Optimization methodologies study for the development of prognostic artificial neural network

ISSN 1590-8844
International Journal of Mechanics and Control, Vol. XX, No. XX, 20XX

OPTIMIZATION METHODOLOGIES STUDY FOR THE DEVELOPMENT OF PROGNOSTIC ARTIFICIAL NEURAL NETWORK

Giuseppe Petti Gaetano Quattrocchi Matteo D.L. Dalla Vedova Paolo Maggiore

Department of Mechanical and Aerospace Engineering, Politecnico di Torino

ABSTRACT

In this work, we discuss the implementation and optimization of an artificial neural network (ANN) based on the analysis of the back-EMF coefficient capable of making electromechanical actuator (EMA) prognostics. Starting from the pseudorandom generation of failure values related to static rotor eccentricity and partial short circuit of the stator coils, we simulated through a MATLAB - Simulink model the values of currents, voltages, position and angular velocity of the rotor and thanks to these we obtained the back-electromotive force which represents the input layer of the ANN. In this paper, we will turn our attention to optimizing the hyperparameters which influence supervised learning and make it more performing in terms of computational cost and complexity. The results are satisfactory dealing with the number of examples present in the available dataset.

Keywords: EMA, prognostic, optimization, ANN, training

1 INTRODUCTION

Aircraft manufacturers are increasingly concentrating on the use of electromechanical actuators as they allow a considerable simplification of on-board systems with respect to alternative technologies used at this time. However, they are a new technology and for this reason, engineers are even investigating failure modes that characterise such systems even though these are already being used in the industrial automation and manufacturing sectors [1]. They must constantly monitor the prognostic and health management (PHM) system and have knowledge of the component’s failures mode because they strongly impact safety and the costs related to the maintenance of the aircraft. The goals are both to maximize equipment up time and to reduce support - operating costs (O&S) and life cycle total ownership costs (TOC) [2]. For example, primary flight controls are designed with a safe-life approach as they are a critical part of the aircraft that must not fail in flight. The future of aviation is certainly based on the "more electric" philosophy. [3] [4]

Electricity is the only indispensable energy source for an aircraft, it is used for actuation systems, for wing ice protection systems, for environmental control systems, and for fuel pumping [5] . Nowadays, the remote hypothesis of "full electric aircraft" is still under study and there are several queries to be resolved yet. However, there are many aspects of electricity that can still be improved in terms of weight, volume cost and reliability.

The objective of this paper is to study a methodology for the recognition of faults within the EMAs. This logic, based on artificial intelligence, principally on artificial neural networks, provides to measure the remaining useful life of the system components starting from a training dataset. The neural network learns autonomously the relationships that link the quantities given as input with those in output [6]. Once trained, this allows to estimate the parameters related to the health status of the system only by measuring some values that we will see.

The model has a reliability that will be discussed and we have improved it by considering some techniques to reduce both underfitting and overfitting problems as well as a deeper network even though it does not always turn out to be an advantage [7] [8]. For this reason, optimizing the work is important.

2 METHODOLOGY

2.1 CASE STUDY

A servo actuator is a device employed to regulate the position or velocity of a mechanical element.
In particular an electromechanical actuator (EMA) converts electrical energy into mechanical energy necessary for the movement of the aircraft's control surfaces.

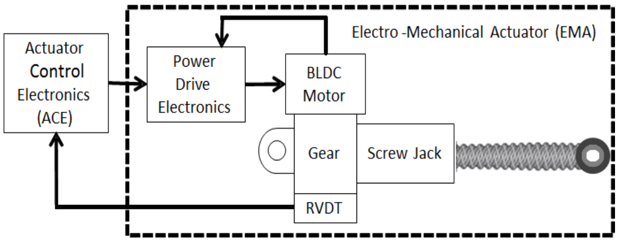

Figure 1: Schematic of the considered EMA [9]

EMAs are used for the implementation of secondary control surfaces such as airbrakes, slats, flaps and the goal is to use them as primary control surfaces because they bring a significant simplification of the aircraft systems in terms of operating costs, maintenance interventions, amount of necessary redundancies, safety and reliability [10].
As shown in Fig. 1, a typical EMA used in a primary flight control is composed by [11]:

- Actuator Control Electronics (ACE): calculate the error deriving from the difference of two signals, i.e. the FBW signal produced by the pilot as a command and the position feedback signal delivered by the sensors.
- Power Drive Electronics (PDE): DC current powers this module and converts the error deriving from the ACE into a modulated 3-phase AC current that will guide the electric motor.
- BLDC Motor: Brushless DC motor converts electrical power into mechanical power.
- Gear: mechanical transmission that converts low torque and high angular speed into high torque and low angular speed to move the actuator.
- Screw Jack: a device adopted to convert rotational motion into linear motion.
- RVDT: Rotary Variable Differential Transducer is an angular position acquisition sensor that allows the closure of the feedback loop [12].

Prognostic purpose is to predict when a certain component loses its functionality and is not able to be fully operative [2] [13]. It is based on the analysis of all the possible failure modes [14]. An EMA can be affected from mechanical, structural and electrical fault modes [15] [16] [17] [18] [19]. In this paper, we discuss only faults related to the electric motor, in particular the partial short circuit of the stator coils expressible through three parameters N_a, N_b, N_c (each for each phase of the electric motor) and the static eccentricity of the rotor through ξ and ϕ, respectively amplitude and phase in polar coordinates [20] [21] [22].
For each example, we generate a fault vector with five values pseudo-randomly using an algorithm known as the Latin hypercube. Exponential spacing is considered for analysing prognostic rather than diagnostic cases so that pseudorandom values satisfy these two rules:

$$\sqrt{N_a^2 + N_b^2 + N_c^2} \leq 0.5 \qquad (1)$$

$$\xi \leq 0.5 \qquad (2)$$

We use each fault vector to generate a simulation and log significant data such as motor currents, voltages, position and angular velocity and we use them to calculate the back-electromotive force coefficient for each phase. [23] [24] [25]

$$k_{bemf,j}(t) = \left(V_j(t) - R_m i_j(t) - L_m \frac{di_j(t)}{dt}\right) \cdot \frac{1}{\dot{\theta}_m} \qquad (3)$$

where j represents one of the three phases and R_m and $L_m$are motor nominal phase resistance and inductance. Finally, we calculate the equivalent for a single phase:

$$k_{bemf} = \sum_{j=1}^{3} |k_{bemf,j}| \qquad (4)$$

In the nominal operating condition, in which faults are not present, this value is unitary. Later on, the signal is appropriately sampled and each curve is expressed by a finite number of points that characterize the features of the artificial neural network.

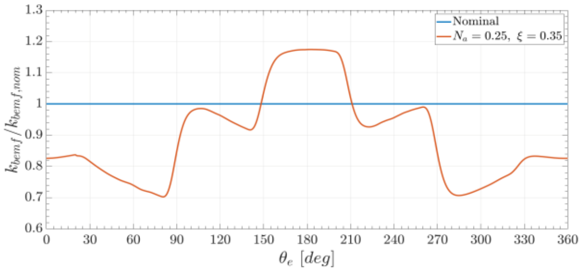

Figure 2: Back-EMF coefficient comparison in nominal and faulty condition

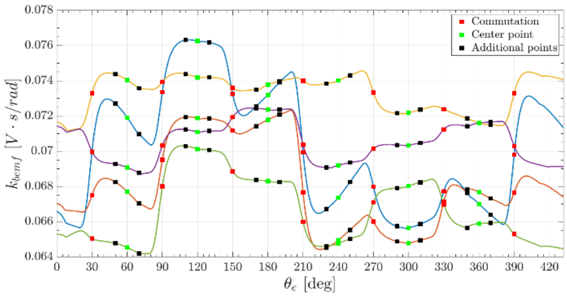

Figure 3: Sampling of five examples of back-EMF signals

We initially considered a simple neural network consisting of a single hidden layer formed by ten neurons and our task is to improve the prediction. [26]

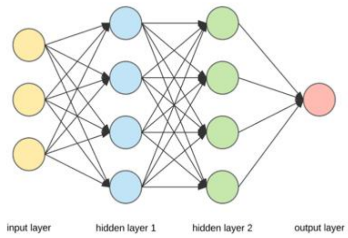

Figure 4: ANN scheme

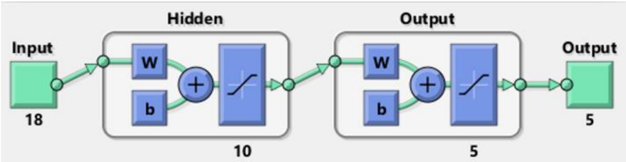

Figure 5: Initial ANN

The dataset is divided into three parts, namely the training, validation and test set. For each of them the cost function mse is calculated which allows to evaluate the network performances.

$$mse_{tr}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \quad (5)$$

Where $h_\theta\left(x^{(i)}\right)$ represents the predicted value while $y^{(i)}$ represents the real value and m is the number of examples [27]. The dataset is made up of 3000 examples and the training function uses the Levenberg-Marquardt algorithm as update rule for weights and biases. Furthermore, the activation function of the hidden layer is the satlins and with this configuration we get all the mean square errors equal to about 7.84 e-04 with 2 seconds for the training time after 77 epochs. MATLAB stops the simulation as it does not notice a significant improvement in the mse values. [25]

Starting from this result we change some hyperparameters of the network: number of features, neurons, hidden layers, activation function and training function.

We conducted the parametric analysis by running three simulations for each case study and considering the average value of the three results, rejecting those out of range.

We estimate a value that represents the performance of the network with (6). The values of K_1 and K_2 reward networks with better mean squared error (mse) at the expense of a longer training time. We execute therefore various simulations inserting 'for' inside the code in way to run it one time and to return to visualize results only when all the simulations are ended. The results we get are specific to the software and hardware on which the training is performed, i.e. MATLAB 2020a, NVIDIA GeForce GTX 1050 Mobile with CUDA technology, Intel Core i5-8300H and 16 GB of DDR3-2666 Mhz RAM.

2.2 VARIATION OF HYPERPARAMETERS

In this section we assume the initial neural network constituted by:

- Features: 18 points of back-EMF coefficient sampling;
- Hidden layers and neurons: one with 10 neurons;
- Training function: Levenberg-Marquardt algorithm;
- Activation function: satlins.

We vary each hyperparameter of the network individually without modifying the others and we analyse the network performance. We then vary the number of neurons, the number of layers considering one, two and three hidden layers, consider different training functions and different activation functions already pre-implemented in MATLAB. The choice of this number of neurons and the number of layers as we will see is related to the results obtained between the different cases.

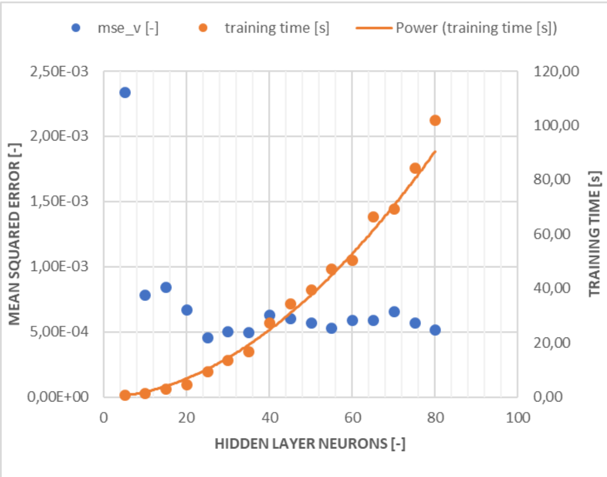

Figure 6: mse and training time with variation in the neurons number

.

$$Perf = \frac{1}{K_1 \frac{mse_v - mse_v.min}{mse_v.max - mse_v.min} + K_2 \frac{training\_time - training\_time.min}{training\_time.max - training\_time.min}} \qquad K_1 = 4, \qquad K_2 = 1 \qquad (6)$$

As an example, during the tests we thought that it did not make sense to complicate the neural network with 1000 neurons in a hidden layer since with 80 neurons we did not notice a significant increase in performance but only a greater time related to training.

In Figure 6 we vary the number of neurons in the network and we notice that an increase in the latter leads to training time growth. Instead, we do not see a significant improvement in network performance when we go beyond the number of 30 neurons and it is visible by an approximately constant trend in the mse relative to the validation set with slight fluctuations due to initialization of the weights and biases.

The next step is to change the number of layers and we do this in Figure 7 and Figure 8. In both cases we note that we do not have a significant improvement in mse, however we achieve performance on the order of $1 \cdot 10^{-4}$ with relatively simple network configurations where the number of neurons is not overly large.

For example, considering the configuration with three hidden layers 15-15-15 we see an increase in training time, however this was not matched by an improvement in mse. As a result, this can lead to overfitting problems.

In Figure 9, the activation function that allow to obtain the smaller values of mse are the 'satlins', the 'tansig' and the 'elliot2sig'. The same applies to the algorithms for updating weights and bias: we can reach the lowest mse with Levenberg-Marquardt algorithm and Bayesian regularization as we see in Figure 10.

3 RESULTS

Having seen how each individual hyperparameter affects the neural network, we present the best hyperparameter trade-offs. We combine them to obtain the lowest mse and for this reason the choice falls on the configurations of a single hidden layer with 25,30 neurons, two hidden layers with 10,20 neurons, trainlm and trainbr as training functions and satlins, tansig and elliot2sig as activation functions.

As visible in Figure 11 and Figure 12, we obtaine the best performance, evaluate through the (6), with the configuration formed by two hidden layers with 10 neurons for each layer, the Levenberg-Marquardt algorithm as training function and the tansig as activation function. This confirmes our initial theory, that is a very simple network does not particularly suffer from overfitting problems and it is the best solution having this dataset available. Furthermore, we achieve the lowest training times by configurations with fewer neurons while in general the mse brings better results with the tansig activation function.

In Figure 13 we compare cost functions of the network previously presented with these obtained and we notice a decrease of all mse, that express a greater accuracy of the ANN. Specifically, the slight increase in training time is completely negligible.

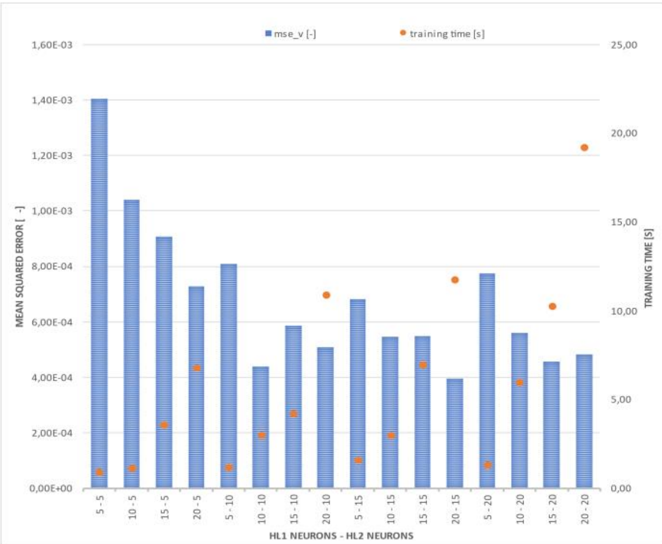

Figure 7: mse and time with variation in the numbers of neurons with two hidden layers

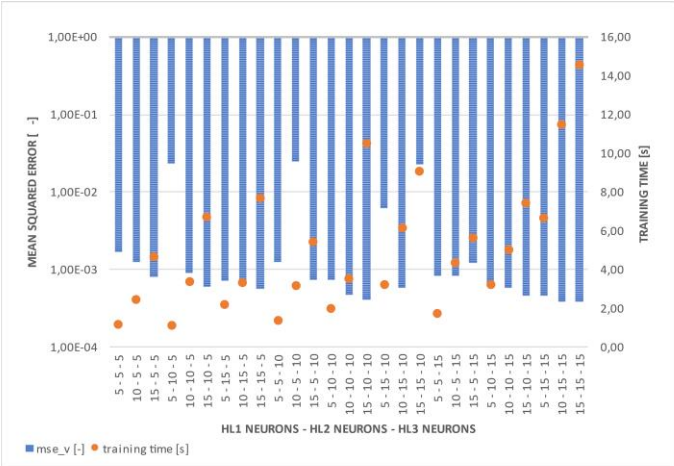

Figure 8: mse and time with variation in the numbers of neurons with three hidden layers

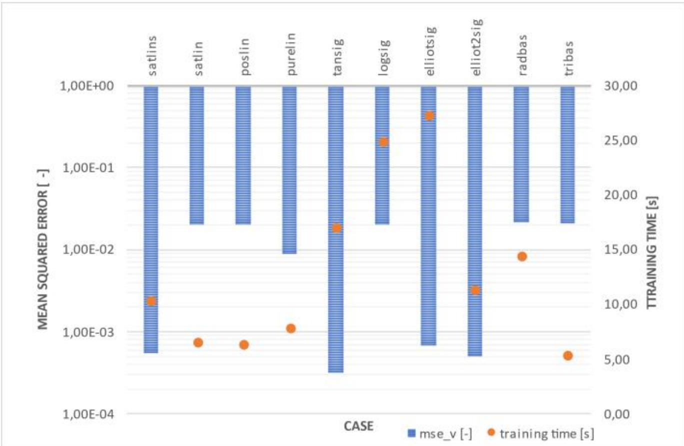

Figure 9: mse and time with variation of the activation function

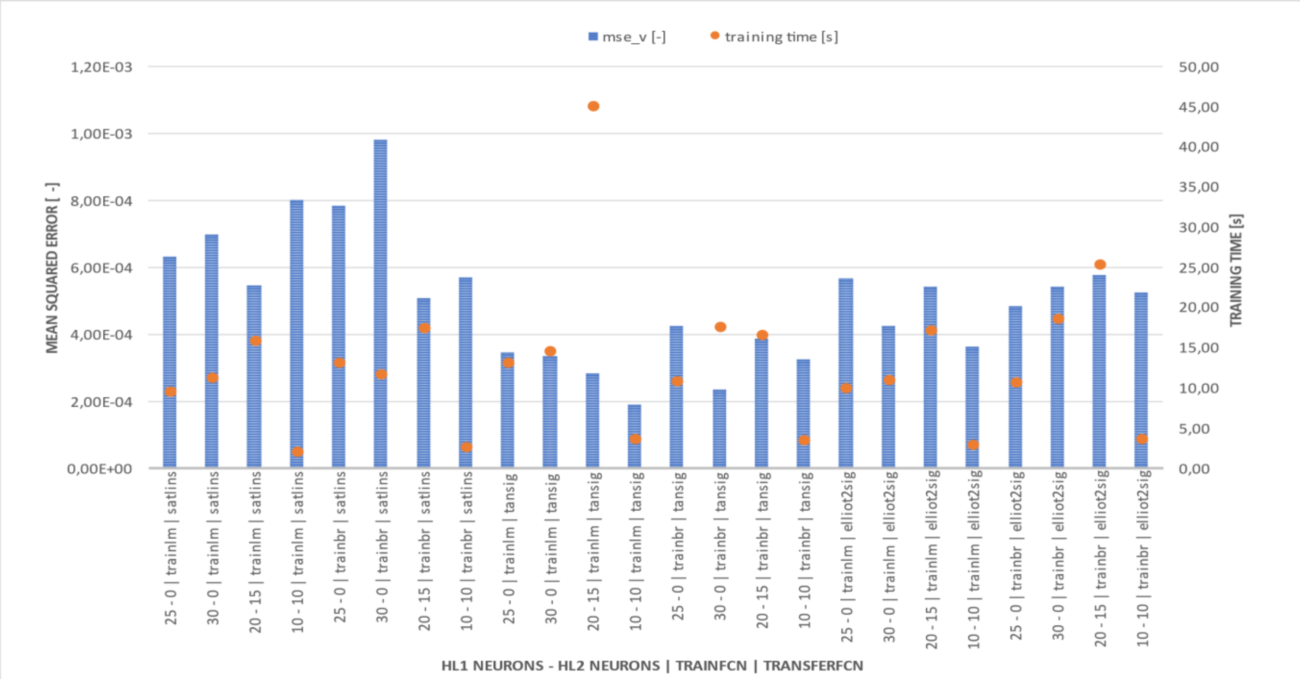

Figure 11: mse and time for best mse combinations

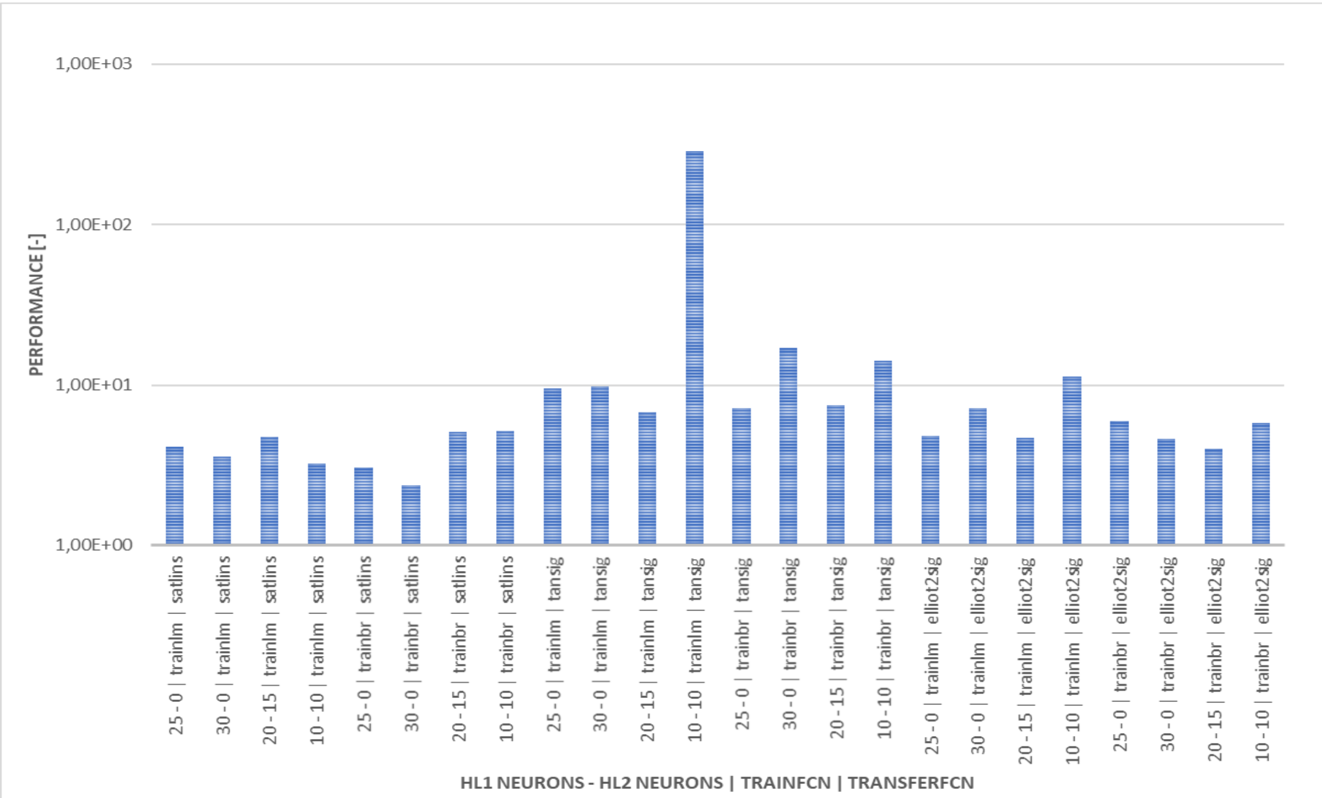

Figure 12: Performance for best mse combinations

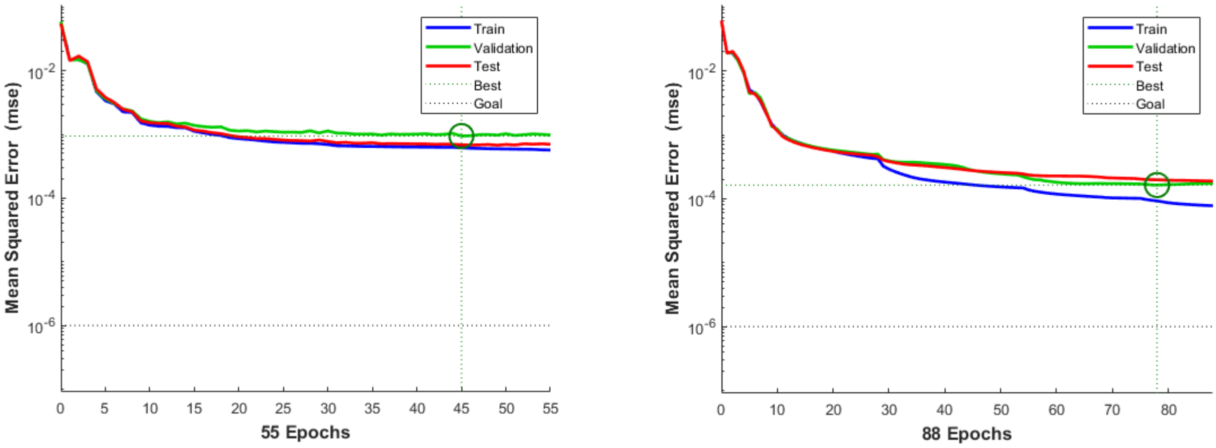

Figure 13: Cost functions over epochs. On the left ten neurons in one hidden layer, on the right ten neurons in the first hidden layer and ten in the second.

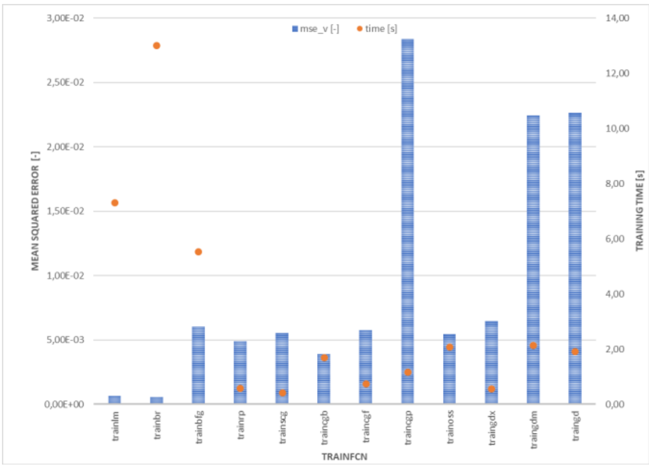

Figure 10: mse and time with variation of the training function

	Before	**After**
mse_{tr} [−]	5.46E-04	1.34E-04
mse_{v} [−]	8.07E-04	1.92E-04
$training\ time$ [s]	2.14	3.77

Table I: ANNs comparison before and after optimization

4 CONCLUSIONS

The study shows that parametric analysis improved the predictive capabilities of the artificial neural network used for prognostic purpose of an electromechanical actuator.
As we see in Table I, we pay the lower mean squared error in terms of the increased network training time, however it is now possible to predict RUL with greater precision from a PHM perspective. Moreover, since the ANN is very simple, there are no overfitting problems and this increases the overall reliability of the network.
The work done in this paper can be seen as a small step towards the implementation of EMA as primary flight controls actuators.
The next step could be the expansion of the training dataset, the use of other types of neural networks to see if and how they work for solving the specific problem.
Remind that in this study, we analyse only the faults due to short circuit of the stator coils and static eccentricity of the rotor. Alternative types of faults can also occur as discussed.
Finally, working on low-level programming language can help the computational cost and test the implementation on On-Board Computers (OBCs) in prospect of large-scale commercialization.

5 REFERENCES

[1] Byington C.S., Watson M., Edwards D. and Stoelting P., *A model-based approach to prognostics and health management for flight control actuators,* 2004.

[2] Vachtsevanos G., Lewis F., Roemer M., Hess A. and Wu B., *Intelligent Fault Diagnosis and Prognosis for Engineering Systems,* Wiley, 2006.

[3] Quigley R.E.J., *More Electric Aircraft,* pp. 906-911, Proceedings of the Eighth Annual IEEE Applied Power Electronics Conference (APEC '93), 1993.

[4] Howse M., *All-Electric Aircraft,* vol. 17, no. 4, pp. 35-37, Power Engineer, 2003.

[5] Wheeler P. and Bozhko S., *The more electric aircraft: Technology and challenges,* IEEE Electrif., 2014.

[6] Géron A., *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems,* 2019.

[7] Tripathi M., *Underfitting and Overfitting in Machine Learning,* Datascience Foundation, 2020.

[8] Burkov A., *The Hundred-Page Machine Learning Book,* 2019.

[9] Dalla Vedova M. D. L., Maggiore P. and Pace L., *A new prognostic method based o simulated annealing algorithm to deal with the effects of dry friction on electromechanical actuators,* vol. 9, no. 236-245, International Journal of Mechanics, 2015.

[10] Van de Bossche D., *More Electric Control Surface Actuation, a Standard for the Next Generation of Transport Aircraft,* 2004.

[11] Dalla Vedova M.D.L. and Berri P.C., *Optimization techniques for prognostic of on-board electromechanical servomechanisms affected by progressive faults,* vol. 12, no. 4, pp. 160-170, International Review of Aerospace Engineering, 2019.

[12] Battipede M., Dalla Vedova M. D. L., Maggiore P. and Romeo S., *Model based analysis of precursors of electromechanical servomechanisms failures using an artificial neural network,* IAA Modeling and SimulationTechnologies Conference, 2015.

[13] Santos L. and Melicio R., *Stress, Pressure and Fatigue on Aircraft Maintenance Personal,* vol. 12, no. 1, pp. 35-45, International Review of Aerospace Engineering (IREASE), 2019.

[14] Byington C.S., Watson W., Edwards D. and Stoelting P., *A Model-Based Approach to Prognostics and Health Managment for Flight Control Actuators,* IEEE Aerospace Conf., 2004.

[15] Churn D. P. M., Maxwell C. J., Schofield N., Howe D., Powell D. J. , *Elkctro-hydraulic actuation of primary fliyht control surfaces,* IEE Colloquium on All Electric Aircraft,1998.

[16] Kenio T. and Nagamori S., *Brushless Motors: Advanced Theory and Modern Application,* 2003.

[17] Chesley J.C., *Handbook of Reliability Prediction Procedures for Mechanical Equipment,* 2011.

[18] Weiss J., *Control Actuation Reliability and Redundancy for Long Duration,* Proceedings of the 2014 Underwater Intervention Conference (UI), 2014.

[19] Mohite M.A. and Kothavale B.S., *Experimental and Analytical Analysis of Crack Propagation on Spur Gear Due to Bending Fatigue and Service Life Estimation,* pp. 88-96, International Review of Mechanical Engineering (IREME), 2018.

[20] Berri P.C., Dalla Vedova M. D. L. and Maggiore P., *On-board electromechanical servomechanisms affected by progressive faults: proposal of a smart GA model-based prognostic approach,* pp. 839-845, 2017.

[21] Dalla Vedova M.D.L., Germanà A., Berri P.C. and Maggiore P., *Model-based fault detection and identification for prognostic of electromechanical actuators using Genetic Algorithms,* vol. 6, no. 9, 2019.

[22] Berri P.C., Dalla Vedova M. D. L. and Maggiore P., *A Smart Electromechanical Actuator Monitor for New Model-Based Prognostic Algorithms,* pp. 59-66, 2016.

[23] Stevens B. L., Lewis F. L. and Johnson E. N., *Aircraft Control and Simulation Third Edition,* Wiley, 2016.

[24] Dalla Vedova M. D. L. and Borello L., *Dry friction discontinuous computational algorithms,* 2014.

[25] Quattrocchi G., Berri P. C., Dalla Vedova M. D. L. and P. Maggiore, *Innovative Actuator Fault Identification Based on Back Electromotive Force Reconstruction,* 2020.

[26] Nielsen M., *Neural Networks and Deep Learning,* 2019.

[27] Ng A., *Machine Learning,* 2020.