Defence Science Journal, Vol. 55, No. 3, July 2005, pp. 313-328

Target Acceleration Estimation from Radar Position Data using Neural Network

A.K. Sarkar
Defence Research & Development Laboratory, Hyderbad–500 058

and

S. Vathsal
Directorate of ER & IPR, New Delhi–110 066

and

Suresh Sundaram
Indian Institute of Science, Bangalore–560 012

and

S. Mukhopadhay
Indian Institute of Technology, Kharagpur–721 302

ABSTRACT

This work is a preliminary investigation on target manoeuvre estimation in real-time from the available measurements of noisy position data from tracking radar using an artificial neural network (ANN). Recently, simulation study of target manoeuvre estimation in real-time from the same position alone measurement using extended Kalman filter has been carried out in a simulated environment using measurements at 100 ms interval. The results reveal that the estimated acceleration consists of substantial error and lag, which is a stumbling block for guidance accuracy in real-time. So, the target acceleration has been estimated using the ANN with less error and lag than the same using Kalman estimator.

Keywords: Kalman filter, artificial neural network, line-of-sight, feedforward neural network, target acceleration estimation, augmented proportional navigation

NOMENCLATURE

(x,y,z) Downrange, cross range, height

$(\dot{x},\dot{y},\dot{z})$ Velocity components along downrange, cross range, height

$(\ddot{x},\ddot{y},\ddot{z})$ Acceleration components along downrange, cross range, height

(R_m,A_m,E_m) Measured range, azimuth and elevation

Received 15 November 2004

(γ, ϕ)	Flight path and heading angles
(η_v, η_h)	Vertical, horizontal load factor
η	$\sqrt{\eta_v^2 + \eta_h^2}$ Total load factor
ω_T	Target weaving frequency
M	Total number of layers including input and output in NN
N_s	Total number of measurement samples to be trained
N_n	Total number of layers in the n^{th} layer
$Node\ (n,i)$	i^{th} node in the n^{th} layer
$x_i(t)$	i^{th} element for sampled data t
$x_i^n(t)$	Output of the node (n,i)
$a_{i,j}^n$	Link-weight from the node (n,j) to $(n+1,\ i)$
θ_i^n	Offset of the node (n,i)

1. INTRODUCTION

The guidance problem of a pursuer flight vehicle (FV) to intercept a moving target is generally solved using well-known proportional navigation (PN) law,in which the line-of-sight (LOS) angular rate between the evader and the interceptor is used for interception[1]. This technique works well in a low-noise environment and also when the target maintains a constant velocity. However,in a noisy scenario in which the target initiates an evasive manoeuvre during endgame, the PN guidance laws are not capable of providing the necessary guidance commands to ensure interception[1] within given structural limit of the flight vehicle.

To improve the performance of the interceptor,it is necessary to accurately estimate target position, velocity, and acceleration components. Augmented proportional navigation (APN) is a popular guidance law which performs better over the PN law because it uses target acceleration levels to generate guidance command[1]. So, for using APN guidance law, estimation of target manoeuvre is mandatory. The present problem addresses estimation of target position, velocity, and acceleration components from the noisy position alone measurements using an artificial neural network (ANN).

In a recent work by Ananthasayanam[2], *et al.* the authors have estimated the target position, velocity, and acceleration from noisy range, azimuth and elevation (R_m, A_m, E_m). The tentative figure of (30 m, 0.2°, 0.2°) has been taken as (one σ) radar measurement noise. In this paper, the sinusoidal types of target manoeuvre at 3g, 5g, and 7g have been estimated using extended Kalman filter (EKF). Through simulation study, it has been seen that while tracking a target with 7 g manoeuvre, even during nonlinear zone, the estimation error becomes close to 2g.

Open literatures on target manoeuvre estimation from noisy position data from tracking radar data are very scanty. A detailed literature survey has been carried out by Ananthasayanam[2], *et al.* Though the Kalman filtering is a fundamental building block for target manoeuvre estimation, the ANN techniques have also been used by some researchers to improve the estimation accuracy of Kalman estimates. Chin[3] has proposed a neural network-aided Kalman filter tracker to improve the accuracy of EK estimated position and velocity. He has demonstrated improvement in estimation using ANN in planar situation.But in his work, he has dealt with non-manoeuvreing target and the details of ANN training is missing. Later based on his work, Vaidehi[4], *et al.* have used ANN-aided Kalman filter for multi-target tracking applications. They have also reported that by aiding ANN with Kalman filter, estimation accuracy in both position and velocity improves considerably.

In the present problem also Chin's idea was tried initially. Both Chin[3] and Vaidehi[4], *et al.* trained ANN with EK estimated gain,innovation sequence and prediction error as input and estimation error wrt truth model as output[3,4]. Subsequently, the required set of elements obtained from EK estimator was used as input to ANN to obtain estimation error wrt truth model as output[3,4]. The EKF estimates were compensated by this ANN output which reduced the estimation error considerably. But unfortunately, this concept didn't work in the present scenario. The main reason behind Chin's idea didn't work[2] as experienced by Ananthasayanam[2] because optimum value of Q-matrix has been used in filter tuning and estimation accuracy is within the statistical bounds of measurement noise. This by no means can be improved. So in this study, the direct radar measurements (R_m, A_m, E_m) have been used as input to train the ANN with actual accelerations as output. From the trained ANN, corresponding to a given (R_m, A_m, E_m), the accelerations have been predicted. Here, the acceleration estimation error is much less than the same obtained by the EKF as discussed by Ananthasayanam[2].

The paper is organised as follows. At first, Feedforward neural network (FFNN) type of architecture used in the present application has been described in brief. Then the generation of radar measurement (R_m, A_m, E_m) also is described. Subsequently, training of the present neural network architecture using different manoeuvre sets is discussed. Then corresponding to a given (R_m, A_m, E_m) prediction of acceleration from the trained network is discussed. Based on the previous and the present study, a combined EKF and ANN architecture has been proposed for traget manoeuvre estimation. At last, the major research activities to be carried out in future to estimate the target acceleration from randomly manoeuvreing target in real-life scenario is discussed.

2. MATHEMATICAL FRAMEWORK

In this section, feedforward neural network type of architecture used in the present application has been described and generation of (R_m, A_m, E_m) corresponding to a typical manoeuvre has been discussed.

2.1 Feedforward Neural Network

Feedforward neural networks (FFNN) are used in a variety of applications because these are adaptive,learn from example, and can provide excellent functional approximations. An FFNN consists of several layers of nodes which express artificial neural units (Fig. 1). Each node, which is connected by the links with all nodes in the adjacent layer, completes a weighted sum of input and then add an offset to the sum. The computed result is output through a nonlinear function. In the present network, no operation is performed in input layer, that is:

$$x_i^1 = x_i(t) \quad 1 \le i \le (N_1-1)$$

Furthermore, the offset bias in each layer is defined by

$$x_{Nn}^n(t)=1 \qquad 1 \le n \le M$$

$$a_{i,Nn}^n = \theta_i^{n+1} \qquad 1 \le n \le M-1 \ \& \ 1 \le i \le N_{n+1}-1 \tag{1}$$

Output of node (n+1, i) is:

$$x_i^{n+1}(t) = f\left(\sum_{j=1}^{N_{n-1}} a_{i,j}^n x_j^n(t) + \theta_i^{n+1}\right)$$

$$= f\left(\sum_{j=1}^{N_n} a_{i,j}^n x_j^n(t)\right) \tag{2}$$

The sigmoid activation function $f(.)$ and its derivative are:

$$f(x) = \frac{1}{1+e^{-x}} \ ; \ f'(x) = f(x)(1-f(x)) \tag{3}$$

Let the input set at layer −1 be defined as $\{x_1^1, x_2^1, x_3^1, ..., x_{N_1}^1\}$ as $r^{(1)}$ and the output set at layer-M as $\{x_1^M, x_2^M, x_3^M, ..., x_{N_M}^M\}$ as $r^{(M)} = z$. Then, the functional relationship between input and output is $z = h(r^{(1)}, a)$. Now referring to Fig. 2, one gets:

$$q^{(1)} = a^{(1)} r^{(1)}$$

$$r^{(2)} = f^{(1)}(r^{(1)})$$

$$q^{(2)} = a^{(2)} \ r^{(2)}$$

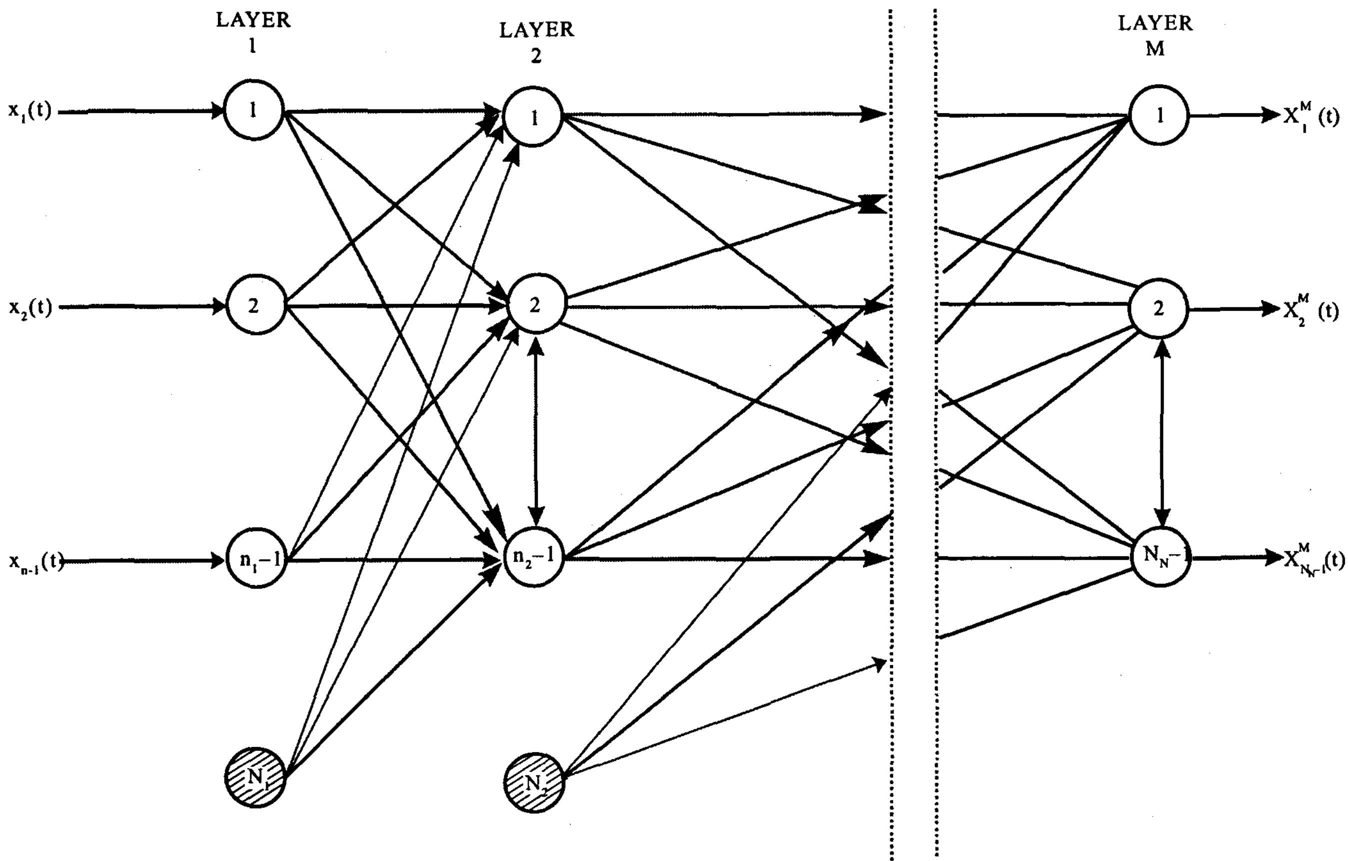

Figure 1. Structure of feedforward neural network

$$r^{(3)} = f^{(2)}(\ f^{(1)}(r^{(1)}))$$

$$\vdots$$

$$q^{(M-1)} = a^{(M-1)}\ r^{(M-1)}$$

$$r^{(M)} = f^{(M-1)}\ (a^{(M-1)}\ f^{(M-1)}$$

$$(\ldots f^{(3)}(a^{(3)}f^{(2)}(f^{(1)}(r^{(1)})))))\qquad(4)$$

$\{a^{(1)}, a^{(2)}, \ldots\ a^{(M-1)}\}$ are the link-weight elements in different layers of network which have to be computed through adaptive learning process. A popular FFNN training algorithm is back-propagation (BP) algorithm.

2.1.1 *Back-propagation Training Algorithm*

Learning of neurons consists of adjusting all link weights such that error measured between the desired output signal, d_m , and the actual output signal, x^M, averaged over all learning examples of M layers will be minimum in least square sense. The back-propagation algorithm uses steepest-descend gradient approach to minimise the error function. The equations for link-weight updation in back-propagation algorithm is:

$$\Delta a_{ji}^{(s)}(k) = \eta\delta_j^{(s)} r_i^{(s-1)} + \alpha\Delta a_{ji}^{(s)}(k-1)$$

$$\Delta a_{ji}^{(s)}(k+1) = a_{ji}^{(s)}(k) + \Delta a_{ji}^{(s)}(k)\qquad(5)$$

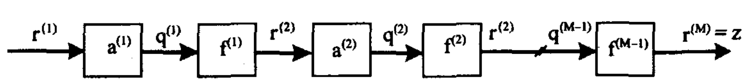

Figure 2. Block diagram representation of feedforward neural network

where $(\eta, \alpha)(\eta > 0$ and $0 \le \alpha < 1)$are learning and momentum update parameters for back-propagation algorithm. where s denotes current layer for link-weight updatement[5].

2.2 Target Modelling, Measurement, Data Generation

Periodic manoeuvre sequence such as sinusoidal or weaving target presents a challenge for a missile guidance system designer[1]. The target manoeuvre in pitch and yaw planes can be of

$$\eta_{ht}=\eta_y \sin\omega_T t,\ \eta_{vt} = \eta_p \sin\omega_T t$$

$$\eta_{ht}=\eta_y \cos\omega_T t,\ \eta_{vt} = \eta_p \cos\omega_T t \qquad (6)$$

A constant-speed target is considered. Governing equations of motion of target are:

$$\frac{dV_t}{dt} = 0 \qquad V_t(t_0)=V_{t0}$$

$$\frac{d\gamma_t}{dt} = \eta_{vt}\frac{g}{V_t} \qquad \gamma_t(t_0)=\gamma_{t0}$$

$$\frac{d\phi_t}{dt} = \frac{\eta_{ht} g}{V_t \cos\gamma_t} \qquad \phi_t(t_0)=\phi_{t0} \qquad (7)$$

$$\frac{dh_t}{dt} = V_t \sin\gamma_t \qquad h_t(t_0)=h_{t0}$$

$$\frac{dy_t}{dt} = V_t \cos\gamma_t \sin\phi_t \qquad y_t(t_0)=y_{t0}$$

$$\frac{dx_t}{dt} = V_t \cos\gamma_t \cos\phi_t \qquad x_t(t_0)=x_{t0}$$

Acceleration vector in cartesian frame is:

$$\begin{bmatrix} \ddot{x}_t \\ \ddot{y}_t \\ \ddot{z}_t \end{bmatrix} = C_I^B \begin{bmatrix} \dfrac{dV_t}{dt} \\ V_t \cos\gamma_t \dfrac{d\phi_t}{dt} \\ V_t \dfrac{d\gamma_t}{dt} \end{bmatrix} \qquad (8)$$

The position and velocity components in cartesian frame $(x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t)$ can be obtained by solving the equations Eqn 7. Tracking radar measurements (R_m, A_m, E_m) can be generated by contaminating (R, A, E) obtained from (x_t, y_t, z_t) by uncorrelated Gaussian noise (Fig.3).

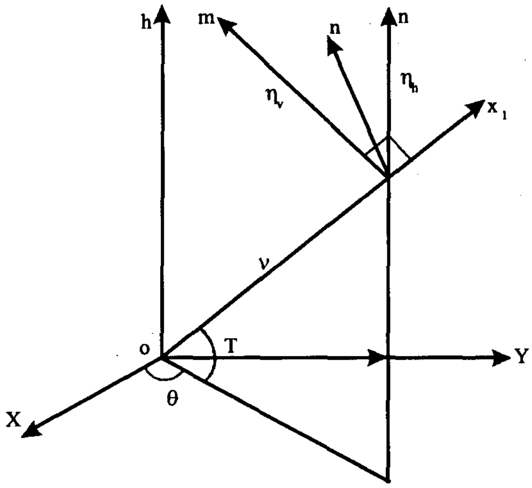

Figure 3. Axes system for 3-D point mass model of target dynamics.

3. SIMULATION STUDIES

Target data is generated as in Section 2.2. After generating the data for 50 s of flight time, it is contaminated by Gaussian noise of (σ_R =30 m/s, σ_A = 3 mrad ≈ 0.2°, σ_E = 3 mrad ≈ 0.2°). Radar data sampling at 0.1 s has been considered.

3.1 Target Acceleration Estimation

In present study, V (0) = 500 m/s, $\phi_t(0)$ = 90°, $\gamma_t(0)$ = 0° , $h_t(0)$ = 10 km has been taken. The target weave frequency, $\omega_T = 2\pi/T$, where T is the time period of manoeuvre. Obviously, choice of T depends on pilot's pull up and aircraft's maximum g capability. In the present context, T =120 s has been chosen for which ω_T = 0.05 r/s. Aircraft has been assumed to be pulling 5g at the given altitude in the yaw plane or in the pitch plane throughout (Table. 1) over 50 s. The input-output pair for the FFNN training is:

$$f_1(R_m(t_k),\ A_m(t_k),\ E_m(t_k),\ R_m(t_k-1),\ A_m(t_k-1),\ E_m(t_k-1)) \rightarrow a_x$$

$$f_2(R_m(t_k),\ A_m(t_k),\ E_m(t_k),\ R_m(t_k-1),\ A_m(t_k-1),\ E_m(t_k-1)) \rightarrow a_y \qquad (9)$$

$$f_3(R_m(t_k),\ A_m(t_k),\ E_m(t_k),\ R_m(t_k-1),\ A_m(t_k-1),\ E_m(t_k-1)) \rightarrow a_z$$

Table 1. Aircraft latax variation with altitude

Altitude (km)	Instanteneous resultant η (g)	Sustained resultant η (g)
0.0	9.00	7.00
5.0	9.00	7.00
10.0	5.00	5.00
15.0	4.00	4.00
20.0	2.00	2.00

In the present context, the ANN maps input space I to output space O through functional approximation f: I $\rightarrow$ O. The primary requirement of functional approximation is that above mapping should be of one-one type. So measurements corresponding to both time t_k and time t_{k-1} have been used in input space of the training vector in Eqn. 9. Network topology consists of 6 input and 3 output of 6-30-30-3 type. Topology selection is based on optimising training time and convergence accuracy. RMS error of 1.0E–05 or 5000 epochs have been used as onvergence criteria.

Different manoeuvres have been selected by generating target trajectory through Monte Carlo simulation using $\eta_p \in [-7,7]g$, $\eta_y \in [-7,7]$ g and manoeuvre start time $\eta_t \in [0,10]s$ (Fig. 4). After training FFNN, a trained network has been used to predict the acceleration corresponding to specific manoeuvre using (R_m, A_m, E_m) measurements.

3.1.1 *Yaw Manoeuvre Case Study*

Total 10 numbers of different manoeuvres were generated and used as input data set. Training data set was generated by randomly picking up measurement samples once in ten times from the input set generated by Monte Carlo simulation. In this case, input data

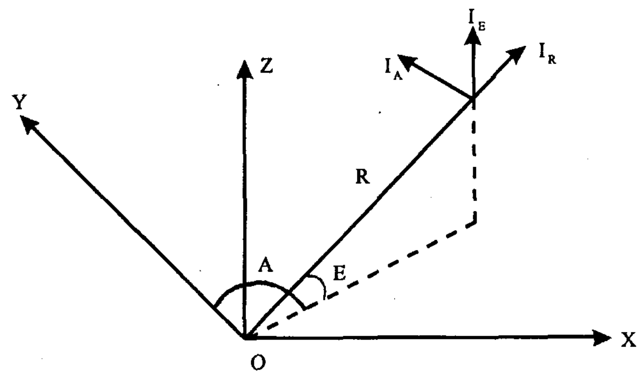

Figure 4. Schematic diagram of target wrt radar location

was generated by randomly selecting η_y from $\eta_y \in [-7,7]g$ basket. From trained network, (a_x,a_y,a_z) were estimated from (R_m,A_m,E_m) corresponding to 6 g yaw manoeuvre. In Fig. 5, the trained network output corresponding to a_x channel is compared with the actual output. The training error is also shown in the same figure. ANN-predicted acceleration $(\hat{a}_x)$ is compared with the true value (a_x) shown in Fig. 6 along with the prediction error $(a_x-\hat{a}_x)$. In a similar fashion, training history, predicted acceleration, training, and prediction error corresponding to a_y is shown in Figs 7 and 8. The training error in a_z channel will be minimum because indifferent yaw manoeuvres, a_z channel time history is invariant wrt different manoeuvres. From these figures it is clear that ANN-predicted a_x and a_y closely match true values with the minimum prediction error.

3.1.2 *Pitch Manoeuvre Case Study*

In this case, input data was generated by randomly selecting η_p from $\eta_p \in [-7,7]g$ basket. From trained network, (a_x,a_y,a_z) were estimated from (R_m,A_m,E_m) corresponding to 6g manoeuvre in the pitch plane. The comparison of estimated target acceleration (a_z) with the true value and the estimation error-time history are shown in Fig. 9.

3.1.3 *Mixed Pitch & Yaw Manoeuvre Case Study*

In this study, 15 numbers of different manoeuvres have been generated and used as input data set. The training data was generated by randomly picking up (R_m,A_m,E_m) samples once in five times from input set generated by Monte Carlo simulation. Input data is generated by randomly selecting η_y

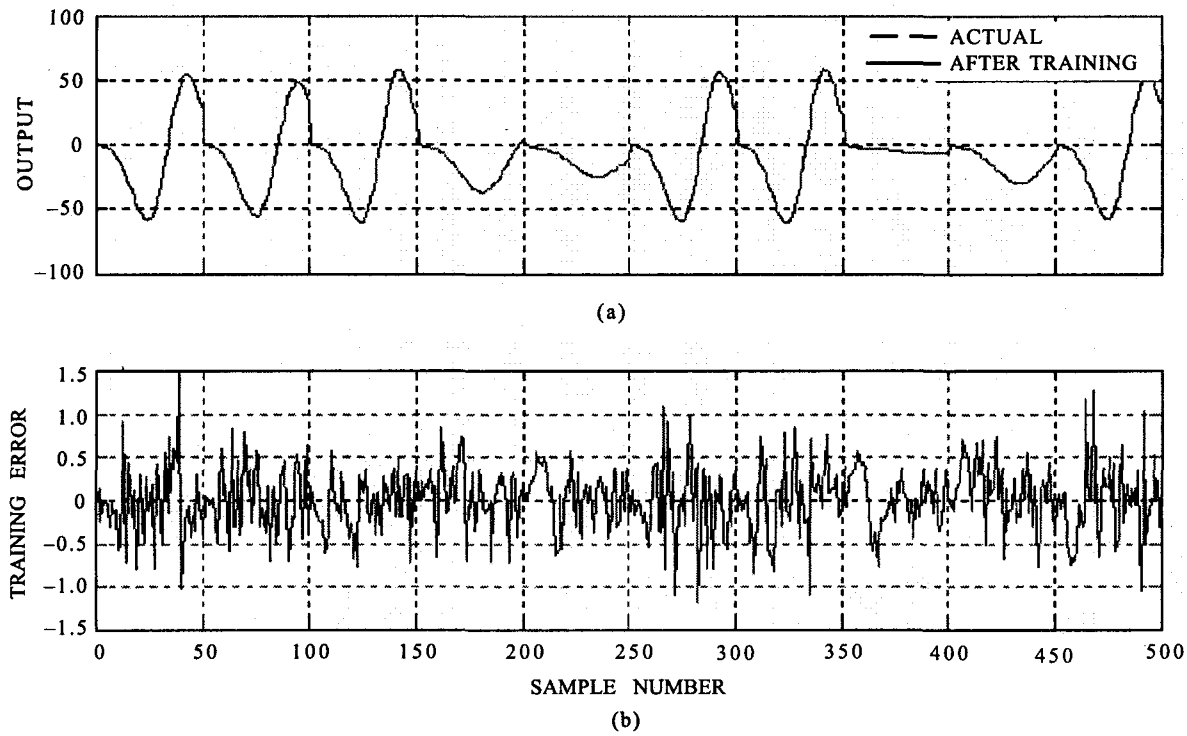

Figure 5. Training of 10 sets of data of $a_x(g)$ ($\eta_h \in [-7,7]$ g, $\omega_T = 0.05$ r/s, and yaw manoeuvre)

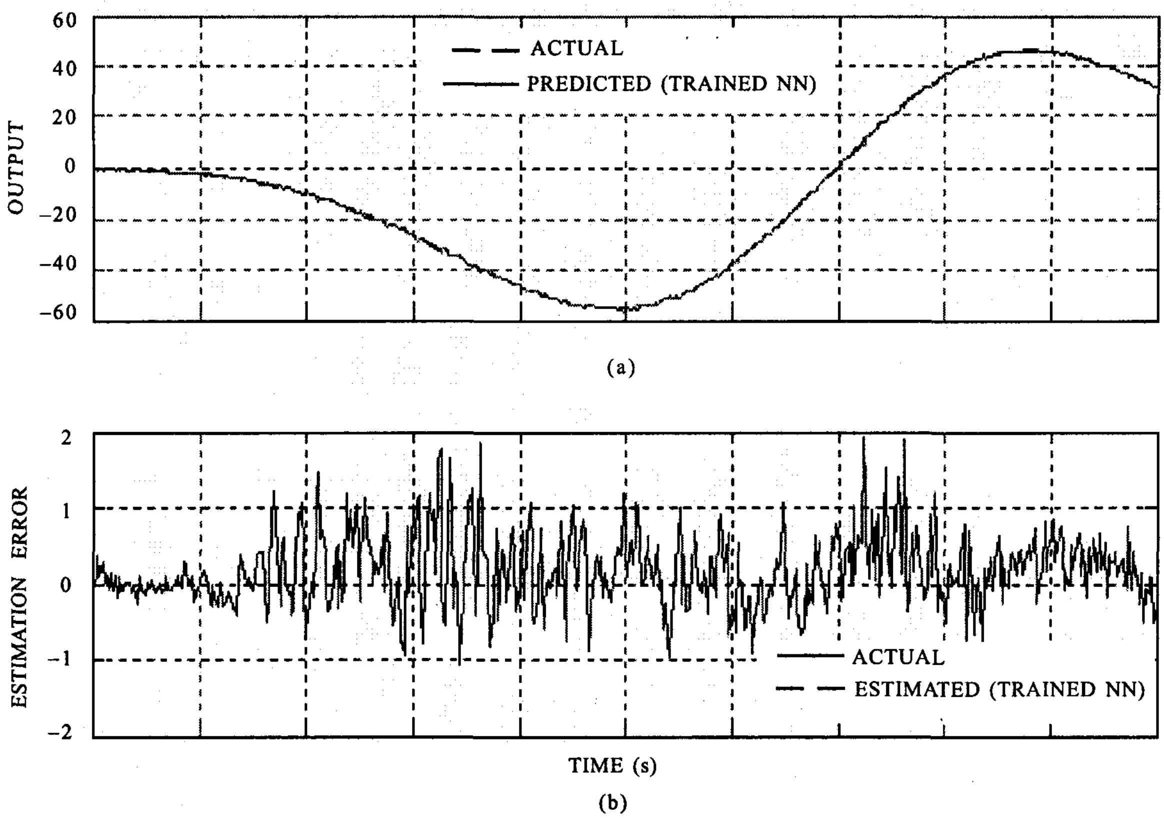

Figure 6. Time history of $(\hat{a}_x, a_x)$ (g) and estimation error ($\eta_h = 6$ g, $\omega_T = 0.05$ r/s, and yaw manoeuvre)

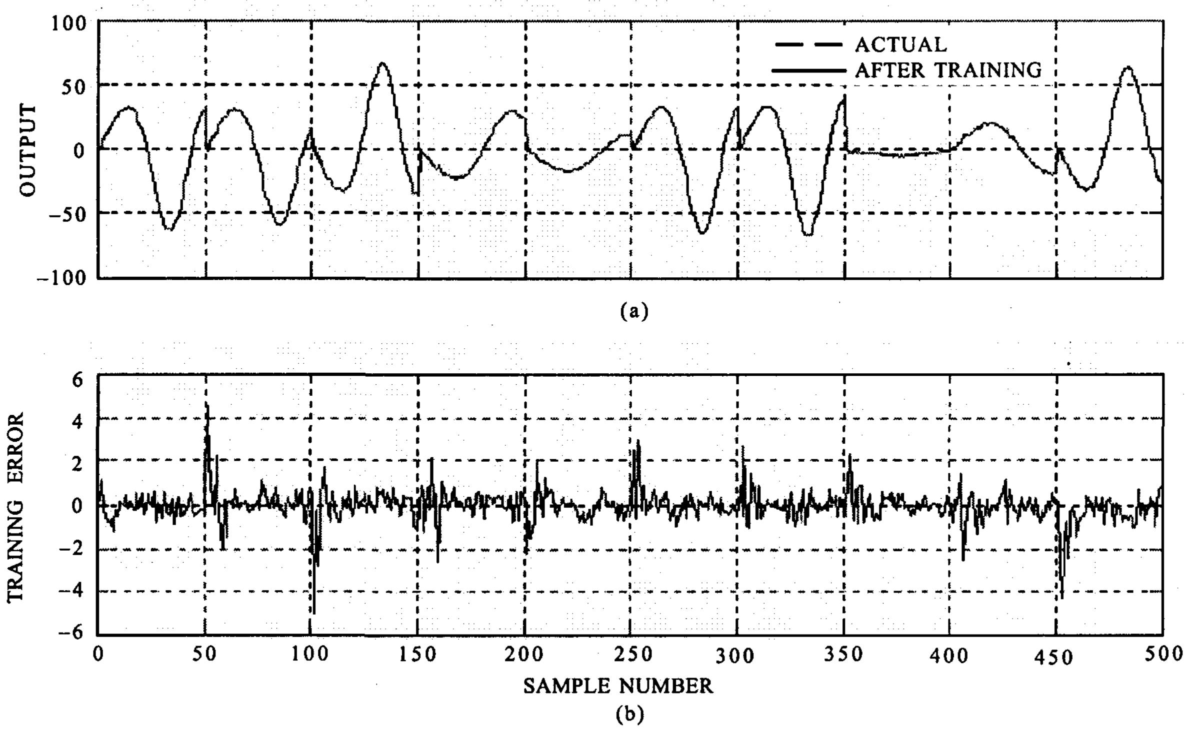

Figure 7. Training of 10 sets of data of $a_y(g)$ ($\eta_h \in [-7,7]$ g, $\omega_T = 0.05$ r/s, and yaw manoeuvre)

from $\eta_y \in [-6,6]g$ basket and η_p from $\eta_p \in [-6,6]g$. Nonlinearity in mixed manoeuvre being more than in pitch and yaw manoeuvres separately, manoeuvres generated by Monte Carlo study is more. Also, more (R_m, A_m, E_m) samples were taken for training from each manoeuvre. From the trained network,

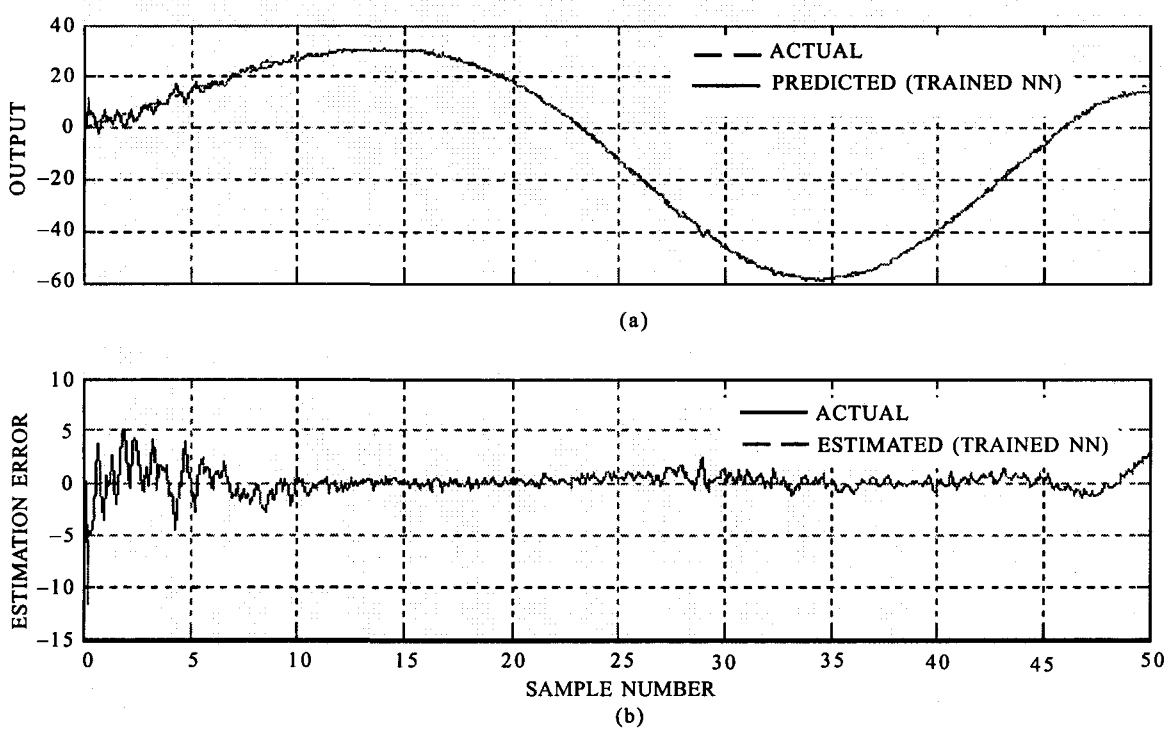

Figure 8. Time history of $(\hat{a}_y, a_y)$ (g) and estimation error ($\eta_h = 6$ g, $\omega_T = 0.05$ r/s, and yaw manoeuvre)

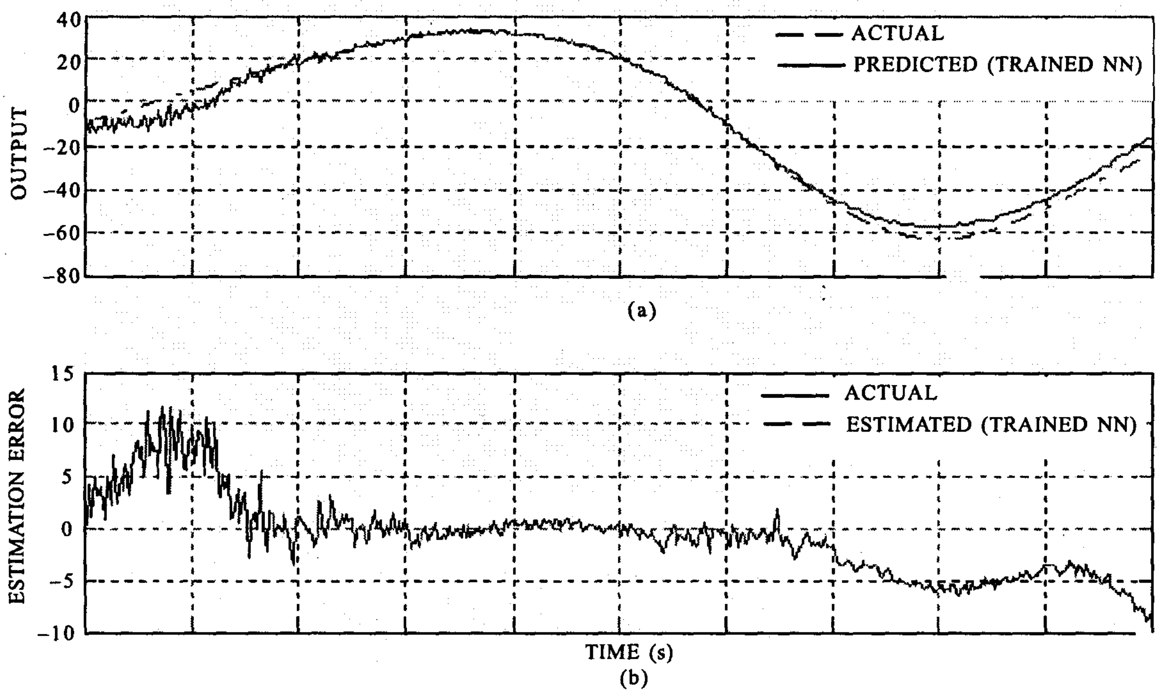

Figure 9. Time history of $(\hat{a}_z, a_z)$ (g) and estimation error (η_v = 6 g, ω_T = 0.05 r/s), $\eta_r \in$ [–7,7] g, and pitch manoeuvre)

for manoeuvre of (η_y, η_p) = (4 .9,–3.5)g, the accelerations were predicted and compared with the true values in Figs 10 to 12. From the figures it is clear that the predicted acceleration follows the true values in all channels. Specifically in a_y channel, there is a sharp discontinuity in the true value which could not be tracked by neural network. But time duration of discontinuity is very small. A more complex manoeuvre has been studied by randomly selecting manoeuvre start

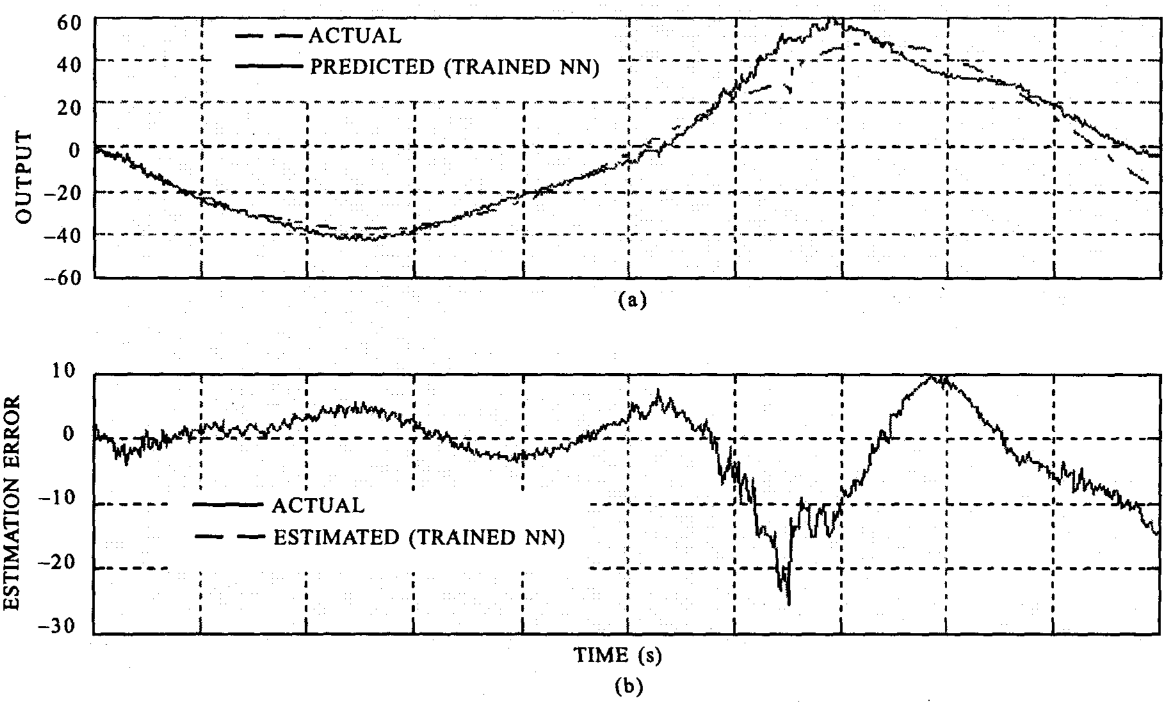

Figure 10. Time history of $(\hat{a}_x, a_x)$ (g) and estimation error (η_h = 4.9 g, η_v = –3.5 g, ω_T = 0.05 r/s, and mixed manoeuvre)

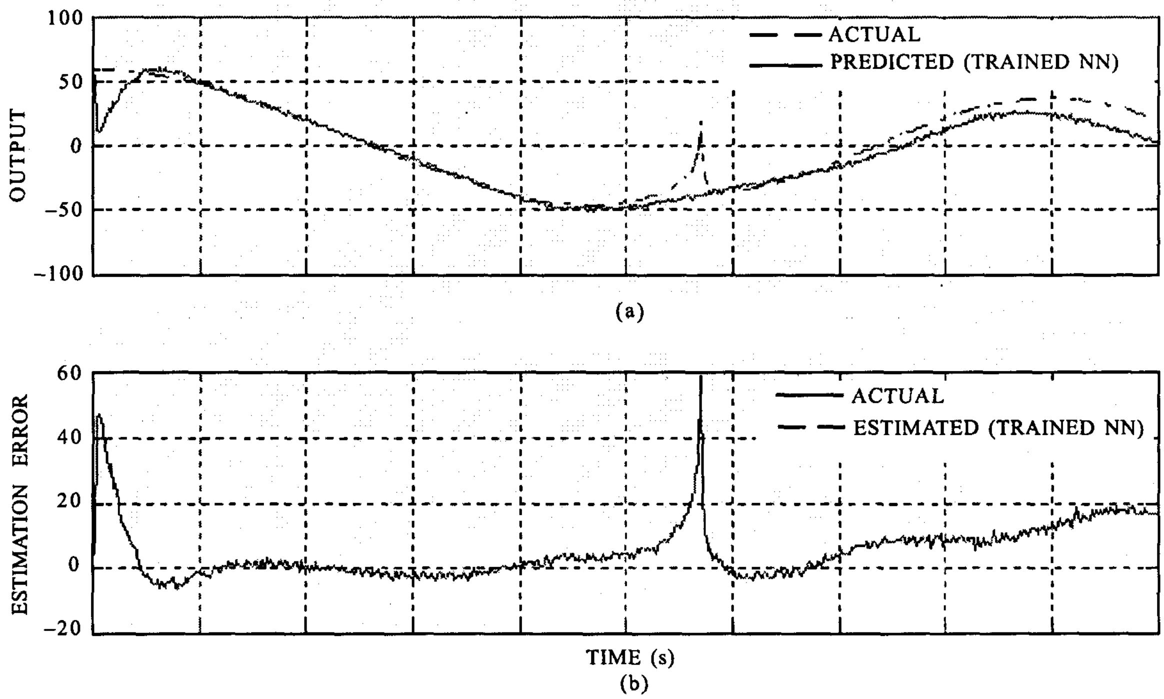

Figure 11. Time history of ($\hat{a}_y$, a_y) (g) and estimation error (η_h = 4.9 g, η_v = –3.5 g, ω_T = 0.05 r/s, and mixed manoeuvre)

time within [0,10]s along with (η_y,η_p) from the domain, as discussed above. The accelerations have been predicted from the trained network for a case study of manoeuvre start time at 5 s and (η_y,η_p) = (4.9,–3.5)g. The estimated accelerations are compared with the true values in Figs 13 to 15.

From all the figures containing estimated acceleration components, it is seen that estimates track true values grow up to 35 s. After that,

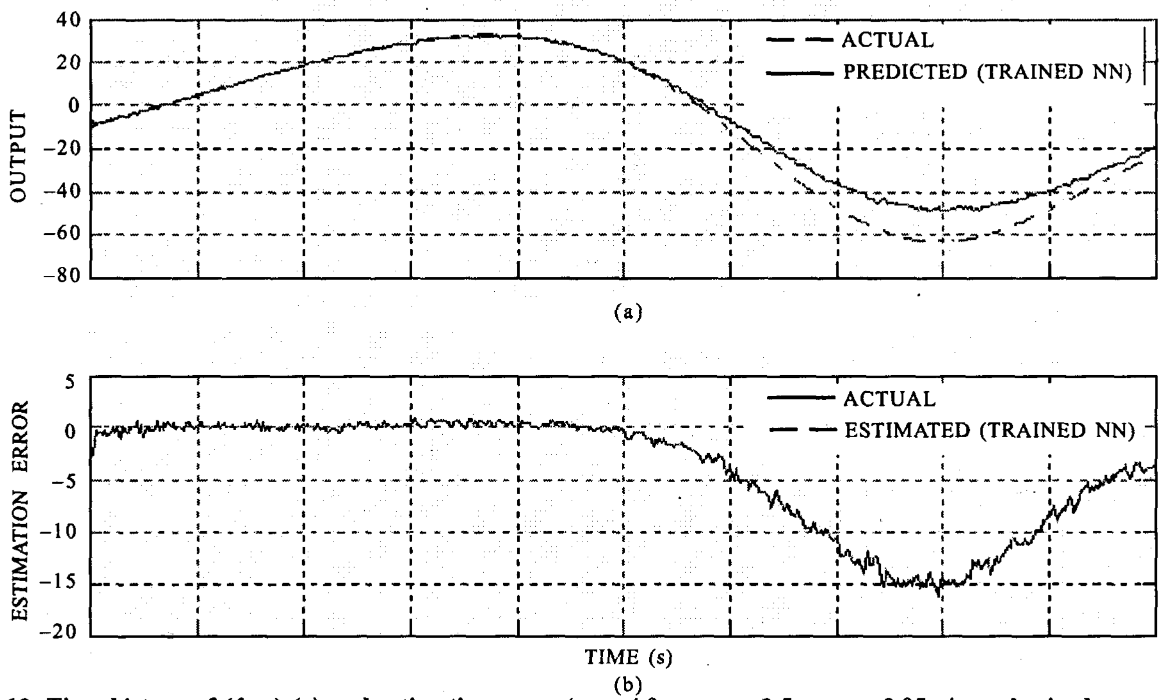

Figure 12. Time history of ($\hat{a}_z$,a_z) (g) and estimation error (η_h = 4.9 g, η_v = –3.5 g, ω_T = 0.05 r/s, and mixed manoeuvre)

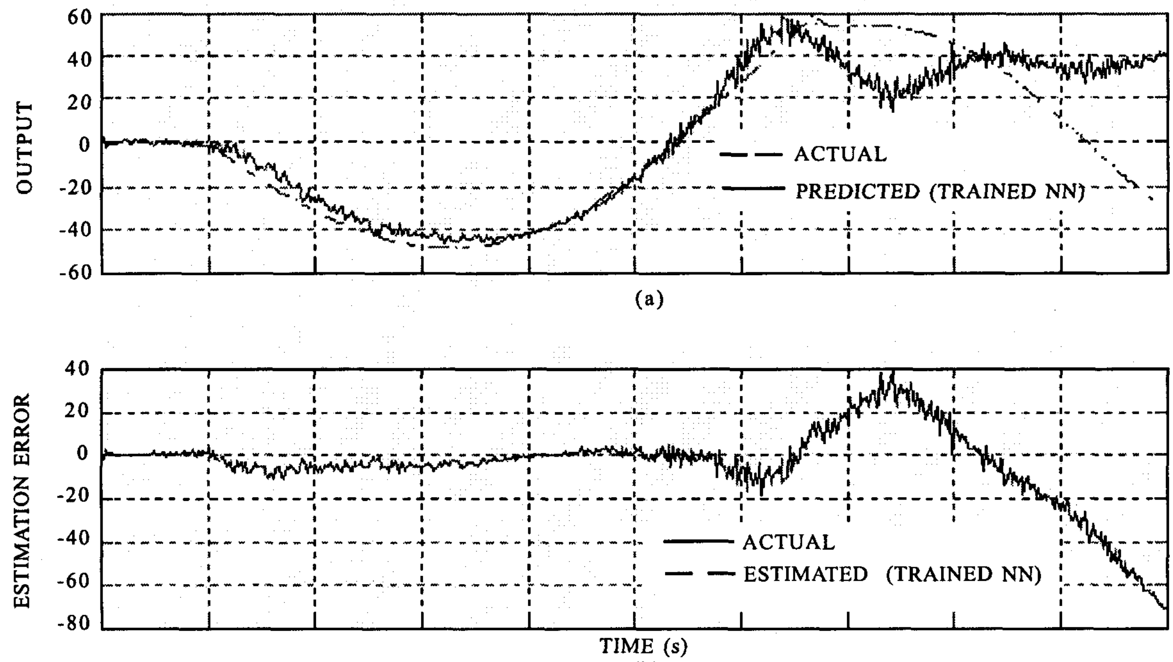

Figure 13. Time history of $(\hat{a}_x, a_x)(g)$ and estimation error (manoeuvre start time = 5 s, η_h = 4.9 g, η_v= –3.5 g, ω_T= 0.05 r/s, and mixed manoeuvre with time delay).

estimation error increases. So, remedy is to take more number of manoeuvres for network training. But in a practical scenario, 35 s is good enough for endgame.

3.2 Comparison of ANN and EKF Estimated Accelerations

In Section 3.1, estimation of target acceleration corresponding to different manoeuvres has been

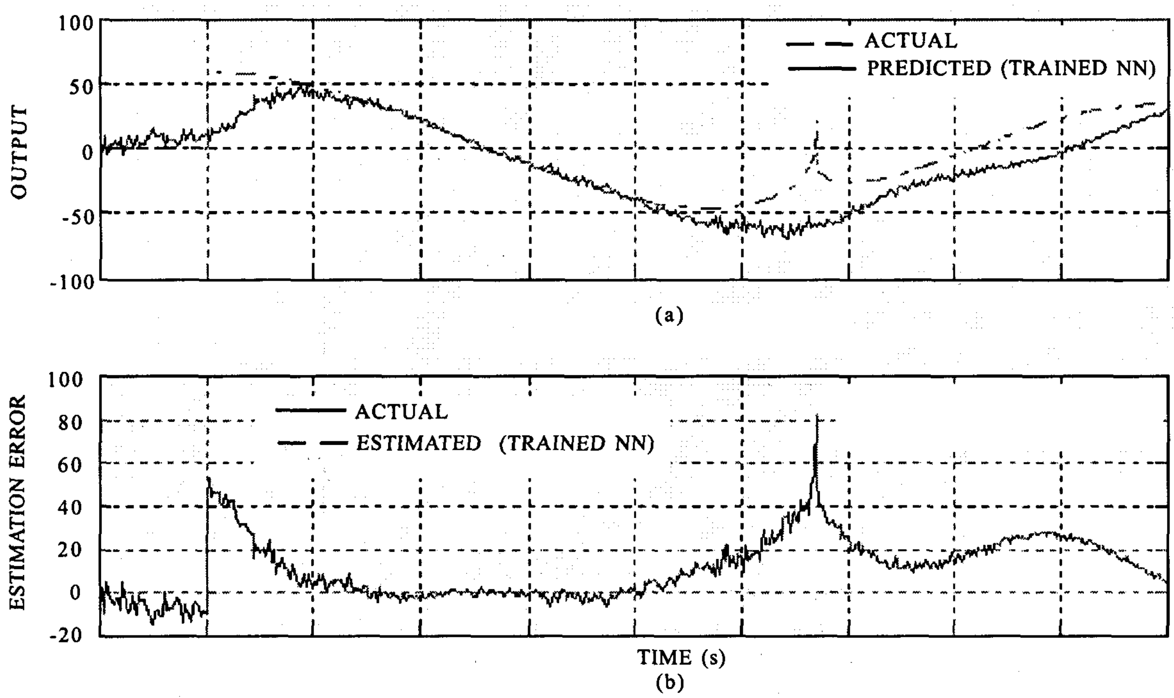

Figure 14. Time history of $(\hat{a}_y, a_y)$ (g) and estimation error (manoeuvre start time = 5 s, η_h = 4.9 g, η_v = – 3.5 g, ω_T= 0.05 r/s, and mixed manoeuvre with time delay).

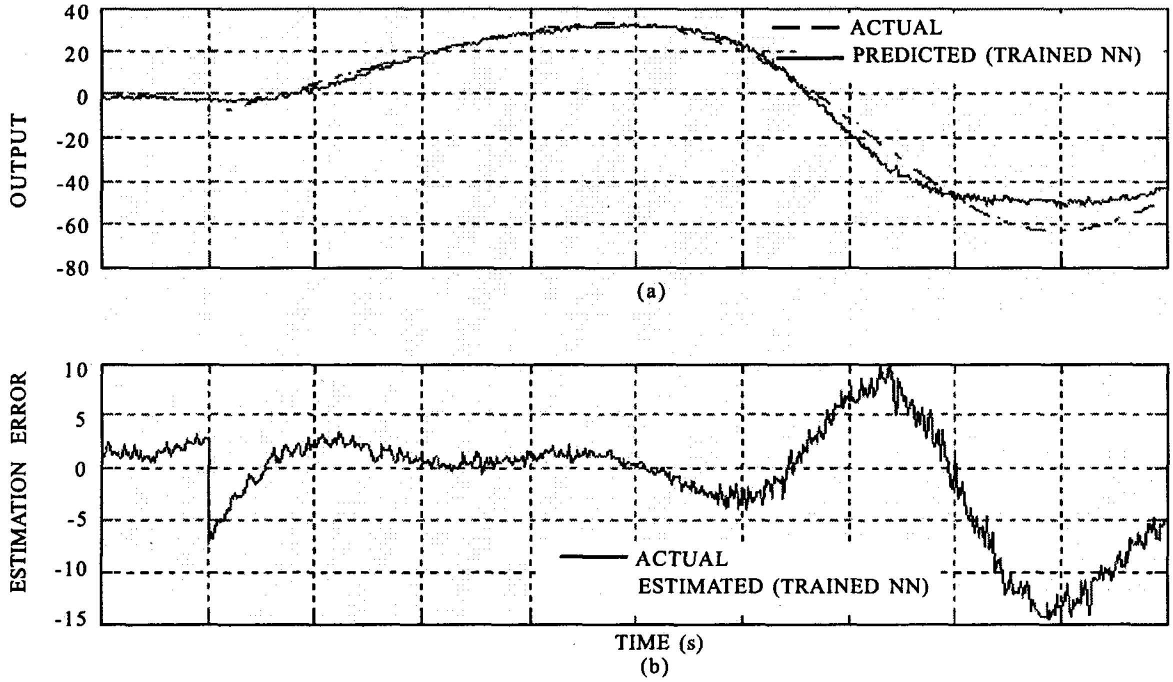

Figure 15. Time history of ($\hat{a}_z$, a_z) (g) and estimation error (manoeuvre start time = 5 s, η_h = 4.9 g, η_v = –3.5 g, ω_T = 0.05 r/s, and mixed manoeuvre with time delay).

discussed with results. Ananthasayanam[2], *et al.* have estimated target acceleration of a manoeuvreing target (Fig. 16). Now the question is:

- Which way and up to what extent ANN prediction is better than the EKF estimation in the context of target manoeuvre estimation
- How much extra price has to be paid for getting better estimate from ANN?

Ananthasayanam[2], *et al.* have estimated the target acceleration from different types of target manoeuvres using EKF (Fig. 17). The EKF-estimated acceleration errors wrt true values corresponding to (a_x,a_y,a_z) channels have been compared with the same obtained from the trained ANN shown in Figs 18 and 19, respectively. From the figures it is clear that estimation errors in acceleration estimates reduce substantially when trained ANN is used

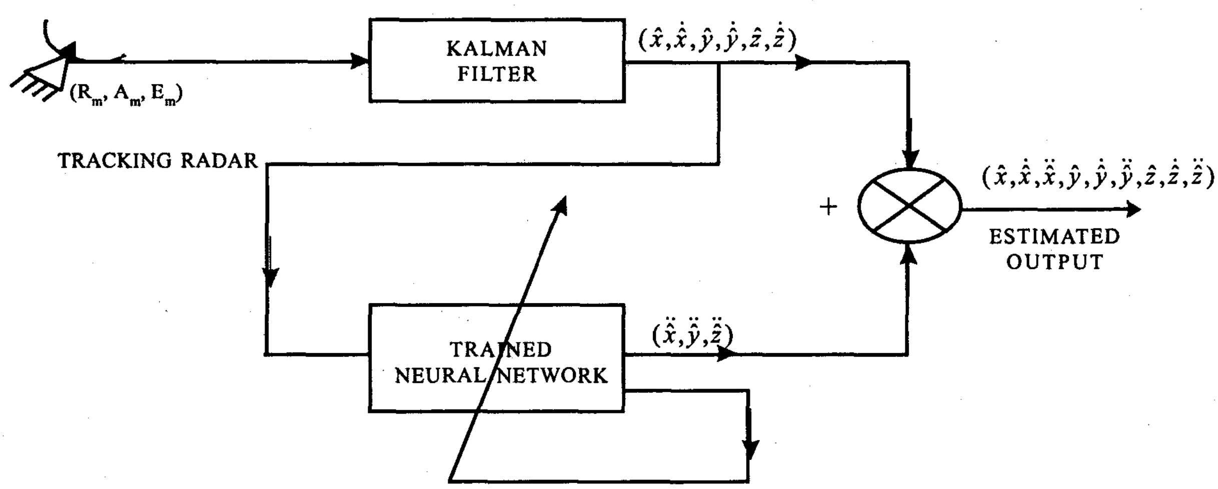

Figure 16. Proposed neural network-aided Kalman filter architecture for target acceleration estimation

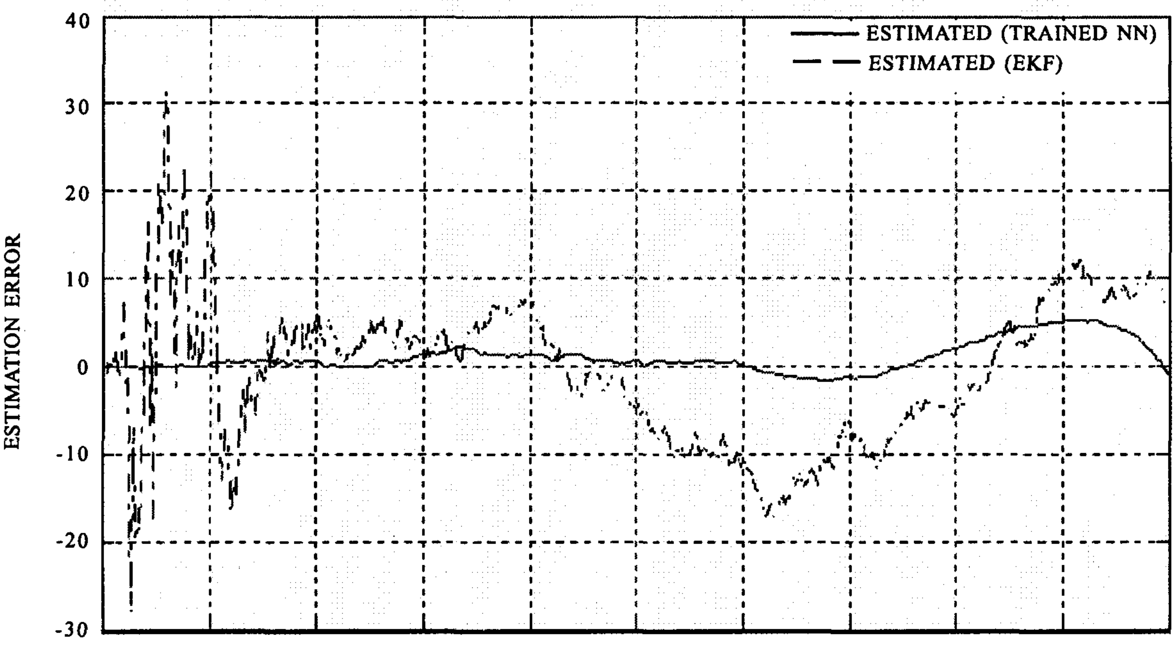

Figure 17. Comparison of a_x (g) estimation error using ANN and EKF corresponding to horizontal manoeuvre ($\eta_h = 5\,g$, $\omega_T = 0.05$ r/s)

compared to the EK route. In Fig.18, it was seen that ANN-predicted estimation error of a_y grows up at the end. This can be reduced if more data points are used while training ANN. In the present context, only 10 different manoeuvres have been used for trained neural network. It is worth to note

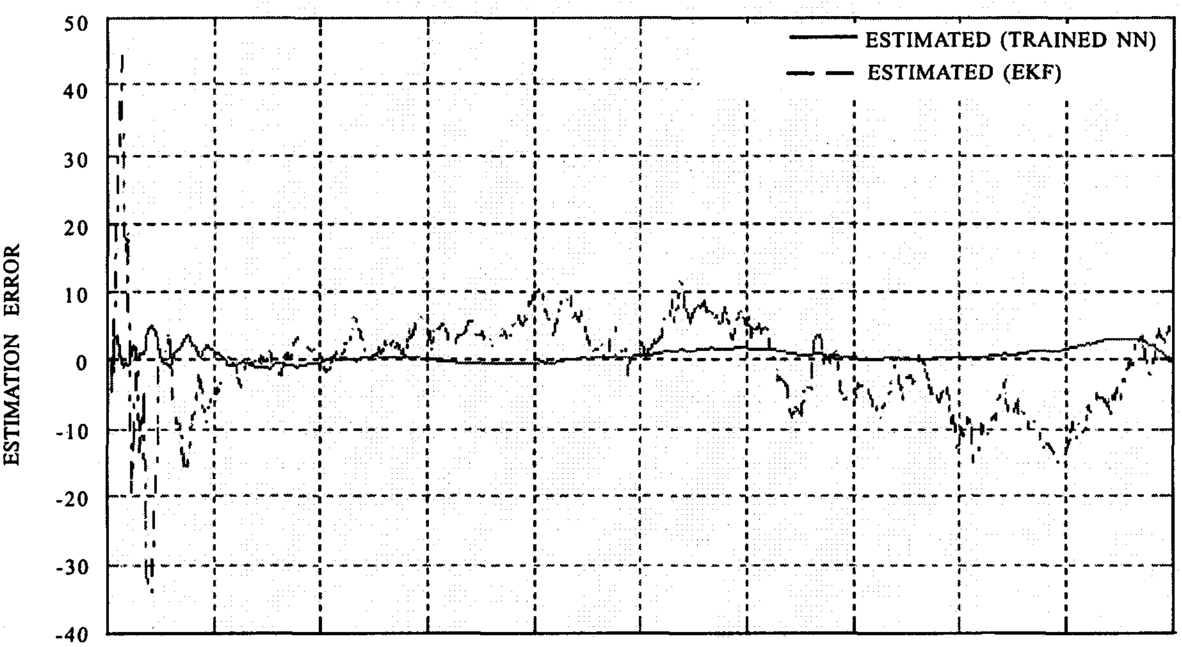

Figure 18. Comparison of a_y (g) estimation error using ANN and EKF corresponding to horizontal manoeuvre ($\eta_h = 5\,g$, $\omega_T = 0.05$ r/s)

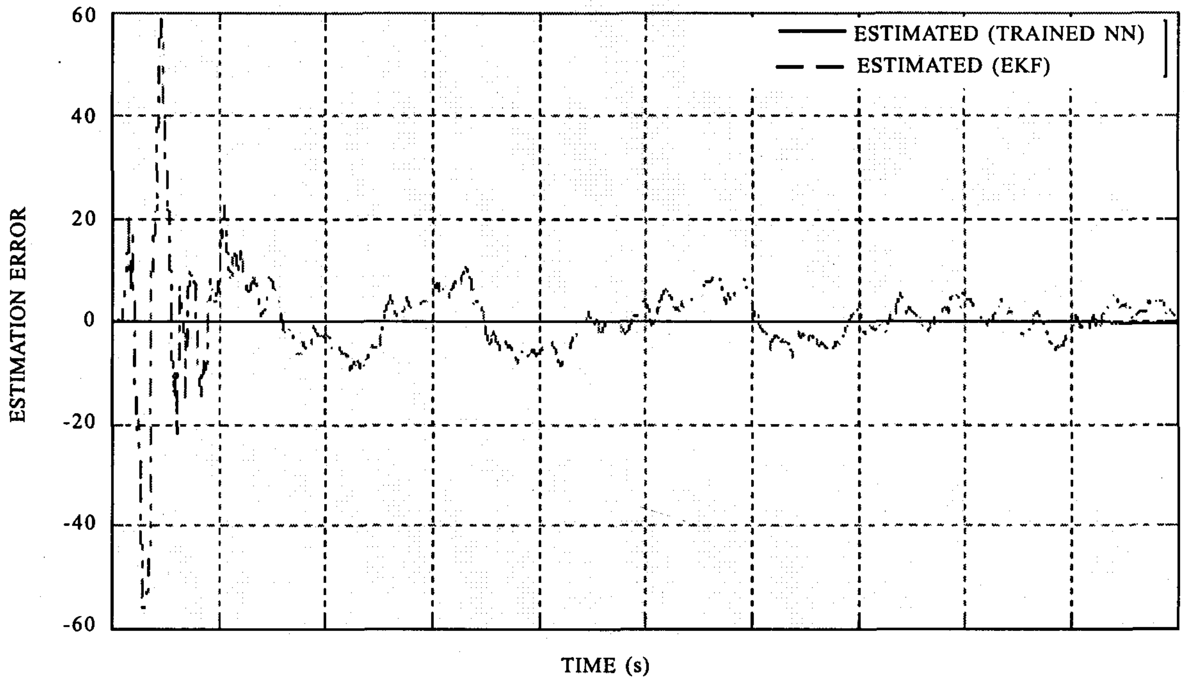

Figure 19. Comparison of $a_z(g)$ estimation error using ANN and EKF corresponding to horizontal manoeuvre ($\eta_h = 5\,g$, $\omega_T = 0.05$ r/s)

that ANN-predicted a_z error is zero, whereas the same using EK estimator is quiet high (Fig.19). This is because of the fact that for different yaw manoeuvres, a_z is invariant and while training ANN learns it and reproduces the same value in the context of a specific manoeuvre. But while processing the data using EKF, there has to be some estimation error based on the statistical uncertainty of the measurement signal.

The next question to be answered is about the extra price to be paid for getting more accurate estimate using ANN. Performance of ANN depends on the choice of input-output pair and selection of train ing points. So, a judicious selection of training pairs to take care of variation of different parameters in the system dynamics is mandatory. This depends mainly on understanding the physics of the problem. Also, as the offline training process takes substantial amount of time, a fast and best training algorithm from functional approximation point of view, has to be chosen.

Based in the present and earlier studies, a combined EK and ANN-based estimator seems to be feasible for target manoeuvre estimation from tracking radar measurements of (R_m, A_m, E_m) in real-time. Using EKF, the position (x,y,z) and velocity components $(\dot{x},\dot{y},\dot{z})$ can be estimated with good accuracy from (R_m, A_m, E_m). Now the acceleration components $(\ddot{x},\ddot{y},\ddot{z})$ from the trained neural network can be estimated from $(\hat{R},\hat{A},\hat{E})$ obtained via EKF estimated $(\hat{x},\hat{y},\hat{z})$. This removes the oscillations in the network-estimated acceleration components as seen in previous figures. The proposed estimator architecture is shown in Fig.16.

4. FUTURE ACTIVITIES & CONCLUSION

The present study is basically an investigation of using ANN to estimate the target acceleration from radar position measurements. Though the total study is of preliminary in nature, the results are very encouraging and throws a new avenue for further practical implementable research in this domain which are as follows:

- In a practical situation, target manoeuvre is of random nature. So, the mathematical modelling of generation of tracking radar data from random target manoeuvre is mandatory. So, if target flight-path angles (γ,ϕ), weave frequency ω_T, target velocity, altitude and latax levels are varied in Monte Carlo simulation, outcome should be a random target manoeuvre. For the selection of lower and upper bounds on Monte Carlo input, interactions with the test pilots are required.

- Learning algorithm used in the present context is the BP algorithm. Use of radial basis network[6,7] (RBN), cerebellar model neural network[8,9] (CMNN) for enhanced training speed can be explored also.

If input space data from a large number of manoeuvres are generated, physical intuition indicates that many of them can be clustered in different groups and rule-based learning should simplify the total learning process. Possibility of study of this aspect while neural network learning, has also to be explored.

ACKNOWLEDGEMENTS

The authors thank Prof V. Mani of Aerospace Engineering Dept, Indian Institute of Science, Bangalore Sarvashri N.V. Kadam and N. Prabhakar, Scientists of DRDL, Hyderabad, for discussions throughout this study.

REFERENCES

1. Zarchan. Tactical and strategic missile guidance. Progress in Aeronautics and Astronautics, Ed. 3. 1997. **176**.

2. Ananthasayanam, M.R.; Sarkar, A.K. & Vathsal, S. Estimation of rapid target acceleration in real time from position alone measurements of tracking radar. *In* Proceedings of National Systems Conference (NSC), IIT Kharagpur, December 2003. pp. 64-69.

3. Chin, L. Application of neural network in target tracking data fusion. *IEEE Trans. Aerospace Electr. Syst.*, 1994, **AES-30**(1), 281-87.

4. Vaidehi, V.; Chitra, N.;. Chokkalingam, N. & Krishnan, C.N. Neural network-aided Kalman filtering for multi-target tracking applications. *In* Computers and electrical engineering. Elsevier Series, 2001, **27**. pp. 217-28.

5. Cichocki, A. & Unbehauen, R.*In* Neural network for optimisation and signal processing. John Willey & Sons, 1993. pp. 122-52.

6. Chen, S.; Cowan, C.F.N. & Grant, P.M. Orthogonal least square algorithm for radial basis function network. *IEEE Trans. Neural Network*, 1991, **2**(5), 302-08.

7. Yingwei, L.; Sundarrajan, N. & Saratchandran, P. Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. *IEEE Trans.Neural Network*, 1998, **9**(2), 308-18.

8. Govindarao, V.M.H. & Sarkar, A.K. Aerodynamic modelling of a flight vehicle using cerebellar model neural network. *In* Proceedings of the Seventh Asian Congress in Fluid Mechanics, IIT Madras, Chennai, 1997. pp. 841-44.

9. Brown, M. & Harris, C. *In* Neurofuzzy adaptive modelling and control. Prentice Hall International (UK) Ltd, 1994. pp. 409-39.

Contributors

Mr A.K. Sarkar graduated in Aeronautical Engg from the Indian Institute of Technology (IIT), Kharagpur, during 1975-80 and did his postgraduation in Computer Science from the Indian Institute of Technology (IIT) Madras, Chennai during 1982-84. He has been working as Scientist at the Defence Research and Development Laboratory (DRDL), Hyderabad since 1984. His areas of research include: Flight mechanics, state and parameter estimation for offline and real-time applications, guidance, and control.

Dr S. Vathsal obtained his BE (Electrical Engg) from the Thaigaraja College of Engineering, Chennai, in 1968, ME (Control System) from the Birla Institute of Technology and Sciences, Pilani, in 1970, and PhD from the Indian Institute of Science (IISc), Bangalore, in 1974. His areas of research include: Control systems estimation, guidance, and neural networks.

Mr Suresh Sundaram obtained his ME (Aerospace Engg) from the Indian Institute of Science(IISc), Bangalore. Presently, he is a research scholar in the Dept of Aerospace Engineering at the IISc, Bangalore. His areas of research include: Control systems and neural networks.