Genetic Algorithm for Topology Optimization of an Artificial Neural Network Applied to Aircraft Turbojet Engine Identification

F. P. Da Costa
Dep. of Mechanical Engineering
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil
fabio13costa@hotmail.com

P. H. L. S. P. Domingues
Dep. of Mechanical Engineering
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil
phd.engmec@gmail.com

Roberto Z. Freire
Industrial and Systems Engineering Graduate Program (PPGEPS)
Pontifical Catholic University of Parana (PUCPR)
Curitiba, Brazil
roberto.freire@pucpr.br

Leandro S. Coelho
Industrial and Systems Engineering Graduate Program (PPGEPS)
Pontifical Catholic University of Parana (PUCPR)
Curitiba, Brazil
leandro.coelho@pucpr.br

A.R. Tavakolpour-Saleh
Dep. of Mechanical and Aerospace Engineering
Shiraz University of Technology
Shiraz, Iran
tavakolpour@sutech.ac.ir

Helon V. H. Ayala
Dep. of Mechanical Engineering
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil
helon@puc-rio.br

***Abstract*—Artificial neural networks (ANN) has attracted attention of the academic community by the current progress that this technique has provided in speech recognition and digital media such as as image, video, audio, and signal processing. Some fields, as industrial process control and product development can be highly benefited by the development of techniques based on the proven potentialities of ANN models, allowing more accurate simulation, better adaptation to changing environments, and greater robustness in model-based fault diagnosis. Along with the advance of ANNs, there is a trend of open-source softwares use for soft computing which facilitates the access of the interested readers to implement their own codes and to explore other applications. Historically evolutionary algorithms such as the Genetic Algorithm (GA) have been implemented to evolve the architectures to search for solutions, in order to solve this fundamental issue that is still an open problem in the general case. Therefore, the present paper investigates the application of ANN to model the nonlinear aircraft turbojet engine through black-box approach. For that purpose it was used real-world measurements of aircraft engine's fuel and rotation as input and output, respectively. In order to facilitate the design, the ANN was optimized aiming to determine the best topology according to the one-step-ahead and free-run simulation. The results obtained encourage the use of automatically generated ANN architectures for dynamic system modeling.**

***Index Terms*—System Identification, Artificial Neural Network, Optimization, Genetic Algorithm.**

The authors thank Fundação Araucária PRONEX (Grant 042/2018) and the National Council of Scientific and Technologic Development of Brazil - CNPq (Grants 303908/2015-7-PQ, 404659/2016-0-Univ, 204893/2017-8, 304783/2017-0 and 430395/2018-3-Univ) for their financial support of this work.

I. INTRODUCTION

Gas turbines have attracted the attention of researchers since the early days of jet propulsion. Nowadays, the gas turbines are present in fields that its operation demands accurate models, as in the aerospace, marine and industrial sectors [1].

System identification is the field of knowledge dedicated to explore ways of extracting a mathematical model from the inputs and outputs measurements of a system of interest. The abstracted model permits e.g. design control algorithms or understand systems' underlying dynamic behaviors.

Briefly, there are two methodologies that guide the implementation of system identification. The first approach is the white-box, where the principles of physics base the structures of the model and the second methodology is the black-box, where little or no knowledge about the system's physics is assumed a priori. The black-box approach presents some advantages, in many cases it is easier to implement than the white-box approach, ease of implementation and is also unbiased and efficient on large and complex systems. To identify a system it is needed to follow four steps: structure definition, where is identified all the terms in the model; parameter estimation, to identify what coefficients the model will assume; prediction, which represents the model adopted to predict the future output; and analysis, the dynamical properties of the system [2].

Artificial Neural Network (ANN) is defined as a massively parallel distributed processor that sought inspiration on the brain learning process and is composed by several simple

978-1-7281-2153-6/19/$31.00 ©2019 IEEE

processing units, known as neurons. Those have a natural propensity for storing experiential knowledge and making it available for use [7]. An ANN is a black-box model that can be used for system identification, but also for signal, image, video and audio processing, nonlinear modelling and control [2]. Due the wide range of applications (see examples in [3]–[6]), ANNs have become an attractive research field in the last decade [2], [8].

An ANN can be made up of multiple layers and feedforward or recursive networks as architectures. Thus, it is possible to vary the number of hidden layers and neurons in each layer, as the neurons activation function, allowing enormous variability in the model creation. The main issue of the ANN is that there is no based theory for optimizing ANN's topology [7].

Another nature inspired technology is the Genetic Algorithm (GA) that can be used to optimize the ANN topology. In the GAs, a population of individuals represented by chromosomes (set of genes) is created and distributed randomly in the search space. Each individual represents a possible solution to the problem in question, which is evaluated according to a objective function that quantifies the solution performance. Individuals with higher fitness are susceptible to be chosen for producing offsprings, recombining their genes through crossover and mutation operators, in order to create a new generation [9].

The deep learning concept was originated from the ANN study. Liu et. al. [8] published a review of deep neural networks in a historical perspective and Schmidhuber [10] revised the variety of architectures. In order to train deep neural networks, the most used methods are based on the stochastic gradient descent [11]. Inside that tool class, the Adaptive Moment estimation (ADAM) [12] was widely employed for the ease of setting the hyperparameters and the results achieved.

Deep neural networks are reaching promising results to solve both system identification and time series forecasting problems, in [13] and [14] the authors used randomized algorithms to perform nonlinear black-box system identification, the method was tested in the Box-Jenkins furnace [15], simulated nonlinear system [16] and the Wiener-Hammerstein Benchmark [17] problems. Genc [18] proposed a new architecture for convolutional neural networks and applied it to identify the building energy load. Gongming et. al. [19] tested in a sinthetic data simulated system a partial least square based regression for deep neural networks. In [20], the authors also proposed a partial least square regression for deep neural networks that was tested in a simulated dynamic system, a chaotic time series benchmark and a wastewater treatment system. Wang et. al. assumed deep neural network with convolutional layers and wavelet transforms to perform wind [21] and photovoltaic [22] power forecasting.

All deep learning applications cited above developed the prediction process considerind the one-step-ahead methodology, not reporting the free-run simulation of the model. As it can be, it is observed, a scarcity of studies involving nonlinear black-box system identification on top of real-world measured data, such as the one detailed in the present paper.

Encouraged by relevant results achieved recently in the use of ADAM learning algorithm [12] and observing the current tendency to use open source software on soft computing, in this paper, free software packages such as R statistical software[1], Keras[2] and GA[3] were tested on the proposition of a optimized topology for system identification. Based on the GA package for a Keras ANN model training through ADAM algorithm, data was obtained in [1] from measurements of an aircraft turbojet engine fuel inputs and rotation outputs, in order to create an ANN model that could yield the best prediction performance. The contributions of the paper are related to the proposition of an encoding architecture for artificial neural networks realizable with the Keras framework. The objective function that was defined for the optimization problem depends on the evaluation of two criteria in one-step-ahead and free-run simulations.

The remainder of the paper is divided as follows. In Section II the theoretical background about system identification through ANNs, validation metrics, and the objective functions associated to the case study adopted in this work are presented. Section III depicts the case study. Section IV presents the results acchieved. Lastly, Section V concludes the paper.

II. Artificial Neural Networks Applied to System Identification

A multilayer ANN can be mathematically described as follows:

$$\hat{y}(t) = \Gamma[x(t)] \tag{1}$$

where $\Gamma[\cdot] : \mathbb{R}^p \to \mathbb{R}$ is a nonlinear vectorial mapping from the model input $x(t) \in \mathbb{R}^p$ to the predicted output $\hat{y}(i) \in \mathbb{R}$ for the i-th sample. Also, $\Gamma[\cdot]$ can be defined as the concatenation of multiple layers, each composed of neurons that perform vector operations known as activation functions, such as threshold, hyperbolic tangent or sigmoid functions [7].

Neurons are simple but nonlinear units with the ability to retain information from a set of data and make it available for use. The neuron structure is shown in Fig. 1.

[1]https://www.r-project.org
[2]https://keras.io
[3]https://cran.r-project.org/package=GA

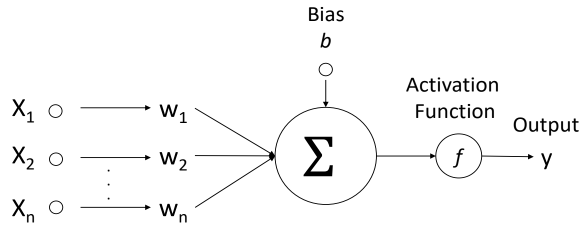

Fig. 1. Neuron structure, w, b are its weights and biases [7].

The complexity of an ANN model is defined by its topology, which is the number of layers and the amount of neurons and nodes in each layer. The proposed architecture is illustrated in 2.

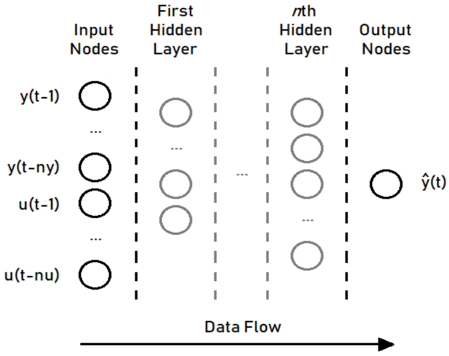

Fig. 2. Artificial neural network multilayer perceptron structure [7].

Once the topology is set, the learning process or training stage starts in order to adjust the free parameters of the model, these parameters are the neurons, weights and biases 1.

In this sense, the stochastic gradient descent has become the most used method for training ANN [11]. The idea of this method is to guide the parameters optimization by the direction of negative gradient, which minimizes the loss function. However, the gradient descent method has a slow convergence and can get strapped into local minima when considering nonlinear or nonconvex problems [2].

The ADAM learning algorithm proposed in [23] is a first-order stochastic gradient descent algorithm. The difference between ADAM and the regular stochastic gradient descent algorithm is that ADAM assigns an individual learning rate for each free parameter in the optimization process, enabling the individualized step adaptation in the search process.

The ANN proposed in the present paper is composed of a multilayer structure and trained through stochastic gradient descent ADAM algorithm, with increasing complexity according to the number of layers that should be trained. In the next topic, it is introduced nonlinear black-box system identification theory required for dynamic systems modeling.

A. Nonlinear Black-box System Identification

The system identification is an experimental task, and the whole procedure can be divided in the following steps: (i) experiment design and data acquisition; (ii) model structure definition; (iii) model construction and (iv) validation. In step (i), the designer should define the excitation to which the system will be subjected, in order to capture and output signal/measurement containing the underlying dynamics that governs the physical phenomena, always respecting the constraints of the studied system. For (ii), it is defined the structure of the model, which means the complexity and the lagged input/outputs terms. In possession of the data, follow the step (iii) where the mathematical model is built; Finally, step (iv) validates the model trying to minimize the error between real and estimated outputs by iteratively changing the free parameters. In case of unfitting models, the process should return to the steps (i) or (ii) until the model fits the accuracy requirement(s).

Among the model structure mostly used figure the nonlinear autoregressive models with exogenous inputs (NARX). Considering the single-input-single-output case, it may be described as follows:

$$\begin{aligned}\hat{y}(t) = F[&y(t-1), y(t-2), \ldots, y(t-n_y),\\ &u(t-1), u(t-2), \ldots, u(t-n_u)]\end{aligned} \tag{2}$$

where $u(t) \in \mathbb{R}$ is the input and $y(t) \in \mathbb{R}$ is the output signal/measurements of the system, respectively, being $\hat{y}(t) \in \mathbb{R}$ the predicted response given by the model. The function $F[\cdot] : \mathbb{R}^p \rightarrow \mathbb{R}$ may be set as any nonlinear vectorial mapping such as higher-order polynomials, neuro-fuzzy systems or artificial neural networks [24]. It is important to note that:

$$\hat{y}(t) = F[\phi(k)] \tag{3}$$

where

$$\begin{aligned}\phi(k) = [&y(t-1), y(t-2), \ldots, y(t-n_y),\\ &u(t-1), u(t-2), \ldots, u(t-n_u)]\end{aligned} \tag{4}$$

it can be used the ANN learning models as mentioned in (1), where ny, nu are the number of lags in the output and the input, respectively. Figure 2 depicts the type of architecture tested in the present work. Also, it is possible to note that the regression vector $\phi(k) \in \mathbb{R}^{n_y+n_u}$ is taken as features of the ANN model. The data is then processed in the different layers of representation until a desired target prediction is achieved.

B. Validation Metrics

The models in system identification are usually validated considering the residuals amplitude and their statistical analysis. Both approaches were depicted below.

The one-step-ahead prediction is the prediction method where it is only considered the last measured data as inputs of the model, as seen in (2) where the arguments of $F[\cdot]$ would be lagged of one sampling time unit. In contrast, the free-run simulation consider measured data only to initialize the model, using the previously predicted output to make a new prediction. Thus, considering $\hat{y}_s(t)$ the free-run simulation output of the NARX model in (2), and the case where the orders of the model are n_y and n_u. The model is initialized with:

$$\hat{y}_s(1) = y(1), \hat{y}_s(2) = y(2), \ldots, \hat{y}_s(n_y) = y(n_y),$$

and then iteratively calculate the simulated predictions with

$$\begin{aligned}\hat{y}_s(t) = F[&\hat{y}_s(t-1), \hat{y}_s(t-2), \ldots, \hat{y}_s(t-n_y),\\ &u(t-1), u(t-2), \ldots, u(t-n_u)].\end{aligned} \tag{5}$$

It is notable that the simulated values in (5) are iterated, so that predictions are based on previous model outputs. Therefore, the prediction errors accumulate during the simulation, becoming harder to develop an accurately prediction.

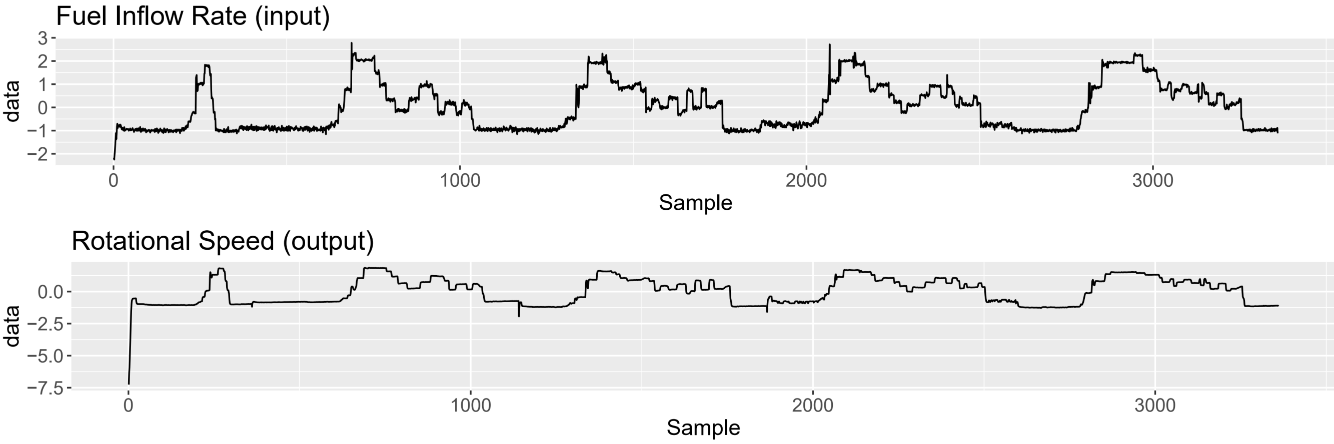

Fig. 3. Aircraft turbojet engine's measured input and output.

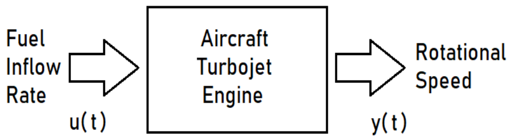

Fig. 4. Schematic representation of the aircraft turbojet engine system.

This error propagation does not happen in the one-step-ahead prediction, as at each prediction the error is not accumulated from previous predictions.

Being one-step-ahead or free-run, the multiple correlation coefficient R^2 is an evaluation metric that can be used to quantify the adherence of the predicitons. The R^2 is defined as follows: [25]

$$R^2 = 1 - \frac{\sum_{t=1}^{N} [\xi(t)]^2}{\sum_{t=1}^{N} [y(t) - \bar{y}]^2}, \tag{6}$$

where the upper bar represents the mean of a sequence, $\xi(t) \in \mathbb{R}$ is the one-step-ahead or free-run simulation residuals. In general, it is recommended to keep R^2 above 0.9 for engineering applications [26]. As R^2 maximum value is 1, it is easy to compare different models through this metric, even considering models based on measurements with different orders of magnitude.

C. Keras ANN Optimization Through Genetic Algorithm

The GA is based on Darwin evolutionary theory. Each individual in the population is a possible solution and is defined by a chromossome which is composed by n decision variables.

The initial population of size P is randomly created respecting the search space defined by the lower and the upper bounds of each decision variable. After that, all solutions are evaluated according to the objective function. The evolutionary process starts with the selection of individual pairs to create a new population of offsprings, where the solutions with greater objective value are more likely to be chosen for crossover. The crossover occurs according to a probability (PC) and also, in case a new individual is generated, there is a mutation probability PM according to which each of the offsprings decision variables can be randomly changed. Finally, the evolutionary process continues untill a stop criterion is reached.

The standard statement of a single objective minimization problem is defined below:

$$\begin{cases} min \quad f(x) & x \in \mathbb{R}^n \\ s.t. & \\ \quad h_i(x) = 0; & i = 1, ..., p \\ \quad g_j(x) \leq 0; & j = 1, ..., m \\ \quad x_n^l \leq x_n \leq x_n^u & r = 1, ..., n \end{cases} \tag{7}$$

where $f(x)$ is the objective function value for the individual x, h_i is the equality contrains, g_j is the inequality constrains and x_n^l and x_n^u are the lower and upper bounds of each decision variable x_n.

III. Case Study: Aircraft Engine

The case study explored in this paper has been investigated in [1] for the purpose of black-box system identification with linear and nonlinear models. The system consists of an aircraft turbojet engine instrumented for providing fuel flow rate and rotational speed measurements as input and output, respectively. Fig. 4 shows a schematic representation of the system's input and output. The environmental conditions details during the data acquisition may be checked in [1].

In this paper, it was created a chromosome of $n = 8$ decision variables, represented by $x = [ny, nu, n1, n2, n3, n4, n5, af]$, where: ny is the number of inputs in the neural network for past y (system output), nu is the number of inputs in the neural network for past u (system input), $n1$, $n2$, $n3$, $n4$ and $n5$ being the amount of neurons in the $1st$, $2nd$, $3rd$, $4th$ and $5th$ hidden layers. Finally, ac is the activation function. In order to maximize the correlation coefficient (R^2) in both one-step-ahead prediction and free-run simulation, the objective function chosen was

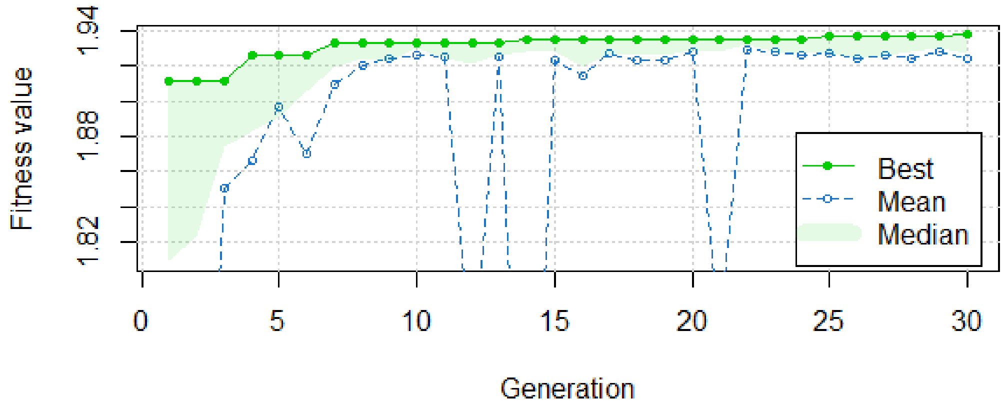

Fig. 5. Genetic algorithm evolution process.

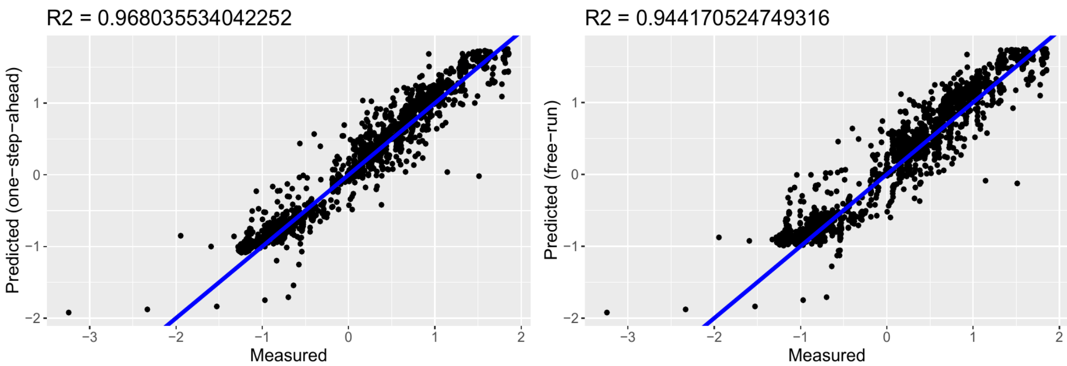

Fig. 6. Multiple correlation metric graphic representation.

set as the sum of the multiple correlation coefficients in both situations.

The neural networks generated from the genetic algorithm could vary from 0 to 5 hidden layers and each hidden layer could have from 0 to 5 neurons. For the networks input, we could have from 1 to 5 inputs from $y(t)$ and $u(t)$. For the activation function, the genetic algorithm could choose between the hyperbolic tangent ($tanh$), the Rectifier Linear Unit ($ReLU$) and the sigmoid function, with af assuming respectively 1, 2 or 3 as value. Also, for the loss function, it was chosen the Mean Square Error (MSE) and for measuring the models accuracy, the Mean Absolute Error (MAE). For the GA parameters it was used a population size of 10 individuals, 30 generations, crossover and mutation probability of $PC = 0.8$ and $PM = 0.1$, respectively.

In order to create the ANN, trained through ADAM optimization, it was used the Keras package and to accomplish the ANN topology optimization the GA package was used, both for R statistical software.

IV. Numerical Results

The aim of this section is to present the obtained results for the GA optimization of the ADAM estimated ANN topology, identifying the nonlinear black-box system represented in Fig. 4.

For the modeling purpose it was used a total of 3,360 samples of measured input and output data pairs, as shown in Fig. 3. The data was divided into two data sets, half for the training set and the other half for the test set. Also, data was normalized with zero mean and unary standard deviation.

A. Simulation Outputs and Discussion

Seeking to obtain the best model for the aircraft turbojet engine [1], an optimization based on genetic algorithm was performed as described in Section II-C on top of the training data, generating the Fig. 5. It was important to note that, as it was considered the objective function as the sum of one-step-ahead and free-run simulation multiple correlation metrics, the maximum value that it can reach is 2. Plus, it is seen that the diversity of solutions was maintained during the optimization process.

With the optimization process finished, the best ANN topology was presented in Table I, and its correlation results for both one-step-ahead and free-run simulations are arrenged in Table II. These tables present results for the prediction considering training, test and the overall data set.

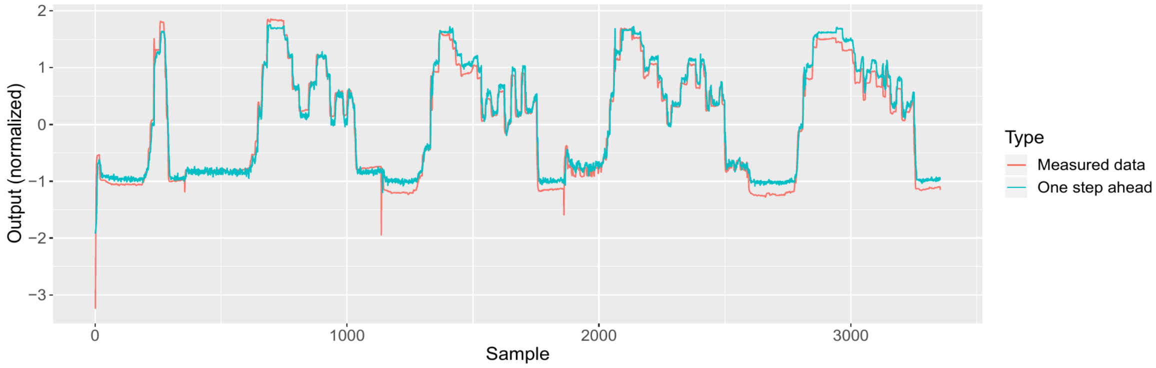

Fig. 7. One-step-ahead and measurements comparison.

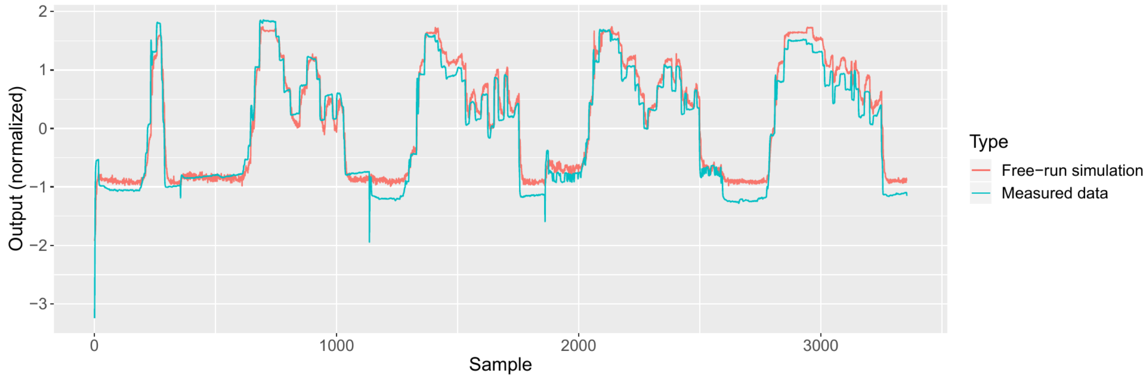

Fig. 8. Free-run simulation and measurements comparison.

TABLE I
BEST ANN TOPOLOGY

ny	nu	n1	n2	n3	n4	n5	af
5	2	1	2	3	3	2	$tanh$

TABLE II
MULTIPLE CORRELATION COEFFICIENTS (R^2) FOR ONE-STEP-AHEAD AND FREE-RUN SIMULATION

Number of layers	All data	Training	Test
One-Step-Ahead	0.9680	0.9703	0.9653
Free-Run Simulation	0.9441	0.9519	0.9360

Visually, Fig. 6 illustrates how well the results obtained for prediction approach the measured values, considering that the predictions were made on top of all data (training + test data sets). Finally, it was plotted the one-step-ahead and free-run simulations. Both data sets could also be compared with the measurements in Fig 7 and Fig. 8, again considering all data for the prediction.

V. CONCLUSION

The present paper investigated the use of genetic algorithm for ANN optimization, considering the ADAM estimation of the ANNs parameters, in order to identify a nonlinear system through black-box approach. It was applied the Keras and GA library in the R statistical software package, which is freely available on the internet and maybe thus tested by the interested readers with few lines of code.

By applying the open source tools cited above for the nonlinear aircraft turbojet engine identification, it was possible to demonstate its use with real world acquired data. Several benchmarks for black-box identification of nonlinear systems with real-world acquired data are available, which draws more attention from the dynamic modeling community in general.

For future works it is suggested the aplication of new ANN architectures, the exploration of other optimization algorithms for the ANN topology optimization and finally, the evaluation of this frameworks for other nonlinear black-box system identification benchmarks.

REFERENCES

[1] A. R. Tavakolpour-Saleh, S. A. R. Nasib, A. Sepasyan, S. M. Hashemi, "Parametric and nonparametric system identification of an experimental turbojet engine," *Aerospace Science and Technology*, vol. 43, pp. 21-29, 2015.

[2] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*, 1st ed, Chichester, West Sussex, John Wiley & Sons, 2013.

[3] P. Ramasamy, S. S. Chandel, A. K. Yadav, "Wind speed prediction in the mountainous region of India using an artificial neural network model," *Renewable Energy*, vol. 80, pp. 338-347, 2015.

[4] W. Chine, et al. "A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks," *Renawable Energy*, vol. 90, pp.501-512, 2016.
[5] L. A. Gatys, A. S. Ecker, M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, CVPR2016, Las Vegas, USA, pp. 2414-2423.
[6] A. Esteva, et al. "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115-118, 2017.
[7] S. Hayking, *Neural networks and learning machines*, 3rd ed, Ontario, Canada, Pearson, 2008.
[8] W. Liu, et al. "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11-26, 2017.
[9] H. V. H. Ayala, L. S. Coelho, "Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator," *Expert Systems with Applications*, vol. 39, no. 10, pp. 89688974, 2012.
[10] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol.61, pp. 85-117, 2015.
[11] S. J. Reddi, S. Kale, S. Kumar, "On the convergence of adam and beyond," *Proceedings of the 6th International Conference on Learning Representations*, ICLR2018, Vancouver, Canada, 2018.
[12] D. P. Kingma, J. Ba, "Adam: a method for stochastic optimization," *Proceedings of the 3rd International Conference on Learning Representations*, ICLR2015, San Diego, USA, 2015.
[13] E. de la Rosa, W. Yu, X. Li, "Nonlinear system identification using deep learning and randomized algorithms," *In IEEE International Conference on Information and Automation*, Lijiang, China, pp. 274279, 2015.
[14] E. de la Rosa, W. Yu, X. Li, "Randomized algorithms for nonlinear system identification with deep learning modification," *Information Sciences*, vol. 364-365, pp. 197-212.
[15] G. E. P. Box, G. M. Jenkins, *Time series analysis, forecasting and control*, Holden Day, San Francisco, USA, 1970.
[16] K. S. Narendra, K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 252-262, 1991.
[17] L. Ljung, J. Schoukens, J. Suykens, "Wiener-hammerstein benchmark," *In 15th IFAC Symposium on System Identification*, St. Malo, France, 2009.
[18] S. Genc, "Parametric system identification using deep convolutional neural networks," *In 2017 International Joint Conference on Neural Networks*, IJCNN2017, Anchorage, USA, pp. 2112-2119, 2017.
[19] W. Gongming, L. Wenjing, Q. Junfei, W. Guandi, "Nonlinear system identification using deep belief network based on plsr, *In 2017 36th Chinese Control Conference (CCC)*, Dalian, China, pp. 1080710812.
[20] J. Qiao, G. Wang, W. Li, X. Li, "A deep belief network with PLSR for nonlinear system modeling," *Neural Networks*, vol. 104, pp. 68-79, 2018.
[21] H.-Z. Wang, G.-Q. Li, G.-B. Wang, J.-C. Peng, H. Jiang, Y.-T. Liu, "Deep learning based ensemble approach for probabilistic wind power forecasting," *Applied Energy*, vol. 188, pp. 5670, 2017.
[22] H. Wang, et al., "Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network," *Energy Conversion and Management*, vol. 153, pp. 409-422, 2017.
[23] D. Kingma, J. Ba, "Adam: A method for stochastic optimization," *In International Conference on Learning Representations*, ICLR2014, Banff, Canada, 2014.
[24] R. Sindelar, R. Babuka, "Input selection for nonlinear regression models," *IEEE Transactions on Fuzzy Systems*, vol. 12, pp. 688696, 2004.
[25] R. Haber, H. Unbehauen, "Structure identification of nonlinear dynamic systemsa survey on input\output approaches," *Automatica*, vol. 26, pp. 651677, 1990.
[26] B. Schaible, H. Xie, Y.-C. Lee, "Fuzzy logic models for ranking process effects,". *IEEE Transactions on Fuzzy Systems*, vol. 5, pp. 545556, 1997.
[27] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.