http://dx.doi.org/10.5755/j01.eee.21.4.12772 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 21, NO. 4, 2015

Incorporation of Neural Network to HPMHT for Tracking Multiple Targets

Ilke Turkmen[1], Harun Celik[1]
[1]*Erciyes University, Civil Aviation School, Department of Aircraft Electrical and Electronics, 38039, Kayseri, Turkey*
titi@erciyes.edu.tr

***Abstract*—In this paper, a hybrid method which combines homothetic multi-hypothesis tracker (HPMHT) and artificial neural networks (ANNs) is presented to solve multiple target tracking problem. The performances of the proposed neural network aided homothetic multi-hypothesis tracker (NNAHPMHT) and the HPMHT are compared for two different test scenarios. It was observed that the estimation performances obtained from the NNAHPMHT are better than those obtained from only the HPMHT. The NNAHPMHT method doesn't require additional complex modeling for tracking multiple targets. The additional implementation time originated from NNAHPMHT is only recall time of the ANN. For this reason, the proposed method is very suitable for real-time implementation.**

***Index Terms*—Target Tracking, neural network, homothetic probabilistic multi-hypothesis tracker.**

I. INTRODUCTION

Multi target tracking is the problem of estimation of the trajectory of objects using one or more sensors. In a cluttered environment, there are many moving objects in the sensor surveillance volume and it is not clear which measurement to which target belongs. The situation is complicated additionally by extra measurements, generated by clutter or noise in the radar receiver. Several methods [1]–[8] have been proposed and utilized for solving the data association problem. In multiple hypothesis tracker (MHT) presented by Reid [1], the obtained measurements at each scan are stated to initialized targets, new targets, or false alarms. A number of hypotheses are generated, every one of which supposes a possible assignment scheme between received in all scans measurements and new targets-confirmed, new ones or false. Pruning and gating techniques are utilized to keep on the most likely hypotheses and so limit their number. The main risk is the possibility of elimination of the correct hypotheses and it is more likely since the main interested sources may be weak and fluctuating.

The other approach called probabilistic data association filter (PDAF) [2], [3] and its developed version to multiple targets, joint PDAF (JPDAF) [2], solve the same data association problem in a simpler way. In the JPDAF, hypotheses are constituted for the measurements and targets only for the current scan. In this way, the number of hypotheses is additionally reduced but the combinatorial explosion in dense target and clutter scenarios doesn't change.

The MHT and JPDAF approaches described above are sub-optimal procedures. Another method called probabilistic MHT (PMHT) [4], [5], starts by thinking all measurements can originate from all targets, which is incorrect assumption. The PMHT is an algorithm which solves the multi-target tracking problem through application of the Expectation Maximisation algorithm. The computational load of the PMHT method linearly increases with all parameters such as window length, measurement number, models, sensors, and targets. The PMHT algorithm assumes that the number of targets is known, and that it is possible to initialize the track states. The problem of initialization was addressed by the introduction of a homothetic measurement model in [6]. Rather than a single Gaussian, the homothetic PMHT (HPMHT) [6], [7] uses a Gaussian mixture where each of the components has the same mean, but different covariance matrices.

Nowadays, some hybrid algorithms to track manoeuvring and multiple targets have been proposed to improve the tracking performance of the traditional tracking methods [9]–[16]. In [9], [10], the neural network (NN) was used to correct the Kalman filter errors for the MTT. The NN and neuro fuzzy system (NFS) were also used in [11] and [12] to increase tracking performance of the JPDAF method. In [13], by using good feature of both NN and traditional statistical model and adaptive algorithm method, an adaptive algorithm was presented to track maneuvering targets. In previous works [14], [15], we proposed hybrid systems combining the genetic tracker with NN [14] and genetic tracker with NFS [15] to solve the target tracking problem without requiring extensive computation. In [16], a tracking algorithm which integrates NN technique with the available knowledge on air combat maneuvering strategies was also presented.

The main contribution of this paper is to incorporate a NN to the HPMHT by taking advantages of this tracker described above for increasing tracking performance. The ANNs have the advantage of learning from examples, generalizability, less information requirement, low computation time requirement, and ease of usage properties [17]. Because of these advantages, in this paper, the learning skill of the neural network (NN) and the tracking skill of the

Manuscript received November 29, 2014; accepted April 24, 2015.

HPMHT are combined. The proposed NNAHPMHT method improves the estimation performance of the HPMHT tracker using the learning and adaptability properties of the neural network. In the following sections, the NNAHPMHT presented in this paper are presented briefly, and simulations used to compare performances of the HPMHT and the NNAHPMHT methods are then presented.

II. NEURAL NETWORK AIDED HPMHT

In the HPMHT, process noise variance is assumed as a white noise, but it is a correlated noise sequence in real applications. For this reason, the estimation accuracy of the HPMHT is degraded in real environment. In order to compensate this error, in this paper, the multi layered perceptron (MLP) NN is incorporated into the HPMHT. The MLPs are commonly used NN architectures because of their simple structure. The important stages of the proposed NNAHPMHT are given below.

A. Selection of the Input and Output Parameters

To obtain good tracking performance, it is very important to choose the most suitable network architecture. The input parameters of the MLP architecture must be chosen carefully for achieving desired tracking accuracy. When the MLP is properly trained using data set including these input parameters, it can be utilized to increase the estimation performance.

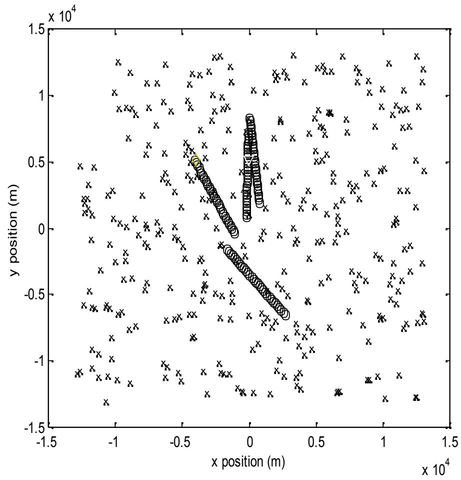
As it can be seen from the HPMHT, the error of the tracker is closely related with the linear moving model and the measurement model. Because the estimation vector and the prediction vector primarily affect the tracker performance, they can be used as the input parameters of the NN. Thus, the first input parameter of the NN is found by subtracting the measured position vector from the estimated position vectors for all positions. Similarly the second input of the NN is obtained by subtracting the estimated position vector from the predicted position vectors. Lastly, the third input of the NN is found by subtracting the estimated velocity vector from the predicted velocity vectors. The output parameter of the NN is correction of the estimation.

B. Generation of the Data Set

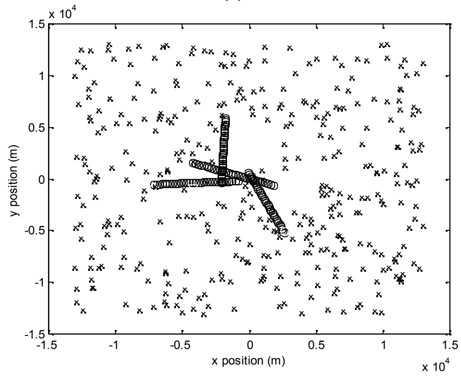
The data set used to train the neural model can be obtained from measurements and simulations for target tracking applications. The selection of these methods related with the problem to be solved and the existence of the data generator. In this paper, 480 data sets used in the training stage were obtained from the training trajectories shown in Fig. 1. After training, the ANN was utilized for improving the estimation accuracy for two test scenarios shown in Fig. 2(a) and Fig. 2(b).

C. Selection of the Network Architecture

In the MLP, transfer function is selected as the linear transfer function for output layer and the hyperbolic tangent functions for the hidden layers. After several trials, the most appropriate network architecture was found as two hidden layers with twelve neurons for each hidden layers. The data sets used to train NN were normalized between 0.0 and 1.0.

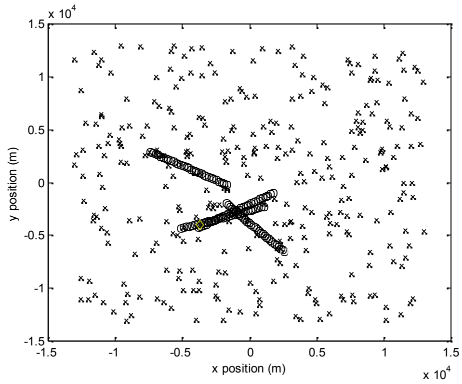


(a)

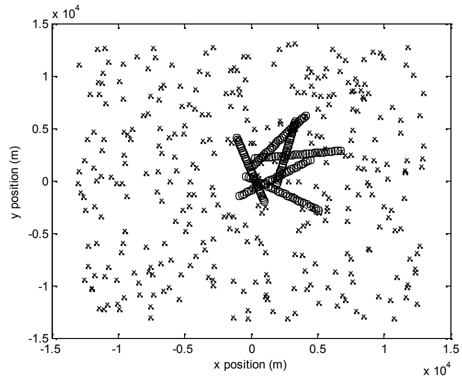


(b)

Fig. 1. Two training scenarios: The original tracks with $M = 4$, $\lambda = 5 \times 10^{-7}$, $P_d = 90$ %, $\sigma_p = 0.005$, and $\sigma_m = 100$. Detections are o's, and only of clutter the final scan is shown, with clutter returns denoted by ×.



(a)



(b)

Fig. 2. Two test scenarios: The original tracks (a) $M = 4$ and (b) $M = 6$ with, $\lambda = 5 \times 10^{-7}$, $P_d = 90$ %, $\sigma_p = 0.005$, and $\sigma_m = 100$. Detections are o's, and only of clutter the final scan is shown, with clutter returns denoted by ×.

The epoch number was selected as 500 for training. The seed number was assigned as 246. The MLPs can be trained with the use of different learning algorithms [17]. In this paper, Levenberg-Marquardt [18], [19] algorithm is preferred to train the MLP because of its fast learning and good convergence capabilities.

III. SIMULATIONS

Two simulation studies are performed to show the performance of the neural tracker that adds the MLP architecture into the HPMHT, and to compare its estimation results with those of the HPMHT. It is assumed that the model is two dimensional and kinematic for all targets and that only position measurement is available. In this situation, discrete-time linear kinematic model of the *s*th of target is given by:

$$x_s(t+1) = F_s(t)x_s(t) + G_s(t)u_s(t) + v_s(t), \tag{1}$$

$$y_s(t) = H_s(t)x_s(t) + w_s(t), \tag{2}$$

where $x_s(t)$ shows the system state of the *s*th target at time t, $y_s(t)$ is measurement vector at time t, $w_s(t)$ is process noise, $v_s(t)$ is measurement noise, $u_s(t)$ is control sequence, and $\{F_s(t), H_s(t), \text{and } G_s(t)\}$ are known matrices. State transition matrix $F_s(t)$ and measurement matrix $H_s(t)$ are defined as:

$$F_s(t) = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

$$H_s(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{4}$$

The covariance matrices of the process noise $w_s(t)$ and measurement noise $v_s(t)$ are given by:

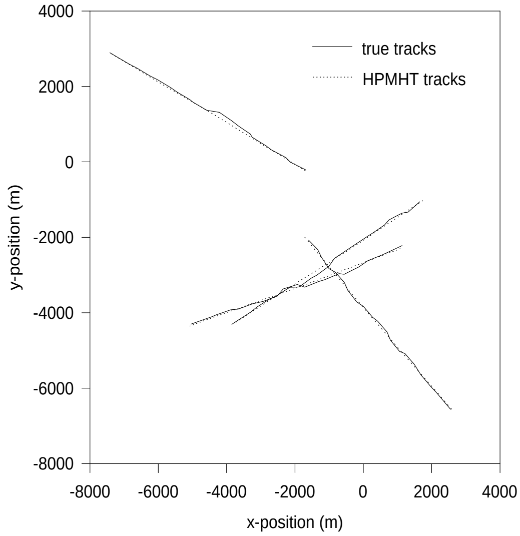
$$Q_s(t) = \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \pm_p^2, \tag{5}$$

$$R_s(t) = \begin{bmatrix} \pm_m^2 & 0 \\ 0 & \pm_m^2 \end{bmatrix}, \tag{6}$$

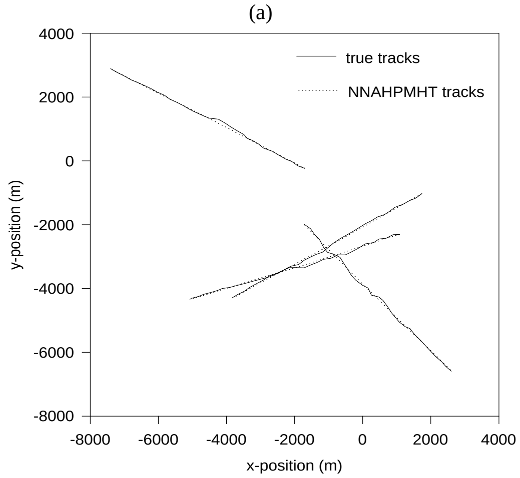
where $s = 1, 2, \ldots, M$ and measurement numbers $T = 30$. The sampling period is selected as $\Delta t = 30$ seconds. The process and measurement noise standard deviations $\pm_p$ and $\pm_m$ are equal to 0.005 and 100, respectively. False alarms are Poisson distributed which have spatial density λ. In the simulation, two-point initialization is used and the detection probability P_d is selected as 90 %. To constitute the M different target trajectories the following method is adopted [5]:

1. Constitute uniformly distributed M points in a preset circle to obtain the beginning points of M targets. The value of initial velocity magnitudes are the same for M targets and directions of these targets are uniformly distributed from 0° to 360°.
2. Next generate the all trajectories from the starting points by using (1).

The performances of the HPMHT and the NNAHPMHT are illustrated in Fig. 3 and Fig. 4 for two test scenarios.

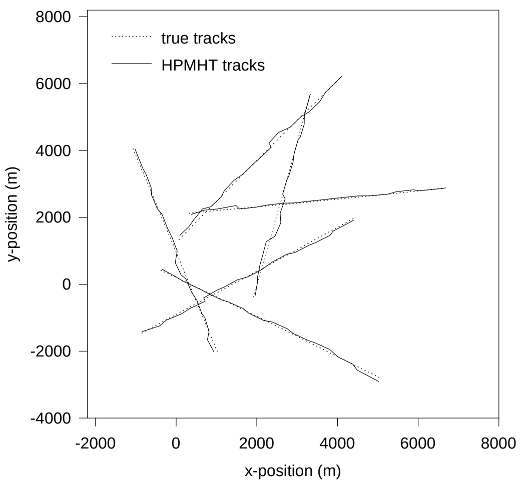


(a)

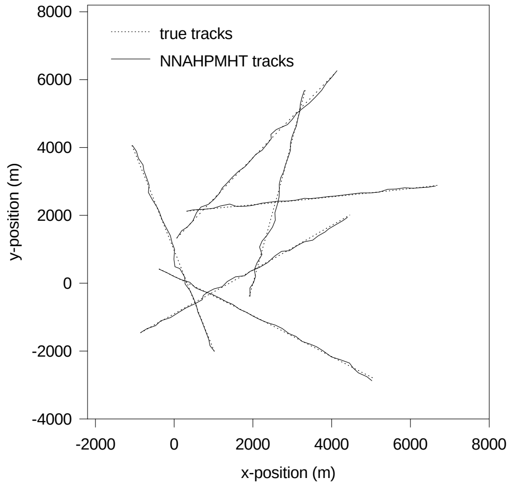


(b)

Fig. 3. True and estimated tracks for test scenario 1 using (a) HPMHT and (b) NNAHPMHT.



(a)



(b)

Fig. 4. True and estimated tracks for test scenario 2 using a) HPMHT and b) NNAHPMHT.

It is clearly seen from Fig. 3 and Fig. 4 that the tracks obtained from NNAHPMHT are more close to the original tracks than the tracks obtained by the HPMHT for two scenarios. Table I illustrates the RMS errors of the HPMHT and the NNAHPMHT methods for different test scenarios. The percentage improvement obtained with the use of NNAHPMHT is calculated as the ratio of the RMS error difference of the HPMHT and the NNAHPMHT method to the RMS error of the HPMHT in percent.

TABLE I. PERFORMANCE COMPARISON BETWEEN THE HPMHT AND THE NNAHPMHT.

Test Scenarios	**Targets**	**RMS Tracking Error (m)**		**Percentage Improvement with NNAHPMHT (%)**
		HPMHT	**NNAHPMHT**	
1	1	67.614	46.313	32
	2	64.452	41.175	36
	3	71.899	47.985	33
	4	62.373	41.471	34
2	1	52.178	40.148	23
	2	88.053	74.762	16
	3	76.639	55.698	27
	4	70.661	51.817	27
	5	64.719	29.227	24
	6	59.181	40.383	32

It is apparent from Table I that the tracking performance of the NNAHPMHT is better than that of the HPMHT for all simulations. The RMS errors given in Table I clearly show that a significant improvement is achieved on the tracking results of the HPMHT. The average percentage improvement is about 28 % when NN is employed. The addition of a NN to the HPMHT provides good accuracy in MTT.

IV. Conclusions

The NNAHPMHT method, that incorporates the learning skill of the NN and the tracking skill of the HPMHT, is presented for tracking multiple targets. In this method, the estimation error of the HPMHT is improved by using the NN. The results of the NNAHPMHT method illustrate a better agreement with the true tracks than the results of the HPMHT. The obtained good results show the validity of NNAHPMHT method. The additional implementation time required for the NNAHPMHT is only recall time of the ANN. For this reason, the real-time implementation of the proposed method is very easy.

References

[1] D. Reid, “An algorithm for tracking multiple targets”, *IEEE Trans. Automatic Control*, vol. 24, pp. 843–854, 1979. [Online]. Available: http://dx.doi.org/10.1109/TAC.1979.1102177

[2] Y. Bar-Shalom, X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, 1995.

[3] H. Quan, A. Xue, “Joint target tracking and classification with heterogeneous sensors”, *Elektronika Ir Elektrotechnika*, no. 4, pp. 61–64, 2012. [Online]. Available: http://dx.doi.org/10.5755/j01.eee.120.4.1454

[4] C. Rago, P. Willett, R. Streit, “Direct data fusion using PMHT”, in *1996 Proc. The American Conf.*, Seattle, Washington, pp. 1698–1702.

[5] Y. Ruan, P. Willett, R. Streit, “Comparison of PMHT and S-D assignment trackers”, in *1999 Proc. of the SPIE Conf.*, Orlando, pp. 280–291.

[6] C. Rago, P. Willett, R. Streit, “A comparison of the JPDAF and PMHT tracking algorithms”, in *1995 Int. Conf. Acoustics, Speech and Signal Processing*, pp. 3571–3574. [Online]. Available: http://dx.doi.org/10.1109/icassp.1995.479758

[7] C, Rago, P. Willett, R. Streit, “A modified PMHT”, in *Proc. The 1995 Conf. Information Sciences and Systems*, Baltimore MD.

[8] D. Huang, A. Xue, Y.Guo, “A particle filter track-before-detect algorithm for multi-radar system”, *Elektronika Ir Elektrotechnika*, vol. 19, no. 5, pp. 3–8, 2013. [Online]. Available: http://dx.doi.org/10.5755/j01.eee.19.5.4363

[9] L. Chin, “Applications of the neural network approaches to multiple target tracking”, *IEEE Trans. Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 1220–1223, 1994. [Online]. Available: http://dx.doi.org/10.1109/7.250437

[10] V. Vaidehi, N. Chitra, M. Chokkalingam, C. N. Krishnan, “Neural network aided Kalman filtering for multitarget tracking applications”, *Computers & Electrical Engineering*, vol. 27, no. 1, pp. 217–228, 2001. [Online]. Available: http://dx.doi.org/10.1016/S0045-7906(00)00013-6

[11] L. Chin, “Application of neural networks aided target tracking”, *IEEE Int. Conf. Neural Networks*, vol. 3, pp. 1634–1639, 1996. [Online]. Available: http://dx.doi.org/10.1109/icnn.1996.549145

[12] I. P. W Ching, L. Yongzhi, L. Chin, D. Mital, “Neuro-fuzzy techniques for airborne target tracking”, *1998 Second Int. Conf. on Knowledge-Based Intellegent Electronic Systems*, pp. 251–257. [Online]. Available: http://dx.doi.org/10.1109/kes.1998.725919

[13] N. Sadati, D. Langary, “A neural network aided target tracking algorithm using angular measurements”, in *Intelligent Sensors, Sensor Networks and Information Processing Conf.*, Melbourne, 2005, pp. 295–300. [Online]. Available: http://dx.doi.org/10.1109/issnip.2005.1595595

[14] I. Turkmen, K. Guney, D. Karaboga, “Genetic tracker with neural network for single and multiple target tracking”, *Neurocomputing*, vol. 69, pp. 2309–2319, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2005.04.014

[15] I. Turkmen, K. Guney, “Genetic tracker with adaptive neuro-fuzzy inference system for multiple target tracking”, *Expert Systems with Applications*, vol. 35, no. 4, pp. 1657–1667, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2007.08.065

[16] A. N. Pineda, L. U. Aragones, J. R. F. C. Diez, M. A. P. Guisado, “Radar tracking system using contextual information on a neural network architecture in air combat maneuvering”, *International Journal of Distributed Sensor Networks*, vol. 2013, p. 11.

[17] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York: Macmillan College Publishing Company, 1994.

[18] K. Levenberg, “A method for the solution of certain nonlinear problems in least squares”, *Quart Appl Math*, vol. 2, pp. 164–168, 1944.

[19] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters”, *J Soc Ind Appl Math*, vol. 11, pp. 431–441, 1963. [Online]. Available: http://dx.doi.org/10.1137/0111030