Coupling of Physical System Modeling with Artificial Neural Networks

Yashwant Kolluru
RWTH Aachen University

Coupling of Physical System Modeling with Artificial Neural Networks

1st Yashwant Kolluru
eBikes Department
Robert Bosch GmbH
Reutlingen, Germany
yashwant.kolluru@de.bosch.com

***Abstract*—Finite element method is widely used to study various physical mechanisms across domains like structural, electro-magnetics, fluid dynamics, thermal etc. However, FE simulations usually require complex procedures to set up and long computing times to obtain final simulation results. In this study, by using of machine learning techniques, an artificial neural network model is developed and coupled with the FE templates to estimate the structure dynamics of the components. The neural network model is designed and trained to take the input of FE and directly output the corresponding structural characteristics like displacements, velocities etc. The trained model is capable of predicting the vibration and acoustic characteristics of the corresponding assemblies. It is build by analyzing different physical mechanisms slowly increasing the complexities in the system and components. As a part of it, a 1-D harmonic oscillator, 2-D cantilever beam and a 3-D cube model are studied. This study marks, one of first major attempt to demonstrate the feasibility and great potential of using the neural network technique as a fast and accurate surrogate of FEA for harmonic and acoustic analysis.**

***Index Terms*—Finite Element, Vibro-acoustics, Artificial Neural Networks.**

I. INTRODUCTION

Modeling and simulation of physical mechanisms include extensive use of software packages depending on the modeling technique chosen. For instance, finite element technique is used when a detailed level of the analysis of structures is required. It basically tries to solve the underlying differential equations considering the appropriate initial, boundary and load conditions. When the problem becomes complex and the model grows large, time needed to solve the resulting systems may shift from hours to weeks. If the input parameters of the FE template need to be modified, the simulations has to be re-executed from scratch. Recent progress in machine learning algorithms and their success in applications across different domains demonstrate that, if rightly chosen, trained and tested, these models can significantly improve conventional techniques. In this project, a possible approach to understand and complement the finite element studies with machine learning is attempted.

The vast majority of analysis in structural mechanics, electromagnetic, fluid simulations, and many other areas is mostly based on the FE for solving the initial and boundary value problems. Using this method, approximate solution of the representing partial differential equation is computed at the discrete points over the computational domain through the analysis of resulting linear or nonlinear system. For the real time examples, the size of the system can be extremely large, i.e. it can have millions of degrees of freedom. This would cause for hours or days of simulation time on a cluster or supercomputer.

Irrespective of the scalability regarding the solver, finite element modeling requires massive computational resources and, except for the end results, the machine experience acquired during the simulation is completely lost. It means that when the input has to be adjusted, even slightly, or one needs to redo the study done elsewhere, in most of the cases the time-consuming analysis has to be done from scratch. On the other hand, when the physical system is properly discretized, the finite element results are accurate and, along with the input parameters of the model templates, can be used to train the machine learning models. One of the options to do this efficiently is to train the model on the large sets of data generated by the well-developed conventional FEM tools on the random basic problems. The training data can also be complemented by the actual measurements as well as by the simulation results for the realistic problems shared by the users of the FEM packages. When properly trained, this model can be used for a wide range of other applications.

This goal of the paper is developing efficient machine learning codes to predict the complex vibrations characteristics present in the physical system. It is done by providing a series of basic examples, gradually increasing the complexities inside the modeling templates.

In the first example, a 1-D oscillator is considered. It is one of the basic physical systems. When excited from its equilibrium it would experience a restoring force proportional to the displacement. An artificial neural network on the data generated from the analytical values of the harmonic oscillator equation is generated. The algorithm is used to predict the amplitude of the oscillator as a function of frequency.

In the second example, a 2-D cantilever beam is constructed and simulated in finite element package. A harmonically varying force is applied as the load condition. The resulting model is solved across different spectrum. In the end, a machine learning code is trained on the generated two-dimensional data to predict the maximum displacements of the system in both of those directions respectively.

In the third example, a 3-D cube model is developed further in the finite element software. The model is calculated for vibrations across frequencies of interest and also validated with help of experimental setups. The resulting data is once again used for developing a back propagation neural network to predict the surface accelerations across various critical areas.

Before discussing the FE models and the associated neural networks, a holistic view of the data flow within the study is illustrated in figure 1. The parameters are generated using various distribution functions like beta, normal depending on the purpose and the problem. Once the parameters are defined, they are passed in to the FE templates. Later, along with the input parameters, the results of FE domain are transferred to machine learning codes to make prediction of underlying physical mechanism.

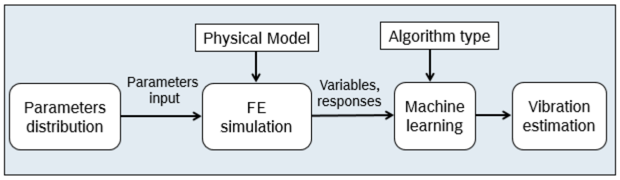

Fig. 1. Schematic view of the coupling between numerical method and neural network

II. 1-D Harmonic Oscillator

This section focuses on a basic level example for analyzing the impact of machine learning algorithm. A simple oscillator with certain mass and stiffness is chosen. Initially, the underlying analytical solution is described, followed with the model development for the artificial neural network. Table I describes the analytical and neural network properties associated with the harmonic oscillator.

TABLE I
Properties of the Model and Neural Network

Analytical properties		**NN properties**	
Mass	1.5 (Kg)	Type	Back propagation
Young' Modulus	210000 (N/mm^2)	Activation function	Sigmoid
Frequency range	0 -100 (Hz)	Structure	1- Inner, 20,20- Hidden, 1- Outer
Damping ratio	0.2	Loss function	Mean square error

A. Analytical Solution

Equation 1 represents the steady state amplitude for the harmonic oscillator. Terms A_s, ζ and ω indicate the amplitudes, damping ratio, frequencies of the system. The analytical solution is solved for 800 discrete frequency points. Detailed information about the oscillator setups can be found in literature like [1].

$$A_s = \frac{1}{\sqrt{(1-\omega^2)^2 + (2 \cdot \zeta \cdot \omega)}} \tag{1}$$

B. Machine Learning Solution

The neural network consisting of multiple hidden layers is trained on a random subset and used to predict the response of the oscillator to a harmonic loading at particular frequency. The network is correctly designed and trained to achieve results with handsome values of accuracy between the analytical solution and the model prediction. For this instance, the network had one independent variable i.e. frequency and one dependent variable i.e. amplitude of steady state solution. The model is equipped with two hidden layers both with the height of 20 neurons. The stochastic gradient descent method is selected for implementation in TensorFlow. Finally, the mean squared error loss function is chosen for the minimization.

Modern FE software codes implement complex algorithms that solve large systems of nonlinear or partial differential equations. These computations produce a large amount of data that is usually discarded for future similar computations. The aim of this section is to leverage these data using machine learning in order to improve numerical simulation performance and get predictions at real time. Figure 2 describes the flowchart for machine learning models.

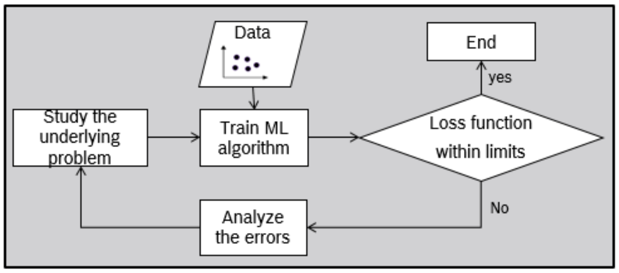

Fig. 2. Machine learning holistic view

Figure 3 describes the results obtained from the neural network. The network is developed with Sigmoid activation functions. It can be clearly seen that the test data fits the analytical solution. These results shown are achieved with 1000 iterations on the training set.

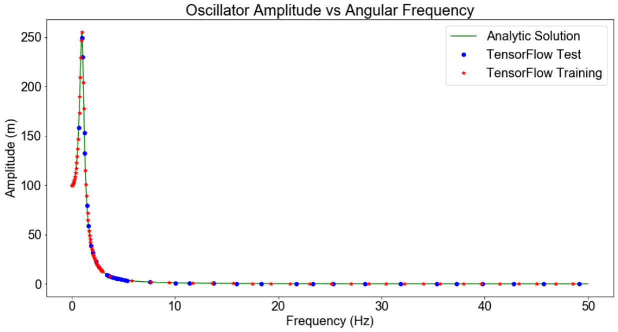

Fig. 3. Model prediction for harmonic oscillator

Figure 4 describes the loss function plotted across the iterations. From this graph it can be inferred that loss decreases to value near zero with growing iterations and hence the predicted values obtained are plausible.

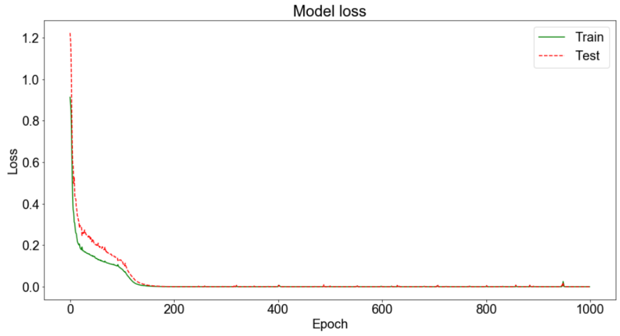

Fig. 4. Loss function of the harmonic oscillator

III. 2-D Cantilever Beam

In the second example, a 2-D cantilever beam model is used for the construction of FE and neural network model. Table II describes the FE and neural network properties used for the cantilever beam model. The simulation is performed with aid of harmonically varying external loading. First, the analysis is executed by using the finite element code for the range of frequencies up to 200 Hz, and then the artificial neural network is trained on the random frequency subset to predict the displacement of the system in directions X and Y. In contrast to earlier example, this model is trained to take into account multiple resonant peaks in different directions.

TABLE II
Properties of the model and neural network

FE Properties		NN properties	
Density	$7.85 \cdot 10^{-6}$ (Kg/mm^3)	Type	Back propagation
Young's Modulus	210000 (N/mm^2)	Activation function	tanh
Frequency range	0 -200 (Hz)	Structure	1- Inner, 50,50- Hidden, 2- Outer
Element function	Linear	Loss function	Mean square error
Step	Steady state dynamic	Epochs	2000

A. FE Modeling of the Beam

Figure 5 describes the FE model built in Abaqus software. Element type of quadrilateral with first order shape functions are used for construction of the model.

B. Artificial Neural Network for the Beam

The network has one input i.e. frequency, two hidden layers both with the height of 50 neurons and two outputs (maximum displacements in X and Y). Additionally, the values are represented in log scale for a better view. The Adam optimization algorithm, extension to stochastic gradient descent implemented in TensorFlow, is used for the minimization of the mean squared error. The results shown in the graphs 6 and 7 are comparison of the beam displacements as calculated with the artificial neural network on the test and training data against the original input from finite element simulator.

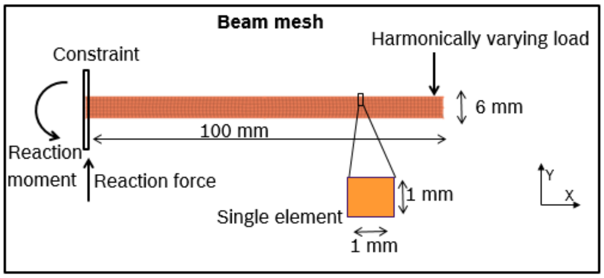

Fig. 5. FE mesh properties for cantilever beam

Figure 6 describes the results obtained from the neural network for the X direction. The network is developed with hyperbolic tangential activation functions. Similarly, figure 7 describes the results obtained from the neural network for the Y direction. It can be clearly observed that the test data fits the analytical solution. Figure 8 illustrates the loss function plotted for the beam model. From this graph it can be interpreted that loss values are close to zero.

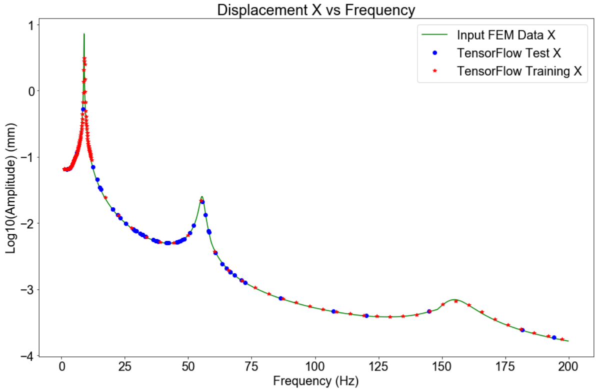

Fig. 6. Model prediction for the beam model (X direction)

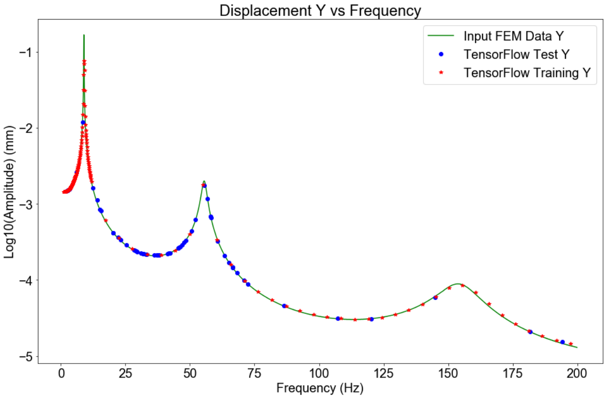

Fig. 7. Model prediction for the beam model (Y direction)

IV. Cube Model

In the third example, a cube structure is investigated for the neural network results on the FE data.

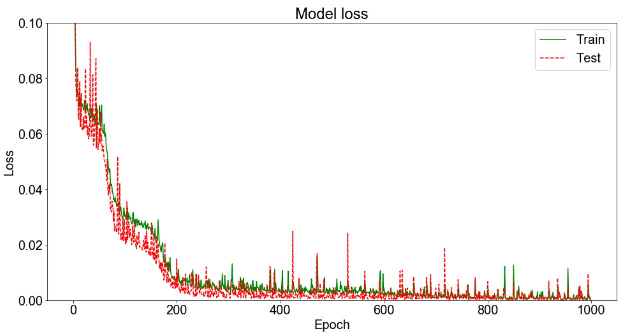

Fig. 8. Loss function of the beam model

A. FE Modeling for the Cube

Figure 9 describes the design of the cube structure. The associated dimensions of this model are presented in table III. The values 55 + 65 in volume tab indicate heights of the two parts respectively.

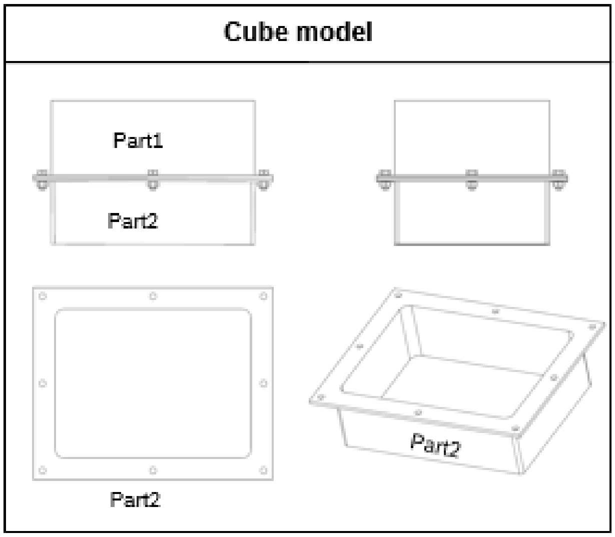

Fig. 9. Design layout of cube model

TABLE III
DIMENSIONS AND MATERIAL INFORMATION (STEEL) OF THE CUBE MODEL

Name	Values	Units
Volume	124 X 164 X (55+65)	mm^3
Thickness	3	mm
Structure - Young's modulus	210000	MPa
Structure - density	7.8 $\times 10^{-6}$	$kgmm^{-3}$
Fluid - bulk modulus	0.142	MPa
Fluid - density	1.3 $\times 10^{-9}$	$kgmm^{-3}$

Table IV represents the FE and neural network properties associated with the cube model. A load of unit magnitude across 0 to 2000 Hz is applied and later compared with the Frequency Response Function (FRF) plots of the experiment. FE of type linear shell with element length 3 mm is used for the construction of the model. Appropriate damping values from the experiments is further inputted in to the FE for proper FRF curves.

TABLE IV
PROPERTIES OF THE CUBE MODEL AND NEURAL NETWORK

FE Properties		NN properties	
Element	Shell	Function	Back propagation
Material Info	Steel (ref tab III)	Activation function	elu
Frequency range	0 -2000 (Hz)	Structure	1- Inner, 40,30- Hidden, 3- Outer
Element type	Linear	Loss function	Mean square error
Step	Harmonic	Epochs	500

B. Artificial Neural Network for the Cube

The modeled network has one input i.e. frequency values in Hz, two hidden layers both with the height of 70 neurons each and three outputs namely, displacements in X, Y and Z coordinates. Due to the features having a broad range of values, a logarithmic scaling is applied. The Adam optimization algorithm, extension to stochastic gradient descent implemented in TensorFlow. Lastly, the activation function of type elu is utilized in the optimizer. Figures 10, 11 and 12 represent the FRF profiles of the cube at a node with maximum accelerations. It can be understood that the trained and test values fit the FE data perfectly.

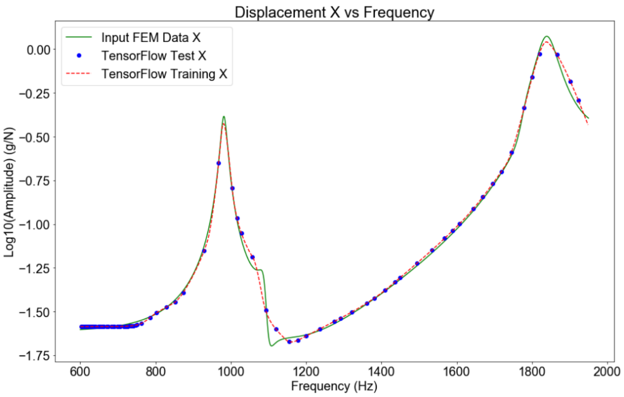

Fig. 10. Model prediction for the cube model (X direction)

Figure 13 demonstrates the error function plotted for the cube model. From this graph it can be understood that loss values are close to zero with growing iterations.

V. CONCLUSION

It is quite obvious that the real time examples would be more intricate than the three instances demonstrated, and might require training on the large data sets generated by the FEM software for various problems. Different parameters, combinations of the boundary and load conditions must be simulated to cover the complete domain. With this approach not just the displacements, but other variables like full field

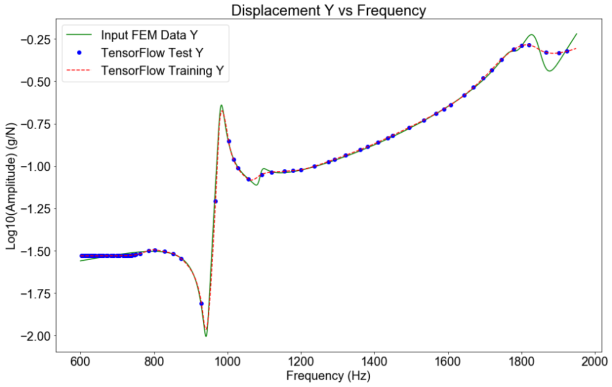

Fig. 11. Model prediction for the cube model (Y direction)

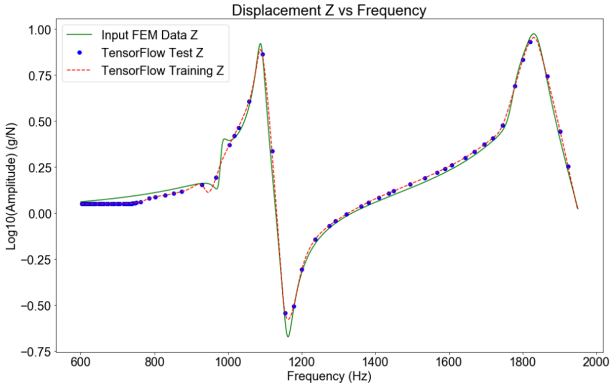

Fig. 12. Model prediction for the cube model (Z direction)

stresses or electromagnetic fluxes on all the nodes of the computational mesh can also be computed. As a large dimensional data is necessary, the neural network design is a big challenge, that can be assisted by including domain experts in the deep learning research. However, without any question, this approach has a potential to solve more realistic problems in time and frequency domains, and, when this kind of the machine learning model is built, it will have an enormous number of applications across all domains.

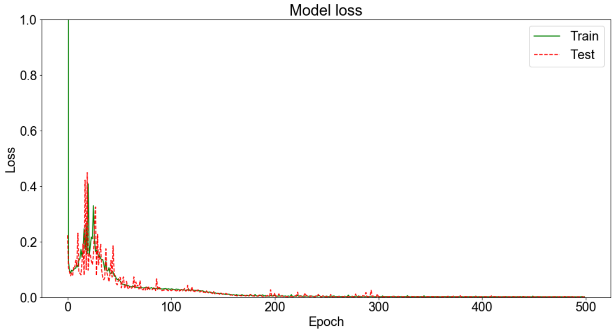

Fig. 13. Loss function of the cube model

References

[1] Shang, Y., Harmonic Oscillators: Types, Functions and Applications, 2019, Nova Science Publishers.

[2] Liang L, Liu M, Martin C, Sun W., A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis, 2018, http://dx.doi.org/10.1098/rsif.2017.0844

[3] Books Llc, Finite Element Software: Ls-Dyna, List of Finite Element Software Packages, Nastran, Nei Nastran, Genoa Software, Z88 Fem Software, General Books LLC, 2013.

[4] Bastian Bohna, Jochen Garcke, Analysis of Car Crash Simulation Data with Nonlinear Machine Learning Methods, International Conference on Computational Science, ICCS 2013, Volume 8.

[5] Yijun Fan, Romain Baudson (2013), Predicting the Acoustic Signature of Electric Vehicle Powertrains, MSC Software.

[6] Kato T, Mizutani R, Matsumoto H and Yamamoto K (2013) Advanced Technologies of traction motor for Automobile. In: Proceedings of the IEEE ECCE Asia Downunder, Melbourne, Australia, 36 June 2013, pp. 147152.

[7] de Santiago, J, Bernhoff, H, Ekergrd, B, Eriksson, S, Ferhatovic, S, Waters, R, Leijon, M (2012) Electrical Motor Drivelines in Commercial All-Electric Vehicles: A Review. IEEE Trans. Veh. Technol 61: 475484.

[8] Ari J Tuononen, Antti Lajunen (2016) Modal analysis of different drivetrain configurations in electric vehicles. Journal of Vibration and Control. Volume: 24 issue: 1, page(s): 126-136

[9] E. Saeed, E. Peter Rigid-elastic modeling of meshing gear wheels in multibody systems, Springer Science 2005.

[10] J.A. Morgan, M.R. Dhulipudi, R. Y. Yakoub, A. D. Lewis, Gear Mesh Excitation Models for Assessing Gear Rattle and Gear Whine of Torque Transmission Systems with Planetary Gear Sets, SAE Noise and Vibration Conference and Exhibition, St. Charles, Illinois, 2007-01-2245.