Strojnícky časopis – *Journal of* MECHANICAL ENGINEERING,
VOL 74 (2024), NO 1, 1 - 8

IMPLEMENTATION OF REDUCED-ORDER MODEL TO DESIGN OF ELECTRIC POWERTRAIN RUBBER MOUNT

DANKO Ján[1], BERNÁTH Martin[1], MAGDOLEN Ľuboš[1], MILESICH Tomáš [1], DOBROVOLNÝ Juraj[1]

[1] *Slovak University of Technology in Bratislava, Faculty of Mechanical Engineering, Institute of Automotive Engineering and Design, Nám. Slobody 17,812 31 Bratislava, Slovakia, e – mail: jan.danko@stuba.sk*

Abstract: This paper deals with possibilities of utilizing neural networks in reduced order modelling. First section of the paper contains theory on reduced order models and their implementations. Following this, subsequent chapter contains methodology concept for obtaining sufficient amount of data and creating reduced order models as well as their evaluation process. Next section describes our findings and summarizes them. Conclusion describes further steps we plan to take in this research.

KEYWORDS: reduced order model, neural network, multilayer perceptron, proper orthogonal decomposition, rubber mount

1 Introduction

Our aim is to devise a methodology that allows us to reduce computational load of a rubber mount design process. Purpose of this paper is to analyse possibilities of using neural network assisted reduced order models in order speed up design process. When designing rubber mounting bushings, it is required to perform multiple finite element analyses in order to find out its stiffness and damping. Doing so with large number of variations can be time consuming and taxing on computational resources. If we utilize a neural network assisted reduced order model, we could decide which variants are feasible to further investigate purely based on preliminary results from the ROM.

2 Reduced order modelling

Reduced order models (ROMs) are a technique for reducing the computational complexity of mathematical models in numerical simulations. They are used to approximate the behaviour of a high-fidelity, complex model while preserving the expected accuracy within a controlled environment [1].

Reduced order models are useful in settings where it is required to perform large number of simulations in a limited time frame. This can be due to limitations in computational resources or the requirements of the simulations setting. For instance, real-time simulation applications or many-query applications in which a large number of simulations needs to be performed at appropriate speed in order to maintain synchronization. Reduced order models can also be created using neural networks for interpolation.

Neural network assisted ROMs are a type of ROM that use deep neural networks to improve the accuracy of the reduced order model. The neural network is trained to learn the relations between the inputs and outputs of the full order model and is then used to predict the outputs of the reduced order model.

DOI: 10.2478/scjme-2024-0001, Print ISSN 0039-2472, On-line ISSN 2450-5471

The main difference between neural network assisted ROMs and normal ROMs is that neural network assisted ROMs use deep learning neural networks to improve the accuracy of the reduced order model. This allows for more accurate predictions of the behaviour of the system being modelled and can be particularly useful in applications where the full order model is highly nonlinear or has a large number of degrees of freedom [1]. Additionally, utilizing deep learning speeds up the creation of ROM, however this is offset by the amount of time required to train the neural network.

Depending on application, neural network assisted ROMs can be more computationally expensive than normal ROMs, as they require training the neural network on the full order model. The accuracy of the neural network assisted ROM is also dependent on the quality of the training data, and the neural network may not be able to accurately predict the behaviour of the system outside of the range of the training data [1].

2.1 Neural Networks used for Reduced order modelling

As mentioned in previous chapter, for the initial stages of this research we are bound to the ANN solutions available to us in CAE Lunar software, however we plan to modify our methods in future to allow us to use MATLAB for creating the reduced order model, which will broaden our options. CAE Lunar currently supports two types of neural networks, being Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM). In the next chapters we will dive deeper in to their advantages and shortcomings.

2.2 Multilayer Perceptron (MLP)

An MLP is a type of artificial neural network (ANN) architecture, belonging to the broader class of feedforward neural networks. The basic structure of an ANN is a network of small processing units, or nodes, joined to each other by weighted connections.

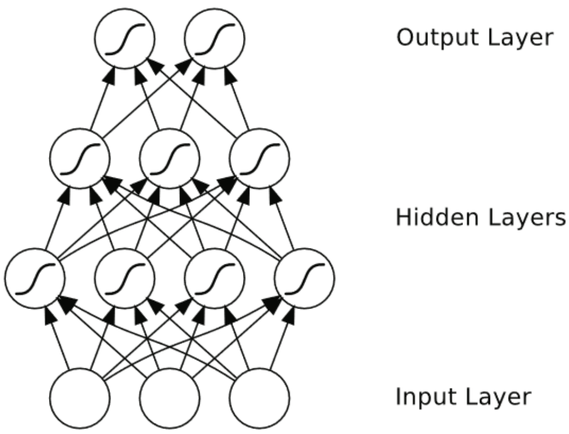

Fig. 1 Multilayer Perceptron

In terms of the original biological model, the nodes represent neurons, and the connection weights represent the strength of the synapses between the neurons. The units in a multilayer perceptron are arranged in layers, with connections feeding forward from one layer to the next. Input patterns are presented to the input layer, then propagated through the hidden layers to the output layer. This process is known as the forward pass of the network. Since the output of an MLP depends only on the current input, and not on any past or future inputs, MLPs are more suitable for pattern classification than for sequence labelling. An MLP with a particular set of weight values defines a function from input to output vectors. By altering the weights, a single MLP is capable of instantiating many different functions. Indeed, it has been proven [3] that an MLP with a single hidden layer containing a sufficient number of nonlinear units can approximate any continuous function on a compact input domain to arbitrary precision. For this reason, MLPs are said to be universal function approximators. [4]

2.3 Proper Orthogonal Decomposition (POD)

Proper Orthogonal Decomposition (POD), also known as Principal Component Analysis (PCA) or Karhunen-Loève decomposition, is a numerical mathematical technique used primarily in data analysis and reduced order modelling. It's often implemented in fields such as fluid dynamics, structural mechanics, and signal processing. [5]

POD works by decomposing a dataset, typically a set of high-dimensional observations or snapshots, into a smaller number of orthogonal modes that capture the dominant features or variability in the data. These modes are ordered in terms of importance, with the first mode capturing the most significant variability, the second mode capturing the next most significant variability, and so on. [5]

The process involves collecting a dataset of snapshots representing the system's states. After preprocessing the data, it's organized into a snapshot matrix. Singular Value Decomposition (SVD) is then applied to this matrix, yielding three matrices: U, Σ, and V^T. The left singular vectors in matrix U represent the POD modes, ordered by significance. Selecting a subset of these modes based on their associated singular values captures the dominant variability in the data. The selected modes can then be used to project and reconstruct the original data, providing reduced-dimensional representations. The modes are analyzed to interpret the underlying patterns or dynamics in the dataset afterwards. Overall, POD serves as a powerful tool for dimensionality reduction, model simplification, and feature extraction in complex datasets. [5]

3 Methodology of creating reduced order models

In order to create a reduced order model, its first important to gather large amounts of input data. The input data in this research was taken from results of finite element method (FEM) structural analysis performed on multiple dimensional variants of the same cross section model of a rubber mount. Following chapter describes this in further detail. The next step is extracting this data from Fem software and inputting it into software where ROMs will be created. In order to create a ROM, its required to split input data into two categories, first is training data and second is validation data. Subsequent chapters describe ROM configuration in greater detail. In order to compare neural network assisted reduced order model (NNROM) and mathematical reduced order model, we have created one reduced order model from each group and compared them against FEM analysis, in order to evaluate each approach both based on accuracy and time required for computation.

Evaluation is done through creating 10 new variants that haven't been used yet and analysing them using the three above mentioned methods. Results gathered via reduced order models will then be compared to FEM results.

4 Creating model of rubber mount and dimension variants

First a basic cross section of a rubber mount was designed. This cross section was then parametrized, meaning that each dimension had its parameter instead of certain dimension. These parameters were varied in a certain span. Variations were created using Monte Carlo method. Creation process was automated using a procedure that generates the shape of the cross section.

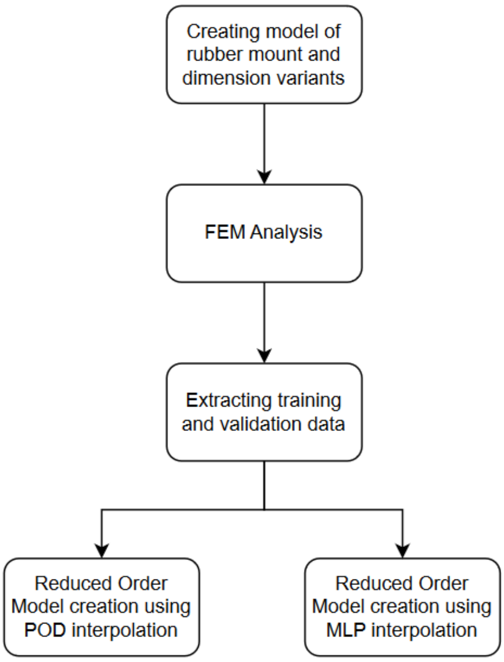

Fig. 2 Method diagram

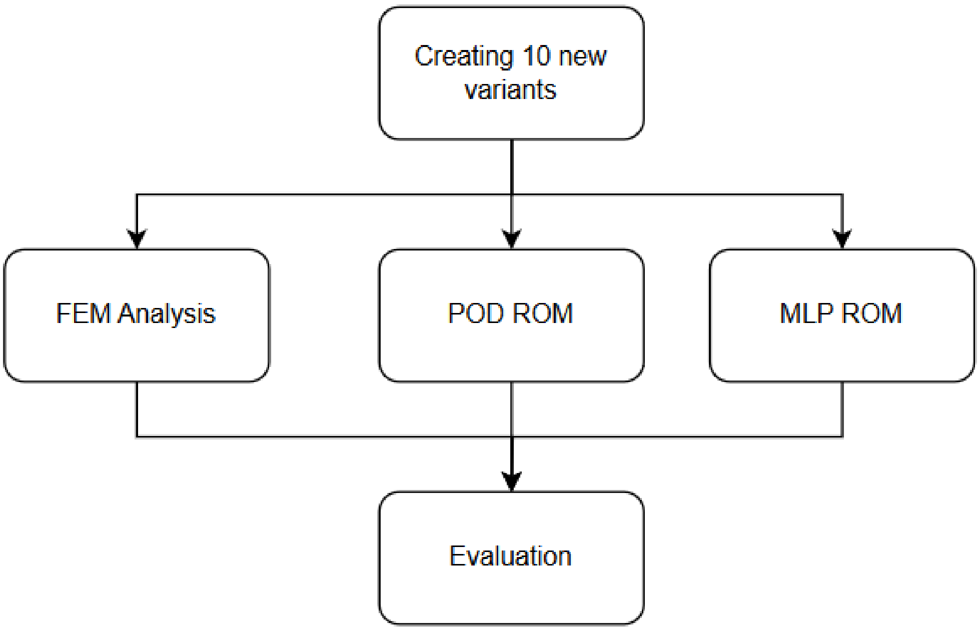

Fig. 3 Evaluation process diagram

4.1 FEM analysis of rubber mount shape variants

In total we created 30 variants using this method. Static structural analysis was performed on these variants using Marc software. The rubber mounts were loaded in Y (vertical) direction with various displacement of 5mm in 0.5mm increments [7]. Since we are only performing a 2D plane strain structural analysis of a cross section, we can gather large amounts of data in a relatively short time. Ideally in future, we intend to perform structural analyses of a 3D model of a rubber mount to give us variations of depth. For simplification, we only analysed a single material the rubber mount can be made of. It is possible to test multiple materials to introduce another variable that can be used to achieve ideal characteristics [6].

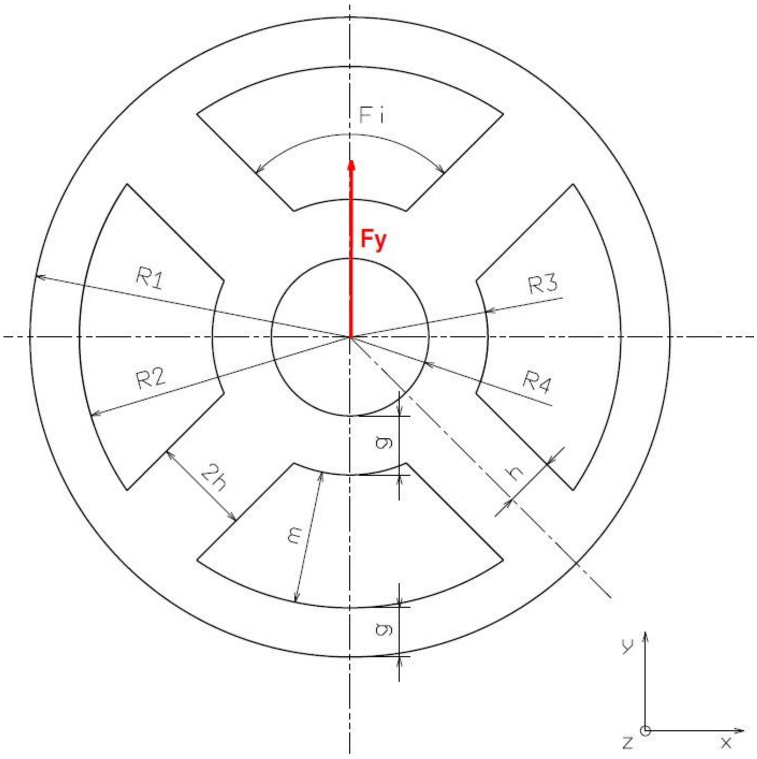

Fig. 4 Cross-section of rubber mount

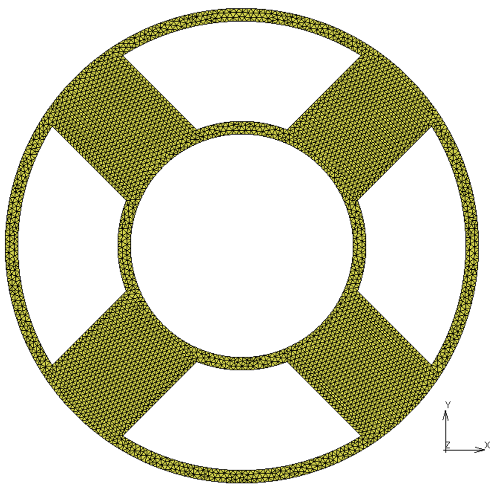

Fig. 5 Variant of rubber mount shape with created mesh

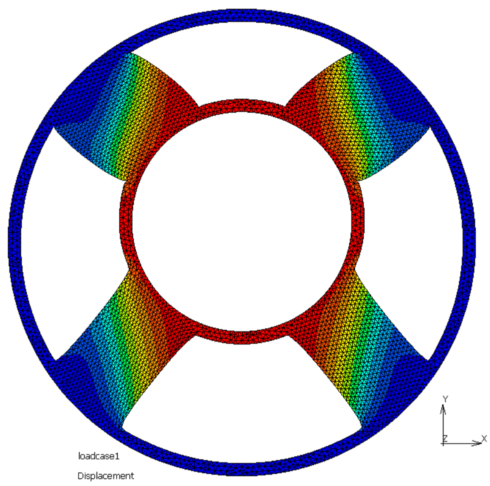

Fig. 6 Deformation of rubber mount

Output data was managed by a procedure that automatically ordered the output values into text file which can then be imported into CAE Lunar software.

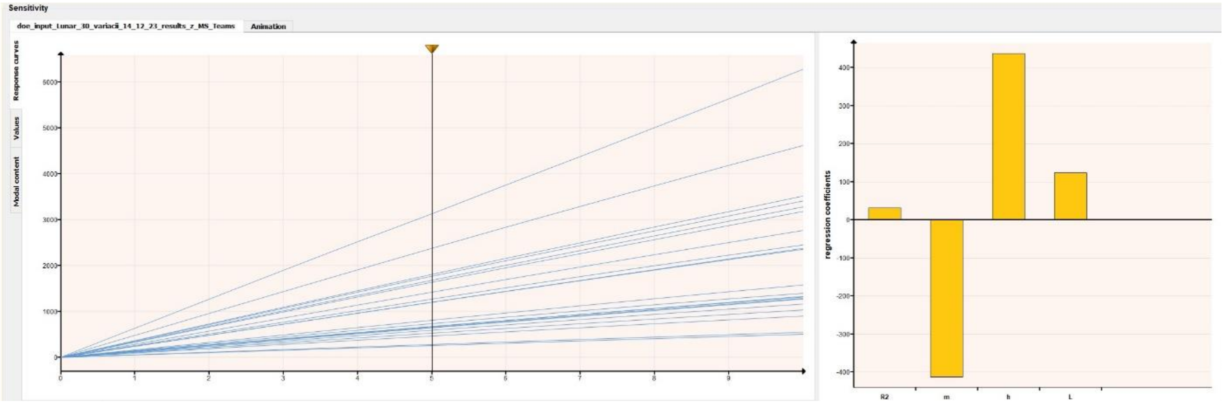

Fig. 7 FEM analysis output data

4.2 Creating reduced order models

Structural analysis data had been exported into a text file through the procedure. Data contains reaction force recorded at maximum displacement of 5mm for a total of 30 shape variations. This data was then imported into CAE Lunar, where split between training and expected data was defined.

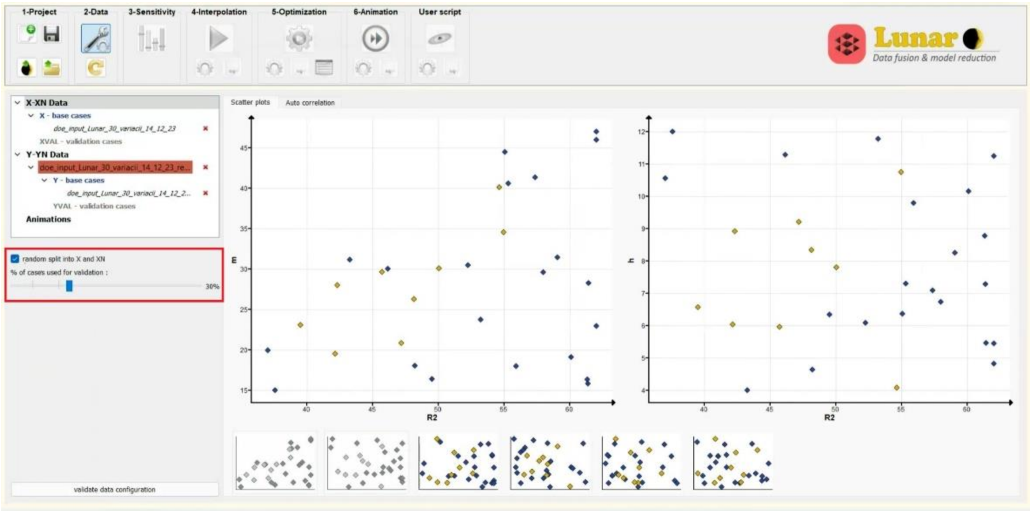

Fig. 8 Scatter plot of imported data, showcasing split ratio

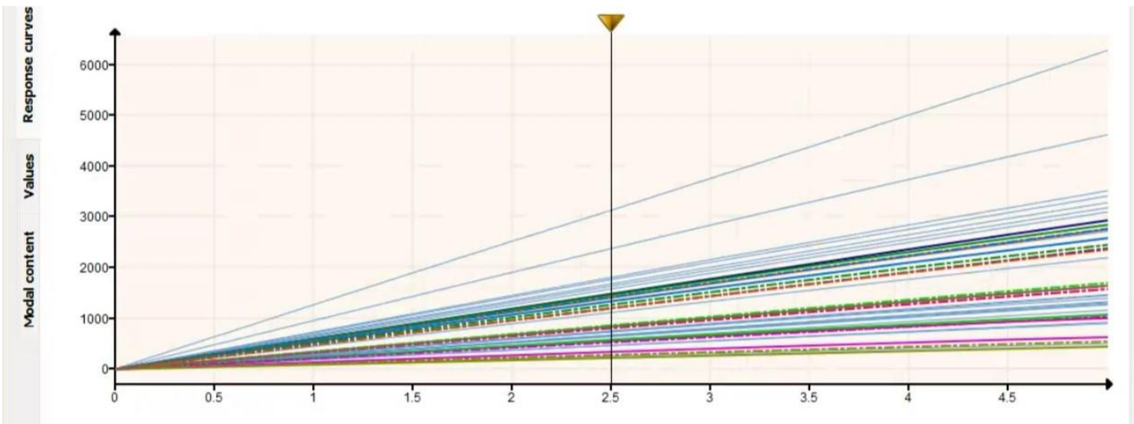

Fig. 9 Imported, validation and expected data curves

A ratio of 70/30 was chosen, therefore 30% of the data will be used for validation. This means that 21 values are used for training and 9 values are used for validation. Validation data then turns into expected data, which is set aside for comparison once simulation is completed.

Two types of solvers were chosen for interpolation of the reduced order model. First solver being Proper Orthogonal Decomposition (POD) (Fig.10) with interpolation method being Adaptive Radial Basis Function (ARBF). Second solver type is using Multilayer Perceptron (MLP) (Fig.11) neural network with two layers. This can give us an insight into how accuracy between mathematical method and neural network compares [8].

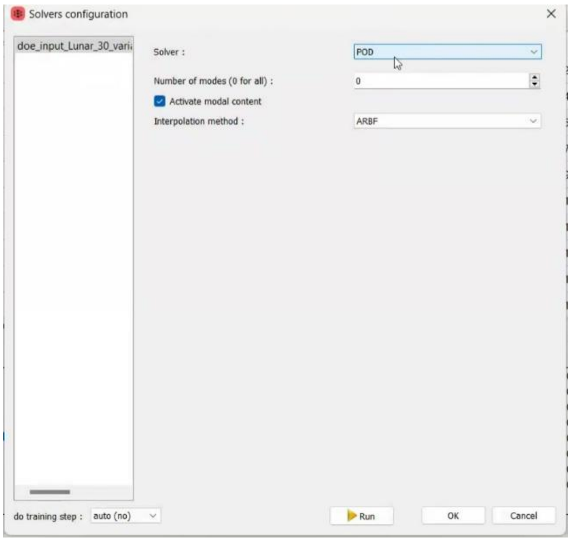

Fig. 10 POD Solver settings

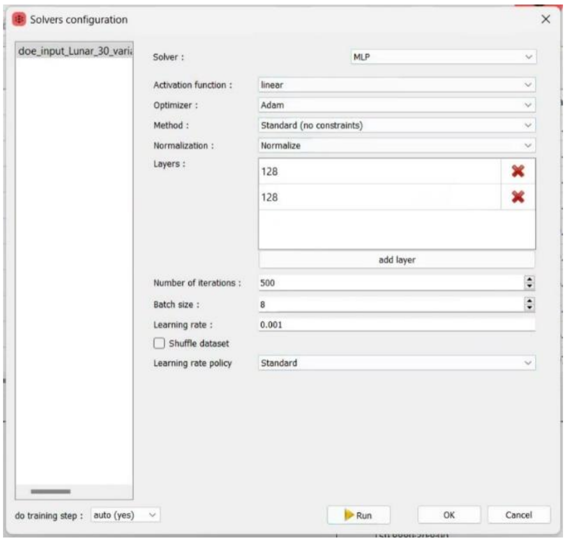

Fig. 11 MLP Solver settings

5 Evaluation

In order to evaluate the methods, another 5 shape variants of rubber mount had to be created, these have been input into FEM analysis and reduced order models. These 5 shape variants showcase the results. Comparison includes static stiffness (C_{stat}) values obtained from Marc FEM structural analysis and POD and MLP reduced order model static stiffness□ from CAE Lunar.

Error was calculated by subtracting the POD and MLP ROM stiffnesses from FEM stiffnesses. Where FEM results were considered 100%. That way we know how much the results obtained from reduced order models, differ from reference values obtained from FEM structural analysis.

Table 1 Evaluation of analysis methods

Shape variant	R2 <43,62>	m	h	L	FEM C_{stat} [N/mm]	POD C_{stat} [N/mm]	Error [%]	MLP C_{stat} [N/mm]	Error [%]
1	43.0	30	8	90	339.68	315.98	6.98	344.76	1.49
2	47.2	30	8	90	351.24	322.00	8.32	346.00	1.49
3	51.4	30	8	90	355.24	330.25	7.03	349.80	1.53
4	55.7	30	8	90	359.05	339.07	5.57	353.59	1.52
5	62.0	30	8	90	361.88	356.94	1.36	359.28	0.72
						Average error:	5.85	Average error:	1.35

CONCLUSION

Based on the results we can conclude that utilizing neural networks in creation of reduced order models allows us to hit higher accuracy with lower errors. Average error with neural network assisted MLP reduced order model was 1.35% whereas the average error using mathematical POD reduced order model was 5.85%. Training a neural network adds to the complexity of process and also requires additional computational resources, but its still a more effective method than finite element structural analysis, depending on the target accuracy we desire. Using reduced order models also brings significant time savings. FEM analysis of 10 evaluation variants took roughly 300 seconds or 5 minutes. POD reduced order model returned all 10 results after roughly 5 seconds and MLP reduced order model after roughly 30 seconds. Continuation of this research will be aimed at repeating this process with data from a dynamic analysis, trying to approximate behaviour of rubber mount at different frequencies. Further research will also explore possibilities of MATLAB to broaden and further adjust neural network options.

ACKNOWLEDGEMENT

This research was supported by the Slovak Research and Development Agency under Contract No. APVV-20-0428.

REFERENCES

[1] https://en.wikipedia.org/wiki/Model_order_reduction

[2] Zambrano, V.; Brase, M.; Hernández-Gascón, B.; Wangenheim, M.; Gracia, L.A.; Viejo, I.; Izquierdo, S.; Valdés, J.R. "A Digital Twin for Friction Prediction in Dynamic Rubber Applications with Surface Textures", Lubricants 9 (5), 57, **2021.** DOI: 10.3390/lubricants9050057

[3] Hornik, K., Stinchcombe, M., White, H. "Multilayer Feedforward Networks are Universal Approximators", Neural Networks 2 (5), pp. 359 – 366, **1989**.

[4] Graves, A. "Supervised Sequence Labelling with Recurrent Neural Networks", Springer Berlin, Heidelberg, 13 April **2014**. eISSN 978-3-642-24797-2

[5] https://en.wikipedia.org/wiki/Proper_orthogonal_decomposition

[6] Danko, J., NEMEC, T., MILESICH, T., MAGDOLEN, Ľ.: "Dynamic Stiffness Determination of the Rubber Bushing Fem Modelwith Viscoelastic Material Behaviour" Strojnícky časopis – Journal of Mechanical Engineering 72 (3), pp. 113 – 120, **2022**, DOI: 10.2478/scjme-2022-0045

[7] Bernáth, M., Danko, J., Magdolen, Ľ., Milesich, T., Nemec, T.: "Analysis and Design of Rear Twin Electric Powertrain Mounting System in E-Up Boost", Strojnícky časopis – Journal of Mechanical Engineering 73 (2), pp. 9 – 14, **2023**. DOI: 10.2478/sc-jme-2023-0018

[8] Bucha, J., Danko, J., Milesich, T., Nemec, T., Magdolen, Ľ. "Comparison of FEM Simulations of Rubber-Metal Mount Element" Strojnícky časopis – Journal of Mechanical Engineering 72 (1), pp. 15 – 22, **2022**. DOI: 10.2478/scjme-2022-0002