumed to be 1 s. The covariance matrix of the process noise can be expressed as

$$Q_t(k) = \begin{bmatrix} \left(\sigma_x^t(k)\right)^2 & 0 \\ 0 & \left(\sigma_y^t(k)\right)^2 \end{bmatrix} \tag{17}$$

The associated variances were selected as $\left(\sigma_x^t(k)\right)^2 = 0.005\,\text{km}^2\text{s}^{-4}$ and $\left(\sigma_y^t(k)\right)^2 = 0.005\,\text{km}^2\text{s}^{-4}$.

The measurement model can be written as

$$z(k) = H(k)X_t(k) + v(k) \tag{18}$$

where $z(k)$ is the measurement vector, H is a known matrix, and v is a zero-mean Gaussian noise vector independent of w_t. It was assumed that only position measurements are available so that

$$H(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ for all } k \tag{19}$$

The covariance matrix of measurement noise $v(k)$ was $R(k) = \text{diag}(0.1, 0.1)\,\text{km}^2$ assuming all the measurement noise to be uncorrelated. The probability of detection was selected as 0.9.

In the simulation studies, there are two and four targets in scenarios (1–4) and (5 and 6), respectively. The sensors can receive eight measurements at every scan for all scenarios. Therefore, there are six and four clutter points at every scan for scenarios (1–4) and (5 and 6), respectively. The clutter points are uniformly located in the measurement space with an average value about two clutter points per validation gate. The false alarm rate of the sensor is chosen as to give the selected number of clutter points. The scan number in every sliding window is 5. As mentioned before, the solutions of the problem are represented by integer valued arrays. This representation can be explained for scenario 1 as follows: For target one in scenario 1, if the corresponding sequence consists of the first measurement of scan 1, the second measurement of scan 2, the fourth measurement of scan 3, the fifth measurement of scan 4, the eighth measurement of scan 5; and for target two, the corresponding sequence consists of the sixth measurement of scan 1, the fourth measurement of scan 2, the seventh measurement of scan 3, the fourth measurement of scan 4, the fifth measurement of scan 5, then the solution is a string [1, 2, 4, 5, 8, 6, 4, 7, 4, 5]. Similar representations were used for other scenarios. There

Table 2. Performance comparison of JPDAF, TST, LRA, CJPDA-ANFISSF, and TST-ANFIS methods.

Scenarios	Targets	RMS tracking errors (km)					Percentage improvement (%)			
		JPDAF [1]	TST [12]	LRA [9, 10]	CJPDA-ANFISSF [3]	Present method (TST-ANFIS)	with respect to JPDAF	with respect to TST	with respect to LRA	with respect to CJPDA-ANFISSF
1	1	0.2905	0.2236	0.2305	0.0918	0.0724	75	68	69	21
	2	0.1858	0.1594	0.1574	0.0852	0.0578	69	64	63	32
2	1	0.1524	0.1086	0.1002	0.0512	0.0412	73	62	59	20
	2	0.1668	0.1602	0.1631	0.1078	0.0727	56	55	55	33
3	1	0.1716	0.1483	0.1498	0.1085	0.0882	49	41	41	19
	2	0.1604	0.1472	0.1433	0.0685	0.0520	68	65	64	24
4	1	0.2776	0.2380	0.2298	0.1356	0.1021	63	57	56	25
	2	0.1719	0.1505	0.1528	0.0875	0.0655	62	56	57	25
5	1	0.1918	0.1676	0.1787	0.0785	0.0585	69	65	67	25
	2	0.2877	0.2513	0.2418	0.1225	0.0974	66	61	60	20
	3	0.2491	0.2023	0.2089	0.0985	0.0824	67	59	61	16
	4	0.2847	0.2037	0.1955	0.1258	0.0961	66	53	51	24
6	1	0.2019	0.1656	0.1663	0.0787	0.0547	73	67	67	30
	2	0.1822	0.1532	0.1513	0.0897	0.0703	61	54	54	22
	3	0.2537	0.1995	02014	0.1015	0.0831	67	58	59	18
	4	0.2477	0.2238	0.2314	0.1158	0.0919	63	59	60	21

are 10 and 20 elements in the solution strings for scenarios (1–4) and (5 and 6), respectively.

For target tracking applications, there are two types of data generators, namely measurement and simulation. The selection of a data generator depends on the application and the availability of the data generator. In this paper, 1240 training data sets were obtained from the neighbourhood of the true trajectories. The true trajectories were corrupted with different clutter points and training data sets were collected by realising the TST for the tracking scenarios. After training, the ANFIS was used to improve the accuracy of the state estimation for six different test scenarios shown in Figs. 3–8. In the test scenarios, the true target trajectories were corrupted with random clutter points which were not used in training. 40 and 35 test data sets were used for each target in scenarios (1, 2, 5, 6) and (3, 4), respectively. Therefore, a total of 620 test data sets were used.

The input and output data sets were scaled between 0.0 and

1.0 before training. The number of epoch was 15 for training. The number of MFs for the input variables, δ_1, δ_2, and δ_3 are 6, 6, and 6, respectively. The number of rules is then 216 ($6 \times 6 \times 6 = 216$). The triangular MF is used for all input variables. It is clear from Eqn. (4) that the triangular MF is specified by three parameters. Therefore, ANFIS used here contains a total of 918 fitting parameters, of which 54 ($6 \times 3 + 6 \times 3 + 6 \times 3 = 54$) are the premise parameters and 864 ($4 \times 216 = 864$) are the consequent parameters.

For comparison, we also obtained the tracking results of the TST, the JPDAF, the LRA, and the CJPDA-ANFISSF for six test scenarios. The tracking performances of the TST and the TST-ANFIS are compared in Figs. 3–8. As it is seen from Figs. 3–8, the TST-ANFIS tracks are closer to the true tracks than the tracks predicted by the TST for all scenarios. The results of JPDAF, the LRA, and the CJPDA-ANFISSF are not shown in Figs. 3–8 for clarity. Table 2 gives the comparative performances of the JPDAF, TST, LRA, CJPDA-ANFISSF, and TST-ANFIS methods in terms of RMS tracking error. The percentage improvement obtained by using the TST-ANFIS is also listed in Table 2. This percentage improvement is calculated as the ratio of the difference between the RMS errors of the TST-ANFIS method and the competing method (JPDAF, TST, LRA, or CJPDA-ANFISSF) to the RMS error of the competing method. It is evident from Table 2 that in all cases the results of the TST-ANFIS are better than those of the JPDAF, TST, LRA, and CJPDA-ANFISSF methods. The RMS tracking error values clearly show that a significant improvement is obtained over the results of the JPDAF, TST, LRA, and CJPDA-ANFISSF methods. When ANFIS is employed, the average percentage improvement with respect to JPDAF, TST, LRA, and CJPDA-ANFISSF 65%, 59%, 59%, and 23% respectively. The incorporation of an ANFIS into the TST leads to good accuracy in tracking multiple targets. It is clear from Table 2 that the RMS error values of the TST are smaller than those of the JPDAF. Table 2 also shows that the tracking results of the LRA algorithm are nearly equal to those of the TST. However the TST takes an average CPU time of 2 seconds whereas the LRA takes an average CPU time of 4.05 seconds for scenarios 1 to 6 running on a Pentium IV 2-GHz PC with 256 MB of RAM memory. Obviously, the TST is faster than the LRA for data association. A prominent advantage of the ANFIS model is that, after proper training, ANFIS completely bypasses the repeated use of complex iterative processes for new cases presented to it. Even if training takes a few minutes, the test process takes only a few microseconds to produce the result. Therefore, CPU times of the TST and TST-ANFIS methods are nearly equal.

6. CONCLUSION

The TST-ANFIS technique is presented for MTT. The data association problem formulated as an N-dimensional assignment problem is solved using TSA. The incorporation of the ANFIS into the TST is then proposed to increase its tracking performance. Performance evaluations of the TST, JPDAF, the LRA, the CJPDA-ANFISSF, and the TST-ANFIS are presented using simulation studies. Six different tracking scenarios are considered for this evaluation. It was shown that the estimation results of the TST-ANFIS are better than those of the TST, the LRA, the JPDAF, and the CJPDA-ANFISSF. The very good agreement between the TST-ANFIS tracks and the true tracks supports the validity of TST-ANFIS technique proposed in this paper.

REFERENCES

1. Bar-Shalom, Y. and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques,* YBS Publishing, Stoors, CT, 1995.
2. Fitzgerald, R. J., "Development of practical PDA logic for multitarget tracking by microprocessor," *Proceedings of the American Control Conference*, Seattle, Washington, 889–897, 1986.
3. Turkmen, I. and K. Guney, "Cheap joint probabilistic data association with adaptive neuro-fuzzy inference system state filter for tracking multiple targets in cluttered environment," *AEU-International Journal of Electronics and Communications*, Vol. 58, 349–357, 2004.
4. Turkmen, I. and K. Guney, "Artificial neural networks for calculating the association probabilities in multi-target tracking," *IEE Proc. Radar, Sonar and Navigation*, Vol. 151, No. 4, 181–188, 2004.
5. Pattipati, K. R., R. L. Popp, and T. Kirubarajan, "Survey of assignment techniques for multitarget tracking," *Multitarget-Multisensor Tracking: Applications and Advances,* Y. Bar-Shalom and W. D. Blair (Eds.), Artech House, 2000.
6. Pattipati, K. R., S. Deb, Y. Bar-Shalom, and R. B. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Trans. Automatic Control,* Vol. 37, No. 2, 197–213, 1992.
7. Poore, A. B., "Multidimensional assignment formulation of data association problem arising from multitarget and multisensor tracking," *Computational Optimization and Applications,* Vol. 3, No. 1, 27–57, 1994.

8. Poore, A. B. and A. J. Robertson, "A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems," *Computational Optimization and Applications,* Vol. 8, No. 2, 129–150, 1997.
9. Deb, S., K. R. Pattipati, and Y. Bar-Shalom, "A generalized S-D assignment algorithm for multisensor-multitarget state estimation," *IEEE Trans. Aerospace and Electronic Systems,* Vol. 33, No. 2, 523–538, 1997.
10. Pattipati, K. R., S. Deb, Y. Bar Shalom, and R. B. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Trans. Automatic Control,* Vol. 37, No. 2, 198–213, 1992.
11. Glover, F. and M. Laguna, *Tabu Search,* Kluwer Academic, Boston, 1997.
12. Turkmen, I., K. Guney, and D. Karaboga, "Tabu search tracker for multiple target tracking," *Journal of Electromagnetic Waves and Applications,* Vol. 18, No. 12, 1573–1589, 2004.
13. Kalman, R. E., "A new approach to linear filtering and prediction problems," *Transaction of the ASME-Journal of Basic Engineering,* 35–45, 1960.
14. Jang, J. S. R., "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Systems, Man, and Cybernetics,* Vol. 23, 665–685, 1993.
15. Jang, J. S. R., C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence,* Prentice-Hall, Upper Saddle River, NJ, 1997.