Integration of Simulation Techniques for the Development and Validation of Artificial Neural Networks in Modeling Complex Dynamic Systems

Kathryn Cloutier
Old Dominion University
Norfolk, VA
kclou002@odu.edu

Dr. Mileta Tomovic
Old Dominion University
Norfolk, VA
mtomovic@odu.edu

Dr. Drew Landman
Old Dominion University
Norfolk, VA
dlandman@odu.edu

ABSTRACT

This paper explores the synergistic application of simulation methodologies to create and validate artificial neural networks (ANNs) for modeling complex dynamic systems. The integration of simulation techniques with ANNs offers a powerful approach to capturing the intricate behaviors of dynamic systems, enhancing the accuracy and reliability of predictive models. The paper discusses the challenges associated with modeling complex dynamic systems and the limitations of traditional analytical approaches. It highlights the increasing importance of artificial neural networks as versatile tools for capturing non-linear and dynamic relationships within such systems. The proposed methodology involves a two-fold process. Firstly, a detailed simulation of the dynamic system is conducted to generate comprehensive datasets that reflect the system's behavior under various conditions. These datasets serve as the training and validation inputs for the artificial neural network. Secondly, an ANN is developed and trained using the simulated data, allowing it to learn the complex relationships within the dynamic system. The paper emphasizes the advantages of this integrated approach, including the ability to handle non-linearity, adaptability to dynamic changes, and improved generalization capabilities. Case studies are presented to demonstrate the application of this methodology in diverse fields, such as finance, manufacturing, and modeling weather patterns. Furthermore, the paper discusses the importance of validation techniques to ensure the accuracy and reliability of the developed ANN. The findings suggest that the integration of simulation techniques in the development and validation of ANNs contributes to a more robust and accurate representation of complex dynamic systems. The paper concludes by highlighting the potential applications and future directions of this integrated approach in areas such as predictive maintenance, process optimization, and decision support systems for dynamic and complex environments.

ABOUT THE AUTHORS

Kathryn Cloutier is a Mechanical Engineering Master's student at Old Dominion University. She is also a Chemical Product Engineer III with 6 years of experience. She graduated with a B.S. in Chemical Engineering from Rochester Institute of Technology. While a student at Rochester Institute of Technology, she completed two internships with Toyota Motor Manufacturing and All Cell Technologies. Kathryn's areas of interest are statistics, lean six sigma principles, mechanical engineering, and computer modeling.

Dr. Mileta Tomovic received BS in Mechanical Engineering from University of Belgrade, MS in Mechanical Engineering from MIT, and PhD in Mechanical Engineering from University of Michigan. Dr. Tomovic is currently serving as Mitsubishi-Kasei Professor of Manufacturing Technology, Batten College of Engineering and Technology, Old Dominion University, Norfolk, VA. Dr. Tomovic has seventeen years of teaching and research experience at Purdue University.

Dr. Drew Landman is a professor of Mechanical and Aerospace Engineering. He is a proponent of the use of statistical engineering in aerospace ground testing based on experience as chief engineer at the Langley Full Scale Wind Tunnel where he conducted aerodynamic testing on ground and flight vehicles, designed, built, and calibrated force measurement instrumentation. Dr. Landman has taught short courses in statistical engineering for over 10 years to the DOD, AIAA, NASA and industry.

Integration of Simulation Techniques for the Development and Validation of Artificial Neural Networks in Modeling Complex Dynamic Systems

Kathryn Cloutier	Dr. Mileta Tomovic	Dr. Drew Landman
Old Dominion University	Old Dominion University	Old Dominion University
Norfolk, VA	Norfolk, VA	Norfolk, VA
kclou002@odu.edu	mtomovic@odu.edu	dlandman@odu.edu

ABSTRACT

This paper explores the synergistic application of simulation methodologies to create and validate artificial neural networks (ANNs) for modeling complex dynamic systems. The integration of simulation techniques with ANNs offers a powerful approach to capturing the intricate behaviors of dynamic systems, enhancing the accuracy and reliability of predictive models. The paper discusses the challenges associated with modeling complex dynamic systems and the limitations of traditional analytical approaches. It highlights the increasing importance of artificial neural networks as versatile tools for capturing non-linear and dynamic relationships within such systems. The proposed methodology involves a two-fold process. Firstly, a detailed simulation of the dynamic system is conducted to generate comprehensive datasets that reflect the system's behavior under various conditions. These datasets serve as the training and validation inputs for the artificial neural network. Secondly, an ANN is developed and trained using the simulated data, allowing it to learn the complex relationships within the dynamic system. The paper emphasizes the advantages of this integrated approach, including the ability to handle non-linearity, adaptability to dynamic changes, and improved generalization capabilities. Case studies are presented to demonstrate the application of this methodology in diverse fields, such as finance, manufacturing, and modeling weather patterns. Furthermore, the paper discusses the importance of validation techniques to ensure the accuracy and reliability of the developed ANN. The findings suggest that the integration of simulation techniques in the development and validation of ANNs contributes to a more robust and accurate representation of complex dynamic systems. The paper concludes by highlighting the potential applications and future directions of this integrated approach in areas such as predictive maintenance, process optimization, and decision support systems for dynamic and complex environments.

INTRODUCTION

With the advancement of technology, artificial intelligence is at the forefront of emerging technology. Artificial Intelligence is typically linked to services and goods provided by companies such as OPEN AI and Tesla. However, many overlook the benefits this type of technology can provide engineers. The key component of artificial intelligence is the series of multilayered algorithms that use the concept of deep learning. The algorithms are organized with weighting factors into nodes. The path that the input data takes from weighted connections to neurodes replicates how data is processed by the brain (1). Neural networks can be modified for different uses by manipulating the number of hidden layers and nodes used to process data (1). Artificial intelligence and neural networks are used for the following engineering applications: visual inspection, trending data, and building enhanced digital twins. When used properly, neural networks can help engineers solve problems and debug equipment issues faster than current engineering techniques (6). In this paper, we will be examining the ability of artificial neural networks to model complex engineering systems and debug problems. Specifically, neural networks were used to model first and second-order engineering problems. The neural networks are compared to a Simulink model to look at the neural network's ability to predict the maximum peak and the settling time of a single mass and a two-mass system. Finally, the neural network is presented with a problem that contains error to give the neural network a more complex model to solve. Overall neural networks are excellent tools for modeling complex engineering systems; however, the engineer needs to be mindful of the noise present in the engineering system. Neural networks are very susceptible to error propagation, especially if the signal used by the neural network is noisy. The engineer must review the results of the neural network carefully and possibly perform an FFT to ensure that the neural network is analyzing the signal of interest.

Basic Spring Mass System

Neural networks have the capability to improve an engineer's ability to solve problems when there is not enough information to create the necessary system of equations to do a thorough analysis. Neural networks can bridge this gap by analyzing the system's output data and predicting future outputs of the system. For the first example, a simple mass-spring-damper system was modeled in Simulink. The Simulink model contained an ideal spring and an ideal damper. The second-order differential equation used as the basis for the model can be seen below in Equation 1. F corresponds to the external force applied to the system, $B*x'(t)$ corresponds to the damping force applied to the mass, $k*x(t)$ corresponds to the force applied by the spring, and $M*x''(t)$ corresponds to the force related to Newton's second law (6).

$$\frac{\partial^2 x}{\partial t^2} * M + \frac{\partial x}{\partial t} * B + k * x = F \quad (1)$$

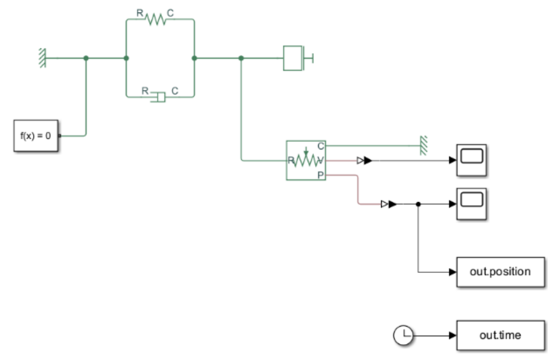

Figure 1: Simulink Model of Ideal Mass-Spring-Damper System.

The ideal Simulink model can be seen in Figure 1. A singular mass is connected to a spring and a damper. The signal is then run through a translational motion sensor allowing the signal to be exported and graphed. Figure 2 shows the position of the single mass versus time. Based on the results of the graph, a user can determine that the spring-mass system was overdamped and that the mass essentially stopped moving after 10 seconds. These results were based on the following initial conditions: $M = 10\,kg,\ B = 200\frac{N*s}{m},\ k = 200\frac{N}{m}$, $v_0 = 5$ m/s, $x_0 = 0$ m, $F_0 = 0$. If an engineer was unable to determine the best-set points or initial conditions of the system, a neural network could determine the unknown parameters. The engineer could use Matlab to create a code that would run this simulation through different spring constants, damping coefficients, and mass amounts. The neural network could construct a section of predicted responses. This would allow the engineer to look at the maximum displacement of the mass and the settling time and select the best set-points to optimize their design.

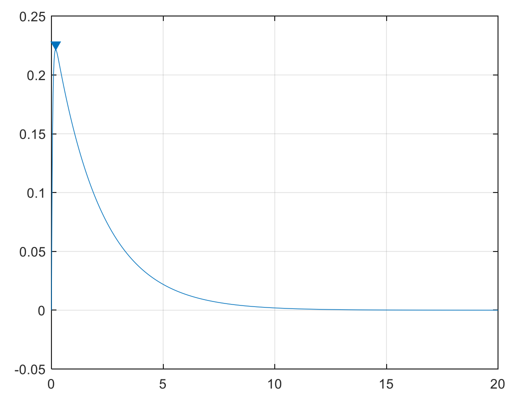

Figure 2: Position of the single mass versus time.

For this example, the Matlab code simulated ranges for mass, spring constant, and damping coefficients. The max deflection and the settling time for each set point were recorded. All the data generated was saved to an Excel file. The data generated from the Simulink model was used to train the neural network. Then the neural network generates a wider set of predicted max deflection and settling time responses. A confirmation data set was run through the simulation and used to check the accuracy of the neural network's prediction. Figure 3 and 4 show the maximum displacement of the single spring-mass system and the accuracy of the neural network. Figure 3 shows the max displacements of the single mass ranging from (0.2 to 0.9) m. The blue line in Figure 3 shows the response of the Simulink model; the neural network predicted responses are shown with the red circles. Based on the results of Figure 3, the neural network performed well in predicting max deflection responses based on the training data. The error histogram, shown in Figure 4, highlights the neural network's performance in analyzing the data. A good model fit is achieved when all the model error is consolidated next to the zero-error line. The error histogram shows that the neural network performed well, fitting the training data.

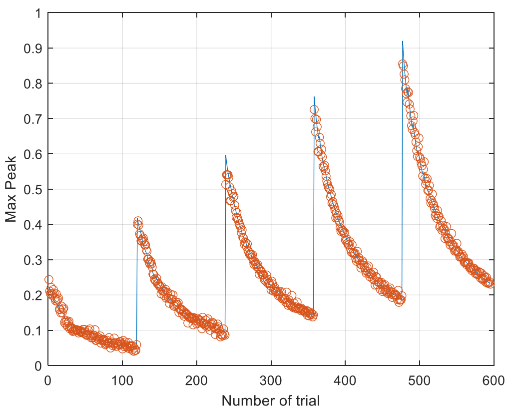

Figure 3: Maximum displacement of single mass system predicted by the neural network.

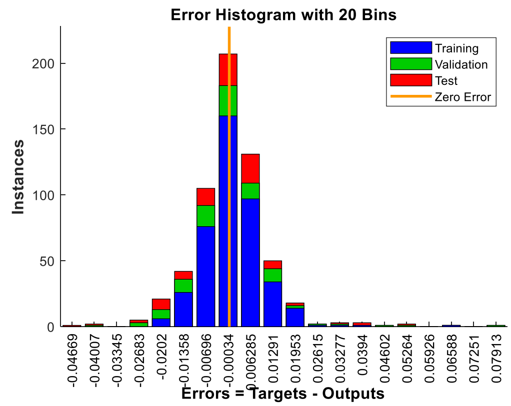

Figure 4: Error histogram of the neural network

One of the reasons the neural network performed well fitting the maximum displacement is because the position of the single mass is a first-order response. The settling time of the single mass is more challenging for the neural network to model because it is a second-order response. The results can be seen below in Figure 5 and Figure 6. The settling time of the single mass can be identified by the blue line in Figure 5. Whereas the neural network's predicted response is represented by the red circles in Figure 5. Figure 6 shows that the neural network generated a good model fit, because the error bars were centered around the zero-error bar.

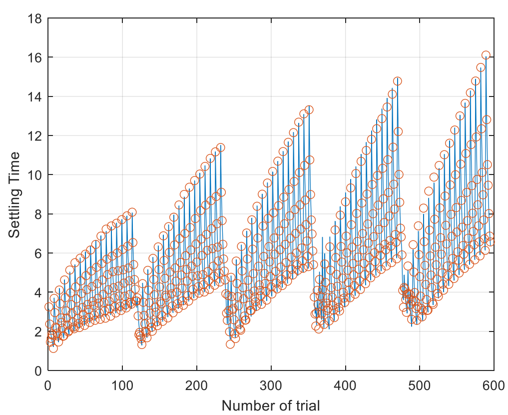

Figure 5: Settling time of the single mass.

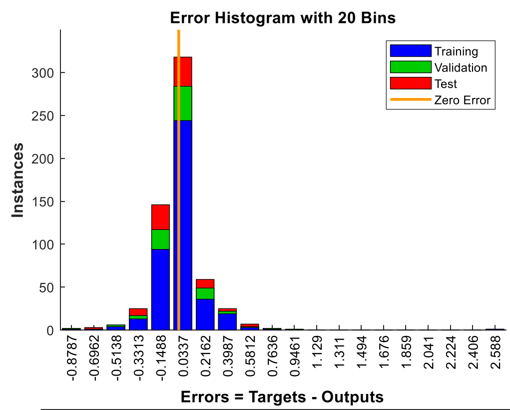

Figure 6: Error histogram for neural network.

After generating the results shown in Figure 5 and Figure 6, the neural network ran through a set of testing data, and the Simulink test responses were compared to the data generated by the neural network. In total, there were 144 confirmation data points. A selection of these confirmation points can be seen below in Table 1.

Set Points	Max Peak Predicted	Max Peak Actual	Max Peak % Error	Settling Time Predicted	Settling Time Actual	Settling Time % Error
M = 15 kg B = 250 N*s/m K = 250 N/m	0.2632	0.2742	4.213	3.315	3.400	2.575
M = 25 kg B = 250 N*s/m K = 450 N/m	0.3872	0.3830	1.074	1.935	1.753	9.395
M = 35 kg B = 850 N*s/m K = 450 N/m	0.1928	0.2028	5.184	5.675	4.481	21.04
M = 45 kg B = 750 N*s/m K = 350 N/m	0.2479	0.2652	6.968	7.205	7.239	0.4758

Table 1: Predicted maximum peak and settling time data from the neural network versus the value achieved by the Simulink model.

Based on the results in Table 1, the neural network was more accurate in predicting the max deflection achieved by the mass. All the predicted values had less than 10% error, which is acceptable for a first attempt without optimization. The predicted settling time was a less successful model. The neural network percent error varied from 0.47% to 21%. The biggest difference between the two responses is that the settling time is a second-order response, making it more difficult to predict. The predictive capability of the neural network can be improved by increasing the range of training data or changing the backpropagation algorithm.

The results generated by the neural network in Figure 5 and Figure 6 highlight the capability of this program to enhance engineering models and problem-solving. An engineer can use the available tools in Matlab and Simulink to go through potential design flaws with a digital twin or feed the data into the neural network to gain an enhanced view of the system. Although the single mass-spring system showcased the neural network's capability, it is a relatively simple model for the neural network to work with. To gain a better indication of the neural network's capability to model complex systems more complex models were generated.

2 Mass 2 Spring 2 Damper System

To gain a fuller understanding of the neural network's capability a two-mass system was modeled in Simulink. The position and settling time of the second mass were exported to Matlab and a neural network was used to predict the position of the second mass and the settling time. Figure 7 shows the Simulink model of the two mass, two spring, two damper system. The goal of this model is to test the neural network's capability of modeling increasingly complex systems and test the accuracy of its predictive capability. Unlike the first model, this model is less damped which can be seen by the oscillation of the second mass' position versus time. The system of equations used to govern the t system can be seen below. The forces acting on the system are separated between the two masses. For this example, we are only focusing on the forces effecting the second mass. F is the forcing function which is applied externally to the system. The force applied by the second spring based on the displacement of both masses is represented by $k_2 * (x_2 - x_1)$. The damping force applied by the second damper is related to the velocity of each of the masses shown by term $b_2 * \left(\frac{\partial x_2}{\partial t} - \frac{\partial x_1}{\partial t}\right)$. The term $M_2 * \frac{\partial^2 x_2}{\partial t^2}$ is related to the acceleration of the second mass.

$$M_1:\ M_1 * \frac{\partial^2 x_1}{\partial t^2} = k_2 * (x_2 - x_1) + b_2 * \left(\frac{\partial x_2}{\partial t} - \frac{\partial x_1}{\partial t}\right) - k_1 * x_1 - b_1 * \frac{\partial x_1}{\partial t} \quad (2)$$

$$M_2:\ M_2 * \frac{\partial^2 x_2}{\partial t^2} = F - k_2 * (x_2 - x_1) - b_2 * \left(\frac{\partial x_2}{\partial t} - \frac{\partial x_1}{\partial t}\right) \quad (3)$$

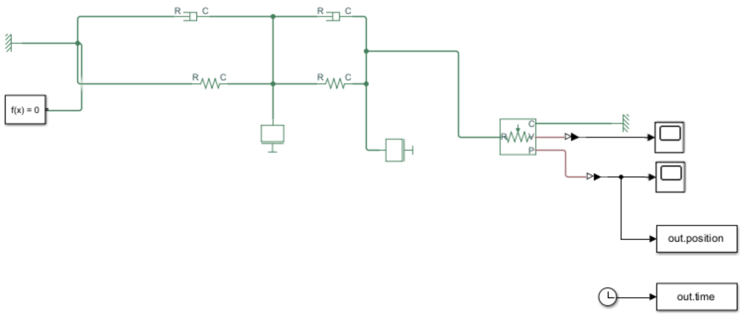

Figure 7: Simulink model of the two mass, two spring, two damper system.

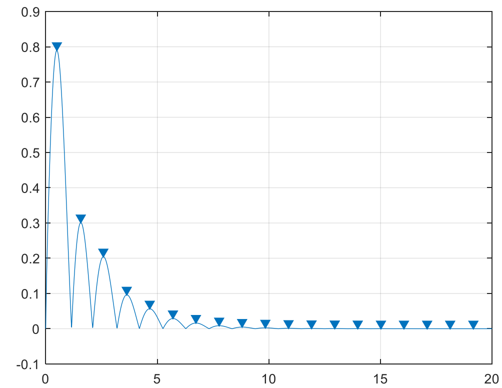

Figure 8: Position of the second mass versus time.

Figure 8 shows the position of the second mass versus time. The additional forces acting upon the second mass cause the oscillation seen in Figure 8. The neural network was used to determine the max peak or position of the second mass versus time. The results were generated after the neural network ran through a range of values for the mass, the spring constant, and the damping coefficient. The results generated by the Simulink model are shown as the blue line in Figure 9. The neural network's responses are the red dots in the same figure. Like with the first theoretical model, the neural network performed well determining the maximum displacement of the second mass. The red circles are laid over the blue Simulink model seamlessly. However, the error histogram, shown in Figure 10, shows significantly more model error than the single mass model. The model error is centered around the zero-error line, which indicates a good model fit. When you compare the range of model error, it is much larger than the single mass model. This would indicate that the neural network has overfitted the maximum peak results, and some edits to the neural network algorithm are necessary.

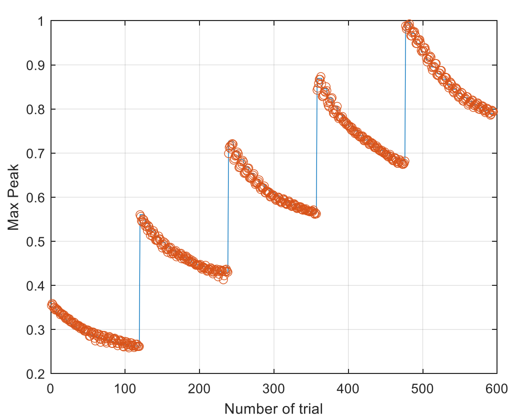

Figure 9: Maximum peak/deflection of the second mass versus time.

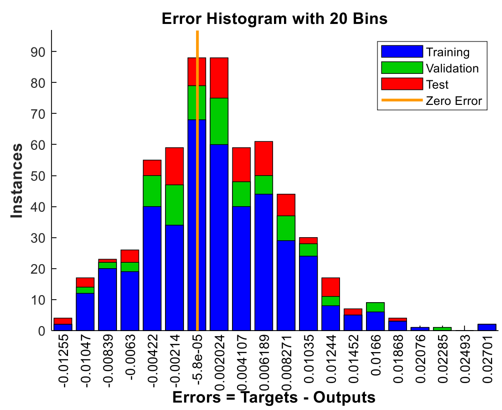

Figure 10: Error histogram for max deflection of the second mass.

The neural network also predicted the settling time of the second mass. For the same ranges of inputs used for the max peak/deflection, the neural network predicted the time it would take for the second mass to stop moving. The results from the neural network and Simulink analysis, can be seen in Figure 11 and Figure 12 below.

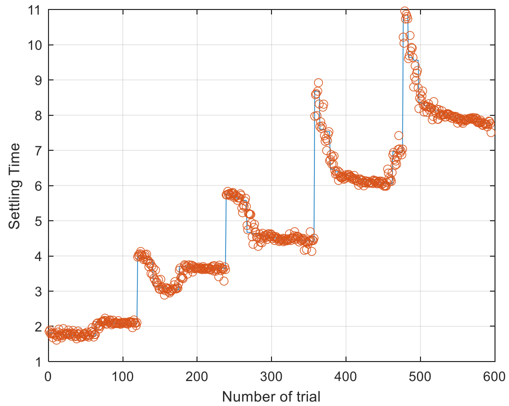

Figure 11: Predicted versus actual settling time of the second mass.

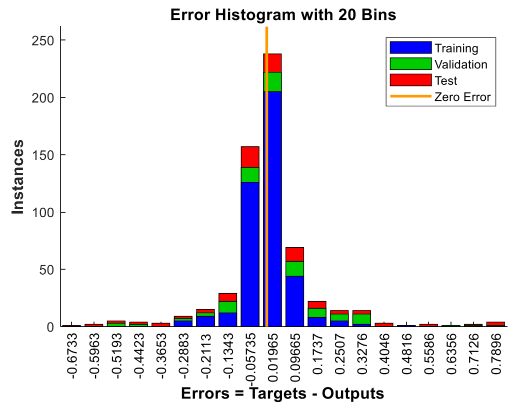

Figure 12: Error histogram of settling time prediction for second mass.

As with the max deflection, the neural network performed well, fitting the settling time of the second mass. Like the previous graphs, the blue line is the result of the Simulink model, and the red circles are the predicted values from the neural network. The neural network-generated result lies perfectly on top of the Simulink-generated response. The error histogram also indicates a good model fit by the neural network, as seen in Figure 12. All the model error is consolidated around the zero-error line, and the model error range is significantly smaller than the max peak response. Based on the results generated in Figures 9 – 12, the neural network shows its adept ability to predict second-order mechanical responses without knowing any of the information necessary to solve the system of equations. However, there are a couple of points that need to be considered before using neural networks for detailed engineering analyses.

2 Mass, 2 Spring, 2 Damper System with Gaussian Noise

While neural networks have proved to be competent at analyzing theoretical engineering systems. All of the engineering systems modeled in Simulink were ideal representations of engineering systems. For neural networks to be useful to engineers when problem-solving, they need to be able to analyze non-ideal systems with unknown amounts of error. To understand the neural network's capability, Gaussian error was added to the two-mass, two spring, two damper system. The Simulink model is shown in Figure 13 below, the AWGN channel simulates feedback error in the sensors that recorded the position and time of the two masses. Figure 14 shows the position of the second mass versus time; the impact of the Gaussian noise can be seen with the markers staggered throughout the position curve. However, the neural network has no issues analyzing the test data and predicting the maximum peak.

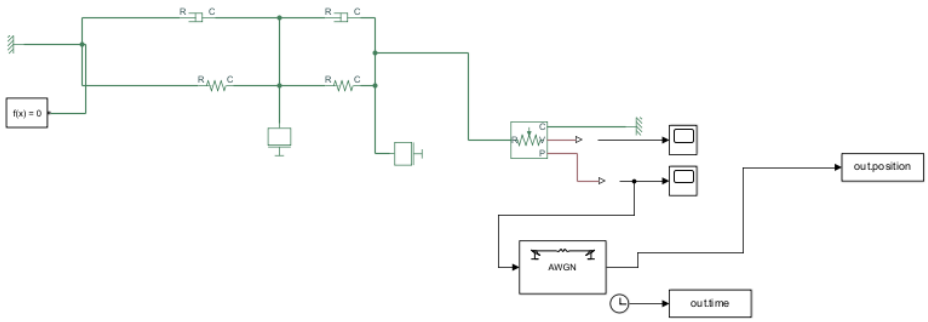

Figure 13: Simulink model of two mass, two spring, two damper system with Gaussian error.

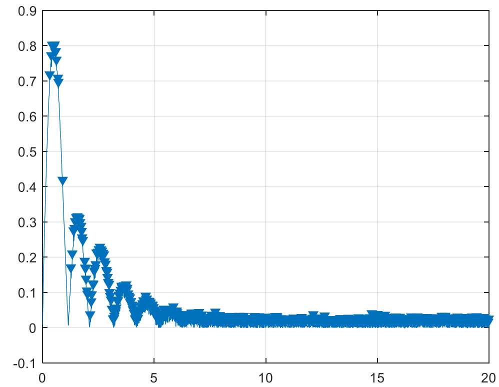

Figure 14: Position of the second mass with Gaussian error versus time.

Figure 15 and Figure 16 show the neural network responses for the predicted peaks of the second mass. Although Figure 14 displays the visible impact from the Gaussian noise, there was no impact on the neural network's determination for maximum peak. Like the previous two mass system models, the neural network fit the data well. However, the range of model error indicates that the neural network may have overfit the model, and a new back-propagation algorithm may be necessary.

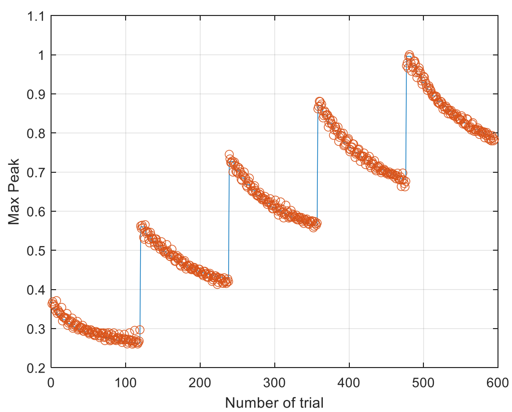

Figure 15: Maximum peak of the second mass with gaussian error.

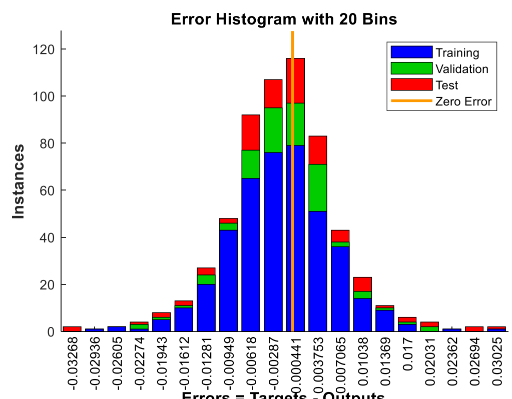

Figure 16: Error histogram for the maximum peak with gaussian error.

Figure 17 and Figure 18 show the settling time and error histogram for the two mass, two spring, two damper system with Gaussian error. Unlike the maximum peak, the settling time is greatly affected by Gaussian error. The main reason is that this is a second-order model and harder for the neural network to predict. This can be seen visually in Figure 17, the responses created by the neural network are spread all over the Simulink response. However, they do not follow the pattern created by the Simulink model. The overall error histogram, shown in Figure 18 displayed a relatively good model fit. Based on the results shown above, neural networks are susceptible to random noise; if an engineer is modeling an unknown system with a neural network, an analysis of collected data should be performed, to minimize the effect of random noise. Otherwise, the accuracy of the results produced by the neural network is limited. For this analysis an FFT was performed to filter out the sensor noise.

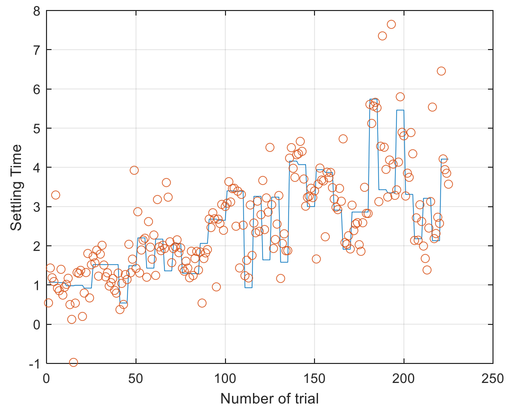

Figure 17: Settling time of second mass with Gaussian error

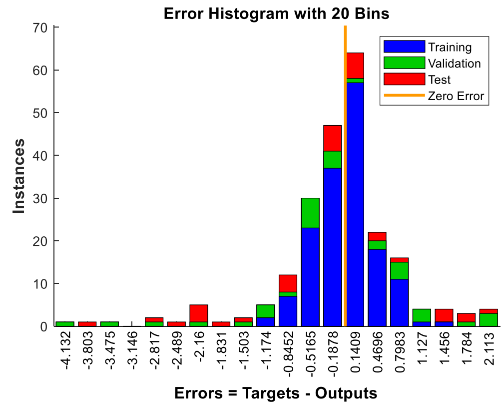

Figure 18: Error histogram for the maximum peak with gaussian error.

As shown in Figure 17, the neural network fit the settling time data with the Gaussian error. However, the neural network results were general and not precise enough for an engineering analysis. To increase the accuracy of a neural network a FFT was performed to remove additional noise from the neural network inputs. Figure 19 shows the position of the second mass versus time, corrupted by sensor noise. Figure 20 shows the FFT performed in Matlab. There is only one signal that is necessary for the neural network.

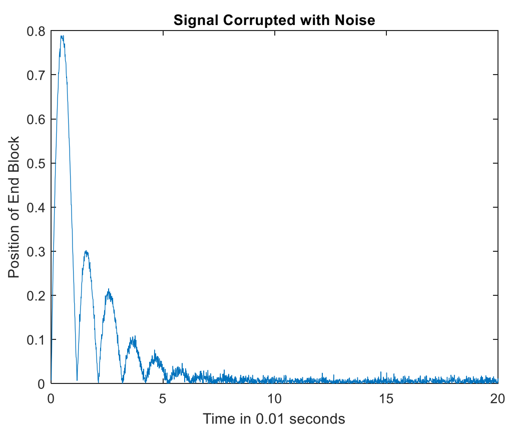

Figure 19: Signal of second mass position corrupted with noise.

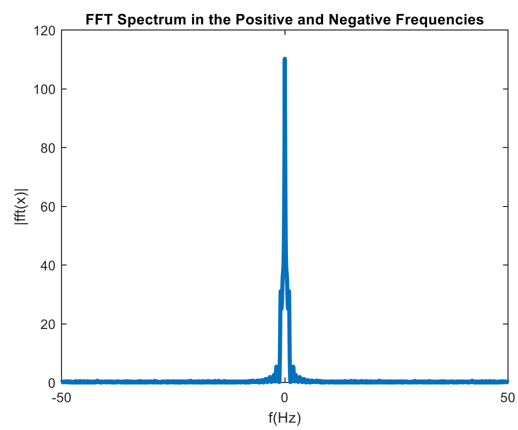

Figure 20: FFT for 2 mass, 2 spring, 2 damper analysis.

After performing the FFT analysis, the neural network was run with the filtered signal. The two mass two spring system with filtered data produce a settling time response shown in Figure 21, and an error histogram in Figure 22. With the filtered data, the neural network was capable of predicting the settling time. The red circles followed all the Simulink data seamlessly. The error histogram also showed the improved performance of the neural network with filtered data. Nearly all the model data is consolidated around the zero-error bar, indicating an excellent model fit. Neural networks are capable and can create precise digital twins. The user needs to be mindful of the type of data being used to create the engineering model. If the data sources contain too much noise, then the generated model will be compromised. It is recommended to use data filters to ensure that model error does not confound the model.

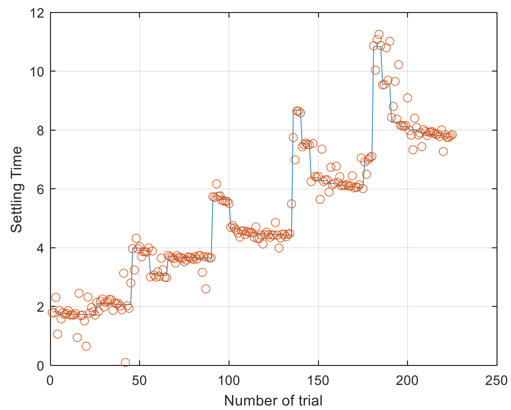

Figure 21: Settling time of the second mass with filtered data.

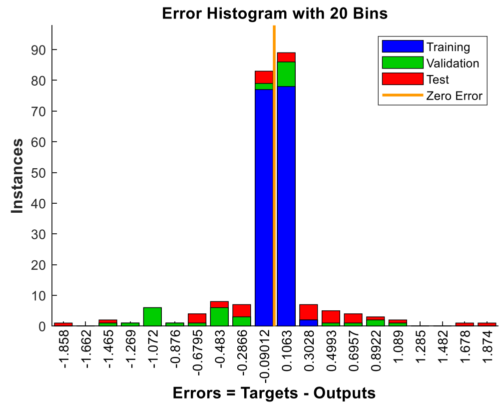

Figure 22: Error histogram for Settling time of the second mass.

CONCLUSION

In conclusion, the integration of simulation techniques for the development and validation of artificial neural networks (ANNs) represents a very useful advancement in the field of artificial intelligence. Through the application of modeling of dynamic systems, it is possible to enhance the robustness, efficiency, and reliability of ANNs across diverse domains. This paper explored the accuracy of ANN based on data sets generated by numerical simulation of simple dynamic system while examining output, settling time, which has non-linear and discontinuous behavior for different sets of system parameters. The results indicate that simulation is very powerful tool in development and verification of ANN.

Looking ahead, the integration of simulation techniques holds significant promise for advancing the capabilities and applicability of artificial neural networks in critical areas such as autonomous systems, healthcare informatics, financial modeling, and environmental science. However, continued research is needed to refine existing simulation methodologies, develop standardized practices for integration with ANN frameworks, and explore novel approaches for leveraging simulation data effectively.

In summary, the integration of simulation techniques represents a transformative paradigm in the development and validation of artificial neural networks, empowering researchers to push the boundaries of AI innovation and unlock new opportunities for addressing complex real-world challenges.

REFERENCES

(1) Walczak, S. and Cerpa, N. (2003) ‘Artificial Neural Networks’, Encyclopedia of Physical Science and Technology, pp. 631–645. doi:10.1016/b0-12-227410-5/00837-1.

(2) What is a digital twin? (no date) IBM. Available at: https://www.ibm.com/topics/what-is-a-digital-twin (Accessed: 02 February 2024).

(3) Tao F., Zhang H., Liu A., Nee Y. C., “Digital twin in industry: State-of-the-art.” IEEE trans. On Industrial Informatics, 15(4), April 2019

(4) Chaturvedi D. K. “Modeling and Simulation of Systems Using MATLAB and Simulink.” CRC Press, ISBN 978-1-4398-0672-2, 2010.

(5) YEH, Y., KUO, Y. and HSU, D. (1992) ‘Building an expert system for debugging FEM input data with Artificial Neural Networks’, Expert Systems with Applications, 5(1–2), pp. 59–70. doi:10.1016/0957-4174(92)90095-a.

(6) 5.7 forced Mechanical Vibrations. Available at: https://www.math.utah.edu/~gustafso/2250forcedOscillations.pdf (Accessed: 02 February 2024).