JOURNAL OF AIRCRAFT
Vol. 35, No. 6, November–December 1998

Estimation of Aircraft Lateral-Directional Parameters Using Neural Networks

A. K. Ghosh,* S. C. Raisinghani,† and Sunil Khubchandani‡
Indian Institute of Technology, Kanpur 208 016, India

A recently proposed method (christened "the Delta method") of estimating aircraft parameters from flight data using feed-forward neural networks is applied for the extraction of lateral–directional parameters from simulated as well as real-flight data. The neural network is trained using aircraft motion and control variables as the network inputs and aerodynamic coefficients as the network outputs; the trained network is used to predict aerodynamic coefficients for a suitably modified input file. An appropriate interpretation and manipulation of such input–output files yields the estimates of the parameters. Flight data for lateral–directional dynamics are analyzed for various combinations and types of control inputs, and suitable control input forms are identified for better estimation via the proposed method. Robustness of the method with respect to measurement noise is demonstrated by its applicability to simulated flight data with pseudomeasurement noise, and to real-flight data.

Nomenclature

a_{ym}	= measured acceleration along y body axis, m/s^2
C_l, C_n	= coefficient of rolling and yawing moment
$C_{l(\cdot)}$, $C_{n(\cdot)}$, $C_{y(\cdot)}$	= nondimensional derivatives
C_y	= coefficient of lateral force
I_x, I_y, I_z	= moments of inertia about x, y, and z axis, kgm^2
I_{xz}	= cross products of inertia, kgm^2
l_μ	= mean aerodynamic chord, m
m	= mass of aircraft, kg
p, q, r	= roll, pitch, and yaw rates, rad/s
$\dot{p}$, $\dot{q}$, $\dot{r}$	= roll, pitch, and yaw accelerations, rad/s^2
$\bar{q}$	= dynamic pressure, N/m^2
S	= reference wing area, m^2
s	= half the wing span, m
u, v, w	= velocity components along x, y, z body axes, m/s
V	= airspeed, m/s
Xa, Ya, Za	= accelerometer offset distances relative to c.g.
α, β	= angle of attack and angle of sideslip, rad
δ_a, δ_r	= aileron and rudder deflections, rad
ρ	= air density, kg/m^3
ϕ, θ	= bank and pitch angles, rad

Superscript

$\cdot$	= derivative with respect to time

Introduction

THE new emerging field of artificial neural networks (ANNs) has attracted and prompted many scientists and engineers to explore its potential in diverse fields such as signal processing, pattern recognition, system identification, and control.[1] Recently, ANN modeling has been attempted for aircraft dynamics where aircraft motion variables and control inputs are mapped to predict the total aerodynamic coefficients.[2–5] There are different types of ANNs classified broadly on the basis of the type of connectivity between the processing elements (neurons), the type of architecture, and the number of layers in the network. Out of these, two types that have found favor for their application to aircraft dynamics modeling are feed-forward neural networks (FFNNs) and recurrent neural networks (RNNs). The FFNNs have neurons arranged in layers like directed graphs, implying a unidirectional flow of variables and, thus, are static in nature. The RNNs are dynamic neural networks incorporating an output feedback. Raol and Jategaonkar[6] showed amenability of RNNs to state-space modeling of aircraft dynamics and thereby demonstrated its applicability to explicitly estimate aircraft parameters (stability and control derivatives) from flight data. However, as stated by the authors,[6] the flexibility of RNNs is limited because of the fixed number of neurons needed for state-space formulation. On the other hand, the universal function approximation capability of FFNNs can be used to construct an unknown model structure to any desired accuracy. However, the FFNNs lead to a block-box type of modeling, where no physical significance can be attached either to the networks structure or to the weights. Although the flexibility of FFNNs has been demonstrated extensively for modeling aircraft dynamics and predicting the total coefficients,[2–5] there is no published work wherein FFNNs have been used for explicitly estimating aircraft stability and control derivatives as they occur in the equations of motion.

Only recently, Raisinghani et al.[7] proposed two new methods for explicitly estimating aircraft parameters from flight data using FFNNs. In Ref. 7, the two methods, christened "the Delta method" and "the Zero method," respectively, have been validated only on simulated flight data for the longitudinal short-period dynamics. In the present work, applicability of the Delta method for the estimation of lateral–directional parameters from simulated as well as real flight data is demonstrated. The longitudinal case had only one control input (elevator) for generating flight data and it required an estimation of only six unknown parameters.[7] In contrast, the lateral dynamics involve two control inputs (aileron and rudder), and the number of unknown parameters is 15 or more. The two control inputs can be used in any number of combinations for generating flight data. A detailed study has been carried out to identify suitable control input forms that would yield better estimates via the Delta method. In Ref. 7, the Delta method has been

Received Oct. 16, 1997; revision received April 20, 1998; accepted for publication July 5, 1998.

*Ph.D. Student, Department of Aerospace Engineering.

†Professor, Department of Aerospace Engineering. Senior Member AIAA.

‡Graduate Student, Department of Aerospace Engineering.

applied to simulated flight data only; in the present paper, for the first time, the method is validated on real-flight data.

FFNNs

FFNNs are composed of groups of neurons that are arranged into an input layer, an output layer, and one or more hidden layers. The number of nodes (neurons) in the input and output layers are determined, respectively, by the number of input and output variables, whereas the number of neurons in the hidden layer is decided by the complexity of the problem. Each neuron of a layer is connected to each neuron of the next layer, and each connection is assigned its individual connective weight. The neurons of the hidden and the output layers have a nonlinear activation function, e.g., a sigmoidal or a hyperbolic tangent function, which provides the network the required nonlinear decision capability for modeling.

For the purpose of lateral–directional aerodynamic modeling on FFNN (Fig. 1), the input variables to the network are the motion variables β, (ps/V) and (rs/V), and the control inputs δa and δr. The output variables are C_y, C_l, and C_n. During the training sessions of the network, the predicted values of the total coefficients C_l, C_n, or C_y are compared with the corresponding known values. The difference between the predicted and known values of the total coefficients at each time point yield the errors that are back-propagated using the method called the back-propagation algorithm (BPA). The BPA essentially treats error function as a function of network weights, and uses an iterative descent gradient algorithm in the weight parameter space to minimize the error between the predicted and the known (desired) values of the output variables. The connective weights are updated during every iterative step. Out of the two most popular BPA algorithms: 1) the batch or sweep and 2) the sequential or pattern learning, we have employed the latter wherein the network weights are updated sequentially as training data are presented. More details about FFNNs and BPA algorithms are available in the open literature.[1,3,8]

A brief study was conducted to determine whether the network should be trained to map the network input variables (β, rs/V, ps/V, δa, δr) to all three network output variables (C_l, C_n, C_y) simultaneously, or to only one output variable at a time. Because the option of one output variable at a time resulted in better training, it was adopted for all of the studies reported herein. Criterion for termination of the iterative process for training was based on computing mean square error (MSE), defined as follows:

$$\mathrm{MSE} = \frac{1}{m \times n} \sum_{j=1}^{n} \sum_{i=1}^{m} [y_i(j) - x_i(j)]^2$$

where y and x are, respectively, the desired (known) and the computed (predicted) outputs of the neural network; n is the number of data points; and m is the number of the output variables. If only one output variable C_l, C_y, or C_n is to be trained, then MSE is defined with $m = 1$. Training sessions are continued until changes in MSE in the successive iterations are less than the prescribed value or the number of iterations exceed the specified number.

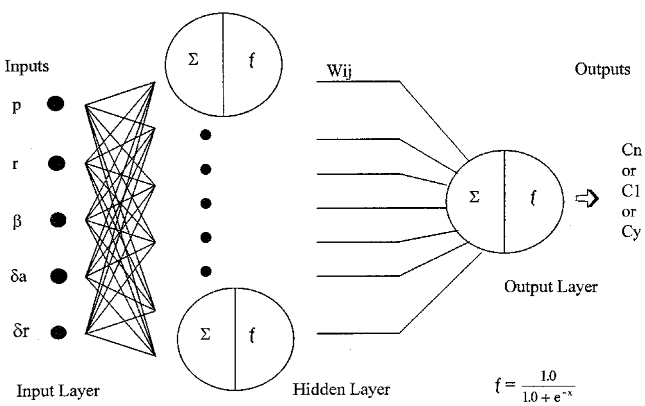

Fig. 1 Schematic of feed-forward neural networks.

A detailed study was carried out for a few sets of simulated and real-flight data to understand the influence of various network parameters (also called the influencing or tuning parameters) on the training and prediction capability of the network. A matrix of tuning parameters such as the number of hidden layers, the number of nodes in each of the hidden layer(s), the learning rate, the momentum rate, the abruptness (gain) factor of the sigmoidal function, the initial network weights, and the scaling of input–output data was generated, wherein each parameter was varied within a prescribed range,[2,4,5] and the network was trained to arrive at the best possible set that led to the minimum MSE for the given flight data. The final set of tuning parameters chosen is as follows: number of hidden layers = 1, number of neurons in hidden layer = 6, learning rate = 0.3, momentum rate = 0.5, logistic gain = 0.85, random seed = 0.6, and number of iterations = 5000.

After the training, the same input data are passed to check the prediction capability of the network. The predicted aerodynamic coefficients are deemed acceptable only if the MSE is less than the specified value. Once the network is trained satisfactorily to map network input variables (ps/V, rs/V, β, δa, δr) to each of the network output variables (C_l, C_n, C_y), it is then used for estimation of lateral–directional stability and control derivatives by using the Delta method[7] described next.

Delta Method

The Delta method[7] is based on the understanding of what a stability/control derivative stands for; the stability/control derivatives represent the variation in the aerodynamic force or moment coefficients caused by a small variation in one of the motion/control variables about the nominal value, whereas all of the other variables are held constant. For example, $C_{l\beta}$ represents a variation in C_l with respect to β, whereas all other variables, rs/V, ps/V, δa, δr, are held constant. Let us consider an estimation of $C_{l\beta}$ via the Delta method. For this purpose, the neural network is first trained such that the network input file variables ps/V, rs/V, β, δa, and δr are mapped to C_l. Next, a modified network input file is prepared wherein β values at each time point are perturbed by $\pm\Delta\beta$ while all of the other variables retain their original values. This modified file is now presented to the trained network and the corresponding predicted values of the perturbed C_l (C_l^+ for $\beta + \Delta\beta$ and C_l^- for $\beta - \Delta\beta$) are obtained at the output node. Now, the stability derivative $C_{l\beta}$ is given by $C_{l\beta} = (C_l^+ - C_l^-)/2\Delta\beta$. Similarly perturbing only, say δa, in the network input file will yield the control derivative $C_{l\delta a}$. Perturbations were given in both increasing (+) and decreasing (−) directions to avoid any bias resulting from one-sided differencing.

Generation of Simulated Flight Data

Because real-flight data were made available for DLR research aircraft advanced technology testing aircraft system (ATTAS),[9] it was found expedient to generate simulated data for the same aircraft. Details of the aircraft mass, moment of inertia, geometric characteristics, etc., are given in Ref. 9. The six-degree-of-freedom equations of motion[9] in a body-fixed axes system are first reduced to three-degree-of-freedom equations of motion pertaining to the lateral–directional dynamics, and then solved using the fourth-order Runge–Kutta method to generate simulated flight data for various types of aileron and/or rudder control inputs. The following decoupled lateral–directional equations of motion have been used:

$$\dot{v} = -ru + g \sin\phi + (\bar{q}S/m)C_y \tag{1a}$$

$$\dot{p} = [1/(I_x I_z - I_{xz}^2)][\bar{q}Ss(I_z C_l + I_{xz} C_n)] \tag{1b}$$

$$\dot{r} = [1/(I_x I_z - I_{xz}^2)][\bar{q}Ss(I_x C_n + I_{yz} C_l)] \tag{1c}$$

$$\dot{\phi} = p \tag{1d}$$

where $\beta = v/V$, $\bar{q} = (1/2)\rho(u^2 + v^2 + w^2)$, and $V = (u^2 + v^2 + w^2)^{1/2}$.

$$C_l = C_{lp}(ps/V) + C_{lr}(rs/V) + C_{l\beta}\beta + C_{l\delta a}\delta a + C_{l\delta r}\delta r \tag{2a}$$

$$C_n = C_{np}(ps/V) + C_{nr}(rs/V) + C_{n\beta}\beta + C_{n\delta a}\delta a + C_{n\delta r}\delta r \tag{2b}$$

$$C_y = C_{yp}(ps/V) + C_{yr}(rs/V) + C_{y\beta}\beta + C_{y\delta a}\delta a + C_{y\delta r}\delta r \tag{2c}$$

The flight condition defined by a landing flap deflection $\delta_f = SP(1 \text{ deg})$, and indicated airspeed $V_{IAS} = 160$ kn was chosen for generating simulated flight data because the real flight data for the same flight condition were also available for analysis.[9] The estimated values of parameters reported in Ref. 9 were used as the true values of the parameters for solving the equations of motion and subsequently preparing the network input file (β, rs/V, ps/V, δa, δr) and the output file (C_l, C_n, or C_y). Such sets of network input–output files for several types of control inputs were separately used to train the FFNN and the parameters estimated via the Delta method. The main findings are illustrated by presenting results for the following cases:

Case 1: multistep 3-2-1-1 type aileron input.
Case 2: arbitrarily varying aileron input.
Case 3: a sinusoidal aileron input.
Case 4: identical multistep 3-2-1-1 type aileron and rudder inputs applied simultaneously.
Case 5: multistep 3-2-1-1 type aileron and rudder inputs; aileron input followed by, after a 1-s interval, rudder input.
Case 6: aileron and rudder inputs taken from real flight data, both nearly 3-2-1-1 type and applied simultaneously.
Case 7: control inputs same as for case 6, but 5% noise added to motion variables (β, p, r) and aerodynamic coefficients (C_y, C_l, C_n).

Parameter Estimation from Simulated Flight Data

For case 1, most of the parameters, particularly the roll derivatives, were well estimated, only weak derivatives such as C_{yp} and C_{yr} being the exceptions. The estimates for case 2 showed considerable deterioration compared with case 1, and for case 3 they deteriorated to an unacceptable level. This only confirmed the well-known result; multistep 3-2-1-1 type inputs are capable of better excitation of aircraft dynamics to increase the information content in the responses and, thereby, leading to better estimates via any of the existing parameter estimation methods,[10] e.g., the maximum likelihood method, the filter error method, etc. Similar observations for the Delta method reconfirm that estimation via the Delta method can also be improved by using optimal control inputs for generating flight data.

The next set of inputs (cases 4–7) used aileron and rudder inputs. The results for these cases are given in Table 1. Results for case 4, having identical δ_a and δ_r inputs, show a pairwise correlation between aileron and rudder control derivatives; i.e., the values of ($C_{l\delta a}$ and $C_{l\delta r}$), ($C_{n\delta a}$ and $C_{n\delta r}$), and ($C_{y\delta a}$ and $C_{y\delta r}$) are of the same order of magnitude, although the corresponding true values are not. It is worth noting that $C_{l\delta a}$ and $C_{l\delta r}$, in addition to having almost equal magnitude (0.097 and 0.098), also have the same sign (negative), even though the true values are of the opposite signs. Thinking from the flight mechanic's point of view, identical aileron and rudder inputs should have had no special significance and, hence, initially, the result seemed very intriguing. On reflection, however, it was obvious that the pairwise values of these control derivatives would indeed be nearly equal; from the point of view of the network, it is being shown as two identical inputs, and it has no way of distinguishing one from the other as being the aileron or the rudder control input, and thereby, any perturbation in either one would yield almost equal values.

Table 1 Parameter estimates from simulated flight data

Parameter	Estimated parameters: Ref. 7	Case 4	Case 5	Case 6	Case 7
$-C_{lp}$	0.978	0.905	1.007	0.934	0.990
	(0.02)[a]	(0.029)[b]	(0.042)	(0.022)	(0.038)
C_{lr}	0.418	0.425	0.372	0.427	0.365
	(0.43)	(0.018)	(0.020)	(0.024)	(0.022)
$-C_{l\beta}$	0.126	0.107	0.125	0.118	0.124
	(0.39)	(0.009)	(0.005)	(0.010)	(0.004)
$-C_{l\delta a}$	0.247	0.097	0.258	0.274	0.251
	(0.20)	(0.001)	(0.012)	(0.013)	(0.009)
$C_{l\delta r}$	0.046	−0.098	0.036	0.065	0.037
	(0.120)	(0.001)	(0.001)	(0.012)	(0.001)
$-C_{np}$	0.115	0.130	0.137	0.128	0.053
	(0.73)	(0.044)	(0.057)	(0.022)	(0.062)
$-C_{nr}$	0.495	0.512	0.480	0.549	0.386
	(0.56)	(0.027)	(0.045)	(0.023)	(0.036)
$C_{n\beta}$	0.281	0.255	0.272	0.258	0.287
	(0.07)	(0.019)	(0.013)	(0.013)	(0.028)
$-C_{n\delta a}$	0.0	0.097	0.003	0.007	0.018
	——	(0.008)	(0.008)	(0.002)	(0.007)
$-C_{n\delta r}$	0.166	0.067	0.165	0.169	0.166
	(0.12)	(0.012)	(0.010)	(0.010)	(0.014)
C_{yp}	0.303	−0.157	0.252	0.182	0.010
	(5.18)	(0.012)	(0.183)	(0.032)	(0.152)
C_{yr}	0.727	0.972	0.459	0.904	0.425
	(2.32)	(0.088)	(0.095)	(0.106)	(0.266)
$-C_{y\beta}$	1.132	1.113	1.044	0.986	1.185
	(0.33)	(0.101)	(0.092)	(0.095)	(0.126)
$C_{y\delta a}$	0.029	0.109	0.039	0.027	0.146
	(13.9)	(0.009)	(0.012)	(0.026)	(0.049)
$C_{y\delta r}$	0.191	0.148	0.193	0.235	0.167
	(2.02)	(0.015)	(0.027)	(0.018)	(0.017)

[a]Standard deviation in percent as in Ref. 9. [b]Sample standard deviation.

It was therefore realized that the control inputs δ_a and δ_r had to be such that the network could easily distinguish them. For example, one could use different forms of aileron and rudder inputs, or introduce a time delay between the application of aileron and rudder inputs. Many such combinations were tried successfully. For illustration, results for case 5 (using identical aileron and rudder inputs with a 1-s delay), given in column 4 of Table 2, show that the accuracy of the estimation improves dramatically as compared to case 4, not only for the control derivatives but also for most of the stability derivatives.

Case 6 uses control inputs similar to those for case 4, but being inputs from real life, the δ_a and δ_r inputs are not identical like case 4. Thus, to some extent, the network is able to distinguish the δ_a and δ_r inputs, which thereby, not only weakens the pairwise correlation between the control derivatives but also improves the accuracy of many other parameters (Table 1). However, the estimated values of parameters are not as good as those for case 5, although they are better than for case 4. These observations suggest that the control inputs that can be easily distinguished by the FFNN are required for estimation via the Delta method.

It was also of interest to see how the accuracy of estimates is affected by the presence of measurement noise in the input and output variables of the network. To this purpose, pseudonoise was added to p, r, β, C_l, C_n, and C_y of case 5. The noise was simulated by generating successively uncorrelated pseudorandom numbers having a normal distribution with zero mean and assigned a standard deviation, the standard deviation corresponding approximately to the designated percentage (1 and 5%) of the maximum amplitude of the corresponding variable. It was observed that the addition of 1% noise had almost no effect on the accuracy of estimates; a higher noise level of 5% affected the strong derivatives marginally, whereas some of the weak derivatives like C_{np}, C_y-derivatives (except $C_{y\beta}$) were significantly affected (case 7, Table 1). This shows robustness of the Delta method with respect to measurement noise and its potential for success on real flight data.

Table 2 Parameter estimates from real flight data

	V_{IAS} = 160 kn			V_{IAS} = 250 kn	
Parameter	Ref. 9	Case 8	Case 9	Ref. 9	Case 10
$-C_{lp}$	0.978	1.040	0.866	1.014	0.958
	(0.20)[a]	(0.044)[b]	(0.013)	(0.22)	(0.098)
C_{lr}	0.418	0.479	0.594	0.319	0.951
	(0.43)	(0.08)	(0.028)	(1.16)	(0.099)
$-C_{l\beta}$	0.126	0.133	0.110	0.116	0.108
	(0.39)	(0.03)	(0.003)	(0.91)	(0.014)
$-C_{l\delta a}$	0.247	0.125	0.209	0.256	0.265
	(0.20)	(0.04)	(0.004)	(0.20)	(0.020)
$C_{l\delta r}$	0.046	0.107	0.027	0.043	0.0325
	(0.12)	(0.012)	(0.007)	(2.16)	(0.025)
$-C_{np}$	0.115	0.263	0.068	0.049	0.097
	(0.73)	(0.04)	(0.003)	(1.20)	(0.017)
$-C_{nr}$	0.495	0.351	0.864	0.599	0.793
	(0.56)	(0.15)	(0.026)	(0.94)	(0.065)
$-C_{n\dot{\beta}}$	0.231	−0.269	0.248	0.284	0.159
	(1.16)	(0.007)	(0.013)	(1.54)	(0.015)
$C_{n\beta}$	0.281	0.234	0.264	0.297	0.233
	(0.07)	(0.04)	(0.005)	(0.11)	(0.016)
$-C_{n\delta r}$	0.166	0.234	0.169	0.169	0.167
	(0.12)	(0.03)	(0.004)	(0.36)	(0.009)
C_{yp}	0.303	0.262	0.111	0.207	0.205
	(5.18)	(0.37)	(0.025)	(8.12)	(0.028)
C_{yr}	0.727	2.942	1.146	0.511	2.533
	(2.32)	(0.22)	(0.603)	(6.96)	(0.158)
$-C_{y\beta}$	1.133	1.034	1.083	1.156	0.923
	(0.33)	(0.17)	(0.049)	(0.69)	(0.032)
$C_{y\delta a}$	0.029	0.321	0.129	0.0037	0.035
	(13.5)	(0.06)	(0.011)	(108.0)	(0.003)
$C_{y\delta r}$	0.191	0.017	0.095	0.222	0.235
	(2.02)	(0.05)	(0.008)	(3.79)	(0.028)
C_{l0}	0.00099	−0.00016	0.00146	0.00069	0.00005
C_{n0}	0.00161	0.00240	0.00147	0.00135	0.00051
C_{y0}	−0.00454	0.00642	−0.00214	−0.00348	0.00132

[a]Standard deviation in percent as in Ref. 9. [b]Sample standard deviation.

Each parameter is estimated at all time points for which the input–output data are utilized for training the network. For example, using the response of a 7-s duration with a sampling interval of 0.1 s will yield 140 estimates (two at each time point corresponding to $+\Delta$ and $-\Delta$ perturbations) for each parameter. To illustrate the spread of estimated values, Fig. 2 shows histograms of estimated C_l derivatives for case 5. Because the histograms show that most of the parameters have a near-normal distribution, the mean is taken as the actual estimate and the sample standard deviation as the measure of its accuracy. It is conjectured that a less than perfect match between the actual and predicted values of aerodynamic coefficients (C_l, C_n, and C_y) is primarily responsible for the observed spread in the estimated values.

Parameter Estimation from Real Flight Data

The flight records supplied by DLR for the ATTAS aircraft had raw data for V, α, β, p, q, r, δ_a, δ_r, $\dot{\beta}$, $\dot{p}$, $\dot{r}$, and a_{ym}: the information about 1) the fuel consumption (to help calculate the mass, c.g., and the moment of inertias of the aircraft at any desired time point), 2) the location of measuring instruments, and 3) the time delays in the measured quantities. The parameter estimates (via the maximum likelihood method) from the same flight data are given in Ref. 9; the estimates are with respect to an experimental axes system with its origin at a chosen reference point.[9] To facilitate direct comparison, raw data were processed to account for 1) the offset of the recording instruments and the reference point and 2) the axes transformation. Additional corrections for the time delays and fuel consumption were done as per information supplied by DLR.[9]

The aerodynamic coefficients C_l, C_n, and C_y were calculated using the following equations obtained from the six-degree-of-freedom equations of motion in body-fixed axes with their origin at the c.g.[9]

$$C_l = (1/\bar{q}Ss)[\dot{p}I_x - \dot{r}I_{xz} - pqI_{xz} + qr(I_z - I_y)] \quad (3a)$$

$$C_n = (1/\bar{q}Ss)[\dot{r}I_z - \dot{p}I_{xz} + qrI_{xz} - pq(I_x - I_y)] \quad (3b)$$

$$C_y = (m/\bar{q}S)[a_{ym} - (pq + \dot{r})Xa + (r^2 + p^2)Ya - (rq - \dot{p})Za] \quad (3c)$$

After applying the necessary corrections and transformations, the network input file with motion and control variables ps/V, rs/V, β, δ_a, and δ_r, and the output file having C_l, C_n, or C_y were prepared. The additional input variables $\dot{\beta}s/V$ and ql_μ/V were considered to see how the training of the network was affected by inclusion/exclusion of one or both the variables. The best possible training was achieved by including ql_μ/V in the input file for training of C_l, and ql_μ/V and $\dot{\beta}s/V$ for C_n.

For illustration, flight data for the flight conditions defined by a landing flap deflection δ_f = SP(1 deg), and perturbations about steady-state flights at indicated speeds V_{IAS} = 160 kn (cases 8 and 9) and V_{IAS} = 250 kn (case 10) are chosen.[9] Case 9 analyzed data for the control inputs shown in Fig. 3a, and case 10 analyzed data for the control inputs of Fig. 3b. Figure 3a inputs consisted of two multistep 3-2-1-1 type δ_a and δ_r input sequences of increasing amplitudes, followed by two aileron step inputs in opposite directions. From Fig. 3a, case 8 used a small segment of data corresponding to the simultaneous application of 3-2-1-1 type δ_a and δ_r inputs in the time interval of 13.0–22.5 s (this segment is shown by dashed lines in Fig. 3b). Table 2 lists parameter estimates via the Delta method for cases 8–10, along with the parameter estimates of Ref. 9 for ready comparison.

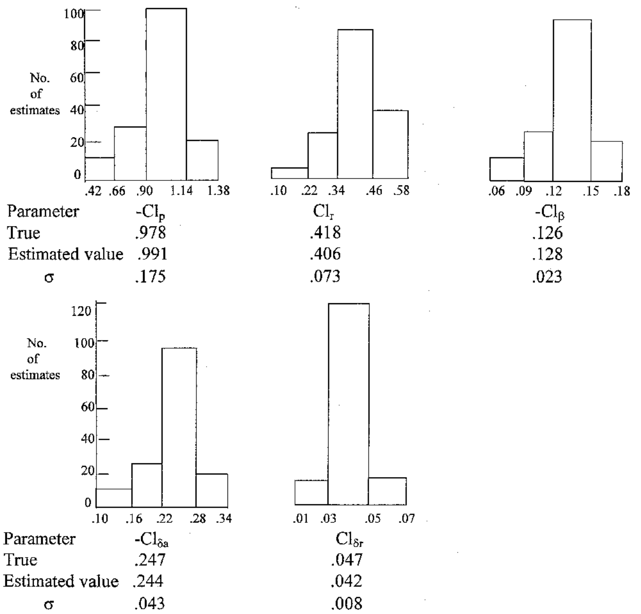

Fig. 2 Histograms of parameter estimates via the Delta method.

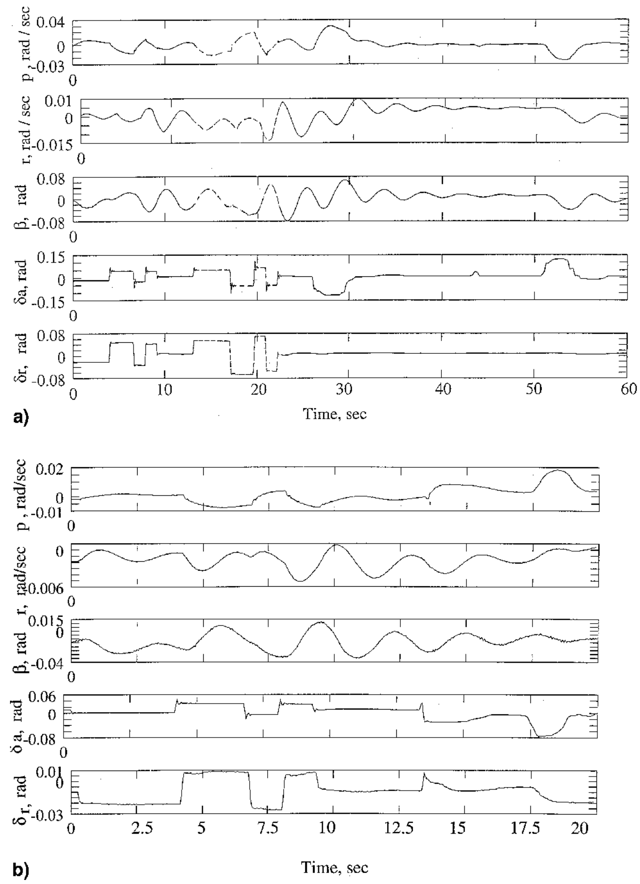

Fig. 3 Real flight data for cases a) 8 and 9 and b) 10.

Case 8: the rolling moment stability derivatives are well estimated, but the control derivatives $C_{l\delta a}$ and $C_{l\delta r}$ show a high correlation. Further, estimates of many C_y and C_n derivatives are not satisfactory. Because the flight data being analyzed correspond to similar looking δ_a and δ_r control inputs applied almost simultaneously, the results reconfirm our earlier contention that such inputs are a poor choice for the estimation of parameters via the Delta method.

Case 9: most of the parameters are fairly well estimated. The presence of large roll maneuver data (generated by the two aileron pulse inputs) has helped in the estimation of $C_{n\beta}$, and also interrupted, to a large extent, the correlation between the control derivatives ($C_{l\delta a} - C_{l\delta r}$) and ($C_{n\delta r} - C_{n\delta a}$). However, estimates of relatively weak control derivatives, $C_{y\delta a}$ and $C_{y\delta r}$, still show correlation. This, we believe, is a result of the first part of the control inputs (up to about the first 22.5 s, Fig. 3a) having similar-looking δ_a and δ_r inputs. This conjecture is further supported by reasonable estimates of these control derivatives for case 10, wherein δ_a and δ_r that were used have dissimilar forms.

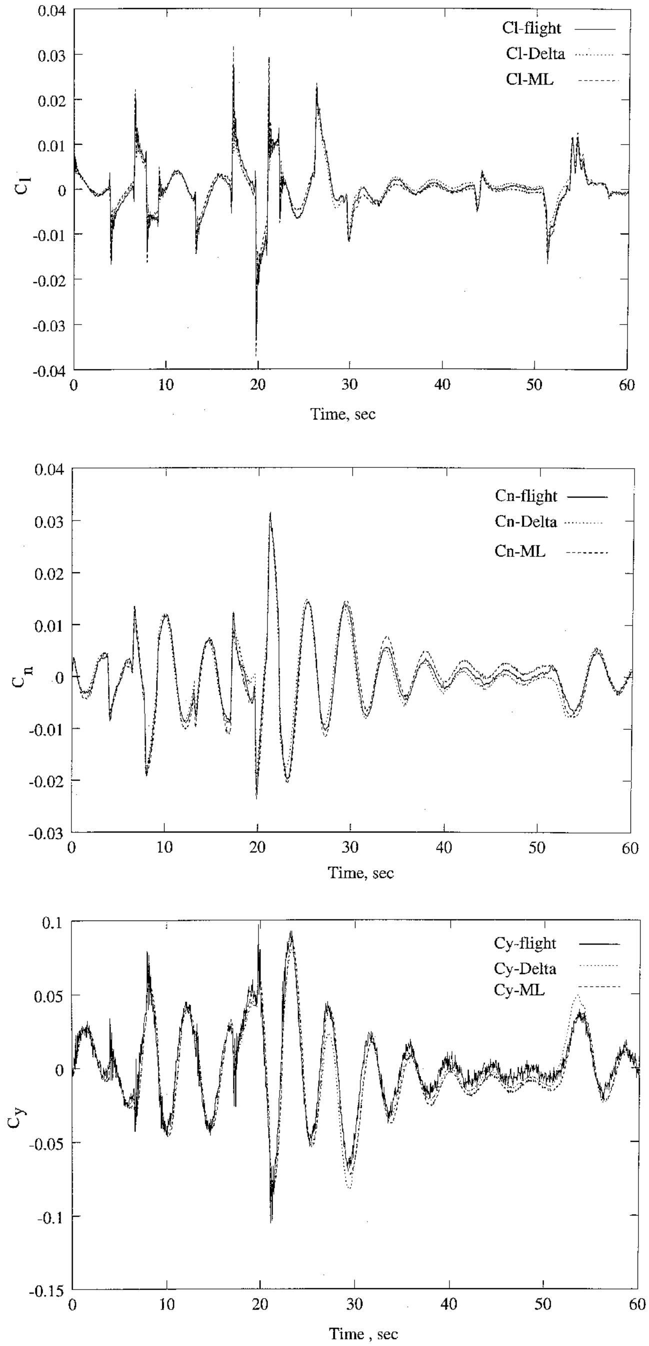

Fig. 4 Comparison of actual and estimated total aerodynamic coefficients.

Case 10: it may be noted from Fig. 3b that the 3-2-1-1 type of δ_a and δ_r inputs for this case are appended by dissimilar-looking δ_a and δ_r inputs and, therefore, we expect better estimates as compared with cases 8 and 9. Results for case 10 confirm this expectation, and most of the parameters are fairly well estimated. Some of the weak derivatives, such as C_{lr}, C_{np}, and C_{yr}, however, have values larger than those reported in

Ref. 9. Also, the estimation of $C_{n\beta}$ is not as good as that reported for case 9. As pointed out by Jategaonkar,[9] the estimation of $C_{n\beta}$ requires maneuvers of a special kind, viz., a large roll maneuver such as that for case 9.

As observed earlier, the number of estimated values for each of the parameters is equal to the number of data points used for training the network. For real flight data, in addition to training being less than perfect, the parameters may not be strictly constants, i.e., the parameters may vary slightly with other motion and control variables. Furthermore, all of the corrections and axes transformations done on the data would introduce their own uncertainties. All of these factors contribute toward different estimates at different time points. Experience with many simulated and real-flight data sets suggested that the estimated values of the parameters be arranged in a descending (or ascending) order and that 25% estimates from both ends of the ordered set be removed before calculating the mean and sample standard deviations. The estimates given in Table 2 incorporate the preceding step. Table 2 also lists estimates of C_{l0}, C_{n0}, and C_{y0}. To estimate C_{l0}, C_{n0}, and C_{y0}, if all the inputs to the trained network are set to zero, the predicted total coefficient C_l, C_n, or C_y are the required estimates.

It is of interest to compare the total coefficients C_l, C_n, and C_y obtained from 1) Eq. (3) using the flight data (C_l-flight, C_n-flight, and C_y-flight); 2) Eq. (2) using estimated parameters via the Delta method (C_l-Delta, C_n-Delta, and C_y-Delta); and 3) Eq. (2) using estimated parameters via the maximum likelihood (ML) method[9] (C_l-ML, C_n-ML, and C_y-ML). Figure 4 illustrates such a comparison for case 9; based on these results, it is difficult to claim that the estimates from either of the methods, i.e., the Delta method or the maximum likelihood method,[9] are definitely superior when compared with the other.

Conclusions

A recently proposed method, the Delta method, for the estimation of aircraft parameters from flight data using the FFNNs, has been validated on simulated and real-flight data. The results from the simulated data clearly establish that it would be a benefit to have flight data corresponding to δ_a and δ_r inputs that can be easily distinguished by the neural network, e.g., inputs of case 5, and avoid inputs that look similar to each other. Notwithstanding our limitation of not being able to generate our own data, the proposed Delta method has been validated on real-flight data.

The proposed Delta method is a straightforward method to apply; the FFNN is trained on the given flight data and the parameters estimated in one try. It is worth emphasizing that, unlike many of the popular estimation algorithms, e.g., the maximum likelihood method, the filter error method, etc., that need initial values of parameters to begin the algorithm, the Delta method does not even require order of magnitude information about the parameters to be estimated. Thus, the method may be viewed as alternative and complementary to the existing methods of parameter estimation. It may also be used for generating initial values of parameters that can be used as starting values for the methods such as the maximum likelihood method that can subsequently further fine tune the estimates.

Future directions of our research include finding schemes for improving the FFNNs training to model the total coefficients and this, we believe, will lead to better estimates. Because the BPA is associated with severe drawbacks, one possible direction for improving the network training (modeling) could be to look for alternatives to the BPA. Work is also in progress to fully explore the advantages of the FFNN approach vs conventional methods, ML method, etc., for estimating parameters of an unstable aircraft, an aeroelastic aircraft, and an aircraft at nonlinear dynamic conditions.

Finally, the method has the potential for on-line estimation of parameters. The coefficient network may be carried on board for online learning (mapping) the coefficients only; the coefficients, in turn, can be accurately estimated using the aircraft sensors and the equations of motion. Such a coefficient network would require a few iterations for on-line learning and, thus, may be used for on-line parameter estimation.

Acknowledgment

The authors are grateful to P. G. Hamel and R. V. Jategaonkar of DLR, German Aerospace Center, Institute of Flight Mechanics, Germany, for supplying and granting permission to use the flight data of ATTAS aircraft. We are grateful to Jategaonkar for his help in interpreting and answering our questions about the flight data. We would also like to thank P. K. Kalra of the Electrical Engineering Department of the Indian Institute of Technology for many useful discussions and the use of the neural network simulator at his laboratory.

References

[1]Hornik, K., Stinchcombe, M., and White, H., "Multi-Layer Feed Forward Networks are Universal Approximators," *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366.

[2]Hess, R. A., "On the Use of Back Propagation with Feed Forward Neural Networks for Aerodynamic Estimation Problem," AIAA Paper 93-3638, Aug. 1993.

[3]Youseff, H. M., "Estimation of Aerodynamic Coefficients Using Neural Networks," AIAA Paper 93-3639, Aug. 1993.

[4]Bassapa, and Jategaonkar, R. V., "Aspects of Feed Forward Neural Network Modeling and Its Application to Lateral-Directional Flight Data," DLR-IB 111-95/30, Braunschweig, Germany, Sept. 1995.

[5]Linse, D. J., and Stengel, R. F., "Identification of Aerodynamic Coefficients Using Computational Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 6, 1993, pp. 1018–1025.

[6]Raol, J. R., and Jategaonkar, R. V., "Aircraft Parameter Estimation Using Recurrent Neural Networks—A Critical Appraisal," AIAA Paper 95-3004, Aug. 1995.

[7]Raisinghani, S. C., Ghosh, A. K., and Kalra, P. K., "Two New Techniques for Parameter Estimation Using Neural Networks," *The Aeronautical Journal*, Vol. 102, No. 1011, 1998, pp. 25–29.

[8]Chen, S., and Cowan, C. F. N., "Parallel Recursive Prediction Error Algorithm for Training Layered Networks," *International Journal of Control*, Vol. 50, No. 6, 1990, pp. 1225–1228.

[9]Jategaonkar, R., "Identification of the Aerodynamic Model of the DLR Research Aircraft ATTAS from Flight Test Data," DLR, DLR-FB 90-40, Braunschweig, Germany, Sept. 1990.

[10]Maine, R. E., and Iliff, K. W., "Application of Parameter Estimation to Aircraft Stability and Control," NASA RP 1168, Jan. 1986.