Identification and Validation of the Cessna Citation X Longitudinal Aerodynamic Coefficients in Stall Conditions using Multi-Layer Perceptrons and Recurrent Neural Networks

Yvan TONDJI[1], Mouhamadou WADE[1], Georges GHAZI[1],
Ruxandra Mihaela BOTEZ*,[1]

*Corresponding author
[1]LARCASE, Department of Automated Production Engineering,
École de Technologie Supérieure, University of Quebec,
1100 Notre Dame West, Montréal Quebec, Canada, H3C 1K3,
yvanschool@gmail.com, georges.ghazi@etsmtl.ca, Ruxandra.Botez@etsmtl.ca*

DOI: 10.13111/2066-8201.2022.14.2.9

Received: 06 February 2022/ Accepted: 06 May 2022/ Published: June 2022

Abstract: *The increased number of accidents in general aviation due to loss of aircraft control has necessitated the development of accurate aerodynamic airplane models. These models should indicate the linear variations of aerodynamic coefficients in steady flight and the highly nonlinear variations of the aerodynamic coefficients due to stall and post-stall conditions. This paper presents a detailed methodology to model the lift, drag, and pitching moment aerodynamic coefficients in the stall regime, using Neural Networks (NN). A system identification technique was used to develop aerodynamic coefficients models from flight data. These data were gathered from a level-D Research Aircraft Flight Simulator (RAFS) that was used to execute the stall maneuvers. Multilayer Perceptrons and Recurrent Neural Networks were used to learn from flight data and find correlations between aerodynamic coefficients and flight parameters. This methodology is employed in here to optimize neural network structures and find ideal hyperparameters: training algorithms and activation functions used to learn the data. The developed stall aerodynamic models were successfully validated by comparing the lift, drag, and pitching moment aerodynamic coefficients predicted for given pilot inputs with experimental data obtained from the Cessna Citation X RAFS for the same pilot inputs.*

Key Words: *Dynamic stall, aerodynamics, modeling, artificial intelligence, neural networks, simulation*

1. INTRODUCTION

The need to improve aircraft safety has been one of the major concerns in the aviation industry [1]. Over the last few years, the loss of control in-flight, one of the primary causes of flight accidents, has been addressed from different perspectives, one of them is using flight simulators to teach pilots how to execute stall recovering maneuvers to developing high-fidelity stall models [2].

A high-fidelity stall model would enable the development of new control strategies [3], that would extend the operational envelope and capabilities of the next generation of aircraft. In fact, flying near-stall conditions could help increase an aircraft's lift capacity, reduce landing distances, and enable safe recovery from stall in case of emergency maneuvers. Today, the most commonly used techniques for modeling aerodynamic coefficients in stall conditions can be classified into the following groups: semi-empirical methods [4], Computational Fluid Dynamics (CFD) methods [5], experimental techniques [6], and system identification techniques [7].

Semi-empirical methods are based on approximations established based on experimental data obtained from flight or wind tunnel tests.

They have the advantage of allowing the rapid modeling of a wide range of aerodynamic coefficients variations with angle of attack. However, in the case of stall modelling, the use of semi-empirical methods has some limitations due to the highly non-linear nature of the stall phenomenon and the quality of the existing database, which might not be sufficiently representative of the system under test [8]. CFD methods are based on solving the fundamental equations of fluid dynamics.

Codes such as Ansys CFX [9] and Ansys Fluent [10], for instance, are based on the resolution of the Navier-Stokes equations and represent practical tools for aerodynamic modeling. However, the accuracy of CFD methods is limited by the assumptions of the mathematical equations used by their algorithms.

Experimental methods based on wind tunnel tests overcome the weaknesses of CFD methods by considering a “real” fluid to reproduce flight conditions over a wide range of the envelope of an aircraft.

Many experimental methods such as pressure measurements [11], Laser Sheet Visualization (LSV) [6], and Time-Resolved Particle Image Velocimetry (TR-PIV) technique [12] have been used to develop stall models by visualizing and characterizing the airflow during wind tunnel experiments.

However, the results obtained are highly dependent on the scale of the model, and require costly and time-consuming experiments [13].

The system identification methods we have chosen to use in our study aim at designing a mathematical model of the aerodynamic coefficients from flight test, under stall conditions.

System identification techniques combine the advantages of the previously presented methods and overcome their weaknesses. They can provide an accurate model based on real flight test data learning.

These types of methods require a large amount of data and a powerful tool to learn the data scheme.

At our LARCASE laboratory, we have used neural network techniques for the morphing wing tip modeling in the CRIAQ MDO 505 project [14], for the wind tunnel calibration [15], for the modeling of the Bell-427 helicopter (Ref. 3) [16], F/A-18 [17,18] and Cessna Citation X Engine Model [19].

The main objective of this paper is to develop a methodology to predict the aerodynamic coefficients of the Cessna Citation X aircraft in stall conditions using neural networks.

The aerodynamic coefficients were estimated from data obtained from flight tests performed on a level-D Cessna Citation X Research Aircraft Flight Simulator (RAFS) designed and manufactured by CAE Inc.

According to the Federal Aviation Administration (FAA), level D is the highest qualification level for flight dynamics.

Therefore, it was assumed that the RAFS was accurate enough to be considered as a test aircraft.

All the collected data was similar to the actual data that would be collected during a flight test on an actual Cessna Citation X Aircraft.

a) Cessna Citation X Aircraft

b) Research Aircraft Flight Simulator (RAFS)

Fig. 1 Cessna Citation X Business Aircraft and its corresponding Level D Research Aircraft Flight Simulator (RAFS)

The rest of this article is organized as follows: Section 2 presents the methodology, including the data acquisition procedure from the RAFS and the data preprocessing of flight parameters, so that they can be used to compute aerodynamic coefficients. Next, we present the method used to select the neural network's inputs and outputs, and its hyperparameters, including the training algorithms and the activation functions. Finally, the numerical results and their comparisons with experimental data obtained from the Cessna Citation X RAFS are presented in Section 3.

2. METHODOLOGY

This section presents the methodology developed at the LARCASE for modeling the lift, drag, and pitching moment aerodynamic coefficients in stall conditions using system identification techniques. The identification was made through neural network optimization using Cessna Citation X flight simulator data.

2.A Flight test procedure and data gathering

Several flight tests were conducted with the Cessna Citation X Level D flight simulator to collect data and prepare these data for the identification process. As shown in Fig. 2 these flight tests were designed to replicate different stall maneuvers, and they included several steps.

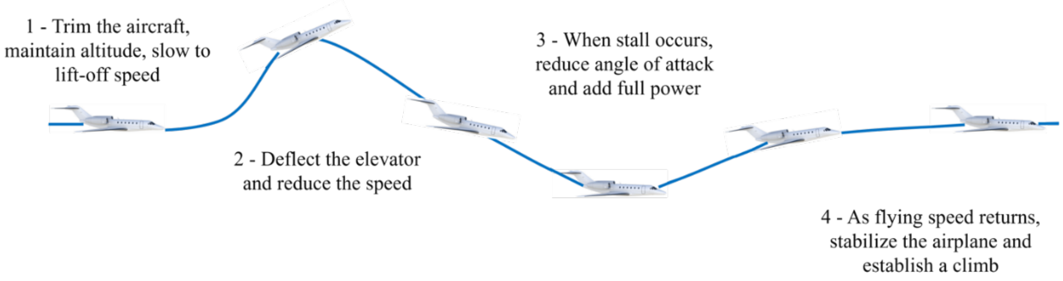
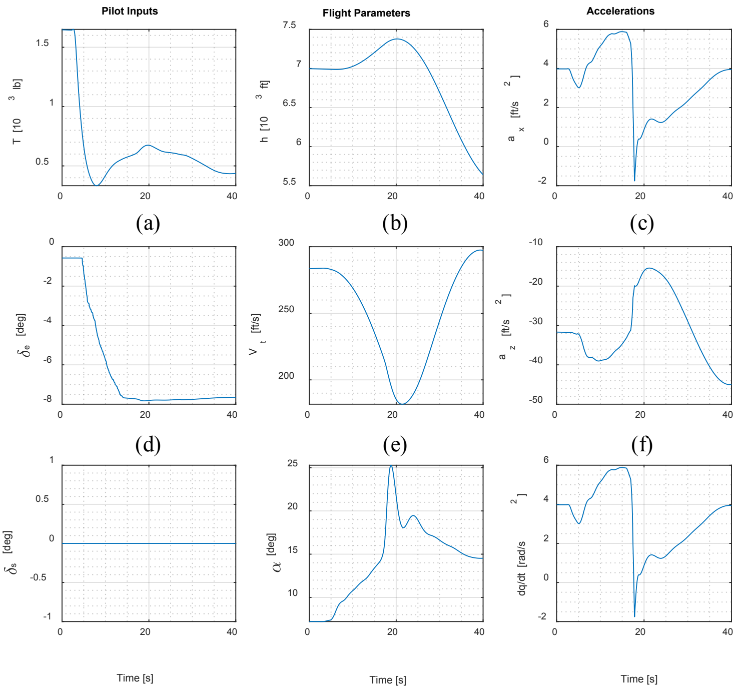

1 - Trim the aircraft, maintain altitude, slow to lift-off speed
2 - Deflect the elevator and reduce the speed
3 - When stall occurs, reduce angle of attack and add full power
4 - As flying speed returns, stabilize the airplane and establish a climb

Fig. 2 Recovery from stall flight test procedure illustration

First, the aircraft was trimmed for a stable flight at a given altitude and airspeed (1). For this purpose, the altitude was maintained using the autopilot's altitude hold mode, while the airspeed was stabilized manually by adjusting the throttle position. Then, to stall the aircraft, the pilot disengaged the autopilot and pulled back on the control column to deflect the elevators while gradually reducing thrust (2). The aircraft's airspeed decreased, and the angle of attack increased until the stall occurred. The aircraft was maintained in stall conditions as long as possible by controlling the elevators in order to better observe the dynamic stall. Finally, a stall recovering procedure was initiated. The engine thrust was increased to gain airspeed, and the elevators were manually controlled to re-stabilize the aircraft (3), which returned to normal level flight conditions (4).

For each flight test, pilot inputs such as elevator angle deflection δ_e and slats angle deflection δ_s, and flight parameters such as altitude h, true airspeed V_t, angle of attack α and finally aircraft longitudinal and vertical accelerations a_x and a_z were recorded at a sampling rate of 25 Hz. Fig. 3 shows an example of recorded data on the RAFS for a flight case carried out at an altitude of 7000 ft, a Mach number of 0.2, and with slats fully retracted (Fig 3.g). In this example, the aircraft was trimmed with an elevator angle deflection of $\delta_e = -0.6^o$ (Fig 3.d), and at a true airspeed $V_t = 284\,ft/s$ (Fig 3.e). At about 3 s, the autopilot was disengaged, the thrust was reduced (Fig 3.a) by adjusting the throttle command. Then, the elevator was deflected (Fig 3.d) to pitch the aircraft up. Consequently, the angle of attack immediately increased until reaching its stall value α_{stall} (Fig 3.h) at about 17 s, leading to a significant reduction of the lift force and, therefore, a drop in vertical acceleration (Fig 3.f). We can also observe the drastic change in vertical and horizontal accelerations a_z and a_x (Fig 3.f and Fig 3.c) at about 17 s, reflecting the significant increase in the drag that happens when stall occurs. Finally, these drops in longitudinal accelerations result in a significant loss in altitude (Fig 3.b). The procedure described in Fig. 3 was replicated 33 times with the Cessna Citation X RAFS for different flight scenarios by varying initial altitudes and slat configurations. Fifteen flight tests were conducted for cases with a slat-in (i.e., retracted) configuration at altitudes ranging from 5000 to 50,000 ft, while 18 flight tests were conducted at altitudes ranging from 15,000 to 50,000 ft for cases with a slat-out (i.e., extended) configuration. When slats are out, they increase the wing's camber and, therefore, change the wing shape. Consequently, slats may have the effect of delaying the stall phenomenon [20].



(a) (b) (c)

(d) (e) (f)

Fig. 3 Example of data recorded for a flight test at 7000 ft, Mach 0.20, and slats δ_s retracted

2.B Data processing and aerodynamic coefficients' determination

The aerodynamic coefficients are not measurable during the flight test. Therefore, once all the flight tests were completed, the next step was to process the measurable flight parameters to estimate the aerodynamic coefficients of the aircraft and then to create a database to prepare for the neural network learning process.

The lift, drag and pitching moment aerodynamic coefficients, expressed in the stability axis, are given by the following equations [21]:

$$CL_s = CL_b \cos(\alpha) - CD_b \sin(\alpha) \tag{1}$$

$$CD_s = CD_b \cos(\alpha) + CL_b \sin(\alpha) \tag{2}$$

$$Cm_s = Cm_b - CD_b z_{cg} - CL_b x_{cg} \tag{3}$$

where $\{x_{cg}, z_{cg}\}$ are the longitudinal and vertical distances, respectively, between the aircraft center of gravity and its aerodynamic center.

CL_b, CD_b and Cm_b are the lift, drag and pitching moment coefficients, respectively, expressed in the aircraft body axis.

The aerodynamic coefficients expressed in the body axis are given by the following expressions:

$$CL_b = \frac{ma_z - T_z}{1/2\rho V_T^2 S_w} \tag{4}$$

$$CD_b = \frac{ma_x - T_x}{1/2\rho V_T^2 S_w} \tag{5}$$

$$Cm_b = \frac{I_{yy}\dot{q} - T_x z_{eng} - T_z x_{eng}}{1/2\rho V_T^2 S_w c_w} \tag{6}$$

where ρ is the air density, $\{T_x, T_z\}$ are the components of the engine thrust, I_{yy} is the aircraft moment of inertia about the lateral axis, S_w is the wing reference area, c_w is the mean aerodynamic chord of the wing, and a_x and a_z are the longitudinal and vertical accelerations of the aircraft, respectively.

Before neural network training, a good practice is to normalize the data. The objective is to ensure that all the neural network's input and output parameters have the same scale and are all centered around zero.

Consequently, the difference in data magnitude will not affect the training results, as only the correlation between the data will be considered during the training process. For this purpose, the input and output parameters (defined in **section 2.c.a**), were normalized using the following equation:

$$Data_{k\,(normalized)} = \frac{Data_k - \mu}{\sigma} \tag{7}$$

where $Data_k$ is the k^{th} value of the considered training data set, μ is the mean of the whole data set, and σ is the standard deviation of the data set. Fig. 4 and Fig. 5 show the normalized aerodynamic coefficients estimated from flight test data for the Cessna Citation X RAFS, with slat-in and slat-out configurations, respectively.

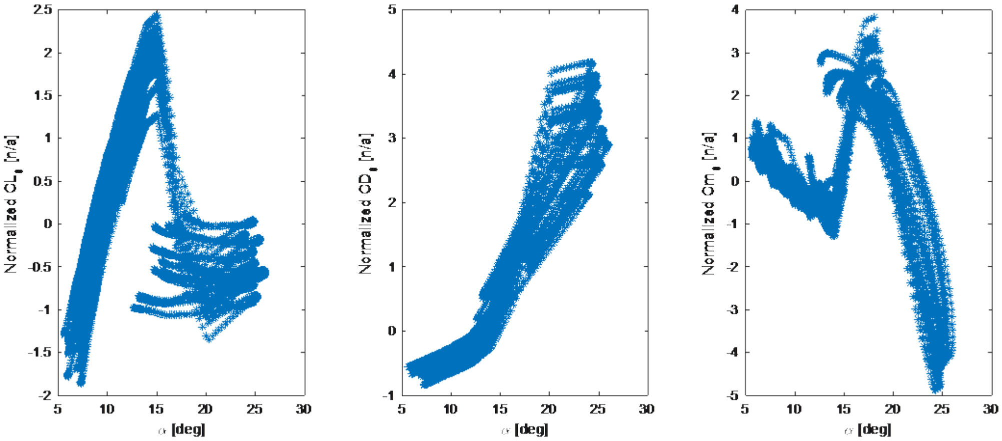


Fig. 4 Normalized aerodynamic coefficients' estimation from flight test data obtained from the Cessna Citation X RAFS (slat-in)

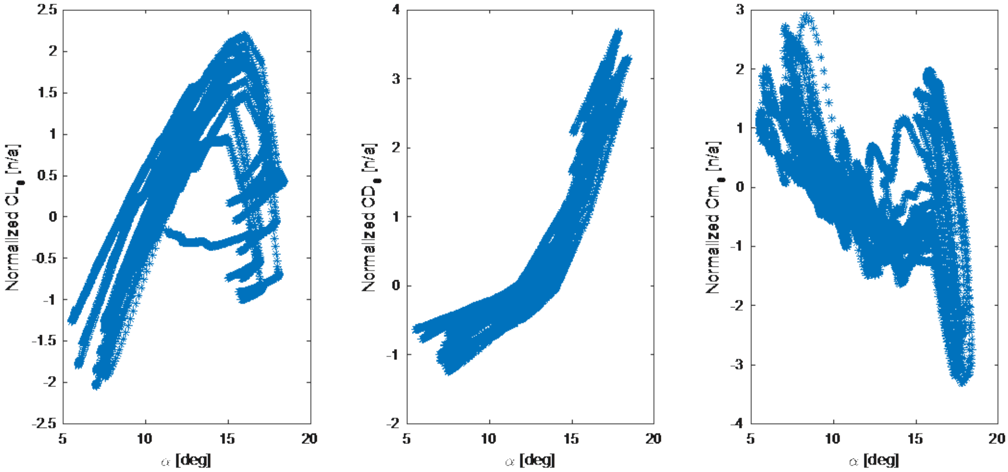


Fig. 5 Normalized aerodynamic coefficients' estimation from flight test data obtained from the Cessna Citation X RAFS (slat-out)

2.C Neural network modeling

a) Choice of neural networks inputs

The first step in defining the neural networks was the determination of the parameters that correlate significantly with the aerodynamic coefficients. These parameters were then used as inputs for the neural network models. Based on the Buckingham Pi's theorem [20] applied to the dimensional aerodynamic analysis, the aerodynamic coefficients depend on the following variables: the angle attack α, its derivative $\dot{\alpha}$, the Mach M and Reynolds numbers R_e. However, as the Reynolds number is not directly measurable in the flight simulator, it has been approximated using to equation (8):

$$R_e = \frac{\rho V_{TAS} c}{\mu} = \frac{c}{\mu} * \rho V_{TAS} = \frac{2c}{\mu} * \frac{Q}{V_{TAS}} \tag{8}$$

where ρ is the density of the air and μ is the dynamic viscosity of air.

Thus, the variables considered as input parameters are the angle of attack α, its derivative with respect to the time $\dot{\alpha}$, the Mach number M, the True Air Speed V_{TAS} and the dynamic pressure Q, resulting from Buckingham Pi's theorem analysis.

Additional variables, such as the elevator angle deflection δ_e, the slat angle deflection δ_s and the stabilizer angle δ_{stab} were also considered as inputs, because these surfaces are used for longitudinal dynamic control. The aircraft center of gravity position $\{x_{cg}, z_{cg}\}$, was considered, as the pitching moment aerodynamic coefficient is dependent on it. The pitch rate q may also affect the aerodynamic coefficients in stall conditions [22], and finally the total engine thrust T was taken into account because the wing airflow may also be affected by the air coming from the engines.

b) Selection of the Neural Network type

Two types of Neural Networks (NNs) were evaluated in this research: the Multilayer perceptron (MLP) and the Recurrent neural network (RNN). MLPs have been widely used to solve regression and function approximation problems in the literature. Therefore, this type of neural network is well suited for predicting aerodynamic coefficients from a given data set [7,21]. Nevertheless, RNNs have also been considered in this paper as they have been effective in designing models from time series data by using information regarding previous states during the network learning process [7,23].

The fundamental element of a neural network, whatever its type, is the artificial neuron (or "perceptron").

Neural networks are composed of neurons organized in layers and linked together by synaptic weights. Fig. 6 presents a typical architecture of a perceptron.

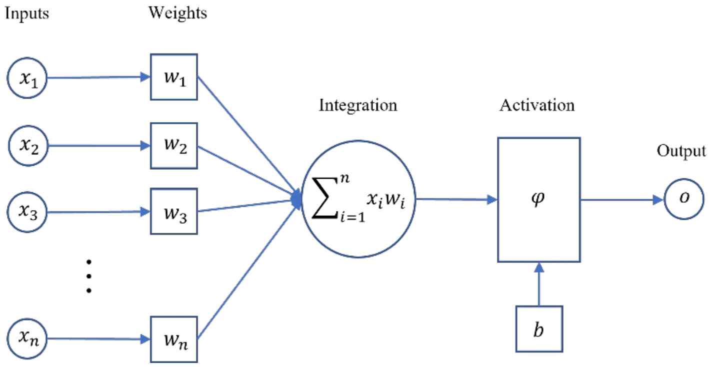


Fig. 6 Architecture of a perceptron

Therefore, as shown in Fig. 6, operating principle of a neuron is essentially composed of two calculation steps.

First, the input signal of the neuron $X = [x_1, x_2, \dots, x_n]$ is multiplied by its corresponding weights $W = [w_1, w_2, \dots, w_n]$, summed up, and then added to a scalar called the bias b, which is used as a "decision threshold".

Secondly, the resulting value is fed into a transfer function called the "activation function", which is used to determine whether the neuron should be activated or not [23, 24]. As a result, the output o of a neuron is given by the following equation:

$$o = \varphi\left(\sum_{i=1}^{n} x_i w_i + b\right) \tag{9}$$

MultiLayer Perceptron (MLP)

The MLP is one of the simplest forms of neural networks and is composed of several layers of perceptron's, called the hidden layers. A representation of a MLP is given in Fig. 7, where each hidden layer is associated with a to vector, and the outputs of the neurons in one layer are used as inputs for the next layer. All neurons of the same layer share the same input vector. The predicted output $\hat{o}$ of a MLP can be computed according to Eq. (10) [24]:

$$\hat{o} = \varphi_m\left(\sum_{k=1}^{k=n_m} w_{m,k} \times \ldots \times \varphi_2\left[\sum_{i=1}^{i=n_2} w_{2,i} \times \varphi_1\left(x_{1,j}w_{1,j} + b_{1,j}\right) + b_{2,i}\right] + b_{m,k}\right) \qquad (10)$$

where X is the input vector and m is the number of network layers, φ_i and n_i are the activation function and the number of neurons, respectively, in the layer i. $W_{i,j}$ and $b_{i,j}$ are the weights and bias, respectively, of the j^{th} neuron of the layer i.

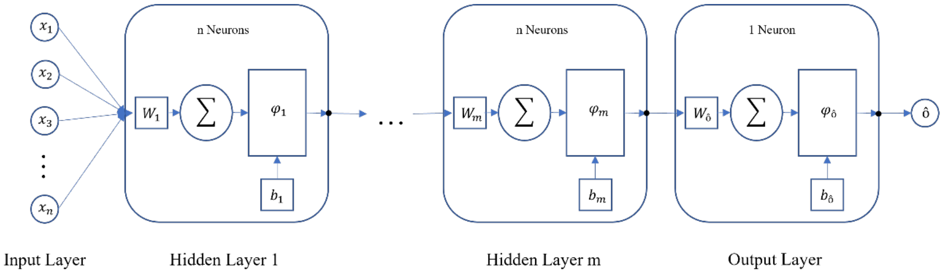


Fig. 7 Graphical representation of an MLP Neural Network

Recurrent Neural Networks (RNNs)

As their name suggests, RNNs are networks that have recurrent interconnections. The idea behind this architecture is to preserve the neurons' information over time. RNNs are often used for processing time-series signals, such as voice or semantic analysis of videos or sentences. They have also demonstrated their ability to learn the behavior of complex dynamic systems, such as the behavior of an aircraft at high angles of attack [7]. Aerodynamic coefficients are in some way presented as a time series signal, where previous states determine the behavior of a future state. A neural network model predicting aerodynamic coefficient variations over time should consider the dynamic nature of the longitudinal behavior of an aircraft. One of the best types of network that meet this criterion is the Elman Neural Network (ENN) proposed by Elman in [25], and which is defined as a feed-forward network with additional memory neurons (called the "context layer") and local feedback. Fig. 8 shows a graphical representation of an Elman Neural Network with one-step delay and a hidden layer h.

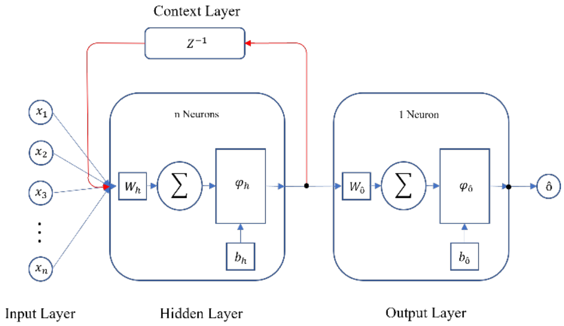


Fig. 8 Graphical representation of an Elman Neural Network

As shown in Fig. 8, the hidden layer h receives its information from both the input and the context layers during the training cycle, which are then combined and analyzed simultaneously. Subsequently, the output of each hidden layer is fed back to the context layers at every time step t to provide additional input to the same hidden layer at time $t+1$. This process is repeated for successive training cycles.

Three different MLPs and three different RNNs were developed for the prediction of the aerodynamic coefficients of the Cessna Citation X (i.e., one MLP and one RNN for each aerodynamic coefficient). This strategy was used because it was found that single output NNs were more accurate than multiple outputs NNs and did not need a complex architecture to learn the correlation between the input data and the target values. Thus, for each neural network model, its output was the coefficient we wanted to predict.

c) Data management:

Building a neural network model necessitates several phases: the training phase, the test phase, and the generalization phase. During the training phase, the neural network weights are updated by a training algorithm to find the inherent relationships among the data in the training data set. Another data set (called the test data set), that was not used for training, is used to calculate the neural network performance. The calculated performance is mainly used to adjust the key model parameters, such as the learning function, the activation function, or the number of hidden layers and neurons. For a given set of n data points and a given set of values of weights $w_{i,j}$, the neural network performance is calculated by the Mean Square Error (MSE), according to the following equation:

$$MSE\ (w) = \frac{1}{n}\sum_{k=1}^{n}\left[\hat{o}_k\left(w_{i,j}\right) - o_k\right]^2 \tag{11}$$

Once the hyperparameters of the neural network are optimized and the neural network has been trained, the “generalization” process can be started. It consists of using the final trained model to predict the aerodynamic coefficients of new flight cases that were not used for training (training set) or hyperparameter tuning (test set). The generalization performance is evaluated by means of the Mean Absolute Relative Error ($MARE$), as follows:

$$MARE = \frac{1}{n}\left(\sum_{k=1}^{n}\left|\frac{\hat{o}_k - o_k}{\hat{o}_k}\right|\right) \times 100 \tag{12}$$

where o_k is the k^{th} experimental data used for validation, and $\hat{o}_k$ is the k^{th} value predicted by the network.

In this study, from the 33 flight cases conducted with the Cessna Citation X RAFS, only 22 were used for identification (training and test phases), while the remaining 11 cases were used for validation. The training set and tests set were selected from the 22 identification flight cases using cross validation method [26].

d) Fine tuning

When training neural networks, three parameters are essential: the training algorithm, the activation function, and the neural network structure (the number of hidden layers and the number of neurons per hidden layer). The procedure for selecting these ideal hyperparameters is given in this section below for the determination of the lift coefficient using a MLP. However, the procedure remains the same for determining other coefficients (CD_s and Cm_s) using both a MLP and a RNN.

Training algorithm

The first analysis consisted of evaluating the performance of the neural network models using several training algorithms. The activation function (*tansig*), the number of hidden layers (2 layers), and the number of neurons in the networks (5 neurons per hidden layer) were fixed and chosen randomly. Only the training function varied. Table 1 presents the list of the tested algorithms.

Table 1: Training Algorithms considered to train the network

Algorithms	Description
LM	Levenberg-Marquardt
BR	Bayesian Regularization
BFG	BFGS Quasi-Newton
RP	Resilient Backpropagation
SCG	Scaled Conjugate Gradient
CGB	Conjugate Gradient Powell/Beale Restarts
CGF	Fletcher-Powell Conjugate Gradient
CGP	Polak-Ribiére Conjugate Gradient
OSS	One Step Secant
GDX	Variable Learning Rate Gradient Descent
GDM	Gradient Descent with Momentum
GD	Gradient Descent

Both MLP and RNN networks were trained with the twelve training algorithms presented in Table 1. During the training, the weights and biases were updated to minimize the network performance (MSE). At the end of the training, the performance of each neural network was evaluated using the test data. Fig. 9 shows the performance in terms of MSE minimization obtained for the prediction of the lift coefficient of the Cessna Citation X using the MLP training process. The performances of the GDX, GDM, and GD were removed for scaling purposes, as their errors were too high compared to that of the results of the other methods.

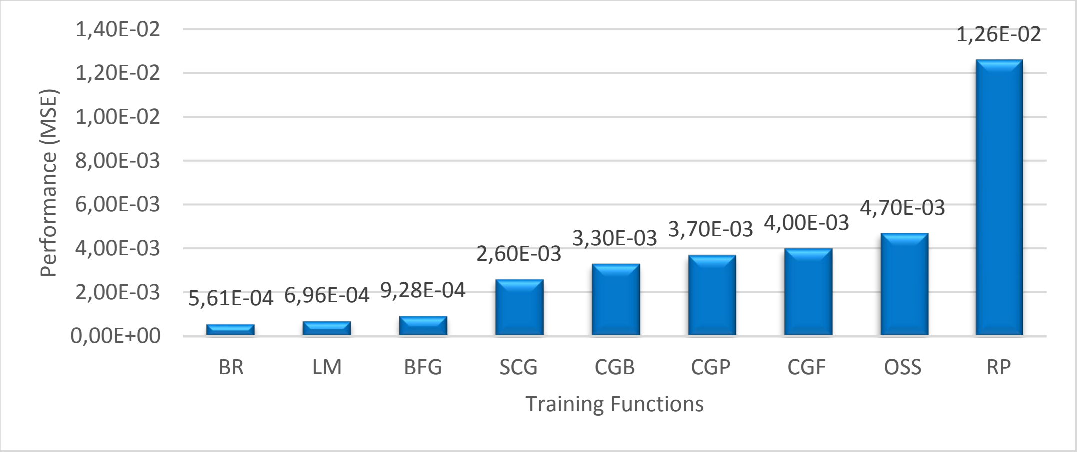


Fig. 9 MLP performance using different training algorithms for the determination of CL_s

Fig. 9 shows that the Bayesian Regularization (BR) and the Levenberg-Marquardt (LM) algorithms provided the lowest MSE. This result was actually expected, as the BR and LM algorithms are well known for their excellent performance in solving nonlinear regression problems.

Both algorithms operate using the same procedure, except that in the BR algorithm, a backpropagation is used to compute the Jacobian of the network performance with respect to the weight and bias variables. However, even if they gave similar results, the BR performed slightly better than the LM, with a $MSE = 5.61 \times 10^{-4}$ for the BR, and a $MSE = 6.96 \times 10^{-4}$ for the LM.

Based on this analysis, the BR algorithm was considered as the most effective training algorithm for the determination of the lift coefficient using the MLP.

Activation function

Once the training algorithm was identified, different activation functions were tested to find the function associated with the BR algorithm that would give the best performance when determining the lift coefficient using the MLP.

The list of activation functions available in MATLAB, that were tested is presented in Table 2.

Table 2: Implemented Activation Function: a is the Neuron's Activation, y is the Neuron's Output

Activation Function	Mathematical Equation
Log Sigmoid (Logsig)	$y(a) = \dfrac{1}{1 + \exp(-a)}$
Hyperbolic Tangent Sigmoid (Tansig)	$y(a) = \dfrac{2}{(1 + \exp(-2 * a))} - 1$
Elliot Symmetric Sigmoid (Elliotsig)	$y(a) = \dfrac{a}{(1 + \lvert a \rvert)}$
Radial basis (Radbas)	$y(a) = \exp(-a^2)$
Normalized radial basis (Radbasn)	$y(a)_i = \dfrac{\exp(-{a_i}^2)}{\sum_{j=1}^{n} \exp(-{a_j}^2)}$ where a is the input vector to a Radbasn function that consists of n elements of n classes, and a_i is the i-th element of the input vector.
Soft max (Softmax)	$y(a)_i = \dfrac{\exp(a_i)}{\sum_{j=1}^{n} \exp(a_j)}$ where a is the input vector to a soft max function that consists of n elements of n classes, and a_i is the i-th element of the input vector.
Saturating linear (Satlin)	$y(a) = \begin{cases} 0, & \text{if } a \leq 0 \\ a, & \text{if } 0 \leq a \leq 1 \\ 1, & \text{if } 1 \leq a \end{cases}$
Symmetric saturating linear (Satlins)	$y(a) = \begin{cases} -1, & \text{if } a \leq -1 \\ a, & \text{if } -1 \leq a \leq 1 \\ 1, & \text{if } 1 \leq a \end{cases}$
Triangular basis (Tribas)	$y(a) = \begin{cases} 1 - \lvert a \rvert, & \text{if } -1 \leq a \leq 1 \\ 0, & \text{otherwise} \end{cases}$
Positive linear (Poslin)	$y(a) = \begin{cases} a, & \text{if } a \geq 0 \\ 0, & \text{if } a \leq 0 \end{cases}$

Fig. 10 shows the MSE error obtained for each activation function when a MLP was trained for predicting the lift coefficient of a Cessna Citation X with the BR algorithm. It can be clearly seen that the *Sigmoid-Type* activation functions performed better than others. In fact, the *Log-Sigmoid*, the *Hyperbolic-Tangent-Sigmoid*, and the *Elliot-Symmetric-Sigmoid* gave lower errors than the other functions, with a MSE of the order of 10^{-4}. The best performance was achieved with *Log-Sigmoid* function, for a MSE of 5.14×10^{-4} as seen on Fig. 10.

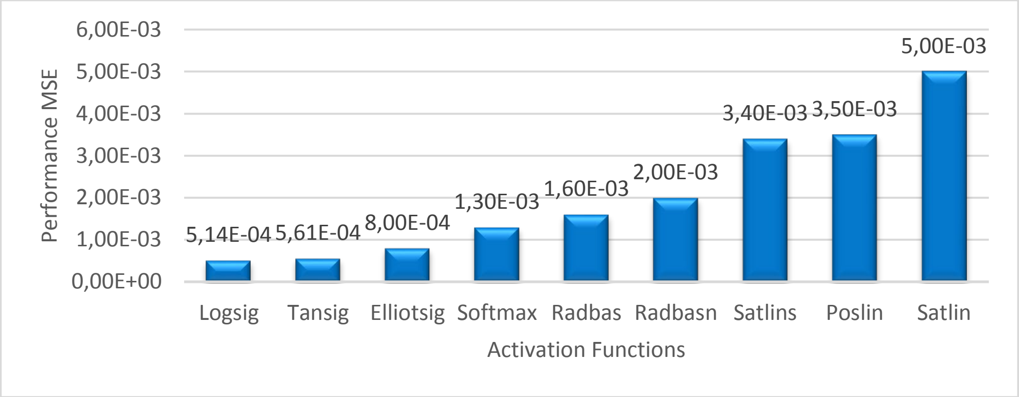


Fig. 10 MLP performance using different activation function for the determination CL_s using BR algorithm

Neural Network structure optimization

The final step in the neural network design process was the selection of the structure that provides the minimum MSE in an appropriate learning time. The structure of the neural network is defined as a combination (m, n), where m is the number of hidden layers of the network, and n is the number of neurons per hidden layer. Similar studies were explored and have demonstrated that the aerodynamic coefficients of a Bombardier CRJ700 in dynamic stall condition can be identified with a MLP model of a maximum of 5 hidden layers and a maximum of 15 neurons per hidden layer [7]. Based on this observation, a minimum number of hidden layers $m_{min} = 1$, and a maximum number of hidden layers $m_{max} = 5$ was assumed. Similarly, the minimum number of neurons per layer was set to $n_{min} = 1$, and the maximum number was set to $n_{max} = 15$.

Once the range of these two parameters was defined, the analysis consisted of training several structures with hidden layers ranging from 1 to 5 and a number of neurons ranging from 1 to 15, and then the resulting *MSEs* were compared. Thus, the number of structures to be trained would be equivalent to 15^5, which is clearly a very large number. To reduce the number of possible structures, it was assumed that all hidden layers should have the same number of neurons. Such an assumption was considered because it was found that varying the number of neurons from one hidden layer to another did not significantly improve the network performance. This assumption reduces the number of possible structures from 15^5 to 75. The *MSEs* of different structures trained to predict the lift coefficient of the Cessna Citation X airplane are shown in Fig. 11.

Fig. 11 shows the $MSEs$ obtained for each tested structure. Structures with a different number of layers are displayed in different colors. For example, the MSE obtained with a structure with $m = 2$ layers are displayed in “orange”, while the MSE obtained with a structure with $m = 3$ layers are displayed in “blue”. Performances obtained for the structure with $m = 1$ hidden layer and structures with less than $n = 7$ neurons per hidden layer have been removed for scaling purposes as the MSE was too high. We can clearly see a convergence of the MSE value, which decreases and tends to zero when the number of neurons increases. The convergence is faster for structures with a higher number of layers. For example, structure with 3 layers converge faster than structure with 2 layers). However, it is challenging to determine which is the optimal network structure, as there is no significant improvement on the network’s performance after 11 neurons for networks with $m = 2$, $m = 3$, $m = 4$, and $m = 5$. Any of these structures could be used to identify the lift coefficient and would give satisfactory results.

However, as a structure with $m = 5$ hidden layers converges even faster than others, a convergence threshold of $MSE = 1.19^{-5}$ is reached even earlier (before $n = 11$) for $n = 9$ neurons. The network structure with $(m = 5, n = 9)$ was therefore considered for the determination of the lift coefficient CL_s of the Cessna Citation X.

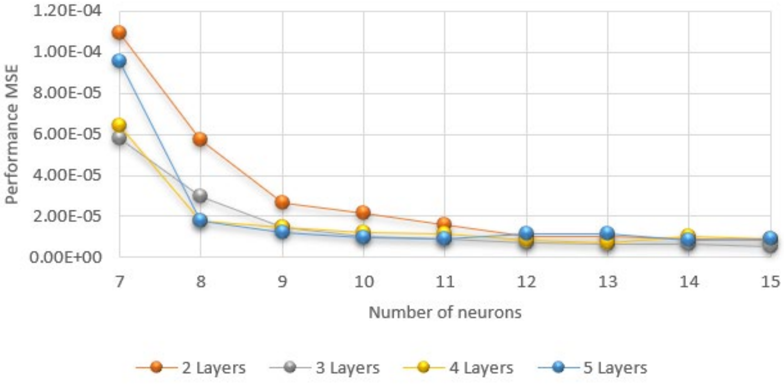


Fig. 11 Performances for various MLP structures for the estimation of the CL_s of the Cessna Citation X

3. RESULTS

This section presents the validation results of the proposed methodology. An evaluation of the accuracy of the two types of neural networks (i.e., MLP and RNN) in terms of the prediction of aerodynamic coefficients is presented. For this purpose, the outputs of both the MLP and the RNN models were compared with experimental aerodynamic coefficients' data obtained from the Cessna Citation X RAFS.

The best performances in terms of MSE obtained for the estimation of the three longitudinal aerodynamic coefficients (CL_s, CD_s and Cm_s) were obtained with the BR algorithms, combined with the *tansig* and *logsig* activation functions. RNN models performed as well as MLP models, but with smaller structures, as the total number of neurons of the hidden layers $(m \times n)$ on RNN structures were smaller than with the MLP structures for the identification of the same coefficient. For example, the identification of the lift coefficient using an MLP model required a structure of 45 (9×5) hidden neurons, while an RNN required only 30 (3×10) hidden neurons. The two models performed similarly ($MSE = 1.2 \times 10^{-5}$ for MLP and $MSE = 1.1 \times 10^{-5}$ for RNN). This faster convergence of RNN models could be explained by the ability of their algorithm to take into account the dynamic behavior of complex systems during the identification process.

Table 3: Optimal parameters obtained for the MLP and the RNN using the BR algorithm

	CL_s		CD_s		Cm_s	
Type of neural network	**MLP**	**RNN**	**MLP**	**RNN**	**MLP**	**RNN**
Activation function	*Logsig*	*Tansig*	*Logsig*	*Tansig*	*Logsig*	*Tansig*
Number of hidden layers	5	3	2	3	4	3
Number of nodes per hidden layer	9	10	14	6	12	12
MSE value	1.2×10^{-5}	1.1×10^{-5}	4.5×10^{-6}	5.4×10^{-6}	1.9×10^{-4}	1.0×10^{-5}

Once the hyperparameter were fixed and the neural networks trained, the accuracy of each model (MLP and RNN) was evaluated using validation cases that were not used for training. For this purpose, the outputs of the MLP and RNN models were compared in terms of the

$MARE$, with experimental aerodynamic coefficients obtained from the Cessna Citation X RAFS. Fig. 12 and Fig. 13 show two examples of the results obtained at respectively 32,500 ft with slats-in and 27,500 ft with slats-out.

In general, the results show good agreement between the experimental data obtained by the RAFS (in "blue"), the data predicted by the MLP (in "red"), and the data predicted by the RNN (in "yellow"). The results showed that the lift and drag coefficients of the two cases were predicted with a Mean Absolute Relative Error ($MARE$) smaller than 1%. For the pitching moment coefficient, the $MARE$ was smaller than 5 %. These very small errors allow us to conclude that both models (RNN and MLP) were able to successfully predict the aerodynamic coefficients for these two flight cases presented on Fig. 12 and Fig. 13. The comparisons presented in these examples were repeated for all the other validation flight cases. Table 4 presents the $MARE$ and the Mean Absolute Residual errors obtained for the prediction of each coefficient with MLP models.

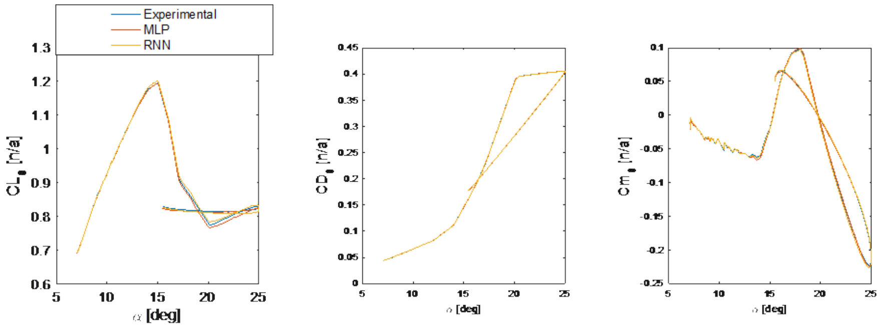


Fig. 12 Validation of predicted aerodynamic coefficients (with MLP and RNN) for a flight test at 32,500 ft with slats-in

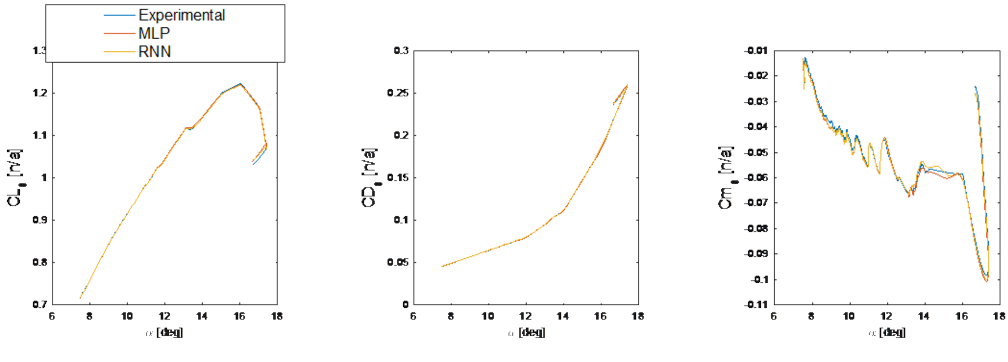


Fig. 13 Validation of predicted aerodynamic coefficients (with MLP and RNN) for a flight test at 27,500 ft with slats-out

Table 4: Mean Absolute Relative Error ($MARE$) and Mean Absolute Residual error obtained between experimental data and values predicted with MLP models

Flight Case		CL_s		CD_s		Cm_s	
Altitude [ft]	**Slat angle**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**
7,000	In	0.998	8.40	0.561	0.68	7.689	2.61
15,000	In	0.325	2.60	0.329	0.60	5.207	1.67
17,500	Out	0.712	6.06	0.535	0.87	5.253	3.80
22,500	In	0.807	7.29	0.390	0.69	14.257	5.49

27,500	Out	0.153	1.52	0.113	0.16	3.011	1.02
32,500	In	0.149	1.27	0.104	0.09	3.625	0.98
37,500	Out	0.386	3.31	0.181	0.16	1.421	1.18
40,000	In	0.540	4.88	0.308	0.63	5.770	2.26
42,500	Out	0.378	4.12	0.263	0.33	8.065	3.42
47,500	In	0.284	2..98	0.203	0.33	3.340	1.51
47,500	Out	0.469	4.79	0.351	0.64	4.529	1.42

Table 5: Mean Absolute Relative Error ($MARE$) and Mean Absolute Residual error obtained between experimental data and values predicted with RNN models

Flight Case		CL_s		CD_s		Cm_s	
Altitude [ft]	**Slat angle**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**
7,000	In	0.340	3.00	0.702	0.15	11.203	3.82
15,000	In	0.351	2.90	0.457	0.86	5.573	1.55
17,500	Out	0.620	5.85	0.570	0.74	7.482	4.65
22,500	In	0.686	6.74	0.392	0.57	14.842	4.22
27,500	Out	0.109	1.09	0.206	0.19	2.921	9.4
32,500	In	0.168	1.57	0.176	0.16	4.250	8.61
37,500	Out	0.478	4.70	0.228	0.27	0.933	7.49
40,000	In	0.413	3.95	0.273	0.41	7.041	1.39
42,500	Out	0.224	2.38	0.396	0.67	6.593	1.15
47,500	In	0.248	2.63	0.170	0.26	3.714	9.85
47,500	Out	0.593	6.42	0.226	0.28	2.318	7.94

We can see that both MLP and RNN methodologies can globally estimate the aerodynamic coefficients quite well, as the obtained errors are verry small. For all 11 flight cases used for validation, the lift and drag coefficients were estimated with a $MARE$ of less than 1 %. For the pitching moment, 9 flight cases were estimated with a $MARE$ of less than 10 %, and only 2 flight cases were estimated with errors between 10% and 15%. The slats position did not affect the models' precision. Aerodynamic coefficients of slat-in and slat-out flight cases were estimated with the same range of precision. The higher error in percentage obtained for the determination of the pitching moment coefficient could be explained by the fact that for most of the flight cases (in both validation and identification cases), the pitching moment changes its sign when varying (as shown on Fig. 4 and Fig. 5). Consequently, at some flight test points, the Cm_s has relatively low values, around zero, which leads to large relative errors ($MARE$). Therefore, even if the $MARE$ errors of the Cm_s seem relatively large (above 10%) for some flight cases, the precision of the estimation remains very good, as the residual errors are very low, even negligible (of the order of 10^{-3}). In summary, the lift coefficient was estimated with an average $MARE$ of 0.5% for MLP and 0.4 % for RNN. For the estimation of the drag coefficient, the average $MARE$ was 0.3% for MLP and 0.35 % for RNN. Finally, the MLP performed better than the RNN for the prediction of the pitching moment coefficient, with an average $MARE$ of 5.6 % comparatively to 6.1% for the RNN. We can also conclude that the RNN models performed similar to the MLP models.

4. CONCLUSIONS

This paper presents a detailed methodology for the lift, drag, and pitching moment aerodynamic coefficients modeling in stall regime using Neural Networks. The linear and nonlinear variations of lift and drag aerodynamic coefficients are estimated along the stall hysteresis curve. This presented methodology was applied to the Cessna Citation X airplane developed by CAE Inc and it gave successful results. A stall recovery procedure was developed and executed on the Cessna Citation X RAFS to obtain flight data. A total of 33 flight cases were conducted for different altitudes ranging from 5000 to 50,000 ft and for the “in” and “out” slat configurations. The obtained data were then processed to obtain the lift, drag, and pitching moment aerodynamic coefficients. Data from 22 (67%) flight cases were used to train the following two types of Neural Network models: Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN). The procedure to select the Neural Network parameters (training algorithms, activation function) was detailed, and the process to optimize the models’ structures was also developed. Both the MLP and RNN models predicted the aerodynamic coefficients with an average Mean Absolute Relative Error ($MARE$) smaller than 0.5% for the lift and drag coefficients and smaller than 6.2 % for the pitching moment coefficients. These minor errors allowed us to conclude that the developed models performed very well.

REFERENCES

[1] K. Cunningham, J. V. Foster, G. H. Shah, E. C. Stewart, R. A. Rivers, J. E. Wilborn and W. Gato, *Simulation Study of a Commercial Transport Airplane During Stall and Post-Stall Flight*, Presented at the World Aviation Congress & Exposition, 2004.

[2] G. K. Ananda and M. S. Selig, *Stall/Post-Stall Modeling of the Longitudinal Characteristics of a General Aviation Aircraft*, Presented at the AIAA Atmospheric Flight Mechanics Conference, Washington, D.C., 2016.

[3] S. Shao, M. Chen and Y. Zhang, Adaptive Discrete-Time Flight Control Using Disturbance Observer and Neural Networks, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. **30**, No. 12, 2019, pp. 3708–3721, https://doi.org/10.1109/TNNLS.2019.2893643.

[4] R. M. Botez, *Une Étude Comparative Des Modèles Semi-Empiriques Pour La Prédiction Du Décrochage Dynamique*, Master Thesis, École polytechnique, Montreal, QC, Canada, 1989.

[5] A. Spentzos, G. Barakos, K. Badcock, B. Richards, P. Wernert, S. Schreck and M. Raffel, Investigation of Three-Dimensional Dynamic Stall Using Computational Fluid Dynamics, *AIAA Journal*, Vol. **43**, No. 5, 2005, pp. 1023–1033. https://doi.org/10.2514/1.8830.

[6] P. Wernert, W. Geissler, M. Raffel and J. Kompenhans, Experimental and Numerical Investigations of Dynamic Stall on a Pitching Airfoil, *AIAA Journal*, Vol. **34**, No. 5, pp. 982–989, https://doi.org/10.2514/3.13177, 1996.

[7] Y. Tondji, G. Ghazi and R. M. Botez, *CRJ 700 Aerodynamic Coefficients Identification in Dynamic Stall Conditions Using Neural Networks*, Presented at the AIAA SCITECH 2022 Forum, San Diego, CA & Virtual, 2022.

[8] M. Albisser, *Identification of Aerodynamic Coefficients from Free Flight Data*, 2015.

[9] W. F. Phillips and D. O. Snyder, Modern Adaptation of Prandtl’s Classic Lifting-Line Theory, *Journal of Aircraft*, Vol. **37**, No. 4, 2000, pp. 662–670, https://doi.org/10.2514/2.2649.

[10] A. Koreanschi, O. Sugar Gabor and R. M. Botez, *New Numerical Study of Boundary Layer Behavior on A Morphing Wing-with-Aileron System*, Presented at the 32nd AIAA Applied Aerodynamics Conference, Atlanta, GA, 2014.

[11] R. A. Piziali, *2-D and 3-D Oscillating Wing Aerodynamics for a Range of Angles of Attack Including Stall*, National Aeronautics and Space Administration, Ames Research Center : US Army Aviation and Troop Command ; National Technical Information Service, distributor, Moffett Field, Calif. : [Springfield, Va, 1994.

[12] K. Mulleners, A. Pape, B. Heine and M. Raffel, *The Dynamics of Static Stall*, 2012.

[13] R. Botez, Morphing Wing, UAV and Aircraft Multidisciplinary Studies at the Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity LARCASE, *AerospaceLab Journal*, Vol. Issue **14**, 2018, p. September 2018; ISSN: 21076596. https://doi.org/10.12762/2018.AL14-02.

[14] A. Ben Mosbah, R. M. Botez, S. M. Medini and T.-M. Dao, Artificial Neural Networks-Extended Great Deluge Model to Predict Actuators Displacements for a Morphing Wing Tip System, *INCAS BULLETIN*, Vol. **12**, No. 4, 2020, pp. 13–24. https://doi.org/10.13111/2066-8201.2020.12.4.2.

[15] A. Ben Mosbah, M. Flores Salinas, R. Botez T. and Dao, New Methodology for Wind Tunnel Calibration Using Neural Networks - EGD Approach, *SAE International Journal of Aerospace*, Vol. **6**, No. 2, 2013, pp. 761–766. https://doi.org/10.4271/2013-01-2285.

[16] S. De Jesus Mota and R. M. Botez, New Helicopter Model Identification Method Based on Flight Test Data, *The Aeronautical Journal*, Vol. **115**, No. 1167, 2011, pp. 295–314, https://doi.org/10.1017/S0001924000005789.

[17] N. Boely, R. M. Botez and G. Kouba, Identification of a Non-Linear F/A-18 Model by the Use of Fuzzy Logic and Neural Network Methods, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. **225**, No. 5, 2011, pp. 559–574, https://doi.org/10.1177/2041302510392871.

[18] N. Boely and R. M. Botez, New Approach for the Identification and Validation of a Nonlinear F/A-18 Model by Use of Neural Networks, *IEEE Transactions on Neural Networks*, Vol. **21**, No. 11, 2010, pp. 1759–1765, https://doi.org/10.1109/TNN.2010.2071398.

[19] M. Zaag and R. M. Botez, *Cessna Citation X Engine Model Identification and Validation in the Cruise Regime from Flight Tests Based on Neural Networks Combined with Extended Great Deluge Algorithm*, Presented at the AIAA Modeling and Simulation Technologies Conference, Grapevine, Texas, 2017.

[20] J. Anderson, *Fundamentals of Aerodynamics*, McGraw-Hill Education, New York, 2010.

[21] G. Ghazi, M. Bosne, Q. Sammartano and R. M. Botez, *Cessna Citation X Stall Characteristics Identification from Flight Data Using Neural Networks*, Presented at the AIAA Atmospheric Flight Mechanics Conference, Grapevine, Texas, 2017.

[22] W. J. McCroskey, *The Phenomenon of Dynamic Stall,* NATIONAL AERONUATICS AND SPACE ADMINISTRATION MOFFETT FIELD CA AMES RESEARCH CENTER, 1981.

[23] R. J. Williams and D. Zipser, A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, Vol. **1**, No. 2, pp. 270–280. https://doi.org/10.1162/neco.1989.1.2.270, 1989.

[24] S. O. Haykin, *Neural Networks: A Comprehensive Foundation*, Pearson, Upper Saddle River, N.J, 1998.

[25] J. L. Elman, Finding Structure in Time, *Cognitive Science*, Vol. **14**, No. 2, 1990, pp. 179–211, https://doi.org/10.1207/s15516709cog1402_1.

[26] M. Stone, Cross-Validatory Choice and Assessment of Statistical Predictions, *Journal of the Royal Statistical Society: Series B (Methodological)*, Vol. **36**, No. 2, 1974, pp. 111–133, https://doi.org/10.1111/j.2517-6161.1974.tb00994.x.