TARGET TRACKING WITH THE USE OF NEURAL NETWORKS

Ilke TITI, Ahmet KAPLAN
Erciyes University, Civil Aviation School, Kayseri, Turkey.

Keywords: *Target Tracking, Neural Networks*

Abstract

In this work, a simple method for target tracking, based on Artificial Neural Networks (ANN), is presented. The backpropagation algorithm is used to train the networks by using the measured position, velocity and acceleration data sets obtained from six different types of aircraft radars such as cargo, bomber, fighter and commercial aircrafts. The test results of ANNs are in very good agreement with the measured results. The results of ANNs are also compared with the results of Kalman filter which is widely used in target tracking. It is shown that the results predicted by using ANNs are better than those predicted by Kalman filter.

1 Introduction

Target tracking is an important issue in military surveillance systems, ballistic missile defense systems, satellite defense systems and air traffic control systems. The objective of target tracking is to partition sensor data into sets of observations, or tracks produced by same source. Once tracks are formed and confirmed, the number of targets can be estimated and parameters such as position, velocity and acceleration can be obtained from each track.

Kalman filter is widely used in the tracking problem [1,2]. It can optimally estimate the target motion from noisy radar data. The optimality of the Kalman filter is based on the assumption of the Gaussian noise. If the assumption is violated, the Kalman filter is no longer the optimal filter. In a radar system, due to the target glint, the measurement noise may present non-Gaussian behavior. If noise is non-Gaussian, tracking performance of the Kalman filter can decrease seriously.

In this study, a method based on ANNs [3-11] that have advantages of ability and adaptability to learn, generalizability, smaller information requirement, fast real time operation, and ease of implementation features will be presented for getting rid of the disadvantages of Kalman filter mentioned above. ANNs in this article are used to estimate target parameters such as position, velocity and acceleration.

Multilayered perceptrons (MLP) and backpropagation algorithm which are used to train MLP will be explained Section 2.Then in Section 3, Kalman filter will be presented and in Section 4, target tracking using artificial neural networks will be explained. And in the last section, results obtained from this study will be discussed.

2 Multilayered Perceptrons

MLPs [7] are the simplest and therefore most commonly used neural network architectures. They have been adapted for the estimation of the position of the six different targets. MLPs can be trained using many different learning algorithms.

In this work, MLPs are trained with a supervised learning algorithm called backpropagation algorithm. As shown Fig 1 an MLP consists of three layers: an input layer, an output layer and a hidden layer. Processing elements (PEs) or neurons in the input layer only act as buffers for distributing the input signals x_i to PEs in the hidden layer. Each PE j in the hidden layer sums up its input signals x_i after weighting them with the strengths of the respective connections w_{ji}

from the input layer and computes its output y_j as a function f of the sum, viz.,

$$y_j = f\left(\sum w_{ji} x_i\right) \qquad (1)$$

f can be a simple threshold function, a sigmodial or hyperbolic tangent function. The outputs of PEs in the output layer is computed similarly.

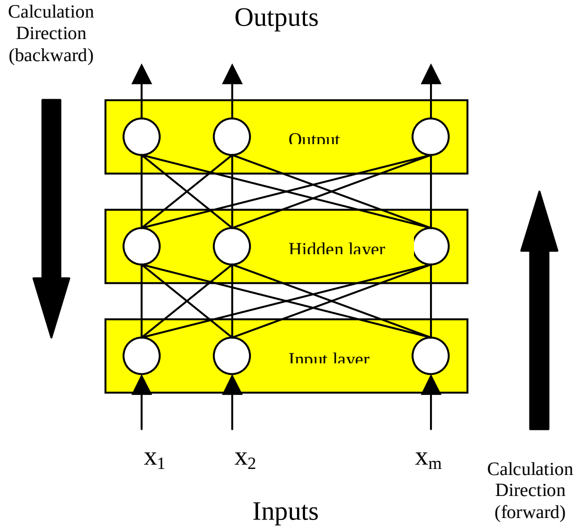


Fig. 1. General form of a backpropagation multilayered perceptron.

Training a network consist of adjusting weights of the network using the different learning algorithms. A learning algorithm gives the change $\Delta w_{ji}(k)$ in the weight of a connection between PEs i and j.

2.1 Backpropagation Algorithm

The algorithm [7] is the most commonly adopted MLP training algorithm. It is a Gradient descent algorithm and gives the change $\Delta w_{ji}(k)$ in the weight of a connection between PEs i and j as follows,

$$\Delta w_{ij}(k) = \eta \delta_j x_i + \alpha \Delta w_{ji}(k-1) \qquad (2)$$

where η is a parameter called the learning coefficient, α is the momentum coefficient, and δ_j is a factor depending on whether PE j is an output PE or a hidden PE.

3 Kalman Filter

The Kalman filter is used to estimate the state $x \in \Re^R$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_{k+1} = A_k x_k + B u_k + w_k \qquad (3)$$

with a measurement

$$z_k = H_k x_k + v_k \qquad (4)$$

where x_k is true system state vector, A is state transition matrix, B is control input matrix, u is system input vector, z_k is true system measurement vector, H_k is output matrix. The random variables w_k and v_k represent the process and measurement noise respectively. They are assumed to be independent of each other, white, and with normal probability distributions

p(w)~N(0, Q)
p(v)~N(0, R).

$\hat{x}_k^- \in \Re^R$ is defined as a prior state estimate at step k given knowledge of the process prior to step k, and $\hat{x}_k \in \Re^R$ as a posteriori state estimate at step k given measurement z_k. The a priori and a posteriori estimate errors can be defined as

$$e_k^- = x_k - \hat{x}_k^-$$

and

$$e_k = x_k - \hat{x}_k .$$

The priori error covariance is then

$$P_k^- = E\left[e_k^- e_k^{-T}\right] \qquad (5)$$

and the posteriori estimate error covariance is

$$P_k = E\left[e_k e_k^T\right] \qquad (6)$$

In deriving the equations for the Kalman filter, an equation that computes an posteriori state estimate $\hat{x}_k$ as a linear combination of a priori estimate $\hat{x}_k^-$ and a weighed difference between an actual measurement z_k and a measurement prediction $H_k \hat{x}_k^-$ is found as shown equation (7).

$$\hat{x}_k = \hat{x}_k^- + K\left(z_k - H_k \hat{x}_k^-\right) \tag{7}$$

The difference (z_k- $H_k \hat{x}_k^-$) in (7) is called the measurement innovation, or the residual. The residual reflects the discrepancy between the predicted measurement $H_k \hat{x}_k^-$ and actual measurement z_k.

The n×m matrix K in (7) is chosen to be the gain that minimizes the posteriori error covariance (6). This minimization can be accomplished by first substituting (7) into the above definition for e_k, substituting that into (6), performing the indicated expectations, taking the derivative of the trace of the result with respect to K, setting that result equal to zero, and then solving for K. One form of the resulting K that minimizes (6) is given by

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \tag{8}$$

The Kalman filter estimates a process by feedback control: after estimating the process state the filter obtains feedback by noisy measurements. The equation for the Kalman filter are grouped in two: time update equations and measurement update equations. The time update equations are used for projecting forward in time the current state and error covariance estimates to obtain the priori estimates for the next step. The measurement update equations are used for the feedback.

The specific equations for the time and measurement updates are presented below. The time update equations are

$$\hat{x}_{k+1} = A_k \hat{x}_k + B u_k \tag{9}$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k \tag{10}$$

and measurement update equations are

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \tag{11}$$

$$\hat{x}_k = \hat{x}_k^- + K\left(z_k - H_k \hat{x}_k^-\right) \tag{12}$$

$$P_k = \left(I - K_k H_k\right) P_k^- \tag{13}$$

The first task during the measurement update is to compute the Kalman gain, K_k. The next step is to actually measure the process to obtain z_k, and then to generate an a posteriori state estimate by incorporating the measurement as in (12). The final step is to obtain an a posteriori error covariance estimate via (13).

4 Target Tracking using Artificial Neural Networks

In this study, a method based on ANNs is presented for six different aircrafts such as cargo, bomber, fighter and commercial aircrafts. Target trajectories are obtained from real aircrafts [12]. Figure 1 shows neural model used in this study. As explained before, BP algorithm is used to train MLPs.

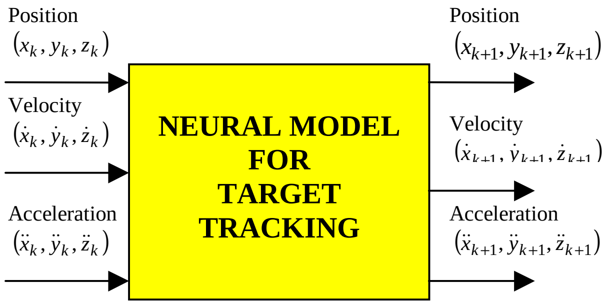


Fig. 2. Neural Model for Target Tracking

Target trajectories are shown in Fig.3. The first target trajectory represents a large aircraft, such as a military cargo aircraft. The second trajectory represents a smaller, more maneuverable aircraft’s, such as a Learjet or other similar high performance commercial aircraft. The trajectories of Target 3 and 4 represent medium bombers’ flying at high speeds with good maneuverability. Target 5 and 6 represent the trajectory of fighter/attack aircraft.

Data sets that are used for training and testing ANN are obtained from radar measurements. In this study, ANN is trained for three different situations. ANN is trained in the first situation with the data sets obtained by the positions of the targets; in the second situation, with data sets obtained by positions and velocities of the targets; and in the third situation, with data sets obtained by positions, velocities and acceleration of the targets.

A set of random values distributed uniformly between -0.1 and +0.1 is used to initialize the weights of the networks. However, the input data tuples are scaled between -1.0 and +1.0 and output data tuples are also scaled between -0.8 and +0.8 before training.

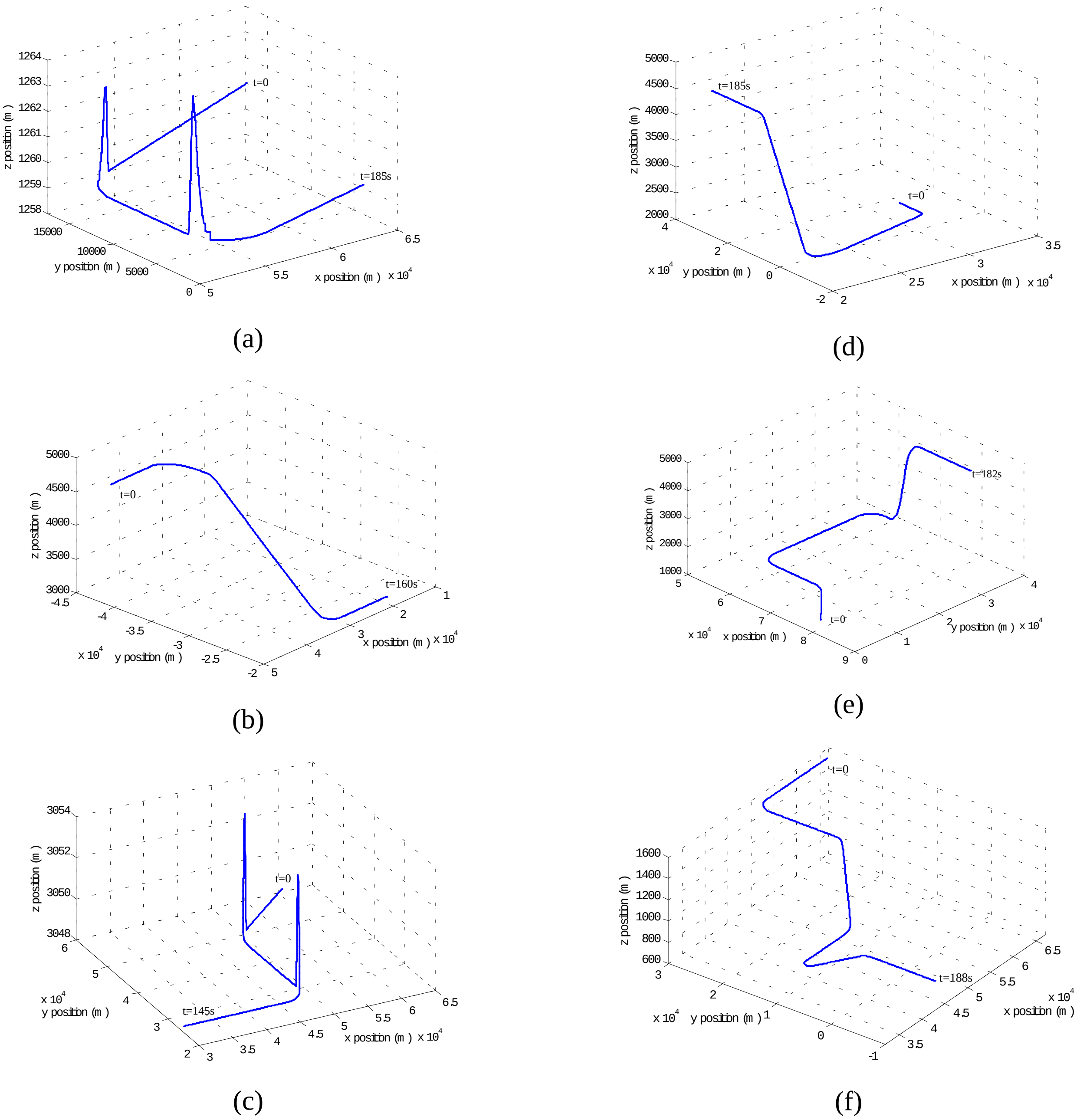


Fig. 3. Target trajectories

(a) Target 1 (b) Target 2 (c) Target 3 (d) Target 4 (e) Target 5 (f) Target 6

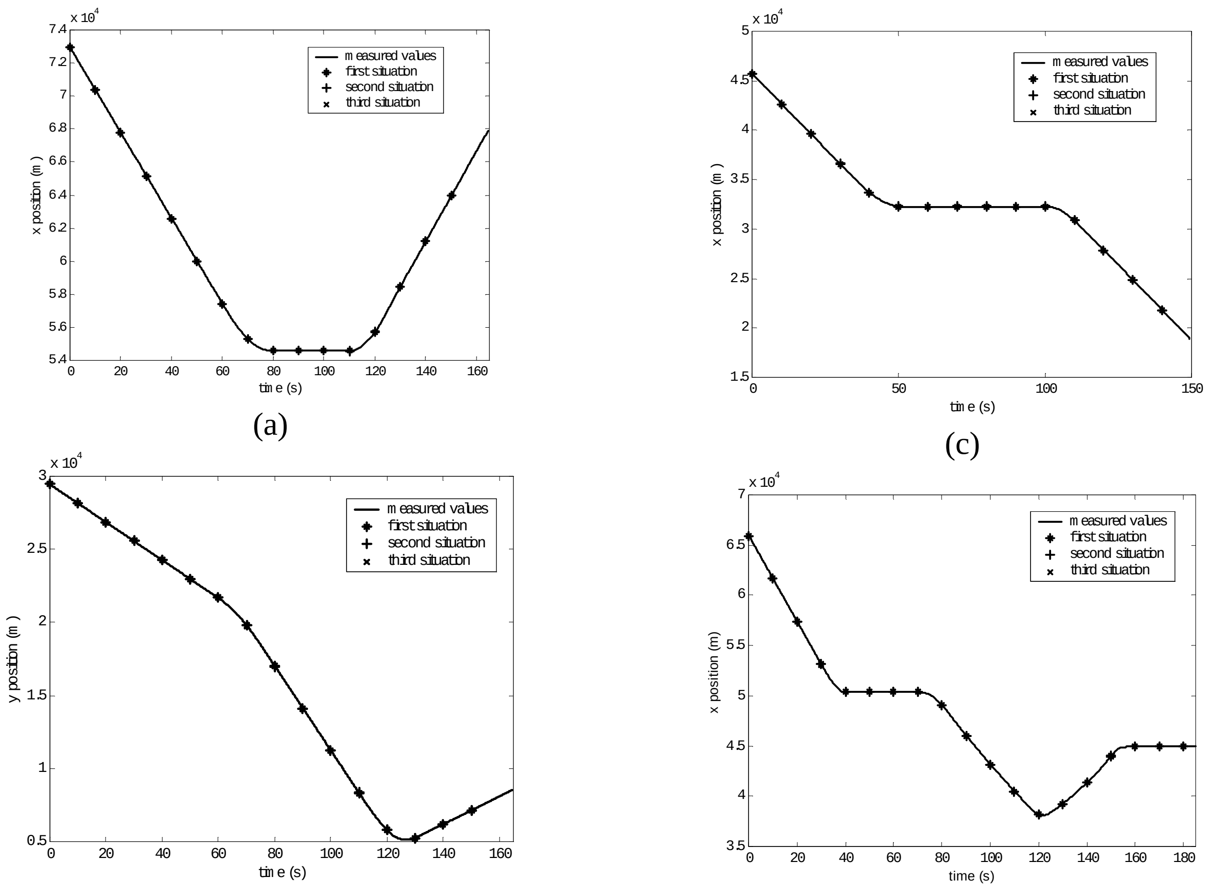
After several trials, it is found that the best estimations are achieved by using two layers in ANN. The most suitable network configuration found is six PEs and three PEs for first and the second hidden layers, respectively.

The parameters of the networks are for BP: the learning coefficients (η) were 0.3 for the first hidden layer, 0.25 for second hidden layer and 0.2 for the output layer, and the momentum coefficient (α) was also set to 0.4.

5 Results and Conclusions

In order to show the performance of the ANNs used in target tracking, position estimation test results of the first, second and third targets are compared with the measured results in the Fig. 4. It is shown Fig. 4. that results obtained from ANNs are in very good agreement with the measured results in the third situation. Only position estimation results of the three targets are given here. For other three targets, similar good results are obtained.

In this study; Kalman filter is also used in order to compare performance of ANNs in target tracking. Figure 5 shows the errors of the Kalman filter and ANNs for the x position of the first target. It can be seen this figure that results predicted by using ANN are better than those predicted by Kalman filter. The advantage of the neural models given here are simplicity and accuracy.



(a) (b) (c) (d)

Fig. 4. Position estimation test results obtained from ANN
(a) x position of the target 1 (b) y position of the target 1
(c) x position of the target 2 (d) x position of the target 6

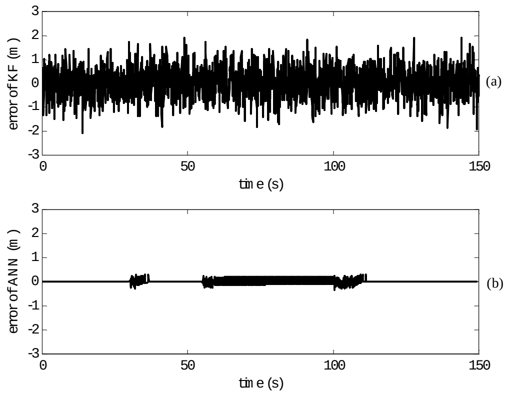


Fig. 5. (a) Error of the Kalman filter
(b) Error of the ANN

References

[1] Welch G., Bishop G., An Introduction to the Kalman Filter, *http:// www. cs. unc. edu/ ~welch/ kalman/ kalman.html.*

[2] Wu W.R., Cheng P.P., A nonlinear IMM Algorithm for Maneuvering Target Tracking, *IEEE Transactions on Aerospace and Electronic Systems*, AES-30, 3, 875-886, 1994.

[3] Sung, T. and Hu, Y., Parallel VLSI Implementation of the Kalman Filter, *IEE Transactions on Aerospace and Electronic Systems*, AES-23, 2, 215-224, 1987.

[4] Baheti, R. S., Efficent Approximations of Kalman Filter for Target Tracking, *IEEE Transactions on Aerospace and Electronic Systems*, AES-22, 1, 8-14, 1986.

[5] Papadourakis, G. M. and Taylor, F. J., Implementation of Kalman Filters Using Systholic Arrays, *In proceedings of the Internaitonal Conference on Acoustics, Speech and Signal Processing*, Dallas, TX, 2, 783-786, 1987.

[6] Kung S. Y., and Hwang, J. N., Systolic Array Desings for Kalman Filtering, *IEEE Transactions on Signal Processing*, 39, 1, 1991.

[7] Rumelhart, D.E. and McClelland, J. L., *Parallel distributed Processing*, Vol. 1, MIT Press, Cambridge, 1986.

[8] Maren, A., Harston, C. and Pap., R., *Handbook of neural computing applications*, Academic Press, London, ISBN 0-12-471260-6, 1990

[9] Haykin, S., “Neural networks: A comprehensive foundation”, Macmillan Collage Co., New York, ISBN 0-02-352761-7, 1994.

[10] Chin,L., Application of Neural Networks in Data Fusion, *In Proceedings of the IEEE Internetional Conference on Intellegent Control and Instrumentation*, Singapure, 1992.

[11] Sagiroglu, S., Güney, K. and Erler, M., Calculation of Bandwidth for Electrically Thin and Thick Rectangular Microstrip Antennas with the Use of Multilayered Perceptrons, *Int. J. of RF and Microwave Computer-Aided Engineering*, 9, 277-286, 1999.

[12] Blair, W.D., Watson, G.A., Kirubarajan, T. et al., Benchmark for Radar Allocation and Tracking in ECM, *IEEE Transactions on Aerospace and Electronic Systems*, 34, 4, 1097-1114, 1998.