Machine Learning Based Prognostics of On-Board Electromechanical Actuators

Machine Learning based Prognostics of On-Board Electromechanical Actuators

Edmondo MINISCI[1[0000-0001-9951-8528]] , Matteo D.L. DALLA VEDOVA[2[0000-0002-3124-2198]], Parid ALIMHILLAJ[3[]], Leonardo BALDO[2[0000-0001-5073-4166]], and Paolo MAGGIORE[2[0000-0003-0218-0730]]

[1] University of Strathclyde, Glasgow G1 1XJ, United Kingdom
[2] Politecnico di Torino, Turin 10129, Italy
[3] Polytechnic University of Tirana, Tirana 1000, Albania
edmondo.minisci@strath.ac.uk, matteo.dallavedova@polito.it, parid.alimhillaj@fim.edu.al, leonardo.baldo@polito.it, paolo.maggiore@polito.it

Abstract. This paper presents a novel machine learning-based prognostic approach for on-board electromechanical actuators. The study is centered around overcoming the limitations of model-based prognostic frameworks that rely on expensive optimization processes. Machine learning techniques were employed to map system signal characteristics directly into parameters related to fault simulation. A first test, utilizing only five of eight implemented fault types, demonstrates a highly promising potential of artificial neural networks to predict and detect faults with minimal error. A second test expand the investigation to include all fault types and provides an analysis of the model's robustness, error rates, and computational costs. The practical outcome of the work is a viable real-time solution for fault detection and characterization in electromechanical actuators, highlighting the efficiency and effectiveness of machine learning techniques for industrial applications.

Keywords: Machine Learning, Prognostics, Electro-Mechanical Actuators, Reliability, Fault Implementation, Failure Analysis

1 Introduction

Electromechanical actuators (EMAs) play a pivotal role in modern technological systems, converting electrical energy into mechanical energy and performing precise motion control tasks. As crucial components in industries such as manufacturing, automotive, and robotics, they underpin the smooth functioning of a myriad of systems. Their importance is even further underscored in the aerospace and aeronautical sectors where EMAs are vital components in aircraft systems including flight control, landing gears, and engine control, with the contemporary trend of transitioning from hydraulic and pneumatic systems to "more electric aircraft" [1]. This transition is driven by the advantages of EMAs, such as weight reduction, simplified maintenance, and overall efficiency improvement[2,3].

These wide-ranging applications accentuate the necessity for early fault detection and system reliability. Failures can lead to substantial financial costs, operational delays, or catastrophic consequences, particularly in safety-critical aerospace applications, fully justifying the carried out exploration into machine learning-based prognostics for on-board EMAs.

Fault detection in EMAs can be conducted through model-based diagnostic methods and approaches, which involve complex and computationally heavy optimization processes[3,4]. These techniques require detailed models of the system and its faults, which can be a resource-intensive and intricate process, often requiring specific expertise, and necessitate exhaustive simulations and the application of heuristic search algorithms to match observed and simulated behavior, making them unsuitable for real-time fault detection and characterization.

Recent research on prognostics and fault detection has seen a shift towards data-driven approaches, notably machine learning techniques. These techniques have demonstrated the potential to identify complex patterns, allowing for the detection and characterization of faults in nonlinear systems like EMAs.

However, there is a persisting gap in the development of methods that facilitate quick and efficient real-time prognostics in EMAs. The carried out research aims to bridge the previous work about the use of metaheuristics by utilizing artificial neural networks (ANNs), a subset of machine learning techniques. ANNs can approximate any nonlinear function, and learn and generalize from provided data, showing great potential in various diagnostic/prognostics tasks. This approach aims to bypass the computationally intensive associated optimization processes and offer a more efficient solution for real-time fault detection in EMAs, especially critical in aerospace and aeronautical systems.

It will be shown that a well-trained and validated ANN model will provide quick and effective real-time prognostics characterization, and it is expected that the proposed approach will significantly contribute to the improvement of system reliability and safety, particularly in industries relying heavily on EMAs.

The rest of the paper is organized as follows. Sec. 2 will detail the models and methods used for this work; Sec. 3 will present and discuss the results obtained for a simplified and a more complex case; and a final section will contain general conclusions and new directions for future research.

2 Models and Methods

In this section, the EMA model and the way the faults have been implemented are detailed. Moreover, the general structure of the artificial neural network used in this work is presented.

2.1 EMA Model and Fault Implementation

Previous work on model-based prognostics required the implementation of two EMA models: a high-fidelity reference model (RM) and a low fidelity monitoring model (MM). The RM was used to create the pseudo-real signals, without having to rely on

test data, while the MM was the model used by the heuristic methods to find the set of fault parameters related to the signal.

Both models, detailed in [3,4,5], implement four particular faults, usually showing a progressive growth: backlash (BLK) and dry friction (FST) affecting the mechanical transmission because of worn components, partial coil-to-coil short circuit in one of the BLDC stator phases (Na) and a drift of the proportional gain of the PID controller. The faults are defined by a set of top level parameters (TLPs), grouped inside a $\boldsymbol{k}$ vector. Each component of the vector is linked and can be traced back to a failure in the EMA, and by modifying these parameters, in the interval [0, 1], it is possible to simulate different failures/degradations. The 8 parameters are as follows:

- k_1: Dry friction.
- k_2: Backlash.
- k_3, k_4, k_5: Short circuit (SC), phases A, B, and C, respectively.
- k_6: Static eccentricity modulus in the rotor.
- k_7: Static eccentricity phase in the rotor.
- k_8: Proportional gain drift (PGD).

The values of k's giving the nominal condition, i.e. no deterioration, are: $\boldsymbol{k}_N = [0, 0, 0, 0, 0, 0, 0.5, 0.5]^T$.

In some previous works [3,4], the fault identification was performed by finding the values of k's, which gave the best match between the MM and RM current outputs. As introduced in the previous section, the optimization process associated with the search, prevented the approach to be used in real time. In this work, the aim is to implement and test a machine learning based approach that could be used in real-time.

2.2 Machine Learning based Approach

Machine Learning (ML) exploits the information contained in the available data to develop systems effectively performing specific tasks, such as regression and classification. For this work, the focus is on the regression application, and ML is considered to build a map linking the current signal to the k parameters, in a sort of inverse modelling. While in the previously used model-based approach the MM was included into an optimization loop to find the input values of the model (the vector $\boldsymbol{k}$) that was producing the current signal matching the reference coming from the RM, here $\boldsymbol{k}$ becomes the output of the ML model, where the input is the current signal (properly sampled). The ML model is very fast to interrogate (hence the real time estimate), but the training phase can be relatively expensive and needs to be performed off line.

Artificial Neural Network Structure

Generic multi-layer perception (MLP) feed-forward Artificial Neural Networks (FF-ANN) with one hidden layer, also known as shallow neural networks (as opposite to a deep neural networks, containing more than one hidden layer) are used in this work to learn the mapping from the current signal data to the *k* values. One-layer MLPs like those used herein are universal function approximators $\boldsymbol{f_g(x)} : R^{n_d} \rightarrow R^{n_o}$ [6], where n_d is the size of the input vector $\boldsymbol{x}$, and n_o is the size of the output vector function $\boldsymbol{f_g}$. The general matrix-vector definition of $\boldsymbol{f_g}$ is:

$$\boldsymbol{f_g}(\boldsymbol{x}) = A2(\boldsymbol{b}^{(2)}+\boldsymbol{W}^{(2)}(A1(\boldsymbol{b}^{(1)}+W^{(1)}\boldsymbol{x}))) \quad (1)$$

where $W^{(1)}$ is a weight matrix of size ($N \times n_d$) and N is the number of neurons in the hidden layer, $\boldsymbol{b}^{(1)}$ is a bias term represented by a column vector of length N, $\boldsymbol{W}^{(2)}$ is a weight matrix of size ($n_o \times N$), $\boldsymbol{b}^{(2)}$ is a bias term represented by a column vector of length n_o, and $A1$ and $A2$ are the activation functions of the hidden layer and the output layer, respectively. The general schematic of a one-layer MLP feed-forward ANN is shown in Figure 1.

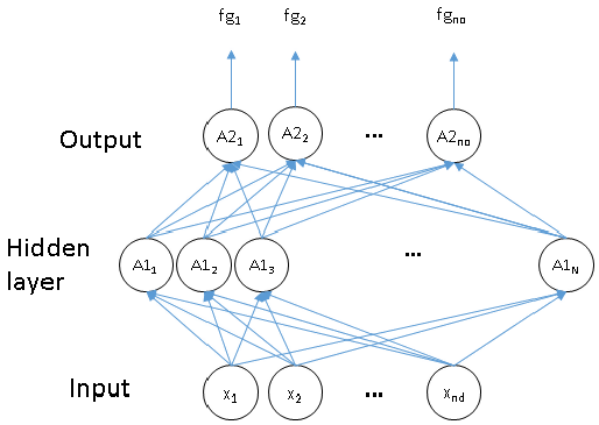

Fig. 1. General scheme of a multi-input multi-output MLP feed-forward ANN [7]

As in the vast majority of cases, for this work $A1$ is the hyperbolic tangent sigmoid function, and $A2$ is linear, and with this choice, function $\boldsymbol{f_g}(\boldsymbol{x})$ becomes:

$$\boldsymbol{f_g}(\boldsymbol{x}) = \boldsymbol{b}^{(2)}+\boldsymbol{W}^{(2)}(tanh(\boldsymbol{b}^{(1)}+W^{(1)}\boldsymbol{x})) \quad (2)$$

Given a set of Ns training samples $\{(\boldsymbol{x}_1,\boldsymbol{y}_1), \ldots, (\boldsymbol{x}_{Ns},\boldsymbol{y}_{Ns})\}$, where y_i is the observed response to the input $\boldsymbol{x}_i$, a learning algorithm seeks the values of the weight matrices and bias vectors that minimize the difference between part or all of the observed Ns sample responses $\boldsymbol{y}_i$ and the Ns responses given by $\boldsymbol{f_g}(\boldsymbol{x}_i)$. Gradient-based approaches are normally used to find the values of the weight matrices and the bias vectors by minimizing the fitting error of the responses. Since the responses are assumed to be smooth functions of the inputs $\boldsymbol{x}$ and the internal weights and biases, the gradients with respect to weights and biases of the difference between sample responses and ML approximations can be computed using the backpropagation method, using the chain rule for derivation. Once the gradients with respect to weights and biases for each layer are computed, the objective function expressing the level of fitting of the training data can be optimized.

If the ANN has a sufficiently high number of neurons or degrees of freedom in the hidden layer, a parameter proportional to the chosen number of neurons, the fitting error can be reduced to machine zero; this results in the system *overfitting* the training

data, and being possibly unable to generalize adequately its predictions to regions of the input space where there are no or insufficient training data.

In this work, the inputs are the time samples of the current signal, c_i, $i=1,\ldots,n_i=n_d$

3 Results and Discussion

As anticipated in the previous sections, the proposed approach has been applied to two different cases: one considering only 5 of the total 8 parameters implementing the degradations, and a complete one considering all the 8 parameters. What follows is a detailed description of the cases, the obtained results, and discussion considering the outcomes of both cases.

Given the already proven correspondence between the current signal of RM and MM, the monitoring model, MM, has been used to save computational time, for both cases and for all the tests.

In both cases, all the considered k parameters have been sampled in [0.25, 0.75], which is the most interesting range for prognostic: 0.25 means that the degradation is at the early stage, but significant enough to need monitoring; 0.75 means that the degradation is at advanced stage, and actions should be performed to avoid total fault. MM produces a current signal from $t=0s$ to $t=0.5s$, with a fine discretization, $\Delta t=1e\text{-}5s$. To reduce the input dimensionality of the ANN model, and with the reasonable idea that not all the points may be needed, a coarser discretization of the signal has been considered, with $\Delta t=5e\text{-}3s$, allowing to have $n_i=n_d=100$. Fig. 2 shows the difference between high and low frequency sampling for three random cases obtained when the first five k parameters are considered (simple case, as described in the following Sec. 3.1). For all the cases, *Sol1=[0.71481, 0.40819, 0.34196, 0.35228, 0.53386]*, *Sol2=[0.54777, 0.73226, 0.57659, 0.62445, 0.57678]*,and *Sol3=[0.62386, 0.73065, 0.25419, 0.30322, 0.39935]*, the low frequency sampling only cuts some of the peaks.

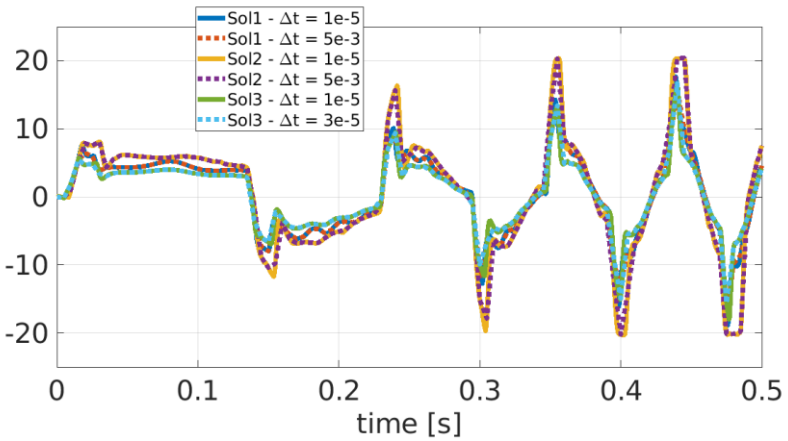

Fig. 2. Random examples of current signals with both high and low frequency sampling.

Moreover, in both cases, the available data (see the coming subsections) have been split 95% for training and 5% for validation, and the Levenberg-Marquardt (LM) backpropagation in the MATLAB implementation has been used.

3.1 Simple Case Results and Discussion

In this case, the first 5 of the total 8 *k* parameters have been considered (the other three *k* parameter keep the nominal values), that are:

- k_1: Dry friction.
- k_2: Backlash.
- k_3, k_4, k_5: Short circuit (SC), phases A, B, and C, respectively.

The design of experiment (DOE) to sample the 5-dimensional space combines a full regular grid with 5 nodes for each dimension (3125 samples), with a Latin Hypercube Sampling of 3000 points, giving *Ns = 6125*. To assess the effect of the number of neurons in the hidden layer, N, the ANN has been set with three different values: N = 40, N = 60, N = 80. The available data have been randomly split 95% for training and 5% for validation, and the Levenberg-Marquardt (LM) backpropagation in the MATLAB implementation has been used.

The three obtained ANNs have been tested against a sample of 100 random signals, i.e., obtained by randomly (uniform) sampling the *k* parameter space (the 100 samples are the same for all the three tests). The obtained statistical results are summarised in Table 1.

Table 1. Prediction errors on the 100 test signals for the case with five *k* parameters. Statistical values (mean value, *mean*, standard deviation, *std*, minimum, *min*, and maximum, *max*) reported for each k parameter, when N varies from 40 to 80.

	k	**mean**	**std**	**min**	**max**
N=40	1	7.79E-05	0.00087	-0.00235	0.002703
	2	-0.00091	0.004006	*-0.03322*	0.00649
	3	-0.00013	0.000925	-0.00318	0.002039
	4	-6.56E-05	0.001332	-0.00441	0.004419
	5	4.17E-05	0.001105	-0.00364	0.00226
N=60	1	-7.99E-05	0.000943	-0.00424	0.002331
	2	-0.0007	0.003556	-0.01496	*0.022563*
	3	-0.00014	0.000838	-0.00261	0.002538
	4	-3.32E-05	0.001214	-0.00367	0.002936
	5	-2.02E-05	0.001132	-0.0041	0.002534
N=80	1	6.96E-05	0.000986	-0.00388	0.002702
	2	-0.00032	0.002315	-0.00762	*0.011558*
	3	-0.00012	0.000774	-0.00189	0.001932
	4	-3.98E-05	0.001133	-0.00246	0.003673
	5	7.63E-06	0.001237	-0.00407	0.006334

The biggest errors for each ANN are highlighted, showing that k_2 is the most difficult parameter to map, consistently across the ANNs. However, the maximum error, never exceeds ~0.03 and is as low as ~0.01 when N=80, which is low enough to be of practical use: even an error of 0.03 would mean that in the worst case scenario the analyst would start tracking k_2 when the model maps $k_2=0.25$ while in reality it is already of 0.28, or that the prediction of the backlash entering in the red zone, i.e. $k_2=0.7$, is either slightly anticipated, and in reality $k_2=0.22$, or slightly delayed, and in reality $k_2=0.28$.

In Figure 3, the TestMM current signal is the one produced by the MM for the random test case showing the maximum error on the k_2 mapping ($k_{1,\dots,5}=[0.71620, 0.7249, 0.74506, 0.31308, 0.7383]$), while the TestANN signal is the one produced by the MM when the input of the model is the output vector of the ANN for N=80 ($k_{1,\dots,5}=[\ 0.71391, 0.71333, 0.74587, 0.313, 0.74237]$). Current signals are shown with both high and low frequency samplings. As it can be seen qualitatively from Figure 3, and more quantitatively from Figure 4, the differences between the signals are very small and certainly within the uncertainty boundaries of the considered MM, confirming that the ANN based approach is ready for practical applications.

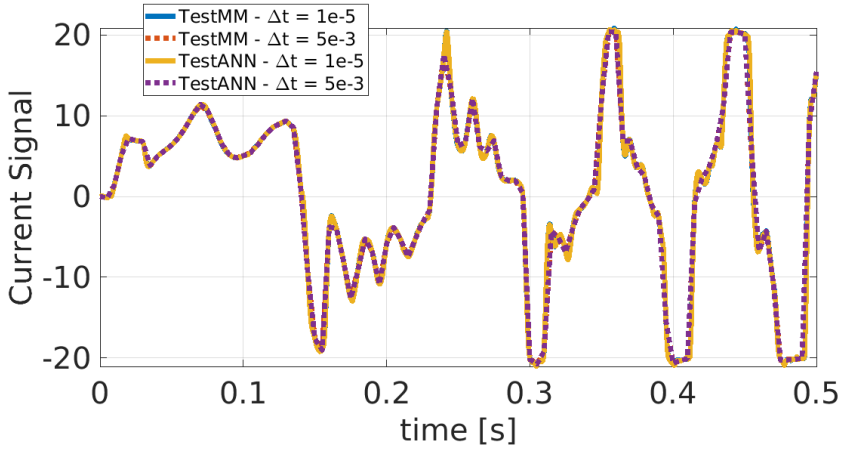

Fig. 3. For the simple case: Current signal produced by the MM for the random test case showing the maximum error on the k_2 mapping, TestMM, and the onee produced by the MM when the input of the model is the output vector of the ANN for N=80, TestANN. (Current signals shown with both high and low frequency samplings)

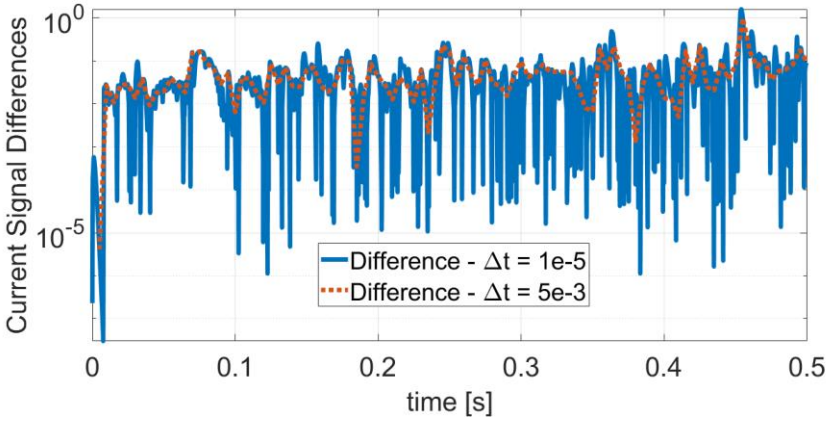

Fig. 4. For the simple case: TestMM and TestANN for both high and low frequency samplings.

3.2 Complete Case Results and Discussion

In this case, all the 8 *k* parameters presented in Sec 2.1 have been considered. The design of experiment (DOE) to sample the 8-dimensional space combines again a full regular grid with 5 nodes for each dimension (390625 samples), with a Latin Hyper-cube Sampling of 400000 points, giving *Ns = 790625*.

To assess the effect of the number of neurons in the hidden layer, N, the ANN has been set with three different values: N = 50, N = 70, N = 90. The three obtained ANNs have been tested against a sample of 100 random signals, i.e., obtained by randomly (uniform) sampling the *k* parameter space (again, the 100 samples are the same for all the three tests). The obtained results are summarised in Table 2. Again, the biggest errors for each ANN are highlighted, and in this case both k_2 (backlash) and k_6 (static eccentricity modulus in the rotor) are the most difficult parameters to map, consistently across the ANNs. However, the maximum error, never exceeds ~0.02, slightly improving when passing from N=50 to N=90. Likely a higher value of *N* could help to reduce the error, but the training with this amount of data has been strongly limited by the computational resources: the training of the N=90 ANN took 7 days on 12 INTEL Skylake 6138 cores, which is the maximum the team could use for this work.

In Figures 5 and 7, the TestMM current signals are the ones produced by the MM for the random test case showing the maximum error on the k_2 mapping ($k_{1,\dots,8}$*=[0.57543, 0.40647, 0.63437, 0.64092, 0.6762, 0.72495, 0.30366, 0.70536]*) and on the k_6 mapping ($k_{1,\dots,8}$*=[0.6549, 0.68609, 0.73232, 0.61184, 0.57124, 0.60872, 0.4837, 0.41279]*), respectively. On the other hand, the TestANN signals are those produced by the MM when the input of the model is the output vector of the ANN for N=90 ($k_{1,\dots,8}$*=[0.57149, 0.426, 0.63765, 0.64145, 0.67431, 0.72874, 0.29977, 0.70928]* when the maximum error on k_2 mapping is considered, and $k_{1,\dots,8}$*=[0.6577, 0.67421, 0.7319, 0.61512, 0.56771, 0.59137, 0.48718, 0.40601]* when the maximum error on k_6 mapping is considered).

Table 2. Prediction errors on the 100 test signals for the case with five k parameters. Statistical values (mean value, mean, standard deviation, std, minimum, min, and maximum, max) reported for each k parameter, when N varies from 40 to 80.

	k	mean	std	min	max
N=50	1	-1.65E-05	0.002746	-0.00745	0.008193
	2	8.32E-05	0.006611	-0.01524	*0.023647*
	3	-0.00013	0.002984	-0.00836	0.007289
	4	0.000189	0.003028	-0.00916	0.005793
	5	0.000259	0.003452	-0.01031	0.007574
	6	0.001182	0.006259	-0.01356	*0.021009*
	7	0.000196	0.004567	-0.01455	0.011524
	8	-0.00048	0.003997	*-0.02224*	0.00672
N=70	1	-9.92E-05	0.002425	-0.00539	0.006919
	2	-0.00053	0.004943	-0.01035	*0.017784*
	3	5.09E-05	0.002121	-0.00484	0.0064
	4	0.000338	0.002705	-0.00719	0.005802
	5	9.49E-05	0.003002	-0.00938	0.011673
	6	0.000531	0.005287	-0.01203	*0.016516*
	7	0.000614	0.0037	-0.01076	0.009334
	8	-0.00014	0.00334	-0.01387	0.011365
N=90	1	8.52E-05	0.002329	-0.00477	0.007887
	2	-0.00108	0.005029	*-0.01954*	0.011878
	3	5.51E-05	0.002208	-0.00589	0.006297
	4	-0.00022	0.002319	-0.00739	0.004484
	5	0.000149	0.002644	-0.0074	0.007928
	6	0.000518	0.004788	-0.01066	*0.01736*
	7	-0.0003	0.003294	-0.00933	0.01076
	8	8.20E-05	0.002778	-0.00888	0.007526

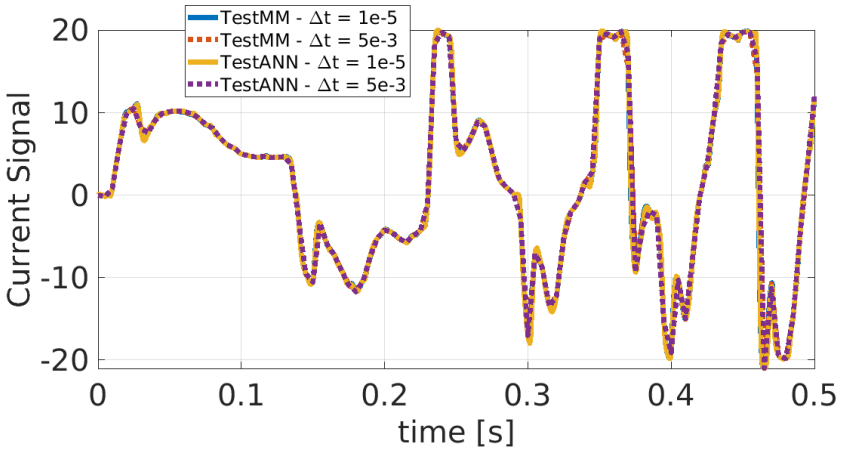

Fig. 5. For the complete case: Current signal produced by the MM for the random test case showing the maximum error on the k_2 mapping, TestMM, and the onee produced by the MM when the input of the model is the output vector of the ANN for N=90, TestANN. (Current signals shown with both high and low frequency samplings)

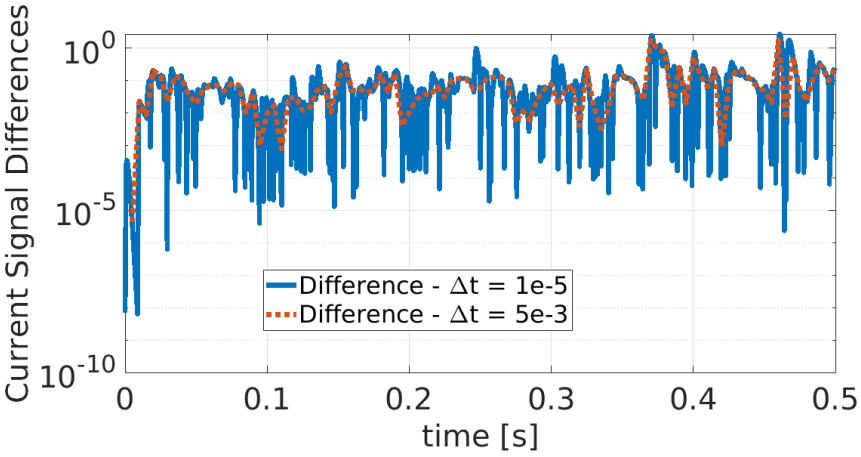

Fig. 6. For the complete case (maximum error on the k_2 mapping**)**: TestMM and TestANN for both high and low frequency samplings.

As it can be seen quantitatively from Figures 6 and 8, the differences between the signals are, again, very small and within the uncertainty boundaries of the considered MM, confirming that the ANN based approach can be considered useful and ready for complex cases too.

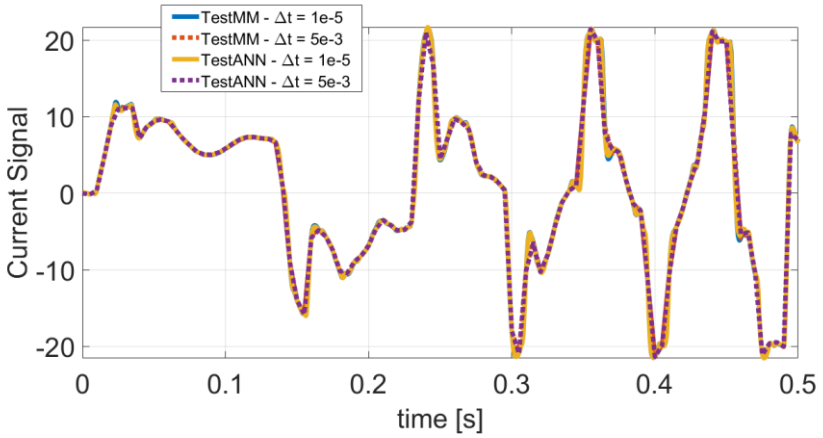

Fig. 7. For the complete case: Current signal produced by the MM for the random test case showing the maximum error on the k_6 mapping, TestMM, and the onee produced by the MM when the input of the model is the output vector of the ANN for N=90, TestANN. (Current signals shown with both high and low frequency samplings)

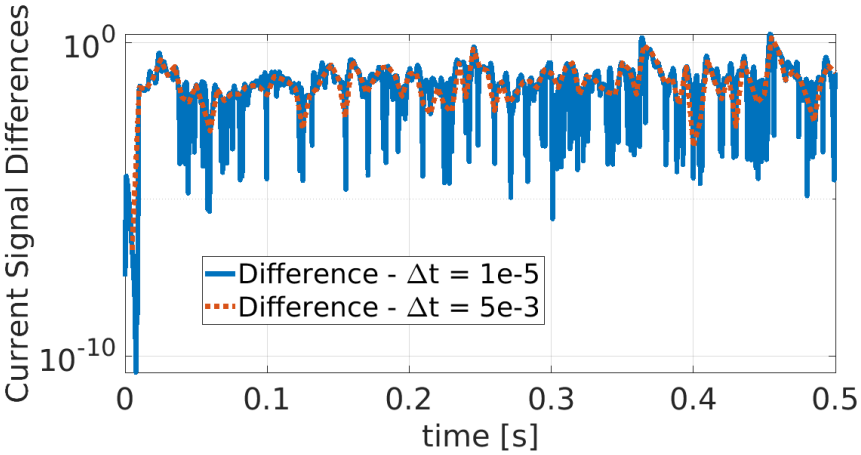

Fig. 8. For the complete case (maximum error on the k_6 mapping): TestMM and TestANN for both high and low frequency samplings.

4 Conclusion

A machine learning based approach has been implemented and tested to provide a real-time prognostic response for a considered electro-mechanical actuator system. In particular, a shallow artificial neural network is used to directly map the electric signal response of the system into the values of the key parameters of the EMA model that implement the different kinds of degradation. The approach has been test both on a simplified case, where only 5 of the total 8 parameters are considered and on the complete case, where all the 8 parameters implemented in the model are considered.

In both cases the proposed approach shows robust performance, with absolute errors on the identification of the simple parameters always ≤2%, which is considered enough for practical applications.

In both cases (i.e., the simplified and the complete ones) the parameter corresponding to the backlash degradation shows the highest identification errors, and for the complete case, the degradation related to static eccentricity modulus in the rotor, shows some relatively bigger errors. This can be related to DOE data considered for the training of the ANNs and/or to higher non-linearities related to that particular degradation in the model, and this aspect will be better analyzed in the future.

Moreover, even if the proposed approach shows good and robust performance, the ANN training in the complete case challenged the available computational capabilities and highlighted the need to sample the electric signal more efficiently to limit the exponential increase of the needed database. This will be considered for future works too.

References

1. Qiao, G., Liu, G., Shi, Z., Wang, Y., Ma, S., Lim, T.C.: A review of electromechanical actuators for More/All Electric aircraft systems. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 232(22), 4128-4151 (2018).
2. Karadotcheva, E., Nguyen, S.N., Greenhalgh, E.S., Shaffer, M.S.P., Kucernak, A.R.J., Linde, P.: Structural Power Performance Targets for Future Electric Aircraft. Energies 14(19), 6006 (2021).
3. Quattrocchi, G., Berri, P.C., Dalla Vedova, M.D.L., Maggiore, P.: An Improved Fault Identification Method for Electromechanical Actuators. Aerospace 9(7), 341 (2022).
4. Dalla Vedova, M.D.L., Germanà, A., Berri, P.C., Maggiore, P.: Model-Based Fault Detection and Identification for Prognostics of Electromechanical Actuators Using Genetic Algorithms. Aerospace 6(9), 94 (2019).
5. Querques, I.: Prognostics of On-Board Electromechanical Actuators: Bio-Inspired Metaheuristic Algorithms; Technical Report, Politecnico di Torino, Torino, Italy, (2021).
6. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems, 2, 303–314 (1989).
7. Campobasso, M.S., Cavazzini, A., Minisci, E.: Rapid Estimate of Wind Turbine Energy Loss Due to Blade Leading Edge Delamination Using Artificial Neural Networks. Journal of Turbomachinery 142(7), 071002-1/11 (2020).