pp 409–428. © Royal Aeronautical Society 2019
doi:10.1017/aer.2019.125

Nonlinear parameter estimation of airship using modular neural network

S. Agrawal
subhamagrwl246@gmail.com
D. Gobiha
gobiha@yahoo.com
N.K. Sinha
nandan@ae.iitm.ac.in
Flight Dynamics and Control Lab
Department of Aerospace Engineering
Indian Institute of Technology Madras
Chennai
India

ABSTRACT

The prime focus of this work is to estimate stability and control derivatives of an airship in a completely nonlinear environment. A complete six degrees of freedom airship model has its aerodynamic model as nonlinear functions of angle of attack. Estimating the parameters of aerodynamic model in a nonlinear environment is challenging as it demands an exhaustive dataset that could cover the entire regime of operation of airship. In this work, data generation is achieved by simulating the mathematical model of airship for different trim conditions obtained from continuation analysis. The mathematical model is simulated using predicted parameter values obtained using DATCOM methodology. A modular neural network is then trained using back-propagation and Adam optimisation algorithm for each of the aerodynamic coefficients separately. The estimated nonlinear airship parameters are found to be consistent with the DATCOM parameter values which were used for open-loop simulation. This validates the proposed methodology and could be extended to estimate airship parameters from real flight data.

Keywords: Parameter estimation; Neural network; Airship

NOMENCLATURE

B	buoyant force, N
b	wingspan, m
b_z	point of action of buoyant force along z axis, m
C_L, C_D, C_Y	lift, drag and side force coefficients

Received 30 March 2019; revised 11 July 2019; accepted 10 September 2019.

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

C_l, C_m, C_n	roll, pitch and yaw moment coefficients
c	wing aerodynamic chord, m
d_z	point of action of thrust along z axis, m
J_x, J_y, J_z	moments of inertia about x, y and z axis, $kg\ m^2$
J_{xz}	cross products of inertia, $kg\ m^2$
m	mass of the airship, kg
m_x, m_y, m_z	components of apparent mass in x, y and z direction, kg
p, q, r	roll, pitch and yaw rates in body axis, rad/s
S	wing planform area, m^2
T_m	maximum available engine thrust, N
V	velocity of the airship, m/s
$\hat{Y}, Y$	predicted and actual values
α, β	angle of attack and sideslip angle, rad
γ, μ	angles along the wind axis, rad
ϕ, θ	Euler bank and pitch angles, rad
η	throttle ratio
ρ	density of air, kg/m^3
$\delta_e, \delta_a, \delta_r$	elevator, aileron and rudder deflection angles, rad

1.0 INTRODUCTION

An airship is a lighter-than-air gas balloon that can be navigated at the design altitude by engine- or battery-driven propellers for longer duration. It has gained popularity in scientific research and industry because of its energy-efficient nature and has a wide range of applications varying from surveillance, weather monitoring and advertising to planetary exploration. For most of these applications, airship has to be positioned in space for extended duration of time or has to track certain desired trajectories. In order to accomplish the above objectives, it would be imperative to estimate system parameters which would be used to design model based controllers. However, more often these parameters are unavailable to control engineers. Thus, in this work, a method is proposed in a completely nonlinear environment to estimate parameters of an airship using flight data.

In the past, model-based methods have been studied for aerodynamic parameter estimation. These approaches require mathematical model of the aerial vehicle to estimate the parameters of the system and can be broadly classified into three categories[1]: (i) equation error method (EEM), (ii) output error method (OEM) and (iii) filter error method (FEM). The EEMs represent a linear estimation problem and cannot handle process noise or measurement noise. maximum likelihood (ML) method, a variant of OEM, can be effectively used to estimate parameters of linear systems and general nonlinear systems[2] and can handle measurement noise but not process noise. The FEMs are the stochastic method for estimation of aircraft parameters which account for both measurement and process noise. These methods have been used to estimate parameters of HANSA-3 aircraft from longitudinal flight data in Ref. (3).

Recently, with the advances in computing power, large amounts of data are being collected during operations of flight. This led to the development of data-based methods using artificial neural network (ANN) for flight vehicle identification[4,5]. Raol and Jategaonkar have used recurrent neural networks (RNNs), a type of ANN, to estimate aircraft parameters from flight data[6]. However, they concluded that RNNs have limited scope and feed forward neural networks (FFNNs) are more flexible and have higher potential for future applications in

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

aircraft parameter estimation[5]. The main advantage of using FFNN is its ability to map any nonlinear function[7]. Raisinghani and co-workers[8–10] proposed two methods, namely, the Delta and the Zero method for estimating aircraft parameters using FFNN from flight data. Both these models do not require a prior postulation of the aerodynamic model and an initial guess of estimates to start with, unlike most of the model-based methods. Further, Peyada and Ghosh[11] proposed a new method christened the Neural-Gauss-Newton (NGN) method to estimate parameters. It can capture nonlinear aerodynamics and estimate aerodynamic parameters from flight data having unknown process and measurement noise[12]. It does not require solving the equations of motion and FFNN is trained to capture the flight dynamic model of an aircraft, whereas in the Delta and the Zero method, FFNN is trained to capture only aerodynamic model.

In most of the above stated works, the aircraft model is linearised around the operating regime, and stability and control derivatives are estimated for that particular trim condition. Though this method holds for cruise flight condition, it fails when aircraft exhibits unanticipated nonlinear manoeuvers. In such scenarios, aerodynamic coefficients cannot be assumed constant, and their nonlinear variation with angle of attack should be promptly considered to cover the entire operating regime of the airship. This consideration complicates the estimation procedure as aircraft equations of motion cannot be decoupled. This is because lateral directional aerodynamic coefficients are functions of angle of attack which is not a state in lateral directional mode. This work overcomes these challenges by combining bifurcation and continuation approach for data generation and neural network algorithm for the estimation procedure.

Bifurcation and continuation approach is used to compute all possible trim states of the airship model, which are taken as initial conditions for open-loop simulation. This results in an exhaustive dataset which covers the entire operating regime of the airship. The obtained dataset is then used for estimation procedure. There are different types of ANNs based on the kind of architecture and connectivity between neurons. In this case, modular neural network[13] was found to be the most promising neural network architecture because of its modular structure. Each module consists of an FFNN which maps angle of attack input to airship parameter. These modules or independent FFNNs are then integrated with a dot product layer to obtain an output layer with one node. The output node represents aerodynamic coefficient to be estimated. Modular networks are then trained using back-propagation algorithm to estimate airship parameters.

This paper is organised in six sections. Section 2 briefs the mathematical model of the considered airship, whereas the data generation procedure is elaborated in Section 3. Section 4 explains the modular neural network scheme for parameter estimation. Simulation results are discussed in Section 5, and Section 6 concludes the paper.

2.0 MATHEMATICAL MODEL

2.1 Equations of motion

A high-altitude airship of pressure altitude 21km is considered in this work. This helium filled aerial vehicle is controlled using forward thrusters and control surfaces at the rear positioned in the cruciform tail configuration. Though stratospheric airship is considered in this work, a picture of low-altitude airship used in our laboratory for experimental purpose is shown in Fig. 1, for the ease of comprehension.

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press



Figure 1. Experimental airship.

The detailed study on the mathematical modelling of the considered airship was carried out in Refs. (14,15) and is presented in this section for the sake of completeness.

$$
\begin{aligned}
\dot{V} &= \frac{1}{m_x}\left(T_m\eta\cos\alpha\cos\beta - \frac{1}{2}\rho V^2 SC_D - (mg - B)\sin\gamma\right)\\
\dot{\alpha} &= q - \frac{1}{\cos\beta}\left(\left(p\cos\alpha + r\sin\alpha\right)\sin\beta\right.\\
&\qquad \left. + \frac{1}{m_z V}\left(\frac{1}{2}\rho V^2 SC_L + T_m\eta\sin\alpha - (mg - B)\cos\mu\cos\gamma\right)\right)\\
\dot{\beta} &= \left(p\sin\alpha - r\cos\alpha\right) + \frac{1}{m_y V}\left(\frac{1}{2}\rho V^2 SC_Y - T_m\eta\sin\beta\cos\alpha + (mg - B)\sin\mu\cos\gamma\right)\\
\dot{p} &= \left(\frac{J_y - J_z}{J_x}\right)qr + \frac{J_{xz}}{J_x}pq + \frac{1}{J_x}\left(\frac{1}{2}\rho V^2 SbC_l - Bb_z\sin\mu\cos\gamma\right)\\
\dot{q} &= \left(\frac{J_z - J_x}{J_y}\right)pr + \frac{J_{xz}}{J_y}(r^2 - p^2) + \frac{1}{J_y}\left(T_m\eta dz\cos\alpha\cos\beta + \frac{1}{2}\rho V^2 ScC_m - Bb_z\sin\theta\right)\\
\dot{r} &= \left(\frac{J_x - J_y}{J_z}\right)pq - \frac{J_{xz}}{J_z}qr + \frac{\rho V^2 SbC_n}{2J_z}\\
\dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta\\
\dot{\theta} &= q\cos\phi - r\sin\phi
\end{aligned}
\qquad \ldots (1)
$$

where

$$
\begin{aligned}
\sin\gamma &= \cos\alpha\cos\beta\sin\theta - \sin\alpha\cos\beta\cos\phi\cos\theta - \sin\beta\sin\phi\cos\theta\\
\sin\mu\cos\gamma &= \cos\alpha\sin\beta\sin\theta - \sin\alpha\sin\beta\cos\phi\cos\theta + \cos\beta\sin\phi\cos\theta\\
\cos\mu\cos\gamma &= \sin\theta\sin\alpha + \cos\alpha\cos\phi\cos\theta
\end{aligned}
\qquad \ldots (2)
$$

2.2 Aerodynamic model

A aerodynamic model of the airship as used in Ref. (16) has been regarded and is tabulated below. As the entire regime of operation of aircraft is considered, all stability and control derivatives are taken as function of α.

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

$$
\begin{aligned}
C_L &= C_{L_0}(\alpha) + C_{L_q}(\alpha)\frac{qc}{2V} + C_{L_{\delta_e}}(\alpha)\delta_e \\
C_Y &= C_{Y_\beta}(\alpha)\beta + C_{Y_p}(\alpha)\frac{pb}{2V} + C_{Y_r}(\alpha)\frac{rb}{2V} + C_{Y_{\delta_a}}(\alpha)\delta_a + C_{Y_{\delta_r}}(\alpha)\delta_r \\
C_l &= C_{l_\beta}(\alpha)\beta + C_{l_p}(\alpha)\frac{pb}{2V} + C_{l_r}(\alpha)\frac{rb}{2V} + C_{l_{\delta_a}}(\alpha)\delta_a + C_{l_{\delta_r}}(\alpha)\delta_r \\
C_n &= C_{n_\beta}(\alpha)\beta + C_{n_p}(\alpha)\frac{pb}{2V} + C_{n_r}(\alpha)\frac{rb}{2V} + C_{n_{\delta_a}}(\alpha)\delta_a + C_{n_{\delta_r}}(\alpha)\delta_r \\
C_m &= C_{m_0}(\alpha) + C_{m_q}(\alpha)\frac{qc}{2V} + C_{m_{\delta_e}}(\alpha)\delta_e \\
C_D &= C_{D_0}(\alpha) + C_{D_q}(\alpha)\frac{qc}{2V} + C_{D_{\delta_e}}(\alpha)\delta_e
\end{aligned} \quad \ldots (3)
$$

Concisely, this work aims in estimating these derivatives from simulated flight data to establish an aero model for the considered airship.

$$
\begin{aligned}
\Theta = \big[& C_{L_0}\ C_{L_q}\ C_{L_{\delta_e}}\ C_{Y_\beta}\ C_{Y_p}\ C_{Y_r}\ C_{Y_{\delta_a}}\ C_{Y_{\delta_r}}\ C_{l_\beta}\ C_{l_p}\ C_{l_r}\ C_{l_{\delta_a}}\ C_{l_{\delta_r}}\ C_{n_\beta}\ C_{n_p}\ C_{n_r}\ C_{n_{\delta_a}}\ C_{n_{\delta_r}}\ C_{m_0} \\
& C_{m_q}\ C_{m_{\delta_e}}\ C_{D_0}\ C_{D_q}\ C_{D_{\delta_e}} \big]^T
\end{aligned} \quad \ldots (4)
$$

3.0 GENERATION OF SIMULATED FLIGHT DATA

A prerequisite for nonlinear estimation using neural network is the availability of an exhaustive dataset ranging through the entire operating regime of the system. A dataset which spans entire operation regime is essential for this work because algorithms can learn and interpolate only within the trained regime. The use of extrapolated values based on the available dataset may lead to erroneous estimates especially when the system is nonlinear. As the real-time data is unavailable for the considered stratospheric airship model, data generation becomes vital for the estimation of airship parameters. Dataset should provide a global picture of trim states of the system as a function of control variables. This is accomplished in this work by carrying out standard bifurcation analysis using MATCONT continuation subroutine in the MATLAB® environment.

The airship equation could be written as a set of nonlinear ordinary differential equations (as shown in Equation 1) of form,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \ldots (5)$$

where $\mathbf{x}$ is the vector of state variables, and $\mathbf{u}$ is the vector of control variables. For the considered airship model, $\mathbf{x} = [V, \alpha, \beta, p, q, r, \phi, \theta]^T$ and $\mathbf{u} = [\eta, \delta_e, \delta_a, \delta_r]^T$. For a fixed set of control variables, $\mathbf{u}$, the system of equations specified in Equation (5) could be integrated to study the time response of $\mathbf{x}$ for a chosen initial condition. But it is extremely cumbersome to repeat the procedure for infinite set of initial conditions and control variables, $\mathbf{u}$. Thus, the continuation algorithm is used to solve the above set of equations represented in Equation (5) as a function of single continuation parameter s chosen from $\mathbf{u}$, while the remaining control variables $\mathbf{p}$ are fixed at the initial values, thus the above equation could be equivalently written as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, s, \mathbf{p} = fixed) \quad \ldots (6)$$

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

This depicts the response of states to change in control parameters, and such procedure is called standard bifurcation analysis. This generates all possible steady-state solutions called trim values throughout the flight regime with δ_e, δ_a and δ_r as continuation parameters as shown in Figs 2, 3 and 4. Since engine thrust is almost maintained constant throughout the flight regime, η continuation is not taken into consideration. The feasibility of the trim values are also accounted by constraining the continuation parameters with the bounds shown in Appendix A.3.

The obtained trim datasets are then used as initial conditions for generating flight data by open-loop simulation using fourth-order Runge–Kutta method. Simulation of the mathematical model in turn requires stability and control derivatives which are approximately computed using missile DATCOM software[16]. Missile DATCOM is a semi-empirical approach to predict the aerodynamics of the aerial vehicles based on its configuration. The modes of airship should be excited to generate time signals for reliable estimation. However, in this case, the response from the computed trim shows low magnitude oscillations as bifurcation analysis gives only the qualitative picture of the system. This makes it redundant to further excite the system for simulation.

Each row of datasets contains input and output sets for training the modular neural network. The input set consists of α, β, $\frac{pb}{2V}$, $\frac{qc}{2V}$, $\frac{rb}{2V}$, δ_e, δ_a and δ_r. The output set consists of aerodynamic coefficient values that are C_L, C_D, C_Y, C_l, C_m and C_n. The rows of datasets which are generated from different initial conditions are no longer time dependent. This is evident from the fact that parameters to be estimated do not change with time. Thus, datasets generated are combined to create training dataset.

The values of aerodynamic coefficients in each row of datasets could be obtained from DATCOM values for simulated flight data. But in this case, aerodynamic coefficient values are obtained by first calculating the rate of change of states using centrally pivoted five-point algorithm specified in Appendix A.2. Then, equations of motion (Equation 1) are solved for aerodynamic coefficients assuming that the true values of parameters are not known, as in case of real flight data. The DATCOM values are only used for generating simulated flight data.

4.0 PARAMETER ESTIMATION METHOD

4.1 Modular neural network

A modular network network is an ANN that constitutes independent neural networks connected by some intermediary. Each independent neural network or module carries out a sub-task of the global task attainable by the entire network. The intermediary integrates the output of each module to produce the required output. Modular neural networks are used when a complex task can be performed by breaking it into smaller viable tasks[13]. In this work, modular neural networks are trained separately to predict each aerodynamic coefficient, and the sub-task of estimating nonlinear parameters from angle of attack is achieved by independent modules using FFNN. FFNN is chosen as modules because they are considered universal function approximators, i.e. any nonlinear function can be computed and learnt by the network[7].

A multi-layer FFNN consists of neurons that are ordered into layers. The first layer is called the input layer, the last layer is called the output layer, and the layers in between are called hidden layers. Nodes in consecutive layers are connected with each other, and each connection is assigned a connective weight. The number of hidden layers and the number of nodes

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

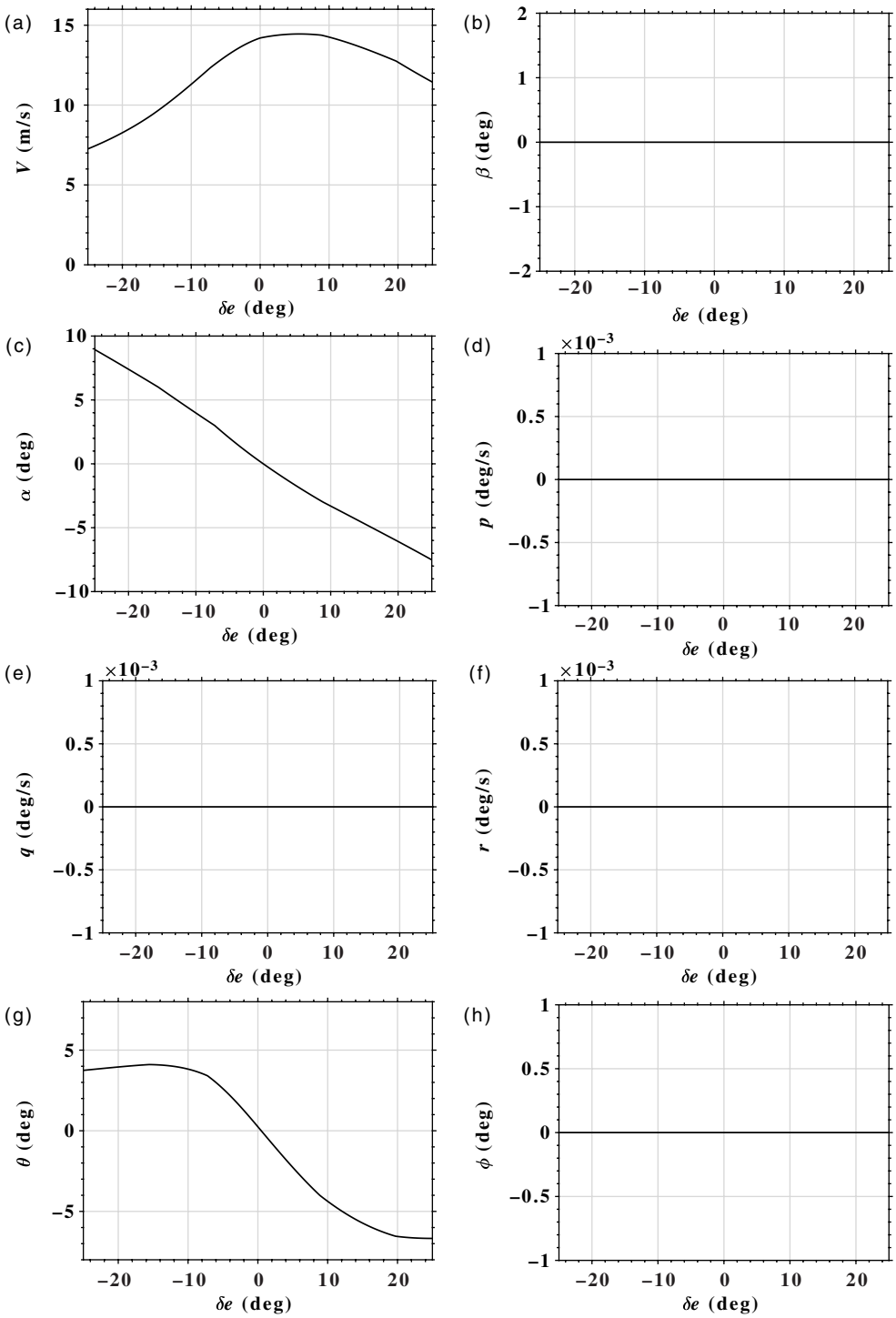


Figure 2. Elevator continuation ($\eta = 0.1836$, $\delta_a = \delta_r = 0$).

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

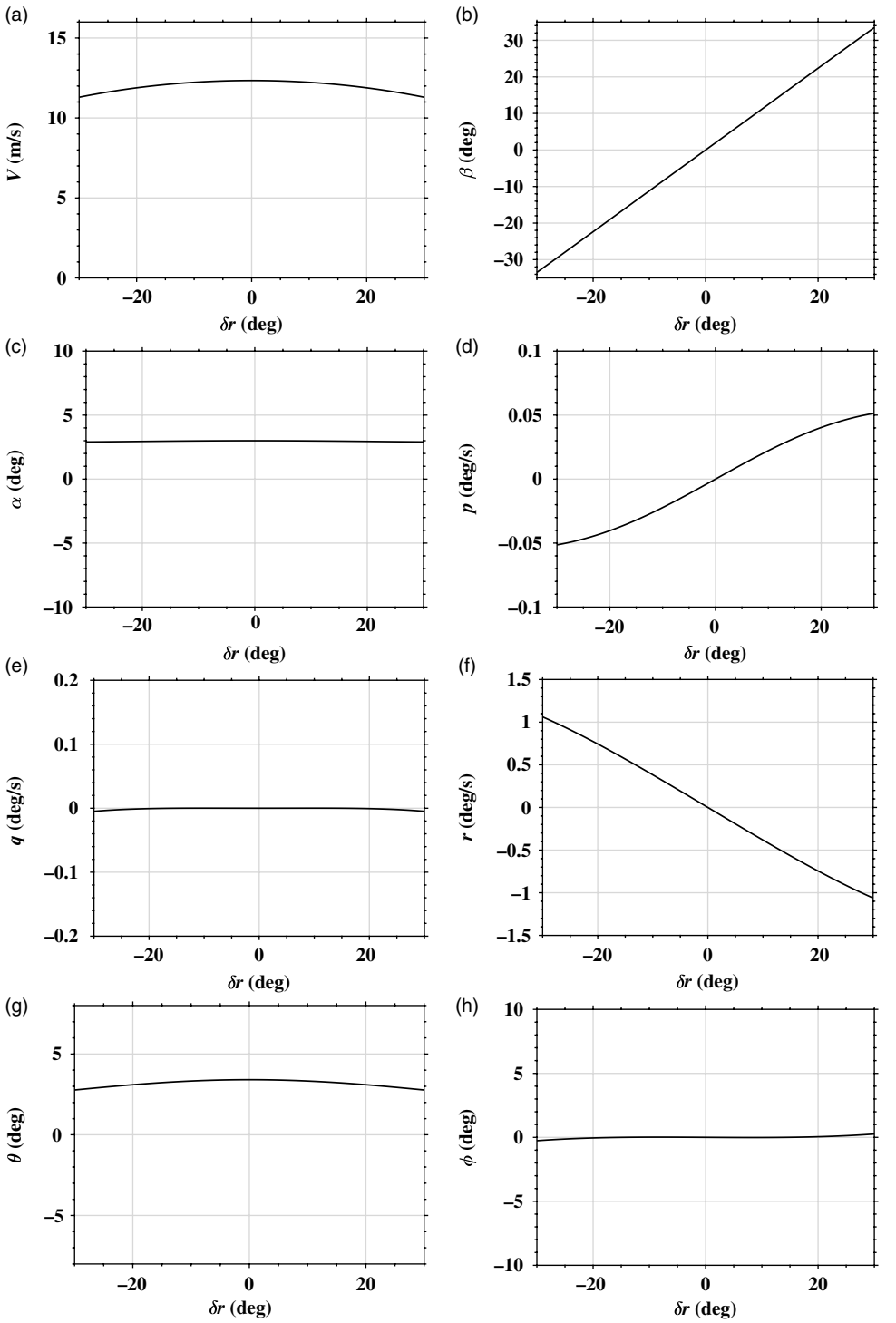


Figure 3. Rudder continuation (η = 0.1836, δ_e = −7.2135 *deg*, δ_a = 0).

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

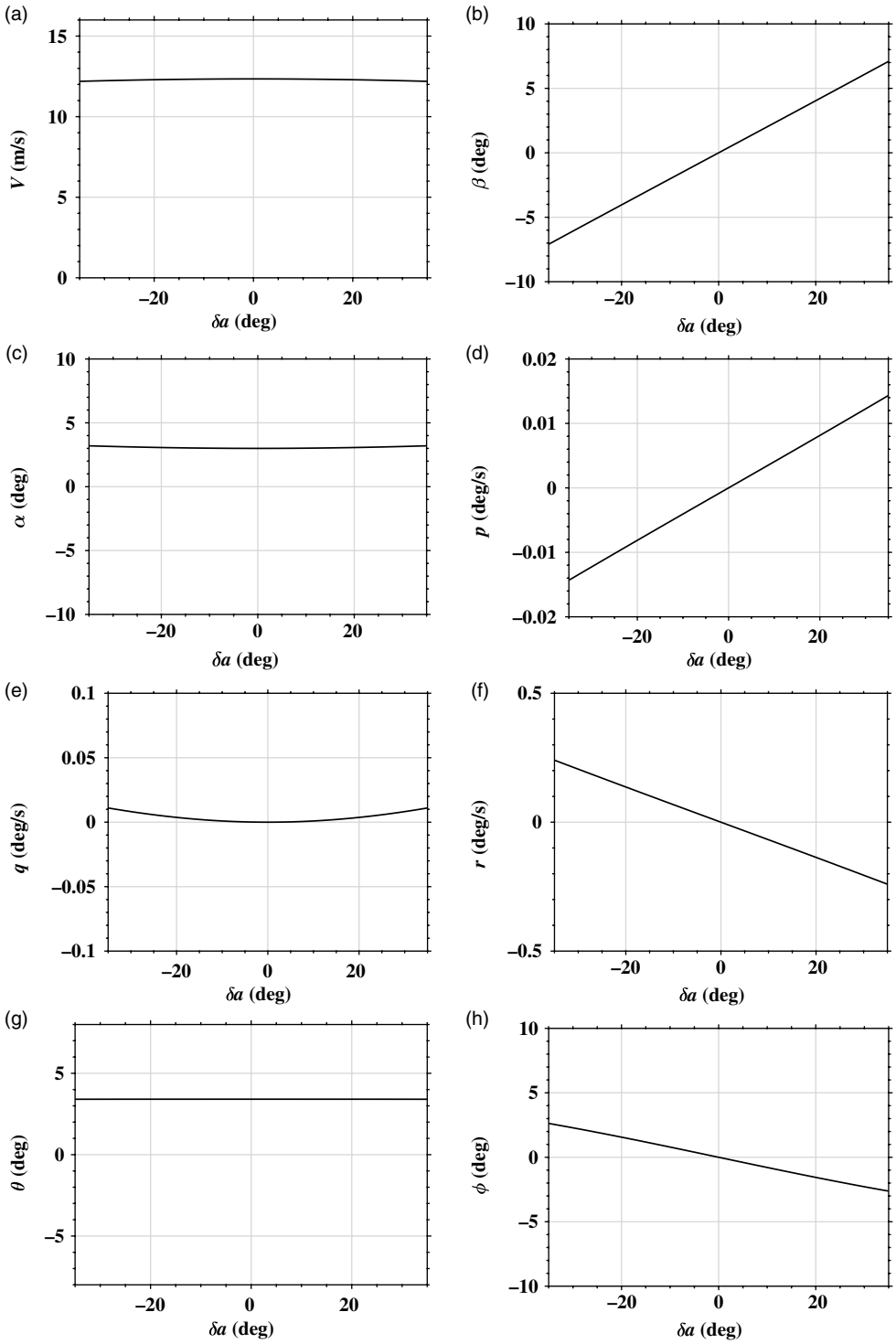


Figure 4. Aileron continuation ($\eta = 0.1836$, $\delta_e = -7.2135\ deg$, $\delta_r = 0$).

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

in each hidden layer is determined by the nonlinearity and complexity of the problem. Output of a layer is calculated by multiplying inputs with their respective weights. The resultants are then summed up, and the activation function is applied. Activation function provides nonlinear decision-making capability and is important for complex nonlinear functional mappings between input and output variables[17].

For parameter estimation of airship, the input variable to the module or FFNN is α and the response variable is unknown airship parameter from Equation (4). FFNN with one hidden layer was sufficient to capture nonlinearity in functions to be estimated. This is because one hidden layer with an arbitrarily large number of units suffices for the universal approximation property[17], and one hidden layer is used for aerospace applications in practice. Thus, each FFNN is composed of 1 node in input layer, 5 nodes in hidden layer and 1 node in output layer. Each node in one layer is connected to each node of the next layer (i.e. fully connected network). In this case, 'softmax' and 'linear' activation functions are used in the hidden layer and the output layer respectively of each module.

$$\begin{aligned} \text{Softmax activation, } \sigma(z)_j &= \frac{e^{z_j}}{\Sigma_{k=1}^{K} e^{z_k}} \text{ for } j = 1, .., K \\ \text{Linear activation, } f(x) &= ax \end{aligned} \quad \ldots (7)$$

where z is a vector (size=K) of the inputs to the output layer and j indexes the output units in softmax activation function. For linear activation function, x is an input and a is a constant.

These modules are then integrated with a dot product layer that is the final layer to predict an aerodynamic coefficient. Inputs for dot product layer are airship states, control inputs and outputs from modules. The number of modules and inputs are determined from Equation (3).

4.2 Training of network

For training, the network is initialised with weights that are small random numbers generated from uniform distribution. The output of the network is then compared with the available DATCOM values which were used for data generation. For comparison, mean squared error (MSE) is considered as the cost function. A cost function is a measure of performance of a neural network with respect to its given training sample and the expected output. The MSE for each iteration is defined by,

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}(\hat{Y}_i - Y_i)^2, \quad \ldots (8)$$

where $\hat{Y}$ and Y are the desired and the predicted output of the modular neural network, respectively, and n is the batch size. Batch size is the number of samples that are to be propagated through the network.

The error calculated is propagated backward to the network using a method called back propagation. Back-propagation algorithm is used to compute gradient of a multi-variable function. In neural network, it computes gradient of the cost function which is a function of network weights and biases and uses gradient-descent algorithm to update the weights and bias terms iteratively such that the cost function is minimised. In this case, Adam optimisation algorithm[18] is used instead of the classical stochastic gradient-descent procedure because it combines best properties of AdaGrad and RMSProp algorithms and can handle sparse gradients on noisy algorithms.

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

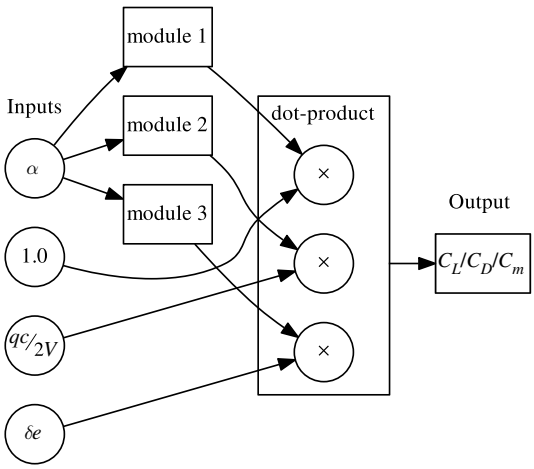


Figure 5. Network architecture for longitudinal coefficients.

The training algorithm is iterative in nature. In each iteration, error is calculated, and the weights are updated using the number of samples in batch size. When all the training samples are passed through the algorithm, an epoch is completed. An epoch is a measure of number of times the entire available training samples are used to update the weights. The training sessions are continued until the MSE in successive iterations is decreasing. The predicted value is acceptable only if MSE is less than the minimum specified value. To achieve the desired level of MSE, hyperparameters of the network are tuned. Hyperparameters are variables that are set before optimising the model parameters. It determines the structure and training properties of the network. Hyperparameters of neural network algorithm are learning rate, loss function, batch size, epoch, number of hidden layer(s), number of units in the hidden layer(s) and type of activation functions which are adjusted by trial and error.

4.3 Estimation of longitudinal parameters

The architecture of modular neural network for estimation of longitudinal parameters of the airship is shown in Fig. 5. Network was trained on generated data for C_L, C_D and C_m individually. Each module in the architecture is composed of 1 hidden layer with 5 nodes, as mentioned in the Subsection 4.1. Modules 1, 2 and 3 output C_{x_0}, C_{x_q} and $C_{x_{\delta_e}}$ parameters, respectively. The final layer computes dot product of the module outputs and airship states according to the following equation:

$$C_x = C_{x_0} + C_{x_q}\frac{qc}{2V} + C_{x_{\delta_e}}\delta_e \quad \ldots (9)$$

Once the modular neural network is trained satisfactorily to represent mapping of the input variables (α, $\frac{qc}{2V}$, δ_e) to the output variable (C_L or C_D or C_m), the network weights are fixed. Then, longitudinal parameters are estimated by passing on α values at each degree in the range $[-15°, 18°]$. For these α inputs, the output of modules 1, 2 and 3 are recorded and plotted to compare with DATCOM values.

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

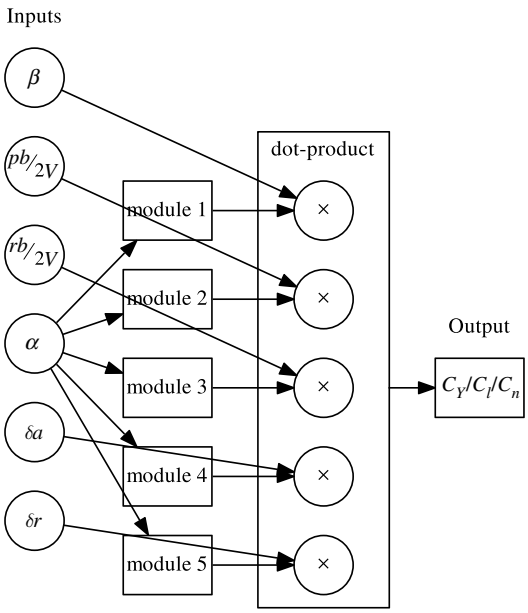


Figure 6. Network architecture for lateral coefficients.

4.4 Estimation of lateral parameters

The network architecture for estimating lateral parameters is shown in Fig. 6. Modular neural network is trained for the aerodynamic coefficients, C_Y, C_l and C_n, separately. Modules 1, 2, 3, 4 and 5 output C_{x_β}, C_{x_p}, C_{x_r}, $C_{x_{\delta_a}}$ and $C_{x_{\delta_r}}$ parameters, respectively. The dot product is computed in the final layer according to the following equation:

$$C_x = C_{x_\beta}\beta + C_{x_p}\frac{pb}{2V} + C_{x_r}\frac{rb}{2V} + C_{x_{\delta_a}}\delta_a + C_{x_{\delta_r}}\delta_r \quad \ldots (10)$$

After networks are trained, parameters are estimated by passing α values at each one degree, similar to the longitudinal case.

5.0 RESULTS AND DISCUSSIONS

Data generation is a crucial phase of parameter estimation, as it has direct implication on performance. In this work, bifurcation-based continuation approach is relied on to compute all possible trim points. These trims are then taken as initial conditions, and airship is simulated for 5,000 seconds with the step size of 1 second. Figure 7 shows one such time simulation with initial condition taken from elevator continuation. The state vector, $\mathbf{x}_i$, and control vector, $\mathbf{u}_i$, corresponding to the considered initial condition are given by:

$$\mathbf{x}_i = [12.3578, 0.0520, 0, 0, 0, 0, 0, 0.0593]^T$$
$$\mathbf{u}_i = [0.1836, -0.1252, 0, 0]^T \quad \ldots (11)$$

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

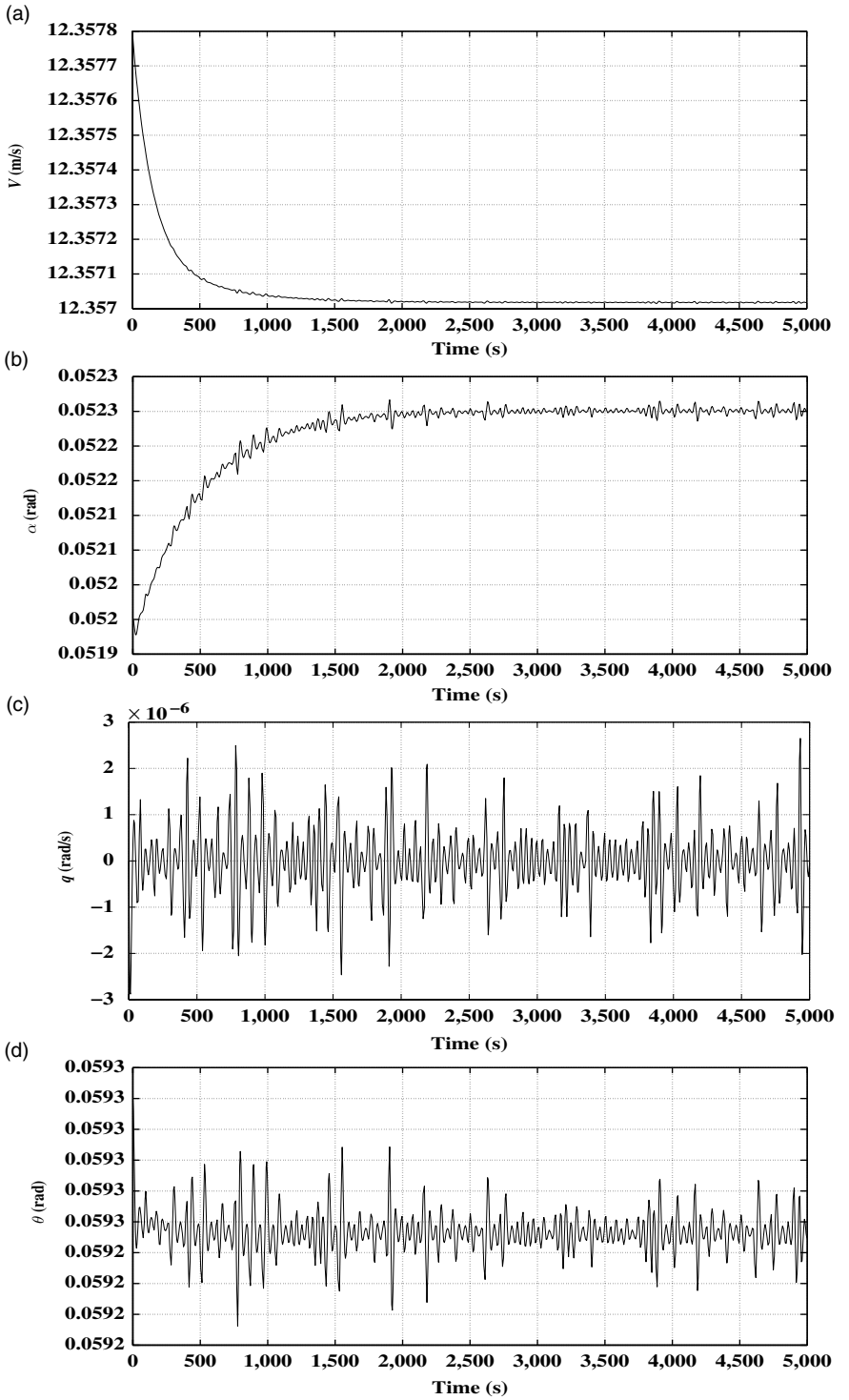


Figure 7. Time simulation plots.

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

As evident from Fig. 7, there are low magnitude oscillations in longitudinal state variables. This excludes further excitation of the trim states, proving to be of major advantage as consideration on the type of excitation could be ignored. As δ_a and δ_r associated with the chosen trims are zero, lateral directional state variables are also zero and are not shown in Fig. 7. Thus, this analysis results in an extensive dataset of 4.5 million points which is used to effectively train the proposed neural network. As the trims are obtained from bifurcation plots, it spans the entire regime of operation of airship. This work is not regime specific and could be conveniently extended to real flight data, if available.

The generated dataset is split into two subsets: i.e. 75% for training and 25% for validation. Hyperparameters of neural network are tuned to obtain least training error for the training set, and the performance of trained network is validated on the validation set. In this work, training epoch and batch size are 1,000 and 5,000, respectively. Also, the training parameters for Adam optimization, i.e. learning rate, exponential decay rate for the first moment estimates (β_1) and exponential decay rate for the second moment estimates (β_2) are 10^{-4}, 0.9 and 0.999, respectively. Modular neural network for longitudinal and lateral coefficients has 48 and 80 trainable parameters, respectively. In every iteration, error is transmitted to modules because there is no trainable parameter in the dot product layer. Each module has 16 trainable parameters to be adjusted such that the MSE for the entire network is minimum. After training modular neural networks for each aerodynamic coefficient, MSE values are found to be in the order of 10^{-6}. Although MSE values are found to be of the same order for each modular neural network, it is higher for lateral aerodynamic coefficients in contrast to networks for longitudinal aerodynamic coefficients. This is because complexity or the number of trainable parameters are higher in modular neural networks for lateral aerodynamic coefficients.

The neural network algorithm is implemented in Keras Python library which provides a clean and convenient way to create a range of deep learning models. The values of estimated and DATCOM parameters are plotted together to compare the performance of the proposed methodology. The DATCOM parameter values are considered as actual values in the plots because these values are used to generate simulated data which are then used for parameter estimation. In Figs 8, 9 and 10, the dashed lines (red) represent DATCOM or actual longitudinal parameter values and the solid lines (blue) represent estimated parameter values for α ranging from $[-15°, 18°]$. Similarly, Figs 11, 12 and 13 present lateral parameters of the airship with solid lines (blue) depicting the estimated parameter values and dashed lines (red) representing DATCOM or actual parameter values.

Since the aerodynamic model (Equation 3) is linear, classical estimation methods (mentioned in Section 1) such as least square could be used for estimation. But, it requires building different linear models for discrete values of α, and interpolation among these estimated values. However, the proposed nonlinear approach generates continuous plots of parameter values. It is worth mentioning that the fully connected FFNN trained on the same data has MSE values in the order of 10^{-4}. Modular neural network reduces the MSE significantly to 10^{-6} because it incorporates the mathematical form of the aerodynamic model in the network, whereas FFNN does not explicitly use aero modelling and rely entirely on a network to learn from data. Thus, the performance of modular neural network was found to be better than other neural network architectures and estimation algorithms.

As observed from the plots of stability and control derivatives, the estimated parameter values at both the ends slightly deviates from the actual value. This is predominantly attributed to insufficient data points in that regime. This evidently ascertains that the performance is greatly influenced by the available dataset or real flight data, if available. This is a limitation of

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

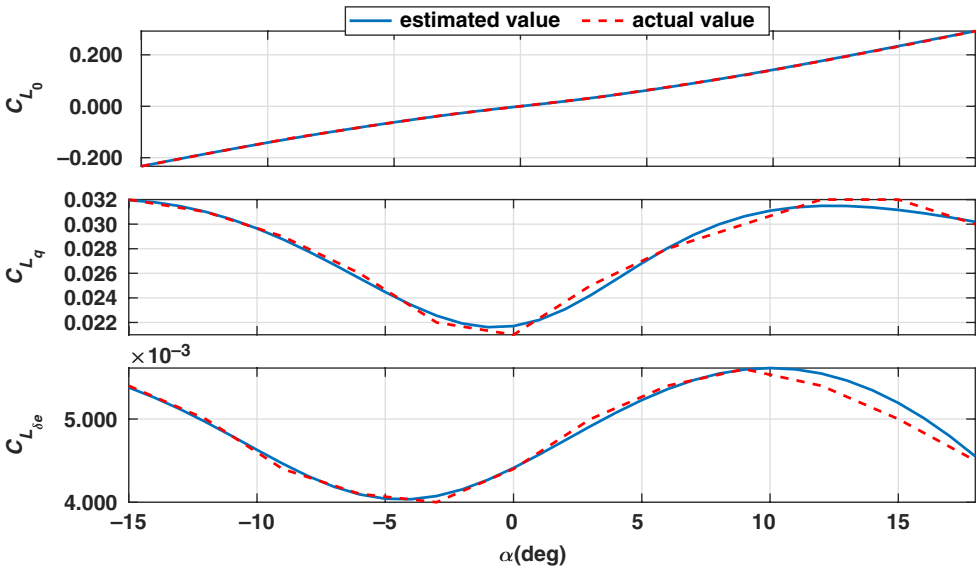


Figure 8. Derivatives of lift coefficient.

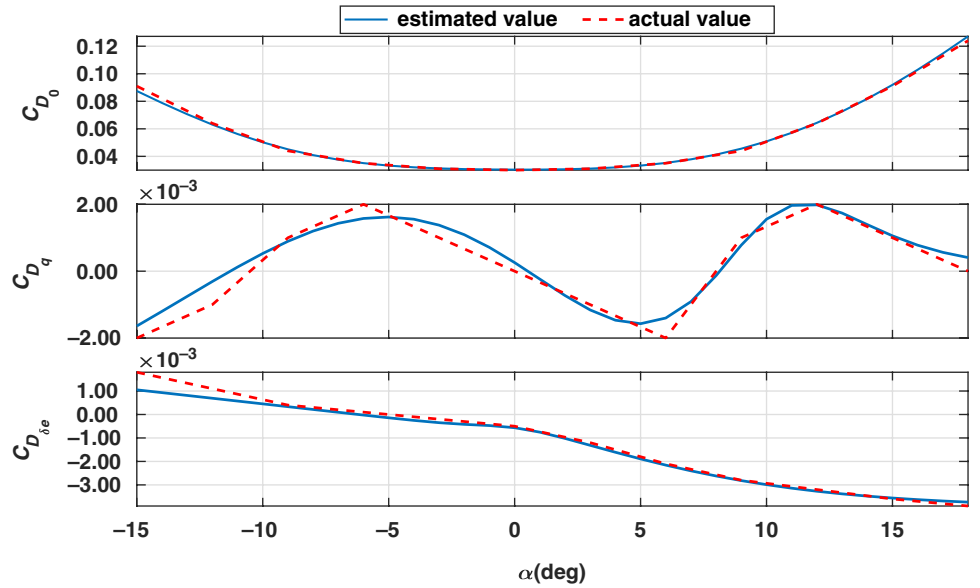


Figure 9. Derivatives of drag coefficient.

the methodology, as real flight data might not be available for critical flight conditions because of safety aspects. However, this limitation is not applicable to unmanned aerial vehicles like an airship.

The estimated values as inferred from the plots are consistent with the assumed parameter values (i.e. missile DATCOM parameter values) used to generate simulated data. This validates the proposed methodology for estimating parameters of airship using simulated data

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

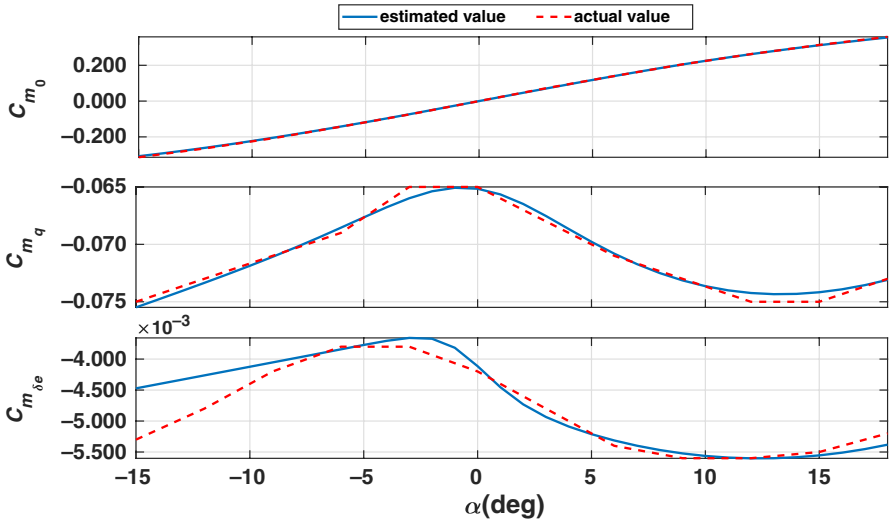


Figure 10. Derivatives of pitching moment coefficient.

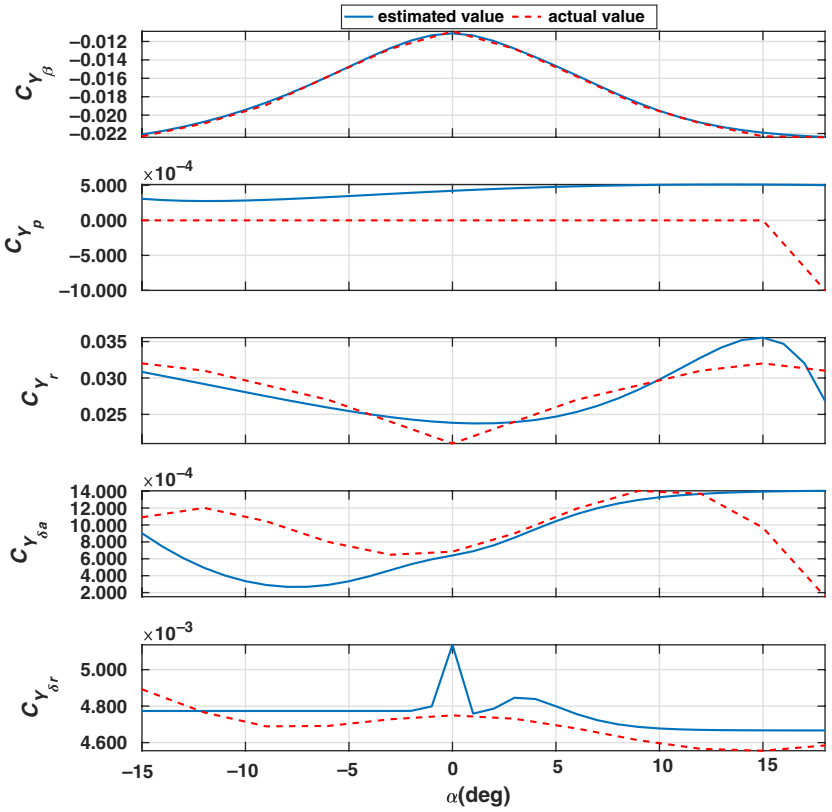


Figure 11. Derivatives of side force coefficient.

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

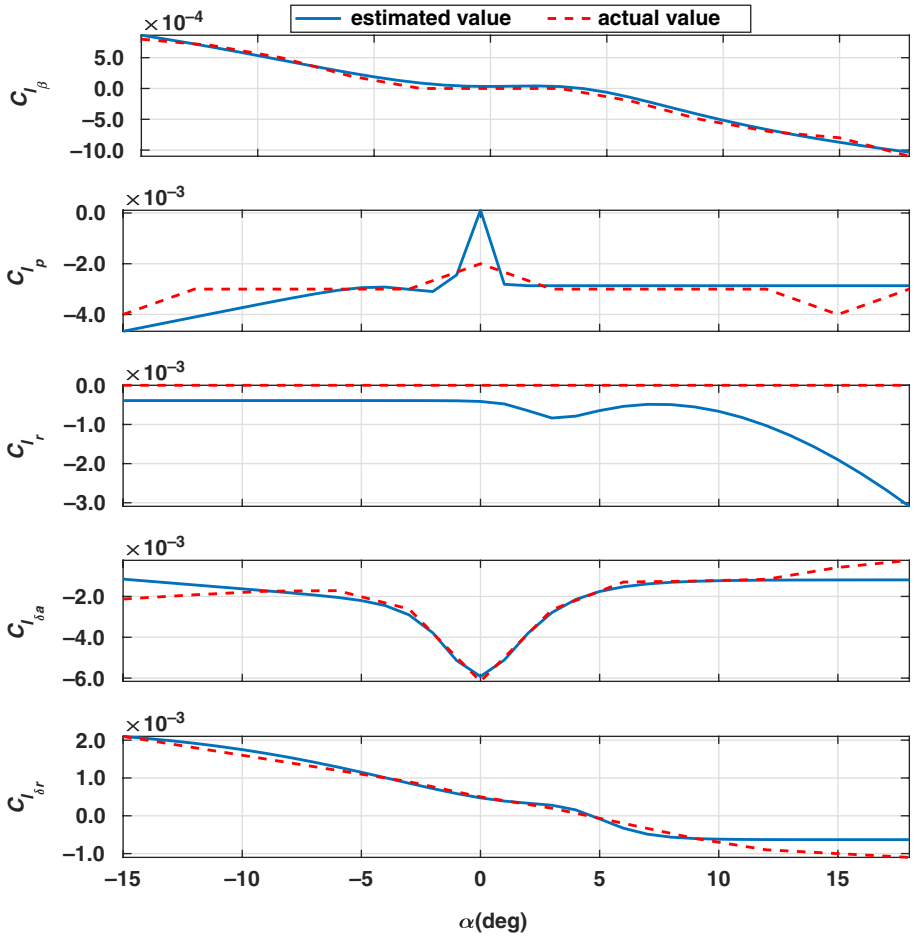


Figure 12. Derivatives of rolling moment coefficient.

and could be extended to real flight data of the airship without any change in neural network architecture. But, this might require addition of filters in the estimation procedure depending on the nature of measurement noise.

6.0 CONCLUSIONS

A numerical scheme for the estimation of stability and control derivatives in a completely nonlinear environment is proposed in this work by combining bifurcation analysis with modular neural network architecture. As the performance of neural network algorithms are highly dependent on the training dataset, data generation is meticulously carried out using bifurcation analysis which generates dataset over the entire operating regime. The preliminary estimates from missile DATCOM has been utilised for open-loop simulation, which further validates the proposed procedure. Though implemented on airship, this numerical approach

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

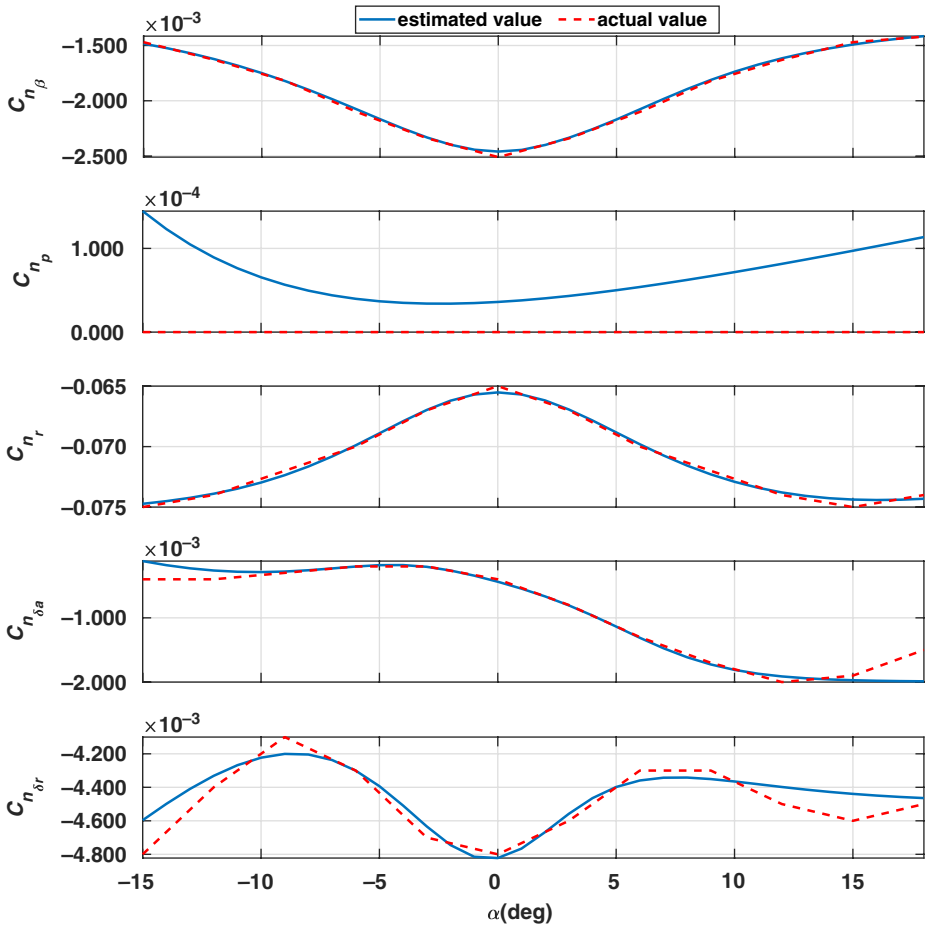


Figure 13. Derivatives of yawing moment coefficient.

could be extended to any aerial vehicle for its aero modelling. This methodology could also be augmented to accommodate real flight data in the estimation procedure.

REFERENCES

1. HAMEL, P.G. and JATEGAONKAR, R.V. "Evolution of flight vehicle system identification," *Journal of Aircraft*, 1996, **33**, (1), pp 9–28.
2. JATEGAONKAR, R.V. *Flight Vehicle System Identification: A Time-Domain Methodology*, Progress in Aeronautics and Astronautics, vol. 216, AIAA, 2015, Reston, VA.
3. PEYADA, N.K., SEN, A. and GHOSH, A.K. "Aerodynamic characterization of HANSA-3 aircraft using equation error, maximum likelihood and filter error methods," Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, 2008, pp 19–21.
4. YOUSSEF, H. and JUANG, J.C. "Estimation of aerodynamic coefficients using neural networks," AIAA Flight Simulation and Technologies Conference, Monterey, CA, 1993, pp 242–245.

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

5. BASAPPA, K. and JATEGAONKAR, R. "Aspects of feed forward neural network modeling and its application to lateral-directional flight data," *DLR IB-111-95a30*, 1995.
6. RAOL, J. and JATEGAONKAR, R. "Aircraft parameter estimation using recurrent neural networks-A critical appraisal," 20th Atmospheric Flight Mechanics Conference, Maryland, 1995, pp 119–128.
7. HORNIK, K., STINCHCOMBE, M. and WHITE, H. "Multilayer feedforward networks are universal approximators," *Neural Networks*, 1989, **2**, (5), pp 359–366.
8. RAISINGHANI, S., GHOSH, A. and KALRA, P. "Two new techniques for aircraft parameter estimation using neural networks," *The Aeronautical Journal*, 1998, **102**, (1011), pp 25–30.
9. GHOSH, A., RAISINGHANI, S. and KHUBCHANDANI, S. "Estimation of aircraft lateral-directional parameters using neural networks," *Journal of Aircraft*, 1998, **35**, (6), pp 876–881.
10. GHOSH, A. and RAISINGHANI, S. "Parameter estimation from flight data of an unstable aircraft using neural networks," *Journal of Aircraft*, 2002, **39**, (5), pp 892–894.
11. PEYADA, N. and GHOSH, A. "Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss–Newton method," *The Aeronautical Journal*, 2009, **113**, (1142), pp 243–252.
12. KUMAR, R. and GHOSH, A. "Nonlinear longitudinal aerodynamic modeling using neural Gauss–Newton Method," *Journal of Aircraft*, 2011, **48**, (5), pp 1809–1813.
13. DING, Y., FENG, Q., WANG, T. and FU, X. "A modular neural network architecture with concept," *Neurocomputing*, 2014, **125**, pp 3–6.
14. GOBIHA, D. and SINHA, N.K. "Autonomous maneuvering of a stratospheric airship," Indian Control Conference, IEEE, Kanpur, India, 2018, pp 318–323.
15. TIWARI, A., VORA, A. and SINHA, N.K. "Airship trim and stability analysis using bifurcation techniques," 7th International Conference on Mechanical and Aerospace Engineering, IEEE, London, 2016, pp 471–475.
16. RANA, V., KUMAR, A., SINHA, N., PAL, A. and SATI, S. "Configuration analysis of stratospheric airship," Symposium on Applied Aerodynamics and Design of Aerospace Vehicles, VSSC, Thiruvananthapuram, India, 2015.
17. SVOZIL, D., KVASNICKA, V. and POSPICHAL, J. "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, 1997, **39**, (1), pp 43–62.
18. KINGMA, D.P. and BA, J. "Adam: A method for stochastic optimization," 3rd International Conference for Learning Representations, 2014.
19. RAOL, J.R., GIRIJA, G. and SINGH, J. *Modelling and Parameter Estimation of Dynamic Systems*, 1st ed. The Institute of Engineering and Technology, 2004, London.

APPENDIX

A.1 Geometric and other data for the considered airship

Parameter	Values	Parameter	Values
m	23,145kg	J_x	1.74×10^7kgm^2
T_m	6,000N	J_y	2.45×10^8kgm^2
S	4,748m^2	J_z	2.45×10^8kgm^2
b	68.9m	J_{xz}	1.92×10^6kgm^2
c	68.9m	m_x	2.50×10^4kg
b_z	16.43m	m_y	4.30×10^4kg
d_z	14.67m	m_z	4.30×10^4kg

A.2 Centrally pivoted five-point algorithm

This is a numerical differentiation scheme to obtain differentiated values of the variables. Past values of data y are denoted by y_1,y_2,.., and future values are denoted by y_{-1},y_{-2},.., with τ as the sampling interval[19]. Then, the derivative $\dot{y}$ of y is given by:

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press

$$\text{Pivotal point, } \dot{y} = \frac{1}{12\tau}\left[-8y_1 + y_2 - y_{-2} + 8y_{-1}\right]$$
$$\text{Initial point, } \dot{y} = \frac{1}{12\tau}\left[-25y_0 + 48y_{-1} - 36y_{-2} + 16y_{-3} - 3y_{-4}\right]$$
$$\text{Second point, } \dot{y} = \frac{1}{12\tau}\left[-3y_1 - 10y_0 + 18y_{-1} - 6y_{-2} + y_{-3}\right] \quad \ldots (12)$$
$$\text{Penultimate point, } \dot{y} = \frac{1}{12\tau}\left[3y_{-1} + 10y_0 - 18y_1 + 6y_2 - y_3\right]$$
$$\text{Final point, } \dot{y} = \frac{1}{12\tau}\left[25y_0 - 48y_1 + 36y_2 - 16y_3 + 3y_4\right]$$

A.3 Constraints in actuator dynamics

Control surface	Position limits (deg)	Rate limits (deg/s)
Elevator	$(-25, 25)$	$(-40, 40)$
Aileron	$(-35, 35)$	$(-100, 100)$
Rudder	$(-30, 30)$	$(-82, 82)$

https://doi.org/10.1017/aer.2019.125 Published online by Cambridge University Press