# CESSNA Citation X Engine Model Identification using Neural Networks and Extended Great Deluge Algorithms

Mahdi ZAAG[1], Ruxandra Mihaela BOTEZ*[,1], Tony WONG[1]

*Corresponding author
[1]Department of Systems Engineering,
École de Technologie Supérieure, University of Québec,
1100 Notre-Dame West, Montréal, Québec, Canada,
mahdi.zaag.1@ens.etsmtl.ca, ruxandra.botez@etsmtl.ca*, tony.wong@etsmtl.ca

DOI: 10.13111/2066-8201.2019.11.2.16

*Received: 8 March 2019/ Accepted: 19 April 2019/ Published: June 2019*


***Abstract**: Accurate numerical engine model is an enabling factor in aircraft performance evaluation and improvement. In this work, nonlinear engine input-output relationships are learned and predicted by two cascading multilayer feedforward neural networks. Machine learning approaches necessitate a great amount of data to achieve efficiency. To satisfy this operational requirement, 441,000 flight cases are designed for a Cessna Citation X turbofan engine using a Level D Research Aircraft Flight Simulator designed and manufactured by CAE Inc. For each flight case, cruise phase data comprising Mach number, altitude, throttle level angle, low-pressure compressor speed, high-pressure compressor speed, engine net thrust and engine fuel flow are recorded. These data are then organized into subsets for training and validation purposes. Each neural network configuration is obtained by means of the Extended Great Deluge algorithm. The latter is also responsible for coordinating neural network training and learning error computation. Analyses based on computer experiments showed a mean relative prediction error upper bound of 4% is achievable for engine output parameters for all cruise phase flight cases.*

***Key Words**: Engine, performance prediction, Cessna citation X, metaheuristics, feedforward neural networks, extended great deluge.*

## ACRONYMS

| | |
|---|---|
| EGD | Extended Great Deluge algorithm |
| FNC | Corrected engine net thrust |
| LM | Levenberg-Marquardt optimization algorithm |
| MI | Maximum number of EGD algorithm iterations |
| ML | Maximum number of neural network layers |
| MN | Maximum number of neurons |
| MRE | Mean relative error |
| N1PC | Low-pressure compressor speed |
| N2PC | High-pressure compressor speed |
| NN | Neural network |
| RAFS | Research aircraft flight simulator |
| TLA | Throttle level angle |
| TSFC | Specific fuel consumption |
| WFC | Engine fuel flow |

## 1. INTRODUCTION AND BACKGROUND

The usual approach in engine modeling consists in linking the engine inputs parameters with engine thrust through analytical treatment of aerodynamic and thermodynamic data from flight tests. Models proposed by Mattingly et al. [1] and Wanner [2] are early representatives of these research efforts. In [1], maximum thrust is expressed in terms of air density and Mach number variation while Throttle Level Angle (TLA) is considered in [2]. Later, Ghazi et al. [3] compared the results of these models to flight data generated by the Cessna Citation X Level D Research Aircraft Flight Simulator (RAFS). They obtained a maximum relative error of 3.4% for Mattingly's model at Mach number $M = 0.4$, and 6% for Wanner's model at $M = 0.1$ at constant altitude.

Similar results were produced using Mattingly's model at different altitudes whereas a relative error in the order of 20% was obtained for Wanner's model at an altitude $h = 30{,}000$ ft. More recently, a generic technique was developed by Rodriguez and Botez [4] to predict turbofan thrust as a function of Mach number, altitude and bypass ratio. Their technique, based on the reference thrust at sea level, can predict the maximum thrust of different turbofan engines by decomposing engine thrust into two multiplicative components: altitude and Mach numbers. Data interpolation and curve fitting were used to determine empirically the values of these components. The reported maximum relative error for thrust prediction was in the range of 6% to 14% depending on the engine type.

The amount of fuel used to generate engine thrust is an important factor in engine performance evaluation. Specific fuel consumption (TSFC) is often used to describe engine efficiency. Torenbeek [5] performed a detailed engine operation analysis and observed that fuel consumption is directly related to engine compressor input/output ratio, Mach number and inversely related to dilution rate.

In [6], TSFC was computed by Mattingly according to altitude, Mach number, engine cycle and temperature. In cruise phase, more specifically at $M = 0.8$ and h = 32,000 ft, Mattingly obtained a TSFC of $1.89 \times 10^{-5}$ (kg/s)/N. Improving upon the analytical works done by Torenbeek, Roux [7] advised resetting the TSFC to $1.75 \times 10^{-5}$ (kg/s)/N for engines with a dilution rate greater than 3 during cruise phase.

Based on experimental data, Roux reported that the new TSFC value produced a mean relative error (MRE) of 3.7% compared to 6.6% for Torenbeek's proposition, and 8% for Mattingly calculations.

Analytical analysis of flight data is a powerful tool when devising engine models. However, airborne data acquisition is still an expansive process involving many dedicated operations and specialized equipment. By extension, the scarcity of published flight test data can affect detrimentally research efforts in engine modelling. In this work, aerodynamics data are generated by the RAFS by programming flight paths using parameters such as altitudes, throttle level angles and Mach numbers.

Engine parameters are then extracted from the RAFS during the cruise phase portion of the flight. Note that the RAFS is equipped with a Level D flight dynamics toolbox which is the highest level certified by the FAA.

This paper is organized in the following manner. Section 2 presents the modeling methodology pertaining to neural network configuration and training through the Extended Great Deluge algorithm.

Section 3 details the engine model, and data preprocessing needed for model training. Numerical results from computer experiments are presented in section 4 and conclusions are drawn in section 5.

## 2. METHODOLOGY

Computational intelligence is related to artificial intelligence with emphasis on metaheuristic algorithms such as neural networks (NN), evolutionary computation and other nature-inspired computation methodologies. These metaheuristic algorithms have the ability to learn from data or experimental observation. They were successfully applied in the determination of aircraft aerodynamic coefficients [8], and aircraft model identification [9, 10]. In particular, the use of a multilayer feedforward NN to identify aeroservoelastic F/A-18 model established interesting research avenues for accurate nonlinear flight model identification [11, 12].

In neural network implementations, the fundamental processing element of a network is a neuron capable of nonlinear mapping $\Re^I \rightarrow [0, 1]$ where $I$ is the number of inputs connected to the neuron. The net input signal to a neuron is often computed as the weighted sum of its inputs. Generally, neurons are grouped together by creating layers, which are connected to one another. In the context of an engine model identification problem, some of the neurons interface to the input data while other neurons provide the model's outputs. All the rest of the neurons are hidden between the input and output layers. Further details on feedforward networks can be found in [13].

### 2.1 Neural network training

Supervised training is used to adjust neuron's connecting weights. The network is provided with a data set consisting of input vectors, and the desired outputs associated with each input vector. This dataset is referred to as the training set. The network processes the training set data and compares its outputs against the desired outputs. The aim of the supervised training is to adjust the weight values such that the mean squared error between the output of the neurons and the desired outputs is minimized. Note that during network training, the training set is processed repeatedly as the connection weights are adjusted [14]. As indicated in section 2, the training set is generated using CAE's Level D RAFS flight cases designed for this research. In [12], Boëly et al. proposed the use of Levenberg-Marquardt (LM) algorithm to train a NN in order to identify F/A-18 flutter behavior. The LM algorithm is a widely used iterative technique for locating local minimum in nonlinear least-squares problems. LM can be considered as a combination of steepest descent with the Gauss-Newton method. As pointed out by Manolis et al. [15], LM has a very appropriate behavior. That is, given a local minimum, if the current solution is distant, then the algorithm will progress slowly but will guarantee to converge. If the current solution is near to a local minimum, then it will exhibit fast convergence.

### 2.2 Neural network configuration

The architectural complexity of a Neural Network depends on the number of neurons on each layer, the total number of layers, and the number of connecting weights [16]. A NN configuration is the set of values representing these properties, and is usually determined based on observations, rule-of-thumb methods or other empirical approaches. Because of this, the effectiveness of a NN may be compromised by its complexity. To alleviate this problem, Mosbah et al. [17] proposed the use of the Extended Great Deluge (EGD) algorithm in order to find an effective NN configuration with minimal complexity. Given as inputs the bounds on the number of neurons in each layer and the number of layers, the EGD algorithm has the task of finding a NN configuration with the smallest training mean squared error. The following describes an algorithm implementing the EGD algorithm for finding minimal complexity NN configuration. An in-depth discussion of the EGD algorithm is given in [18].

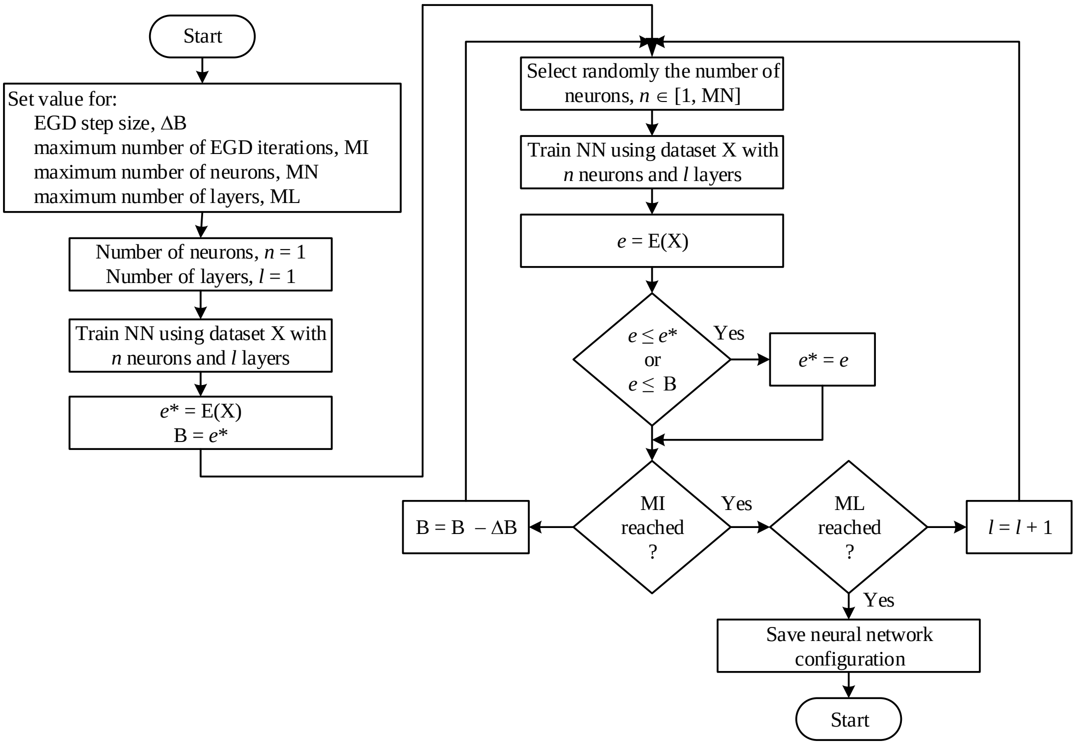


Fig. 1 – Algorithm implementing EGD in order to find a NN configuration with minimal complexity [17]

As shown in Fig. 1, the first step is to assign initial values to the algorithm's controlling parameters and to NN configuration parameters. Starting with one neuron and one layer, the algorithm trains the NN by using training set X, and computes its mean squared error E(X). This initial error is also assigned to B, a variable acting as a dynamic acceptance level for the search procedure. Next, the algorithm choses a random number of neurons $n \in [1, \text{MN}]$ and assigns these neurons to the current layer. The new NN configuration is then trained and again its training error computed. The new configuration is kept if it has a lower or equal training error than the previous configuration. However, the new configuration is also kept when its output error is lower or equal to the current acceptance level. This is the non-greedy nature of the EGD algorithm. The search procedure continues by randomly selecting another value for $n$, and by lowering the acceptance level B by an amount represented by ΔB. When the number of iterations exceeds MI, the algorithm adds a layer to the current configuration and restarts the search procedure for this new layer. The algorithm stops when the maximum number of layers is reached. The next section presents the NN engine model and provides details on the datasets used in the model training procedure.

## 3. NEURAL NETWORK ENGINE MODELING

A turbofan engine operates by compressing, combusting, expanding and exhausting air from the engine creating thrust to propel the aircraft. Fig. 2 shows the major components of a turbofan engine. Ambient air enters the engine through the fan. Part of this accelerated air enters the core of the engine while the other portion bypasses the core creating most of the thrust. Both low-pressure and high-pressure compressors act to pressurize air, and thus increasing its temperature before entering the combustion chamber. The purpose of combustor is to add energy to the air propelling the flow towards the rear turbines. This flow energy is converted into mechanical energy by the turbines enabling the compressors and the fan to spin. Continuous thrust is thus generated by repeating the above described cycle.

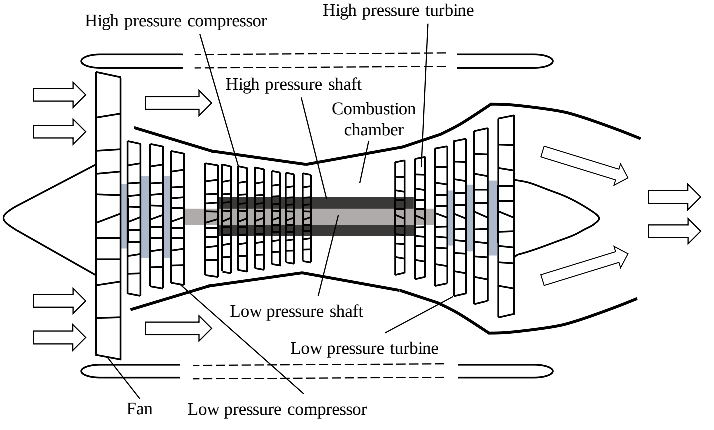


Fig. 2 – Simplified side view of a turbofan engine

In their work on engine model identification, Ghazi et al. [3] compared several analytical models based on the following input/output scheme.

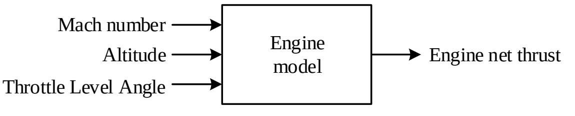


Fig. 3 – Selected input/output for engine modeling from [2]

As shown in Fig. 3 the inputs of the model are the Mach numbers *M*, altitude *h* in feet (ft), TLA in degrees, and the output is the engine net thrust in pounds (lb). Engine control in those models is implicit, and is therefore embedded within the input-output relationship. This model can be detailed to represent a more practical operational context. In the proposed NN engine model, the existence of an electronic engine controller is explicit. The Full Authority Digital Engine Control (FADEC) is a digital electronic system that controls many parameters regarding engine performance. Specifically, FADEC controls the low-pressure compressor rotation speed as function throttle level angle (TLA), Mach number (*M*) and altitude. In addition, fuel consumption (WF) in pounds per hour (lb / h), the high-pressure compressor speed (N2P) and the engine net thrust (FN) in pounds (lb) can be determined using the low-pressure compressor rotation speed (N1P) in rotations per minute (rpm) and *M* [19]. Fig. 4 shows the complete NN engine model.

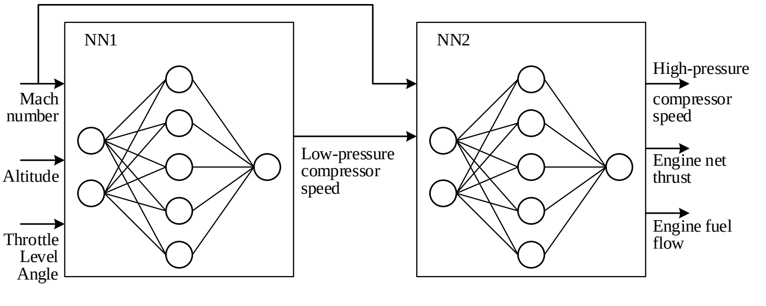


Fig. 4 – Selected input/output for engine modeling from [2]

The proposed engine model is composed of two cascading neural networks NN1 and NN2 models. The first network NN1 provides the low-pressure compressor speed (N1P) while the second network NN2 provides the engine net thrust (FN), the fuel flow (WF) and the high-pressure compressor speed (N2P).

### 3.1 Neural network data preprocessing

The parameters describing an engine's internal operation, such as flows, pressures, temperatures, and speeds, vary with ambient conditions at the engine's inlet. It is important to characterize engine performance by taking into account these atmospheric conditions. This entails that raw data from the RAFS should be preprocessed before NN model training and validation.

This preprocessing is accomplished using the so-called θ and δ exponent correction [20]. Usually, the correction is carried out by first determining the θ (temperature) and δ (pressure) ratios:

$$\begin{aligned}\theta &= T/T_0,\\ \delta &= P/P_0\,,\end{aligned} \tag{1}$$

where $T$ and $P$ are the ambient temperature and pressure. $T_0$ is the temperature at sea level and $P_0$ is the pressure at sea level. Assuming $T_0 = 518.69$ degrees Rankine (15 °C), and $P_0 = 2116.2$ lb/ft$^2$, these dimensionless ratios are used to derive the following correction factors with isentropic air flow [21]

$$\begin{aligned}\theta^* &= \theta(0.2M^2 + 1),\\ \delta^* &= \delta(0.2M^2 + 1)^{3.5},\end{aligned} \tag{2}$$

where $M$ is the Mach number. Using Eq. (2), the corrected low-pressure compressor speed N1PC is

$$\mathrm{N1PC} = \mathrm{N1P}/\sqrt{\theta^*}, \tag{3}$$

and the corrected high-pressure compressor speed N2PC is

$$\mathrm{N2PC} = \mathrm{N2P}/\sqrt{\theta^*}. \tag{4}$$

While the engine net thrust FN is corrected by

$$\mathrm{FNC} = \mathrm{FN}/\delta^*, \tag{5}$$

and finally, the fuel consumption WFC is

$$\mathrm{WFC} = \mathrm{WF}/\delta^*. \tag{6}$$

These preprocessed data are then used in model training and validation. Two set of experiments are conducted to evaluate the NN engine model performances. Experiment protocol and results are given in the next section.

## 4. COMPUTER EXPERIMENTS

When applying multilayer neural networks, the usual practice is to first divide the data randomly into two subsets. The first subset is the training set. It is used for computing the gradient and updating the network connecting weights.

The second subset is the validation set and is used to evaluate the network's generalization capability.

Note that, in machine learning literature, training and validation errors are often referred to as learning and prediction errors.

For engine model identification, the error on the validation set and on the training set usually decrease during the initial phase of training.

The error on the validation set typically begins to rise when the NN begins to overfit the training data [13].

For this reason, the NN configuration and error rates are reported at the minimum of the validation set error.

Data are generated by the RAFS system by varying the altitude, Mach number and TLA in order to train and validate the NN engine model.

Recall that the RAFS is a level D flight simulator which reproduces accurate flight dynamics according to the FAA certification authorities. Table 1 summarizes the parameters and their values.

Table 1 – Engine model input parameters and their ranges

| Parameter | Values | Steps |
|---|---|---|
| Altitude | 5,000 ft to 45,000 ft | 5,000 ft |
| Mach number | 0.2 to 0.9 | 0.0001 |
| Throttle Level Angle | 35 degrees to 65 degrees | 5 degrees |

In this work, a total of $N$ = 441,000 flight cases are designed, simulated on the RAFS, and the corresponding engine model output parameters recorded.

These flight test cases are organized according to seven TLA values ranging from 35 to 65 degrees with steps of 5 degrees.

Thus, there are 63,000 flight cases per TLA value. Of the 63,000 flight cases, 47,250 (75%) of them are randomly assigned to the training set, and the remaining 15,750 (25%) flight cases are assigned to the validation set.

The NN engine model is then trained and validated using these datasets. To evaluate the NN model performances, the mean relative error (MRE) measure is used. The relative error of an engine output parameter for flight case *i* is defined as

$$e_i = \left| \frac{\tilde{o}_i - o_i}{o_i} \right| \tag{7}$$

where $\tilde{o}_i$ is the output parameter computed by the NN model and $o_i$ is the corresponding output parameter value from a flight test dataset.

From Eq. (7), the MRE for $n \subseteq N$ flight cases is given by

$$\bar{e} = \frac{100\%}{n} \sum_{i=1}^{n} e_i. \tag{8}$$

Two sets of computer experiments are executed, and evaluated using the MRE performance measure of Eq. (8).

The first of these experiments assesses the capability of the first neural network NN1 to reproduce accurate low-pressure compressor speed N1PC with varying TLA, altitude and Mach number as detailed in Table 1.

Fig. 5 shows a typical flight envelope for the flight test cases where TLA is equal to 35 degrees.

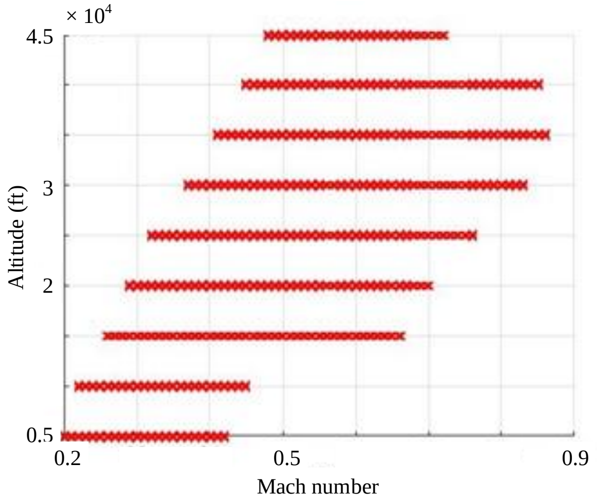


Fig. 5 – Typical flight envelop for TLA = 35 degrees

The second set of experiments verify the combined NN1 and NN2 models performance. High-pressure compressor speed N2PC, engine net thrust FNC and fuel flow WFC are measured according to TLA, altitude and Mach number.

For each TLA value, 47,250 flight cases are randomly selected for training and the remaining 15,750 cases are used for validation.

Thus, each training set is used by the EGD algorithm to determine the least complex configuration in terms of number of layers, and number of neurons per layer.

The performance of the selected NN configuration is then verified using the validation set. The maximum number of layers ML, the maximum number of neurons per layer MN and other EGD parameters are given in Table 2.

Table 2 – Engine model input parameters and their ranges

| Parameter | Values |
|---|---|
| Maximum number of layer | ML = 4 |
| Maximum number of neurons per layer | MN = 10 |
| Acceptance level step size | $\Delta B = 0.00001$ |
| Maximum number of EGD iterations | 10 |

### 4.1 Results and comment

The first set of experiments are performed to evaluate the first neural network block NN1, and its ability to learn low-pressure compressor speed N1PC.

Fig. 6 shows an excerpt from NN1 output using training set containing TLA = 35 degrees, altitudes ranging from 20,000 ft to 40,000 ft and Mach numbers varying from 0.2 to 0.9.

In Fig. 6, the outputted N1PC values actually overlap the values from the training set. This confirms the ability of NN1 to learn N1PC for different altitudes and Mach numbers.

Table 3 presents the learning and prediction error in terms of MRE for each TLA value. It is observed that the maximum prediction MRE of the N1PC for different TLA values is less than 0.05%.

This indicates a good mapping between the inputs and outputs, and the prediction capability of the NN1 block.

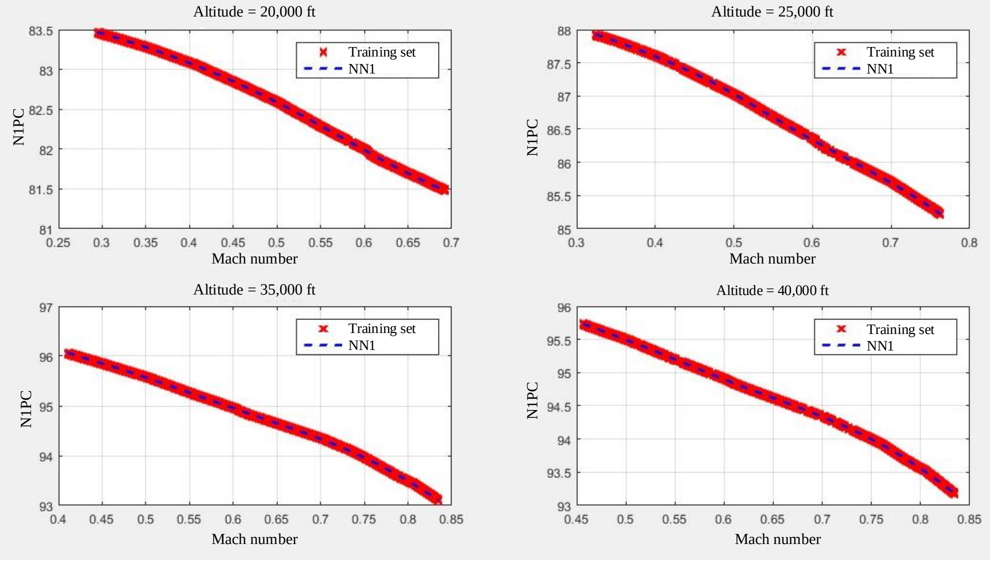


Fig. 6 – NN1 training: N1PC learning given altitude, Mach number, and TLA = 35 degrees

Table 3 – NN1 training and validation MRE:

| TLA degrees | Training MRE (%) | Validation MRE (%) |
|---|---|---|
| 35 | 0.0037 | 0.0048 |
| 40 | 0.0036 | 0.0035 |
| 45 | 0.0086 | 0.0099 |
| 50 | 0.0115 | 0.0089 |
| 55 | 0.0401 | 0.043 |
| 60 | 0.0212 | 0.0096 |
| 65 | 0.0029 | 0.0021 |

Fig. 7 presents a graphical comparison between learning and prediction MRE. It shows that the prediction MRE follows closely the learning MRE. The largest deviation occurred at TLA = 60 degrees, but it only amounts to 0.011%.

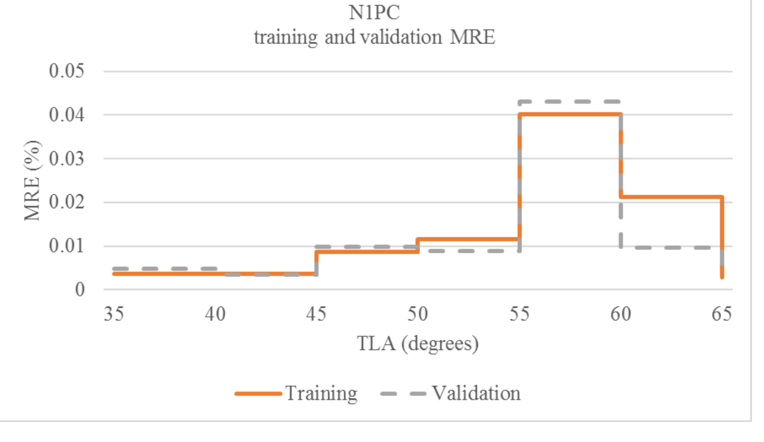


Fig. 7 – NN1 training and validation MRE comparison

The second set of experiments combined both the NN1 and NN2 blocks of the NN engine model. The outputs of the NN2 block are the engine net trust (FNC), the engine fuel flow

(WFC) and the high-pressure compressor speed (N2PC). Fig. 8 to Fig. 9 show the NN2 training at TLA = 65 degrees for different altitudes and Mach numbers.

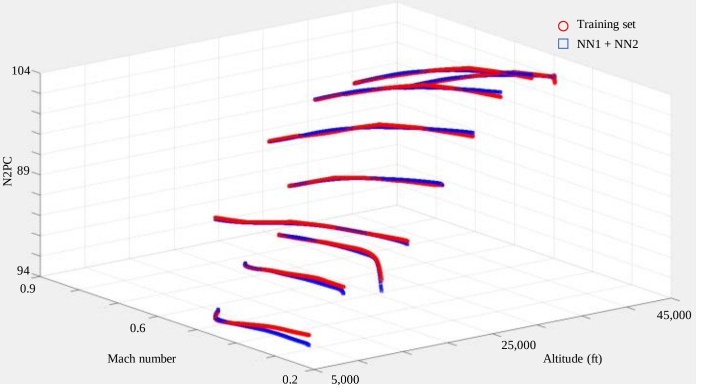


Fig. 8 – NN2 training: N2PC learning given altitude, Mach number, and TLA = 65 degrees

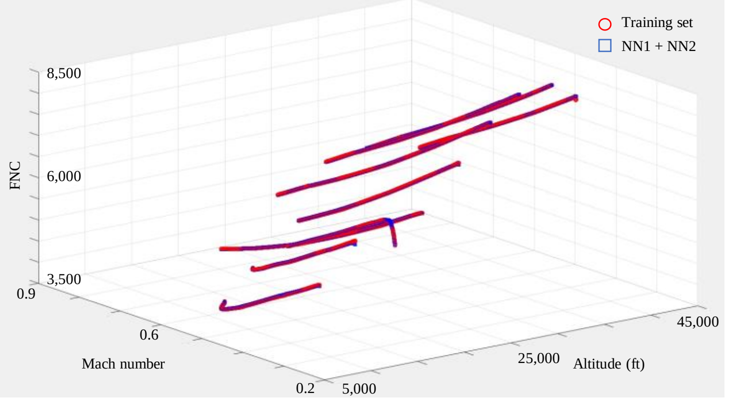


Fig. 9 – NN2 training: FNC learning given altitude, Mach number, and TLA = 65 degrees

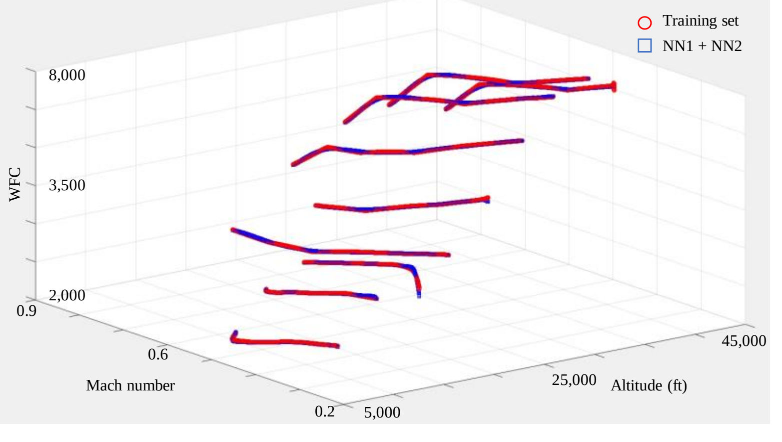


Fig. 10 – NN2 training: WFC learning given altitude, Mach number, and TLA = 65 degrees

Again, the learned output values mostly overlapped the values from the training set. The MRE of the NN engine model combining NN1 and NN2 are given in Table 4.

Table 4 – NN engine model training and validation MRE

| TLA degrees | N2PC Training (%) | N2PC Validation (%) | FNC Training (%) | FNC Validation (%) | WFC Training (%) | WFC Validation (%) |
|---|---|---|---|---|---|---|
| 35 | 0.0976 | 0.0721 | 0.08 | 0.0389 | 0.1165 | 0.0722 |
| 40 | 0.0975 | 0.0814 | 0.3408 | 0.6185 | 0.4501 | 0.7304 |
| 45 | 0.0759 | 0.0635 | 0.1209 | 0.2158 | 0.4191 | 0.3711 |
| 50 | 0.000001 | 0.2077 | 0.0000015 | 0.073 | 0.0000018 | 0.1233 |
| 55 | 0.5601 | 0.8132 | 1.54 | 1.5645 | 3.4273 | 3.2867 |
| 60 | 0.5674 | 0.6794 | 0.4058 | 0.4576 | 1.6301 | 1.5323 |
| 65 | 0.0473 | 0.0233 | 0.0479 | 0.0811 | 0.1363 | 0.1124 |

For the NN engine model, the highest prediction errors are obtained at TLA = 55 degrees. However, the MRE differences are quite small: 0.82% for N2PC, 1.56% for FNC and 3.29% for WFC. Fig. 11 to Fig. 13 present graphical comparisons of the learning and prediction MRE. It is observed that the MRE differences in learning and prediction MRE never exceed 0.3% for all TLA values.

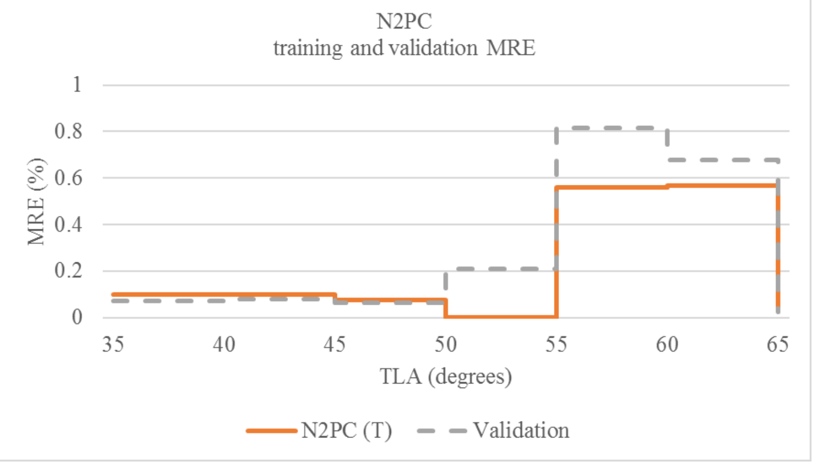


Fig. 11 – N2PC training and validation comparison

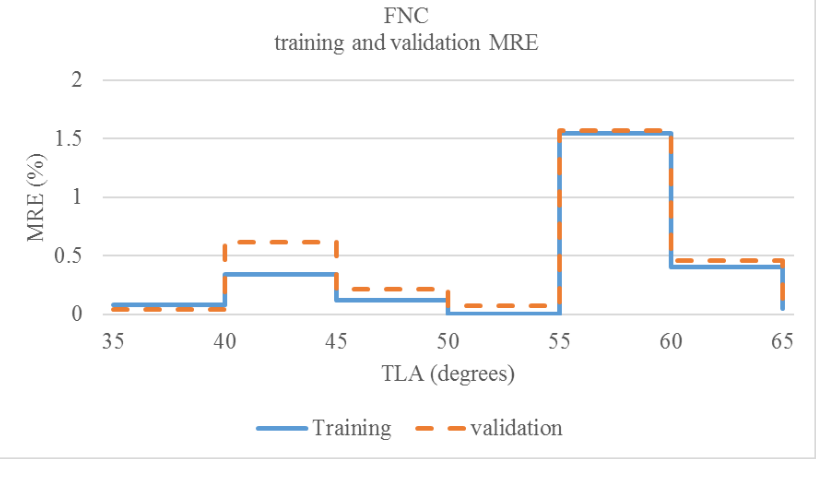


Fig. 12 – FNC training and validation comparison

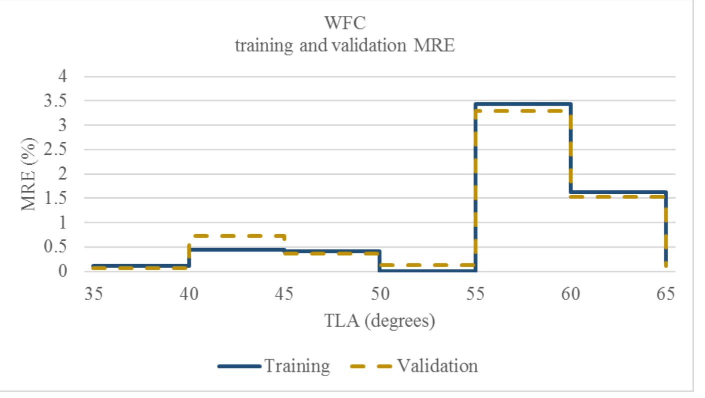


Fig. 13 – WFC training and validation comparison

Lastly, Table 5 provides a comparison between the NN model prediction and previously published results. For the engine net trust FNC, the NN model produced comparable MRE to the analytical model of Ghazi et al. Interestingly, the NN model outperformed models by Roux and Torenbeek in fuel flow prediction. The accurately of the NN model is an indication that engine non-linearity behavior can indeed be modeled by the NN approach.

Table 5 – NN engine model performance compared to previously published results

| | Proposed NN model | Analytical model [2] | Generic model [3] |
|---|---|---|---|
| FNC | 1.56% | 1.58% | Between 6% to 14% depending on engine type |
| | Proposed NN model | Roux's model [7] | Torenbeek's model [8] |
| WFC | 3.42% | 8% | 6.6% |

## 5. CONCLUSIONS

Accurate engine models are critical to evaluate aircraft performances and machine learning approach can contribute in the refinement of engine parameters. Through appropriate neural network training, it is possible to adequately predict engine net thrust and fuel consumption bypassing complex analysis and development. The resulting engine model is encapsulated within the network's configuration expressed in terms of neuron interconnection and weighting values. Thus, network configuration determination is a significant task in the modeling process. The usual ad-hoc configuration selection can be replaced by an automated procedure such as the extended great deluge algorithm. Neural network modeling can achieve engine parameters prediction under cruise flight conditions. Additional research encompassing climbing and landing phases is needed in order to yield a functional model capable of representing engine behavior during different phases of aircraft operation.

## ACKNOWLEDGEMENTS

This work is part of the Canada Research Chair in Aircraft Modeling and Simulation Technologies research program performed at the Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity Research (LARCASE). The Research Aircraft Flight Simulator (RAFS) was obtained by Dr Ruxandra Botez under a grant by the Canadian Foundation of Innovation, and le Ministère du Développement Économique, de l'Innovation et de l'Exportation of Québec (MDEIE), and from contributions by CAE Inc.

## REFERENCES

[1] J. D. Mattingly, W. H. Heiser, and D. H. Daley, *Aircraft engine design*, Washington, D.C: American Institute of Aeronautics and Astronautics, 1987.

[2] J. C. Wanner, *Dynamique du vol et pilotage des avions*, Châtillon, 1983.

[3] G. Ghazi, R. Botez and A. J. Messi, Cessna Citation X Engine Model Identification from Flight Tests, *SAE International Journal of Aerospace* **8**, no. 2: 203-213, 2015

[4] L. F. Rodriguez, R. M. Botez, *Civil turbofan engines thrust generic model*, IEEE Industrial Electronics Conference (IECON 2012), pp. 5444-5450, 2012.

[5] E. Torenbeek, *Synthesis of Subsonic Airplane Design*, Delft University Press, 1982.

[6] J. D. Mattingly, *Elements of Gas Turbine Propulsion*, McGraw-Hill series in mechanical engineering, 1996.

[7] E. Roux, *Pour une approche analytique de la dynamique de vol*, Ph. D. thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace. France, 2002.

[8] M. Zaag, 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*, in AIAA Modeling and Simulation Technologies Conference, AIAA 2017-194, 2017.

[9] G. Kouba, R. M. Botez, and N. Boëly, Fuzzy Logic Method Use in F/A-18 Aircraft Model Identification, *Journal of aircraft* **47**, No. 1: 10-17, 2010.

[10] S. De Jesus Mota, R. M. Botez, *New identification method based on neural network for helicopters from flight test data*, in AIAA atmospheric flight mechanics conference, AIAA 2009-5938, 2009.

[11] N. Boëly, R. M. Botez, 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* **225**, no. 5: 559-574, 2011.

[12] N. Boëly, 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* **21**, no. 11: 1759-1765, 2010.

[13] S. S. Haykin, *Neural Networks and Learning Machines*, New York: Prentice Hall/Pearson, 2009.

[14] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd ed. Wiley Publishing, 2007.

[15] M. I. A. Lourakis, A. A. Argyros, Is Levenberg-Marquardt the Most Efficient Optimization Algorithm for Implementing Bundle Adjustment?, In *Proceedings of the 10th IEEE International Conference on Computer Vision*, vol. **2**, pp. 1526-1531, 2005.

[16] D. K. Chaturvedi, *Soft Computing Techniques and its Applications in Electrical Engineering*, Springer-Verlag, 2008.

[17] A. B. Mosbah, F. Salinas, R. M. Botez and T. Dao, New methodology for wind tunnel calibration using neural networks-EGD approach, *SAE International Journal of Aerospace* **6**, no. 2: 761-766, 2013.

[18] E. Burke, M. Gendreau, M. Hyde, et al., Hyper-heuristics: a survey of the state of the art, *J Oper Res Soc* **64**, no. 2: 1695–1724, 2013.

[19] A. F. El-Sayed, *Performance Parameters of Jet Engines*, In: Fundamentals of Aircraft and Rocket Propulsion, Springer, London, 2016.

[20] A. J. Volponi, Gas Turbine Parameter Corrections, *ASME. J. Eng. Gas Turbines Power* **121**, no. 4: 613-621, 1999.

[21] D. G. Hull, *Fundamentals of airplane flight mechanics*, Berlin: Springer, 2010.