COBEM 2021 26th International Congress of Mechanical Engineering
November 22-26, 2021, Virtual Congress, Brazil
ABCM Associação Brasileira de Engenharia e Ciências Mecânicas

COB-2021-0615
Development of a structural metamodel in MDO for wind blades based on neural network

Julia da Silva Maschietto
Juan Pablo Julca Avila
Federal University of ABC , UFABC, Av. dos Estados, 5001, Bangú, Santo André, SP, Brasil, 09210-580
julia.maschietto@aluno.ufabc.edu.br
juan.avila@ufabc.edu.br

Geraldo Majella Nunes Junior
Federal University of ABC , UFABC, Av. dos Estados, 5001, Bangú, Santo André, SP, Brasil, 09210-580
geraldo.junior@aluno.ufabc.edu.br

Abstract. *According to data from the World Economic Forum, 2019 was the biggest growth to date in the offshore wind energy sector with installed power of 29GW globally. In Brazil, installed power is increasing every year, In 2019, the capacity factor as it measures wind productivity was 42.7%. Thus, the need to study these technologies related to wind turbines is evident. The objective of this research is to contribute to the scientific advancement in the field of structure dynamics through an implementation of structure tracking in a Multidisciplinary Design Optimization (MDO) applied to a real problem, in the optimization of parameters of the wind turbine blade. For this, the blade was simplified to an equivalent beam, thus disregarding parameters such as the complexity of the geometry. Thus, it was possible to vary only the beam parameters such as height, width and thickness . The material chosen for the study was aluminum, widely used in industry due to its good mechanical properties. Multibody model was used to output the displacements, which were used as input for the training of the neural network. The results obtained after this phase were used to generate a metamodel and implemented the MDO integrated with wind blade technologies. The result was a metamodel that when inserting the input parameters, provided as output the structural response in a short time, having good accuracy and dispensing with more complex analyses.*

Keywords: *Multibody model, neural network , metamodel, wind energy*

1. INTRODUCTION

The wind energy sector is constantly growing in Brazil, obtaining investments from companies, employing professionals from different areas of engineering. Furthermore, Brazil is a country favored by the quality of the winds, which are stable, with adequate intensity and without sudden changes in speed or direction.

The continuous development of wind turbines is aimed at greater production of effect and reduction of purchase and maintenance costs for customers. This requires that the components of wind turbines be optimized as much as possible, but in engineering, in general, optimization processes are not simple to use directly, they require time and great computational cost.

This paper aims to overcome these difficulties with the use of metamodels, that is, simple approximations of expensive computational analysis, but which keep the results faithful. The result was a metamodel that, when inserting input parameters provided as output the structural response in a short time, having good accuracy and dispensing with more complex analyses.

2. CONTEXTUALIZATION

MBD (Multibody Dynamics) modeling is a relevant topic, since many mechanical and structural systems, such as vehicles, space structures, robotics, mechanisms and aircraft, consist of interconnected components that undergo large displacements of translation and rotation.

A wind turbine is a complex structure consisting of several rotating structural components, where especially the blades exhibit large geometric deformations.

When reviewing the literature, referencing Shabana (2005)it is observed that the multibody systems that consist of interconnected rigid and deformable components, approach this work. Problem modeling can be considered a generalization of structural and rigid body analysis methods.

According to (Jin Xu et al,2020), unlike the flexible MBD method, the wind turbine blade can be divided into several rigid bodies connected by restriction hinges (such as ball joints, cylindrical hinges, universal joints, etc.), springs and dampers on the MBD model. In this way, it greatly reduces the degrees of freedom (DOFs) of systems and improve computational efficiency.

As a result, it is especially suitable for quickly evaluating the displacement response of blades in the time domain. But the strain and deformation at the specified position of blades cannot be predicted. As these last two parameters will not be analyzed in this work, the modeling of rigid bodies is adequate.

The "Figure 1" shows a generic multibody system

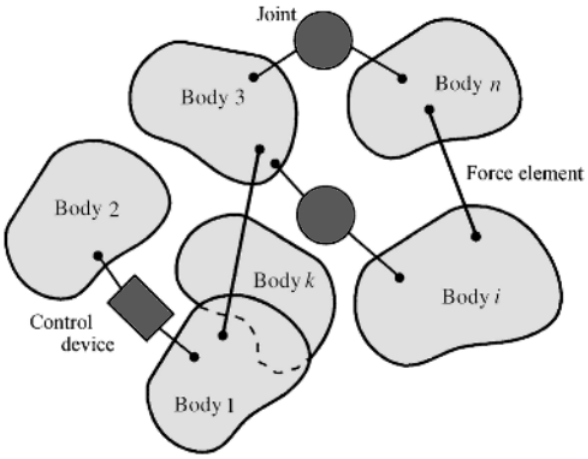

Figure 1. Generic multibody system-Adapted from Shabana (2005) pag.3

Once the displacements are calculated by the MBD method, the results will be used in the neural network and later in the metamodel in which the values are always updated until the process is finished.

Referencing the Learning (2020),inspired by the functioning of biological neurons in the nervous system of animals, a computational model of a neuron was established in the field of Artificial Intelligence.

With just one neuron it is not possible to perform many tasks, but it is possible to combine them in a layered structure, each with a different number of neurons, forming a neural network called Multilayer Perceptron (MLP). The vector of input values passes through the initial layer, whose output values are linked to the inputs of the next layer, and so on, until the network provides the output values of the last layer as a result. The network can be arranged in several layers, making it deep and capable of learning increasingly complex relationships.

The "Figure 2 shows the structure of the MPL Regressor model.

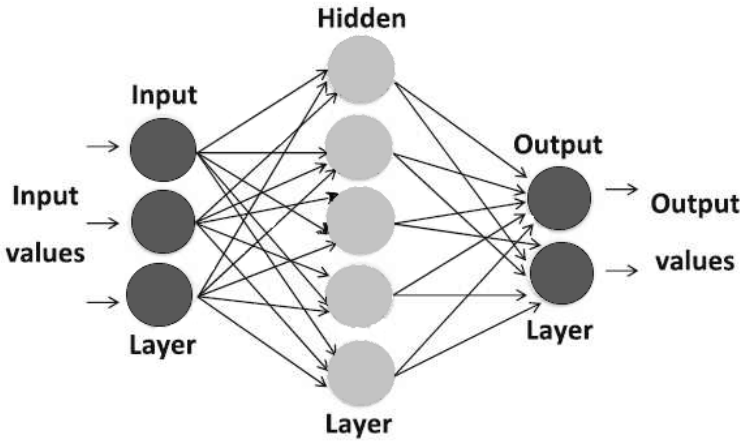

Figure 2. Example MLP Structure

For the network to work, it needs to be trained. The training of an MLP network is inserted in the context of supervised machine learning, in which each data sample used has a label informing which classification it belongs to.

MLP Regressor trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters.This implementation works with data represented as dense and

sparse numpy arrays of floating point values.

The metric used for evaluation was the R-squared (R^2), suitable for regression problems. Referencing Services (2020), the R-squared is a statistical measure of how close the data is to the fitted regression line, it is also known as the coefficient of determination or the coefficient of multiple determination for multiple regression.

The definition of the R^2 is the percentage of the variation of the response variable that is explained by a linear model, when the fit is good, the model explains the total variation and, consequently, the value of R^2 is close to 1.

In order to reduce the time needed to train the neural network, the Stochastic Gradient Descent (SDG) method was used to update the weights, making it possible to calculate the predictions, errors and back-propagation of a sample.

In short, the steps for implementing the Neural network are:

1. Initialize all network weights with small random values.

2. Provide input data to the network and calculate the value of the error function obtained.

3. In an attempt to minimize the value of the error function, the gradient values are calculated for each weight of the network.

4. Once the gradient vector is calculated, each weight is updated iteratively, always recalculating the gradients in each iteration step, until the error decreases and goes below some pre-established threshold, or the number of iterations reaches a maximum value.

5. At the end of the algorithm, the network is trained

After training, a neural network model fits perfectly with the metamodeling concept, preserving the characteristics of the modeled system in terms of an approximate function.

Referencing Júnior (2011), metamodels are substitutes for other models, which assumes simplified mathematical forms to approximate functions, often looking for improvements in computational cost.

Typical MDO (Multidisciplinary Optimization) often face high computational cost issues, which make modeling and integrating disciplines difficult. An alternative to deal with the challenge of reducing effort computational is the use of the concept of metamodeling, these techniques are able to improve the understanding of relationships between the input and output variables, as well as provide tools for optimization that are faster than conventional computational analysis.

For this paper, the metamodel used was based an optimization module where was used the lingprog model (with simplex revised) implemented in PYTHON.Referencing UFPR (2020), in mathematical optimization theory, George Dantzig's simplex algorithm is a popular technique for giving numerical solutions to linear programming problems. The revised simplex method, which is a modification of the original method, is computationally more economical as it calculates and stores only relevant information needed to test and improve the solution.

In general, the project flowchart can be summarized in "Fig.3"

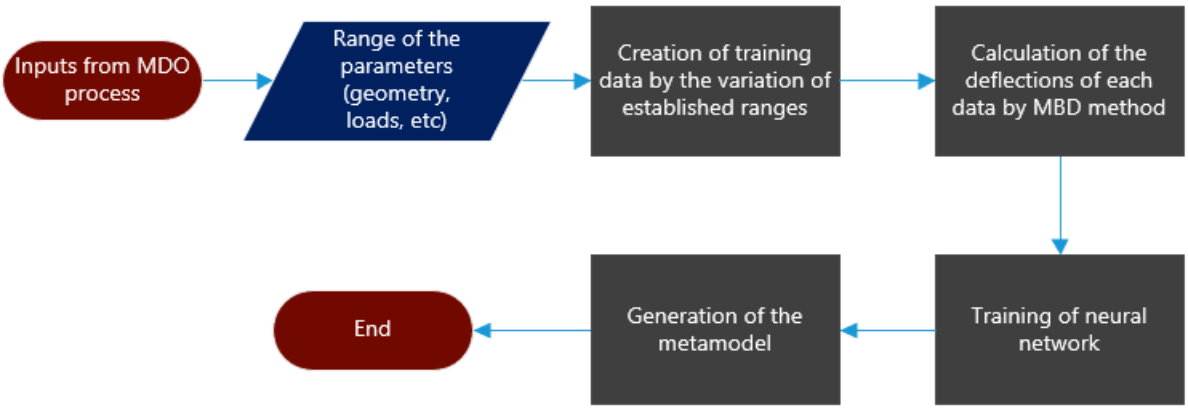

Figure 3. Project Flowchart

3. METHODOLOGY

3.1 MBD- Multibody Dynamics - Blade discretization

The problem to be modeled is the simplification of an eolic blade, more specifically hollow rectangular section of stringer. The focus of modeling is to divide this stringer into small pieces, using a technique called multibody model. A multi-body system as seen in "Fig. 4" consists of rigid bodies interconnected by mechanical joints with adjacent bodies. Geometric parameters of height, width and thickness of the beam were changed. A parameter for the beam root and tip was chosen and linearly varied. In this paper, K is the elastic constant of the spring, q_1, q_2 q_n are the the body position vectors, α the inclination angle , m the block mass , g the gravitational constant and P the acting force.

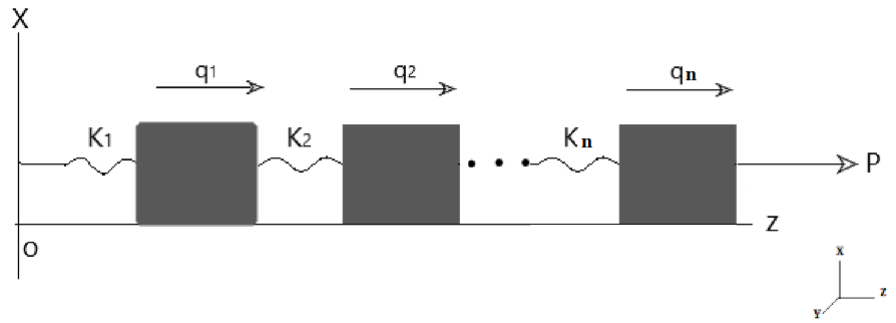

Figure 4. Problem model

The movement of each component of the system is influenced by the movement of the others through kinematic constraints as discussed in Shabana (2005). The system needs a set of coordinates, certain degrees of freedom.

This work will consider a model with n rigid bodies connected by a spring of stiffness K, according to the "Fig. 5". A force acting on the Z axis, at n angles, as seen in the "Fig. 6", simulating a possible blade positioning, this way the blade can be modeled as a rigid body, similar to the one performed in the studies of Songyi Jiang (2011), in which the blade is divided into four rigid bodies. The model in question was carried out in one dimension in order to simplify the problem. Thus, stiffness parameters are only related to the z direction.

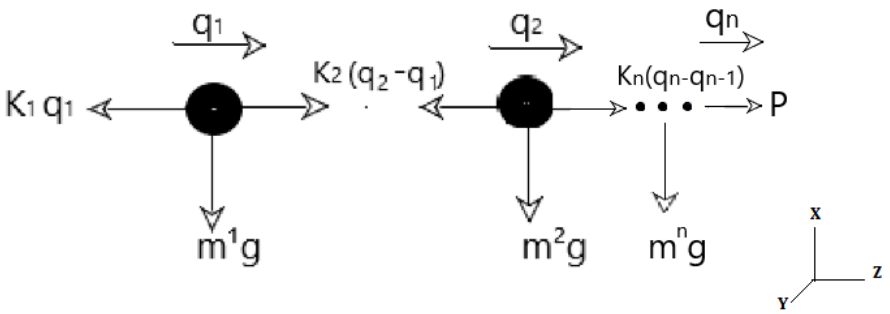

Figure 5. Problem Diagram

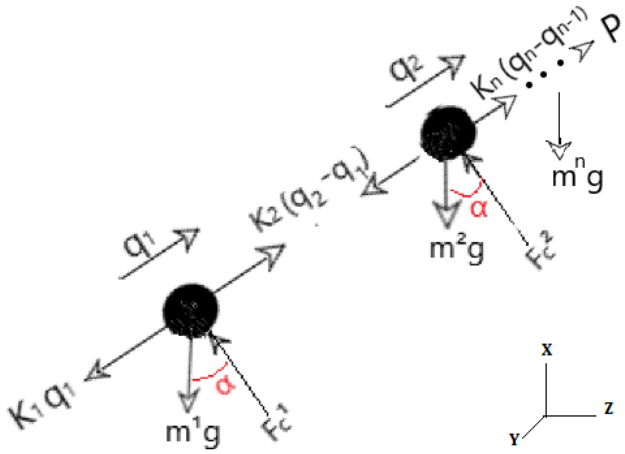

Figure 6. Problem Diagram-Angled case

F_c is force that act on the n particles are defined in the Cartesian coordinate system.

3.2 Mathematical model

The mathematical development to find the displacements is given following chapter 3 of Shabana (2005). The system has n independent coordinates q_1 until q_n which are the position vectors of the body in the Cartesian plane. For this paper n=10 was chosen, however, it should be noted that the discretization varies with each project.

According to Jin Xu (2020) spring stiffness related to the application of a force in the z direction and the external force vectors are given by the "Eq (2)" and "Eq (3)".

The springs have equal lengths, and the same material, then the values are equal and can be calculated to the "Eq (1)".

$$K_z = \frac{EA}{L} \tag{1}$$

E is young material module, A is section area and L, beam length.

$$F_e^1 = \begin{bmatrix} (k_2(q_1 - q_2) - k_1 q_1)cos(\alpha) \\ (k_2(q_2 - q_1) - k_1 q_1)sin(\alpha) - m_1 g \\ \dots \\ (k_n(q_{n-1} - q_n) - k_{n-1} q_{n-1})cos(\alpha) \\ (k_n(q_n - q_{n-1}) - k_{n-1} q_{n-1})sin(\alpha) - m_n g \\ 0 \end{bmatrix} \tag{2}$$

$$F_e^n = \begin{bmatrix} (P - k_2(q_1 - q_2))cos(\alpha) \\ (P - k_2(q_2 - q_1))sin(\alpha) - m_2 g \\ \dots \\ (P - k_n(q_{n-1} - q_n))cos(\alpha) \\ (P - k_n(q_n - q_{n-1}))sin(\alpha) - m_n g \\ 0 \end{bmatrix} \tag{3}$$

The principle of virtual work for equilibrium states that the virtual work of externally applied forces of a particle system in equilibrium with workless constraints is equal to zero. The virtual work of external forces can be written as "Eq (4)" and "Eq (5)"

$$\delta W_e = \delta W = \sum_{i=1}^{np} F_e^{iT} \delta r^i = F_e^{1T} \delta r^1 + F_e^{2T} \delta r^2 ... + F_e^{nT} \delta r^n \tag{4}$$

So,

$$[K_2(q_2 - q_1) - K_1 q_1(cos^2(\alpha) + sin^2(\alpha) - m_1 g sin(\alpha)]\delta q_1 + ...[P - (q_n - q_{n-1})(cos^2(\alpha) + sin^2(\alpha) - m_n g sin(\alpha)]\delta q_n \tag{5}$$

Using the trigonometric identity $sin^2 + cos^2 = 1$ and knowing that q_1 until q_n are linearly independent, the coefficients

can be set equal to zero, as can be seen in the "Eq (6)".

$$\begin{aligned} Q_1 &= K_2(q_2 - q_1) - K_1 q_1 - m_1 g sin(\alpha) = 0 \\ &\dots \\ Q_2 &= K_3(q_3 - q_2) - K_2 q_2 - m_2 g sin(\alpha) = 0 \\ &\dots \\ Q_n &= P - K_n(q_n - q_{n-1}) - m_n g sin(\alpha) = 0 \end{aligned} \tag{6}$$

Finally, equations can be written in matrix form, according to the "Eq (7)"

$$\begin{bmatrix} -m_1 g sin(\alpha) \\ -m_2 g sin(\alpha) \\ \dots \\ P - m_n g sin(\alpha) \end{bmatrix} = \begin{bmatrix} K_1 + K_2 & -K_2 & \dots & Kn \\ -K_2 & K_2 & \dots & Kn \end{bmatrix} . \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{bmatrix} \tag{7}$$

Isolating q from the "Eq.(7)" and with "Equation (8)" it is possible to find displacements.

$$[r] = \begin{bmatrix} cos(\alpha) \\ sin(\alpha) \\ \dots \\ 0 \end{bmatrix} . \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{bmatrix} \tag{8}$$

4. IMPLEMENTATION

4.1 MATLAB-MBD

The code in MATLAB for calculating displacements was based on the mathematical modeling covered in the previous ones, with the same sequence of steps. The equations were used for implementation in MATLAB, with the final formula denoted in "Eq (8)".

Input parameters like stiffness, height, width and thickness and another were chosen according to the "Tab. 1". In order to generate analysis cases, the Latin Hypercube Sampling (LHS), a statistical method for generating random samples from initial values was used. This method can be used in several programming languages.

Table 1. MATLAB Input Parameters

Variable	Value
Number of elements (n)	10
Angle (α)	90º
Modulus of Elasticity (E)	75x10^6 (MPa)
Area (A)	[0.175 : 0.325] (m²)
Length (L)	60 (m)
Mass (m_1 ... m_n)	2500 (Kg)
Force on main body	10 x 10^6(N)
External width (b_{ext})	[1.4 : 2.6] (m)
Internal width (b_{int})	[1.05 : 1.95] (m)
External height (h_{ext})	[2.1 : 3.9] (m)
Internal height (h_{int})	[1.75 : 3.25] (m)
thickness (t)	[0.35 : 0.65] (m)

The program generated several cases, calculating the values of the displacements. A set of these outputs can be seen in "Tab. 2".

Table 2. Vertical displacement values for each node

Displacements (m)
0.0
0.08072
0.30775
0.66490
1.13585
1.704353
2.35415
3.06899
4.62873
5.44113

4.2 PYTHON

4.2.1 Neural Network

In order to analyze the best balance of performance and prediction of the neural network, the main characteristics were modified, such the number of layers, neurons, activate function and control parameters, where the main objective is the obtain satisfactory prediction with a good performance.

For the implementation of the neural network, the Multilayer Perceptron (MPL) Regressor model was implemented in PYTHON with 1 hidden layer and 15 neurons.

The input data for this study are the possible geometric configurations upon established design ranges for every part of blade discretization, where was varied the height, width of stringer and their thickness, and the output data is the displacements obtained from MBD. With these data, a supervised training can be done, which the neural network can estimate the displacements given a geometric configuration and generate a metamodel. The metric used for evaluate was the R^2.

For the training data, was generated 3000 cases of different structural configurations, where 80% is for the training set and 20% for test set. As the case treated is non-linear, a hyperbolic tangent was used for the activation signal, this function is responsible for performing the calculations from the input variables. Also, The solver used was the SDG with an adaptative leaning rate. The evolution of training can be seen on "Fig. 7".

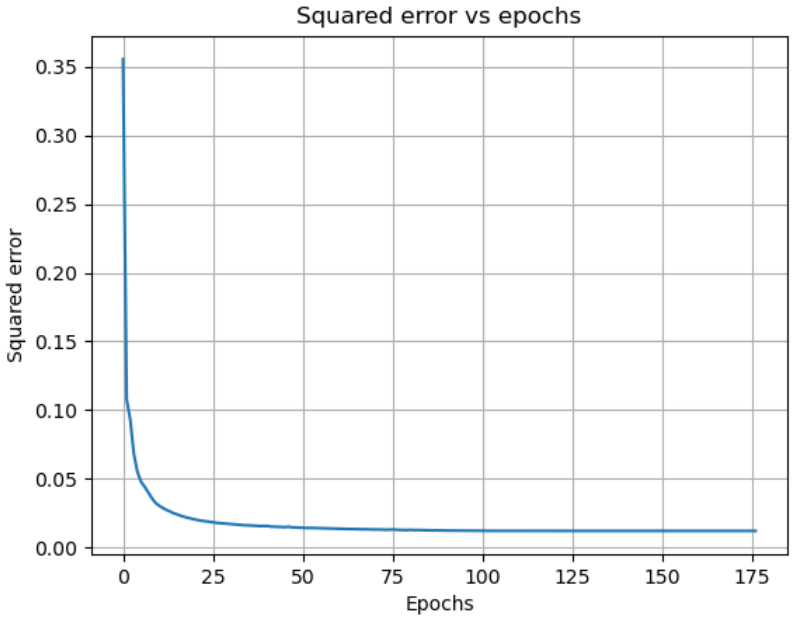

Figure 7. Squared error of trained neural network

After getting the best neural settings, the weights of neurons links was obtained and the trained neural network was mathematic implemented in a single function. Finally, this function constituted the metamodel, which can calculate the displacements.

4.3 Metamodel

The metamodel was constituted by the trained neural network (mathematical implemented) in PYTHON.

The objective function of the optimization process is minimize the weight metamodel function where calculates the weight of the structure, the restrictions are the design criteria of the blade.

So the input of the metamodel is geometric configuration of the stringer and the output is a delta-weight (δ_w),in other words, the new weight calculated by varying the geometry.

After the metamodel is ready, the user must proceed with the analysis choosing a critical design deflection value, if the input data generate larger values of critical criteria, the optimization module changes geometric values to reach an acceptable optimal value, ie , the one that generates the minimum weight. Thus, the new values will be recalculated in the trained displacement neural network. With the new values, the new weight of the structure can be calculated and compared with the nominal one, generating the δ_w.

In this paper, a critical design value will not be determined, as the study is restricted to the generation of the metamodel.

5. RESULTS AND DISCUSSIONS

With the output data generated by MBD in MATLAB, the neural network was trained with 3000 geometric cases, the estimated values vs real values can be seen on "Fig. 8", which was obtained a R^2 of 0.985.

The continuous line represents the R^2 and the dotted line the results obtained, it is observed that both are very close.

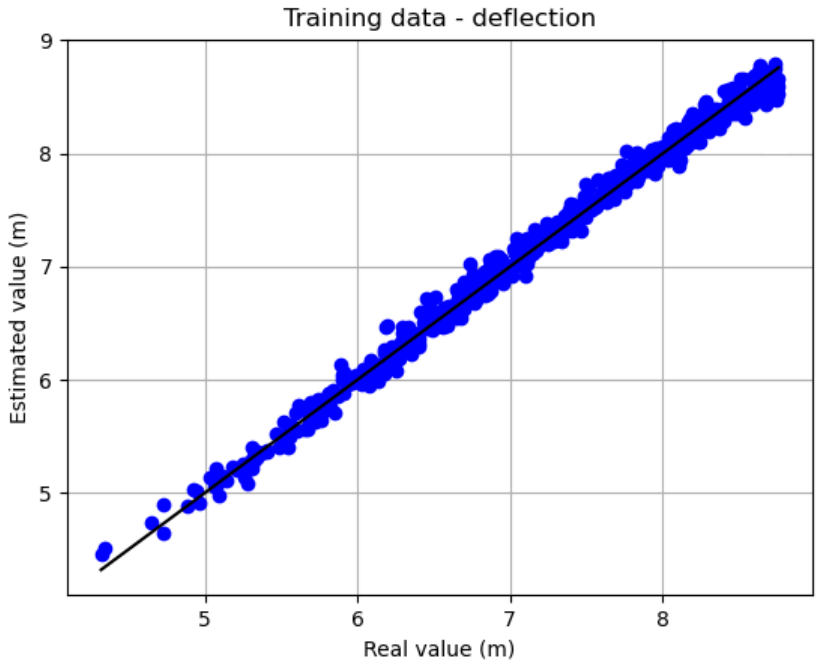

Figure 8. Real vs estimated values - Training data

For validate the training data, a test data was evaluated, which can be seen on "Fig. 9", with a R squared of 0.978.

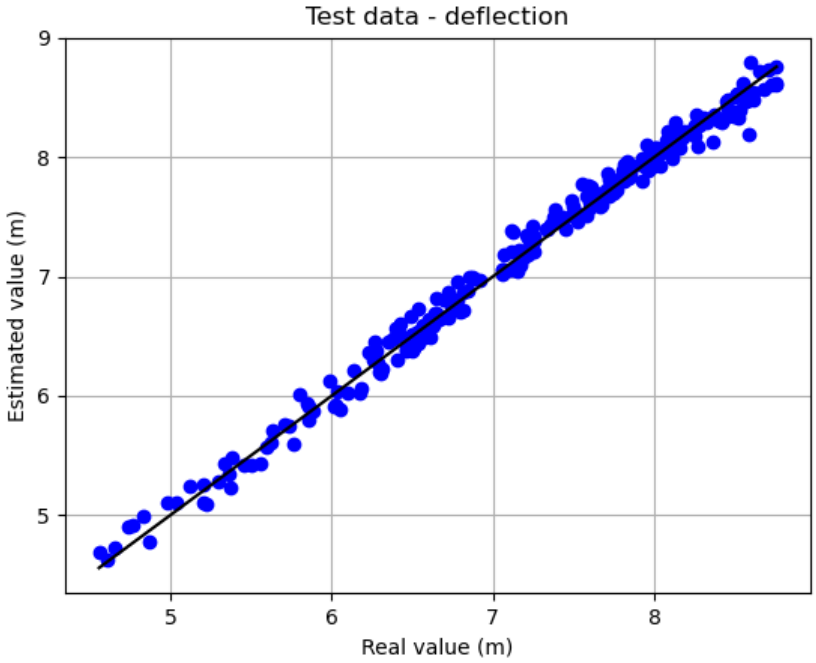

Figure 9. Real vs estimated values - Test data

Observing the results, the neural network has a good estimation accuracy, thus, the metamodel was made by the coefficients obtained by this trained neural network.

6. CONCLUSION

The analyzes in neural network and metamodel proved to be good considering the simplicity of the model. A relatively high speed for generating cases was observed as well as the calculation of deflections for each case by MBD method. Also, it was noted a high speed for training the neural network for the implementation on the MDO. The obtaining values was very close to those calculated, thus, having a good estimate within the given range.

7. ACKNOWLEDGEMENTS

Thanks to Federal University of ABC for the support provided.

8. REFERENCES

Jin Xu, Lei Zhang, X.L.S.L.K.Y., 2020. "A study of dynamic response of a wind turbine blade based on the multi-body dynamics method". *Renewable Energy*, Vol. 155, pp. 358–368.

Júnior, P.R.C., 2011. "Otimização multidisciplinar em projeto de asas flexíveis utilizando metamodelos".

Learning, S., 2020. "Sklearn neuralnetwork mlpregressor".

Services, A., 2020. "How to interpret r-squared in regression analysis".

Shabana, A.A., 2005. *Dynamics of Multibody Systems*. Mechanisms and Machine Science. Cambridge edition 3. doi: 10.1017/CBO9780511610523.

Songyi Jiang, S.S.D., 2011. "A four-rigid-body element model and computer simulation for flexible components of wind turbines". *International Mechanical Engineering Congress Exposition IMECE2011*, p. 7. doi: https://doi.org/10.1016/j.renene.2020.03.103.

UFPR, 2020. "Departamento de engenharia de produção –(simplex revisado)".

9. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.