Sensitivity Analysis Using Neural Network for Estimating Aircraft Stability and Control Derivatives

Rohit Garhwal[a], Abhishek Halder[b] and Dr. Manoranjan Sinha[c]

Department of Aerospace Engineering, Indian Institute of Technology, Kharagpur – 721 302, INDIA
e-mail: [a]ur.rohit2003@gmail.com; [b] halder.abhishek@gmail.com; [c]masinha@aero.iitkgp.ernet.in

***Abstract*-This paper presents a method for aircraft parameter estimation using neural sensitivity analysis. The results are found to be superior to other ANN based methods.**

I. INTRODUCTION

The stability and control derivatives of an aircraft are of engineering significance for making flight simulator database, designing control law, expanding flight envelope and meeting airworthiness requirements. There are two broad approaches to estimate stability and control derivatives from flight data: direct approach and indirect approach. Widely applied parameter estimation techniques like output error method, filter error method and equation error method belong to the first category. In indirect approach, a non-linear filter is used to estimate the parameters which are defined as artificial state variables [1]. Furthermore, some methods can work in real time (on-line parameter estimation) and some deal with post-flight data (off-line parameter estimation). However, these conventional methods need knowledge about initial values of the parameters (often supplied by computational fluid dynamics (CFD) simulations and wind tunnel tests). Poor initialization may result in divergence of the algorithm. Additionally, conventional methods require a priori knowledge of the aerodynamic model (model identification problem). This poses a severe limitation to the applicability of these methods for complex aerodynamic situations like stall hysteresis, large-amplitude time-dependent maneuvers and high angle of attack flight where coming up with a realistic non-linear aerodynamic model is not so obvious [2].

The ability of artificial neural network (ANN) to act as universal function approximator offers significant advantages for aircraft parameter estimation problem. It can capture highly non-linear complex phenomena in global sense without a priori knowledge about the dynamic model. Also initial estimates of the parameters are not necessary. These two qualities render ANN a natural choice for estimating stability and control derivatives from flight data. An early application of ANN for aircraft parameter estimation can be found in [3] where motion and control variables were mapped to aerodynamic forces and moments using back propagation algorithm. Various aspects of feed forward neural network for identification of aerodynamic coefficients were discussed in detail in [4], [5] and [6]. Raisinghani, Ghosh and Kalra [7] proposed two techniques, called delta method and zero method, for computing the stability and control derivatives using feed forward neural network with back propagation learning. In the first technique, the derivatives were interpreted as variations in the aerodynamic co-efficients due to a small (delta) variation in one of the motion or control variables in such a fashion that only that variable undergoes incremental change while the rest remain at their nominal values (hence the name delta method). In the second technique, the derivatives were interpreted as the ratio of variation in the value of the aerodynamic co-efficients to the incremental variation in one of the motion or control variables while the rest of them remain identically zero (hence the name zero method). Both the techniques employ central difference approximation for computing the derivatives.

TABLE I
PARTICULARS OF THE MULTI-LAYER FEED FORWARD NEURAL NETWORK

No.	Attribute	Best Choice
1	No. of Hidden Layers	2
2	No. of Neurons (Hidden1 – Hidden2 – Output)	5 – 13 – 3
3	Activation Function (Hidden1 – Hidden2 – Output)	tanh – tanh – Identity
4	Scaling Range (Input & Output)	- 0.9 to + 0.9
5	Initial Random Weights	- 0.1 to + 0.1

In this paper, a novel neuro-computing approach is presented to address the problem of aircraft parameter estimation by means of sensitivity analysis of the aerodynamic forces and moments with respect to motion variables and control inputs. The next section outlines the methodology. Simulation results are presented next, followed by conclusion.

II. METHODOLOGY

The rationale behind applying sensitivity analysis for estimating stability and control derivatives lies in interpreting them as the sensitivity of the aerodynamic forces and moments on the motion and control variables. The motion and control variables are taken as the input to the ANN and the force and moment co-efficients are taken as the output of the network. A high (low) value of a parameter (derivative) implies that output variable of the ANN is highly (weakly) sensitive to the corresponding input variable.

A multi-layer feed forward neural network (MFNN) with two hidden layers was used in this study for input-output mapping. The particulars of the network are summarized in Table I. The sensitivity analysis of the ANN is discussed next.

1-4244-1355-9/07/$25.00 @2007 IEEE

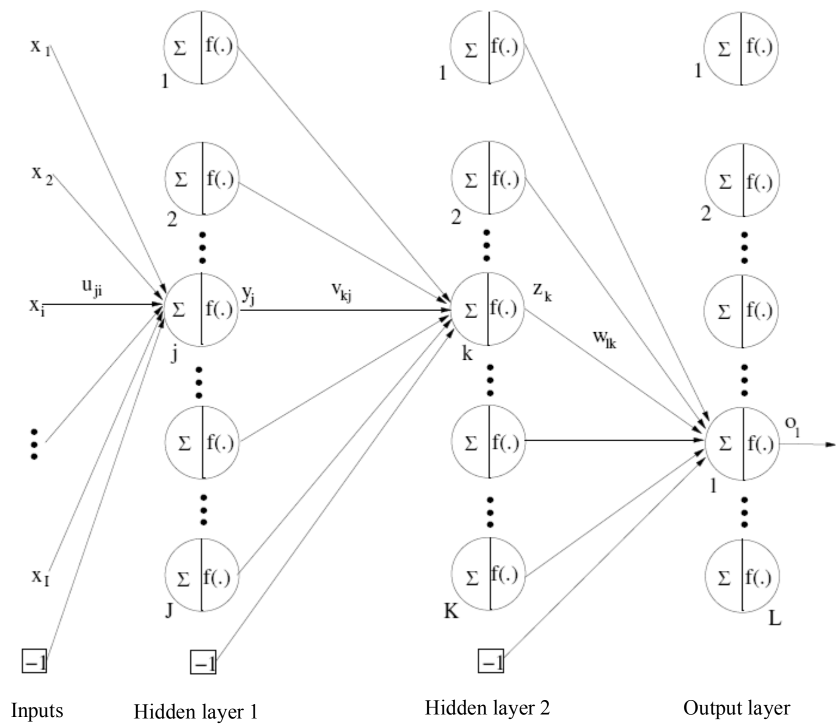

Fig.1. Neural network architecture

The Neural network model used consists of two hidden layers and one output layer as shown in figure 1. The input to the network is designated as $\tilde{x}$, which is a vector of size $I+1$ including the bias. The inputs are fed to the first hidden layer; output of the first hidden layer is fed to the second hidden layer; and output of the second hidden layer is fed to the output layer which consists of linear neurons i.e., just summation functions. The hidden layers neurons consist of tangent hyperbolic function as a neuronal nonlinearity. The outputs of the neurons for various layers are designated as $\tilde{y}, \tilde{z}, \tilde{o}$, as shown in the figure. There are J number of neurons in the first hidden layer, K number of neurons in the second hidden layer, and L number of neurons in the output layer. Therefore, the following equations can be written for the outputs of the first hidden layer, second hidden layer and the output layer of neurons.

For the first hidden layer,

$$y_j = f\left(\sum_{i=1}^{I+1} x_i u_{ji}\right) = f\left(net_j\right) \tag{1}$$

where, x_i is the i^{th} input; u_{ji} is the weight connecting i^{th} input to the j^{th} neuron in the first hidden layer; $f(.)$ is the tangent hyperbolic function and is defined as

$$net_j = \sum_{i=1}^{I+1} x_i u_{ji} \tag{2}$$

$$f(x) = \frac{1-e^{-\lambda x}}{1+e^{-\lambda x}}$$; λ being the steepness factor.

Similarly, the output of the second hidden layer can be written as

$$z_k = f\left(\sum_{j=1}^{J+1} y_j v_{kj}\right) = f\left(net_k\right) \tag{3}$$

where all the notations and subscripts follow the same connotation as explained above. The output layer output can be written as

$$o_l = \sum_{k=1}^{K+1} z_k w_{lk} = net_l \tag{4}$$

where the subscript l varies from 1 to L. Method of back propagation with scaled conjugate learning is used to train the neural network. Once the network is trained then it represents the function which is contained in the input-output data, i.e. network has identified the function embedded in the data the the partial derivatives of the outputs with respect to the inputs represent the sensitivity and can be derived as follows.

$$\frac{\partial o_l}{\partial x_i} = f'(net_l) \times \frac{\partial(net_l)}{\partial x_i} \tag{5}$$

since there is no nonlinearity in the output layer of neurons, therefore $f'(net_l) = 1$, where

$$\frac{\partial(net_l)}{\partial x_i} = \sum_{k=1}^{K} \frac{\partial z_k}{\partial x_i} w_{lk} \quad (6)$$

$$\frac{\partial z_k}{\partial x_i} = f'(net_k) \frac{\partial(net_k)}{\partial x_i} \quad (7)$$

$$\frac{\partial(net_k)}{\partial x_i} = \sum_{j=1}^{J} \frac{\partial y_j}{\partial x_i} v_{kj} \quad (8)$$

and

$$\frac{\partial y_j}{\partial x_i} = f'(net_j) \frac{\partial(net_j)}{\partial(x_i)} \quad (9)$$

$$\frac{\partial(net_j)}{\partial(x_i)} = u_{ji} \quad (10)$$

The input data are normalized before use and therefore, equation for the sensitivity can be written as

$$\frac{\partial o_l}{\partial x_i} = \frac{\partial o_l}{\partial o_{nl}} \frac{\partial o_{nl}}{\partial x_{ni}} \frac{\partial x_{ni}}{\partial x_i} \quad (11)$$

where the subscript n refers to the normalized value. The following equation is utilized to normalize the input and the output

$$o_{nl} = o_{l\min} + diff \times \frac{o_{l\max} - o_l}{o_{l\max} - o_{l\min}} \quad (12)$$

similarly

$$x_{ni} = x_{i\min} + diff \times \frac{x_{i\max} - x_i}{x_{i\max} - x_{i\min}} \quad (13)$$

Here, o_{nl} , x_{ni} are the normalized l^{th} output and i^{th} input respectively. $o_{l\max}$ and $o_{l\min}$ stands for the minimum and the maximum value of the l^{th} output. Similar connotation stands for $x_{i\max}$ and $x_{i\min}$ also. $diff$ stands for the range of normalization used. In this case because the normalization was done in the range $[-0.9, 0.9]$ hence the value of $diff$ is (1.8). Therefore,

$$\frac{\partial x_{ni}}{\partial x_i} = -diff \times \frac{1}{x_{i\max} - x_{i\min}} \quad (14)$$

$$\frac{\partial o_{nl}}{\partial o_l} = -diff \times \frac{1}{o_{l\max} - o_{l\min}} \quad (15)$$

An investigation was done to reach the optimal combination for the number of neurons in each hidden layer. Tangent hyperbolic functions were used as activation functions for each of the neurons. A scaled conjugate gradient algorithm was used for fast supervised learning [8].

The flight data used to train the network was for lateral-directional dynamics of Advanced Technology Testing Aircraft System (ATTAS) from DLR, German Aerospace Center, Germany. The data was of total 85 seconds duration with the sampling time of 0.04 second. There were three distinct maneuvers performed by the pilot: short period mode (for first 25 seconds, using multi-step elevator input), bank-to-bank maneuver (for next 30 seconds, using aileron input) and dutch roll (for last 30 seconds, using rudder doublet input). Once trained, the MFNN was subjected to testing data set, which was taken to be the entire length (2128 samples) of training data set. Simulations were performed for varying number of iterations to assess the accuracy of mapping. The measured (flight derived) force and moment co-efficients were then compared with ANN predictions.

After predicting the aerodynamic co-efficients, the network was used to estimate the stability and control derivatives using sensitivity analysis [9]. Unlike delta method and zero method, neural sensitivity analysis computes the derivatives without applying finite difference approximation. Since partial derivatives are point functions, the proposed method yields mathematically more accurate estimates. The superiority of the derivatives, estimated by the method proposed here, is evident from the results presented in the next section.

III. Results And Observations

Numerous simulations revealed some important aspects of the MFNN. It was observed that reducing the number of neurons in the hidden layer lowers the order of the prediction curve resulting poor estimation of the aerodynamic co-efficients. It was also noted that using a tangent hyperbolic function in the output layer degrades the prediction. Hence only weighted sum was performed at the output layer.

To make a direct comparison, the same flight data was used to train and test delta method (which gives better estimates than zero method). Table II clearly shows that, smaller mean square error (MSE) and cost function value is achieved in less number of iterations by scaled conjugate gradient algorithm used in this study. Beyond 5000 iterations the MFNN shows no significant improvement in predictions.

TABLE II
MSE And Cost Function

Number of Iterations	MSE		Cost Function	
	Delta Method	Sensitivity Analysis	Delta Method	Sensitivity Analysis
Starting Value	5.27e-3	6.71e-5	3.00e-7	2.48e-13
500	6.29e-4	4.49e-6	5.00e-13	1.90e-19
5000	8.50e-5	3.40e-7	4.70e-13	1.02e-19
15,000	4.50e-5	-	3.80e-13	-

In Fig. 2, 3 and 4, the measured (blue) and ANN predicted (red) aerodynamic co-efficients (side force co-efficient, rolling moment co-efficient and yawing moment co-efficient respectively) are plotted with time. These figures show much superior tracking of the aerodynamic co-efficients compared to results obtained by employing output error method, filter error method, equation error method and delta method on the same flight data.

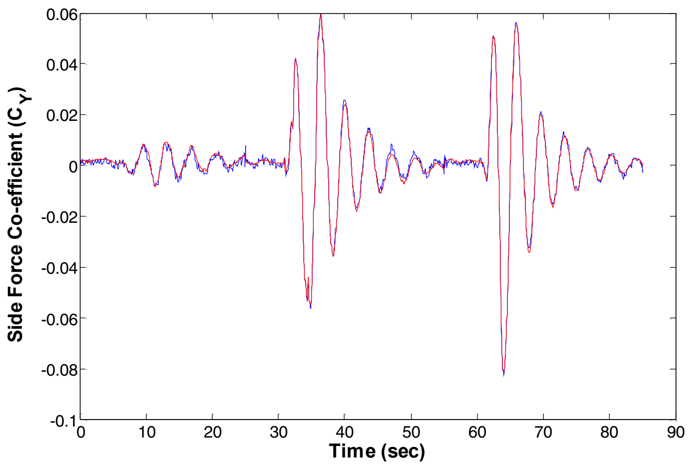

Fig. 2. Measured (blue) and ANN predicted (red) side force co-efficient

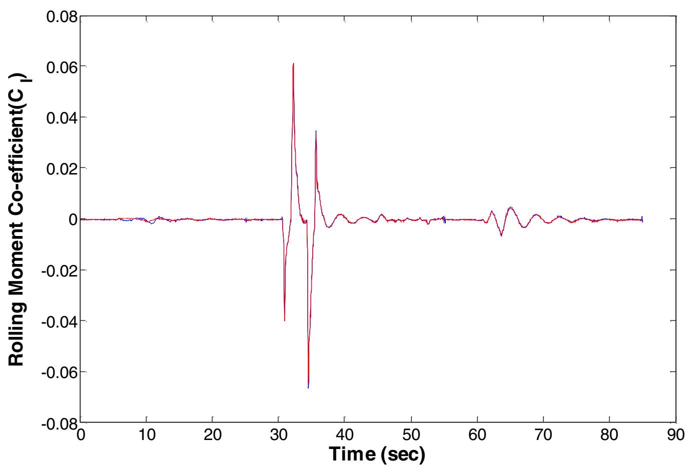

Fig. 3. Measured (blue) and ANN predicted (red) rolling moment co-efficient

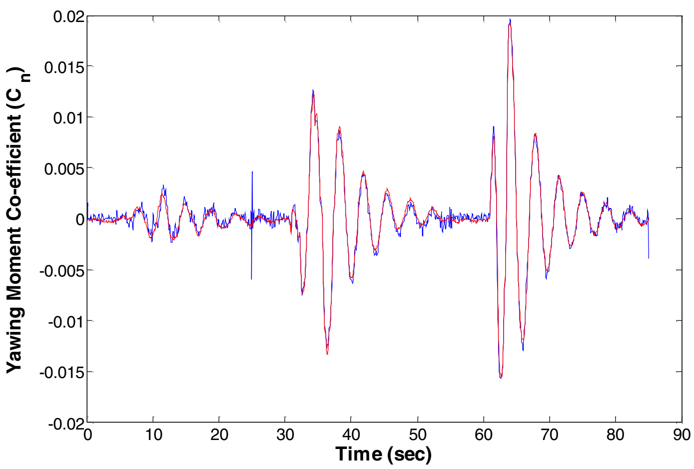

Fig. 4. Measured (blue) and ANN predicted (red) yawing moment co-efficient

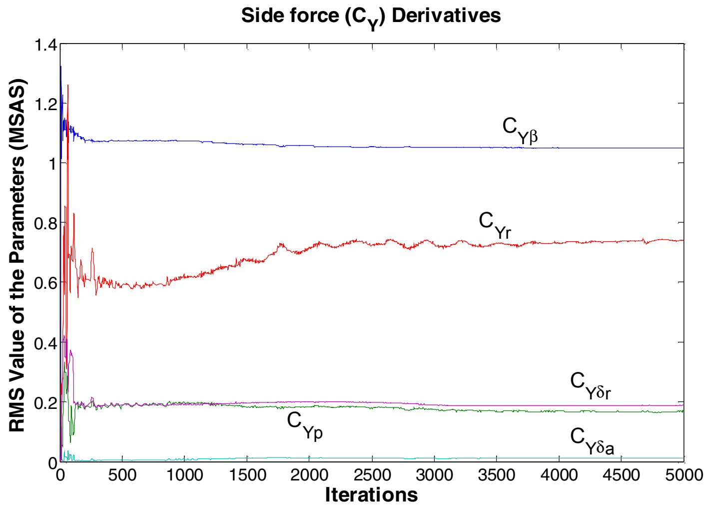

Fig.5. RMS values of side force derivatives over the entire training set

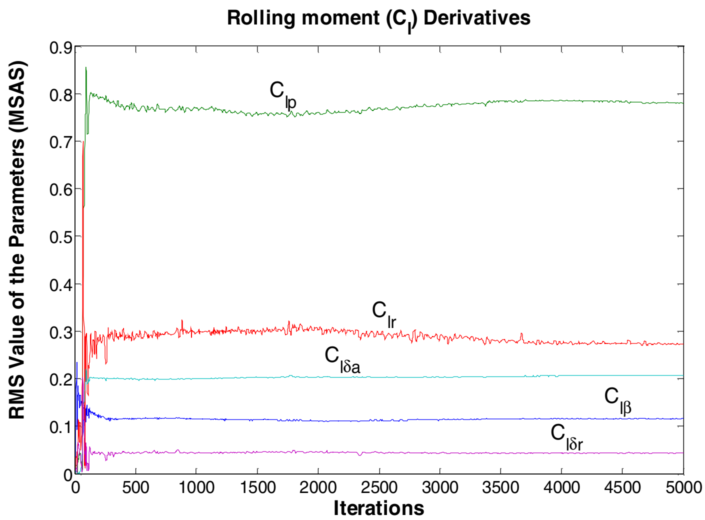

Fig.6. RMS values of rolling moment derivatives over the entire training set

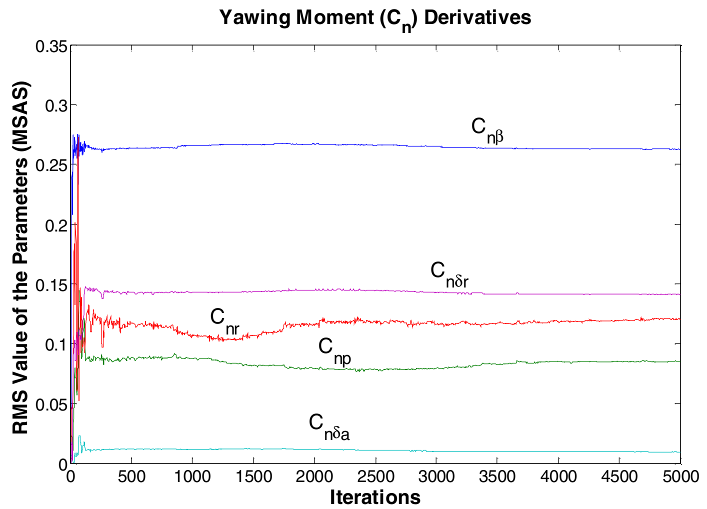

Fig.7. RMS values of yawing moment derivatives over the entire training set

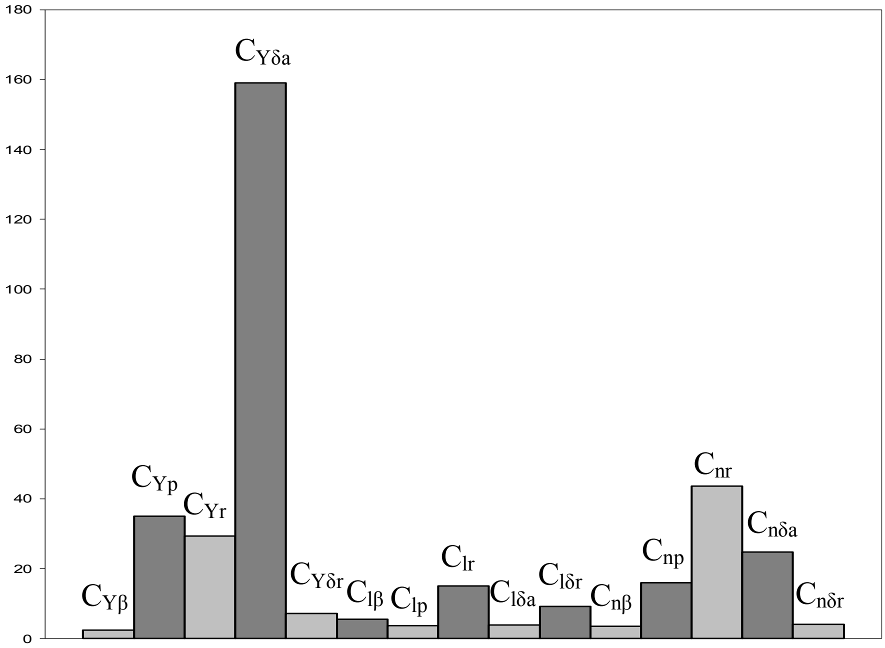

Fig.8. Relative standard deviations of the aerodynamic derivatives

During training cycle, weights of a network get adjusted. Once trained, the weights of an ANN remain fixed but the activation values of the neurons still change across the training set. Corresponding to each training vector one sensitivity matrix will result. In this particular case, 2128 samples will result 2128 sensitivity matrices. To get an average measure of the parameters over the entire training set, RMS values of the stability and control derivatives were obtained. These values are often [9] termed as mean square average sensitivities (MSAS). Fig. 5, 6 and 7 depicts the convergence of the RMS values of the parameters (MSAS) over the iterations.

One would expect the side force to be more sensitive to angle of sideslip and yaw rate and less sensitive to roll rate and aileron input. Its sensitivity on rudder input should be moderate. These are indeed the case as shown in Fig. 4. Again rolling moment should be highly sensitive on roll rate and aileron input and weakly sensitive on angle of sideslip and rudder input. Fig. 5 justifies these arguments. The significant dependence of rolling moment co-efficient on yaw rate is a result of roll-yaw coupling. Similarly, the trends of the parameters shown in Fig. 6 corroborate well with those expected from the knowledge of flight dynamics. Thus the RMS values of the parameters not only unify the gradients of all the training vectors of the ANN, but also provide an insight to the mechanics of flight. The histogram distribution (not shown in this extended abstract) of the parameters show near normal distribution and hence their arithmetic means were taken as the average values of the parameters.

IV. Conclusion

A novel method for estimating aircraft stability and control derivatives is proposed in this paper based on neural sensitivity analysis. The proposed method shows much superior estimation of aerodynamic forces and moments compared to all existing methods for aircraft parameter estimation using neural network.

References

[1] K. W. Iliff, "Parameter estimation for flight vehicle," *J. Guidance, Control and Dynamics*, vol. 12, no. 5, pp. 609-622, 1989.

[2] R. V. Jategaonkar, *Flight Vehicle System Identification: A Time Domain Methodology*, 1st ed., vol. 216, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2006.

[3] R. A. Hess, "On the use of back propagation with feed forward neural networks for aerodynamic estimation problem," *AIAA Paper 93-3638*, August 1993.

[4] H. M. Youseff, "Estimation of aerodynamic coefficients using neural networks," *AIAA Paper 93-3639*, August 1993.

[5] Basappa and R. V. Jategaonkar, "Aspects of feed forward neural network modeling and its application to lateral-directional flight data," *DLR-IB 111-95/30*, Braunschweig, Germany, September 1995.

[6] D. J. Linse and R. F. Stengel, "Identification of aerodynamic coefficients using computational neural networks," *J. Guidance, Control and Dynamics*, vol. 16, no. 6, pp.1018-1025, 1993.

[7] S. C. Raisinghani, A. K. Ghosh and P. K. Kalra, "Two new techniques for parameter estimation using neural networks," *The Aeronautical Journal*, vol. 102, no. 1011, pp. 25-29, 1998.

[8] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, pp. 525-533, 1993.

[9] J. M. Jurada, A. Malinowski and I. Cloete, "Sensitivity analysis for minimization of input data dimension for feedforward neural network", *IEEE International Symposium on Circuits and Systems*, vol. 6, pp. 447-450, 30 May-2 June, 1994.