Prediction of Vessel RAOs:

Applications of Deep Learning to Assist in Design

James Frizzell

Texas A&M University

Dr. Mirjam Furth

Texas A&M University

***Proceedings of the 27th Offshore Symposium, February 22nd, 2021, Houston, Texas***

***Texas Section of the Society of Naval Architects and Marine Engineers***

***Copyright 2021, The Society of Naval Architects and Marine Engineers***

**ABSTRACT**

In an age of high-paced design, a need arises for engineers to quickly estimate the feasibility of their ideas without spending weeks developing a computer model.  At the same time, the use of machine learning models, or neural networks, in the maritime industry has grown substantially over the past years.  By further extending the use of these predictive models in the design phase, marine engineers and naval architects can expedite their work.

This paper focuses on the creation of a neural network that can estimate the Response Amplitude Operators (RAOs) of a vessel given its characteristic properties such as length, beam, and draft.  A dataset was collected through a parametric design analysis of box barges using ANSYS AQWA, and the resulting RAO data saved for all 6 degrees of freedom. A critically damped spring equation of the form was generated for each degree of freedom in each dataset, and the A, B, and C parameters tabulated. Lastly, a Keras Neural Network Model was trained on the three parameters and the wave heading angle, with the hidden layers and neuron count being adjusted to optimize the loss and maximize the R-squared.

To validate the results, a series of box barges with dimensions that were not a part of the training dataset were simulated in ANSYS, while the virtual model with identical characteristics was simulated with the Neural Network.  The resulting RAOs were compared to validify the accuracy of the Neural Network.

With this predictive model, engineers can quickly determine a hullform’s RAOs, and compare the response with the common sea states along the intended route. Additionally, the model can assist in design iteration. As the hull shape gradually changes, the new RAOs can be estimated to ensure that the design is progressing in an appropriate direction.

**Keywords: SNAME, Offshore Symposium, Machine Learning, Deep Neural Network, RAO**

# Introduction

Finding the Response Amplitude Operators (RAOs) of a vessel is a numerically expensive process. Industry standard programs such as ANSYS AQWA use complex procedures to solve Equation 1 and return a precise solution.

(1)

This methodology has benefits and drawbacks, however. While it is a highly scalable process and can be applied to an object of any shape, the computational time is long. Additionally, the quality of results is directly coupled to the quality of the input geometric model. In an iterative design environment where an engineering is constantly adjusting a hullform to suit a client’s needs, the corresponding digital model must be updated as well in order to maintain the integrity of the numerical results. This method, although highly accurate when performed correctly, is time-consuming and by extension, costly.

To eliminate the need for complex mathematical equations, neural networks (NN) can be implemented to derive the correlation between inputs and outputs. In this case, a deep neural network (DNN) has been created to predict the RAOs of a vessel with respect to frequency given only the characteristic dimensions of length, beam, draft, and the wave heading angle. With this approach, the calculation of approximate RAOs only requires a few seconds of computational time.

Simply inserting a neural network into a complex problem is never an ex-machina solution, though. To obtain accurate results, a comprehensive dataset must be collected during development to train the NN on. Hundreds or thousands of vessels must have their RAOs solved in the conventional method to provide learning material for the predictive model. Additionally, the model is only able to accurately predict vessel with similar body shapes as those given in the training dataset. To create a robust, accurate model would take a long time to develop.

The purpose of this paper is to detail the process of creating a proof-of-concept neural network that can predict the RAOs of a simple box barge. Future work will be needed to expand the capability to complex hull structures.

# Literature Review

To gain a better understanding of the potential ways to approach this project, a previous study implementing deep learning techniques was referenced.

# Methodology

## **Data Collection**

The first step in developing any neural network is to collect a dataset. The dataset must include:

1. The model inputs
2. The model outputs

The inputs are easy to define and mark the bounds of viable prediction. Table 1 outlines the four parameters that were used in this study. The boundary of accuracy is the upper and lower limits of inputs, meaning that this model will work best for box barges with characteristic dimensions that fall within these ranges. The model will still work for barges outside of these values, but the predicted RAOs are subject to higher inaccuracies. In total, 214 box barges were included in the dataset.

Table 1: Inputs to Neural Network Model

|  |  |  |
| --- | --- | --- |
| Input Type | Minimum Value | Maximum Value |
| Vessel Length (meters) | 0.3 | 25 |
| Vessel Beam (meters) | 0.1 | 16 |
| Vessel Draft (meters) | 0.033 | 1.23 |
| Wave Heading Angle (degrees) | 0 | 180 |

To collect the model outputs – the actual RAO values, ANSYS AQWA was used. First, a parametric study was setup in DesignModeler, in which a box barge with a certain length, beam, and draft would be created and passed into AQWA. Once in AQWA, a point mass was inserted at the volume centroid to ensure that the weight of the barge would be appropriate to produce the given draft. This weight was determined by Equation 2, where the density of freshwater was selected.

(2)

For each barge, a molded depthwas selected to ensure that waves would not overtop the vessel. This value ranged from 0.1 meters for the smallest barge to 2 meters for the larger barges.

The point mass was also given specified moments of inertia based on the dimensions of the box barge. It was assumed that the mass was evenly distributed throughout the vessel, so finding the mass moments of inertia were able to be found with Equations 3, 4 and 5.

(3)

(4)

(5)

This completed the setup process for the ANSYS model. The calculation was executed for all datapoints. The computation time averaged 5 minutes per vessel, for a total of about 17 hours.

ANSYS AQWA stores RAO data in a text table format enumerated by frequency. This style of presentation is good for determining the maximum and minimum RAO, but it is hard to train a neural network on this data. A sample of the raw data can be seen in Figure 1.

Table

Description automatically generated

Figure 1: Sample Raw Data Output from ANSYS AQWA

## **Data Pre-Processing**

Data preprocessing took several steps. First, the frequency-dependent RAO value for a datapoint was collected. Then, a Python script fit the frequency-dependent RAO value of each degree of freedom to a curve of the form in Equation 6. The critically damped spring equation was selected due to its natural similarity to the data curves. Other curves were considered, including 3rd, 6th and 9th order polynomials. These were eventually discarded because the curves generated by the neural network were poorly fit to the actual data.

(6)

The free parameters A, B and C were then collected and stored. These became the model outputs.

This process was repeated for each degree of freedom for each wave heading of each barge size. The result of the data processing was 1070 datapoints, each with 18 values that described the shape of the RAO curve with respect to frequency. The check that the curve fitting was accurate, the R-Squared between the true data and the fit curve was evaluated, and the summary is shown in Table 2. The RX and RY degrees of freedom tended to have trouble fitting to the actual data. This is most likely since the actual Roll and Pitch responses are 0 for certain wave directions, and the curve fit failed to make the exponential equation equal to 0, which in turn pulled the R-Squared score down significantly for those two degrees of freedom.

Table 2: R-Squared Results of Curve Fit to Raw Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | X (Surge) | Y (Sway) | Z (Heave) | RX (Roll) | RY (Pitch) | RZ (Yaw) |
| Average R-Squared | 0.773267 | 0.649511 | 0.871257 | 0.457635 | 0.421508 | 0.645575 |
| Median R-Squared | 0.960781 | 0.954595 | 0.925898 | 0.462877 | 0.437471 | 0.72613 |

Now all the components needed to create a neural network had been collected. The inputs – the barge dimensions and wave headings, were compiled with the corresponding outputs – the A, B, and C values for each degree of freedom in a single spreadsheet.

## **Neural Network Architecture**

The creation and optimization of neural networks has been made simple with modern packages. Keras and Tensorflow were the two packages used in this study. This paper will not detail the inner workings of machine learning algorithms, as there are better sources to reference for more information on that subject.

Using the spreadsheet that was created earlier, the data was read into Python and split into a training and test dataset with the typical 80/20 split. Simple pre-processing methods were employed to prevent any errors – namely dropping any rows with NaN values. The number of rows dropped was small in comparison to the overall dataset size.

The neural network itself consists of an input layer, multiple hidden layers, and an output layer, each having a set number of neurons. The input and output layers are dictated by the data that will be passed through the model, while the hidden layers are more variable and can be adjusted to optimize the model.

A parametric study was done to select the optimal architecture for the neural network and can be seen in Figure 2. Ultimately, 2 hidden layers, each having 256 neurons was selected to be the best model. Table 3 lists the array shape of each layer in the model.

Chart, scatter chart

Description automatically generated

Figure 2: Results of Neural Network Architecture Parametric Study

Table 3: Selected Neural Network Architecture

|  |  |  |  |
| --- | --- | --- | --- |
| Input Layer | Hidden Layer 1 | Hidden Layer 2 | Output Layer |
| (,4) | (,256) | (,256) | (,18) |

Training time is another parameter that can be adjusted to optimize the model. A shorter training time tends to avoid the risk of over-fitting, but if the training time is too short the model may not be fully fit at the end of the training. To determine whether the model has been properly fit, a visual inspection of the Loss-Epoch graph (Figure 3) can be done. If the line is horizontal and mostly unchanging by the final epoch, the model is well-fit. To check that the model is properly predicting the A, B, and C parameters, the true values can be plotted against the predicted values, as seen in Figure 4.

|  |  |
| --- | --- |
| Chart, line chart  Description automatically generated  Figure 3: Model Loss During Training Progression | Chart, scatter chart  Description automatically generated  Figure 4: True Values Plotted Against Predicted Values, R2 = 0.717 |

# Results

With the model creation and training completed, it can be benchmarked against random datapoints. Figure 5 shows the similarity between the predicted values and true values. There are two metrics used to quantify performance. Relative percent difference (RPD) assigns a number between 0 and 100 and explains the difference between the predicted curve and actual curve. A simple error calculation cannot be used since the true value is often 0 and would result in a division by 0 error. In the cases where the true value is 0, the RPD is often very high, so the analysis has been supplemented with a raw error measurement. This value shows the actual difference between the true and predicted value. In Figure 5, the Sway degree of freedom has an RPD of 100, but a raw error of 0.039 m/m response. So, even though the percent difference is large, the response is only incorrect by about 4 centimeters for every meter of wave height – which is still very accurate for a barge 15 meters in length.

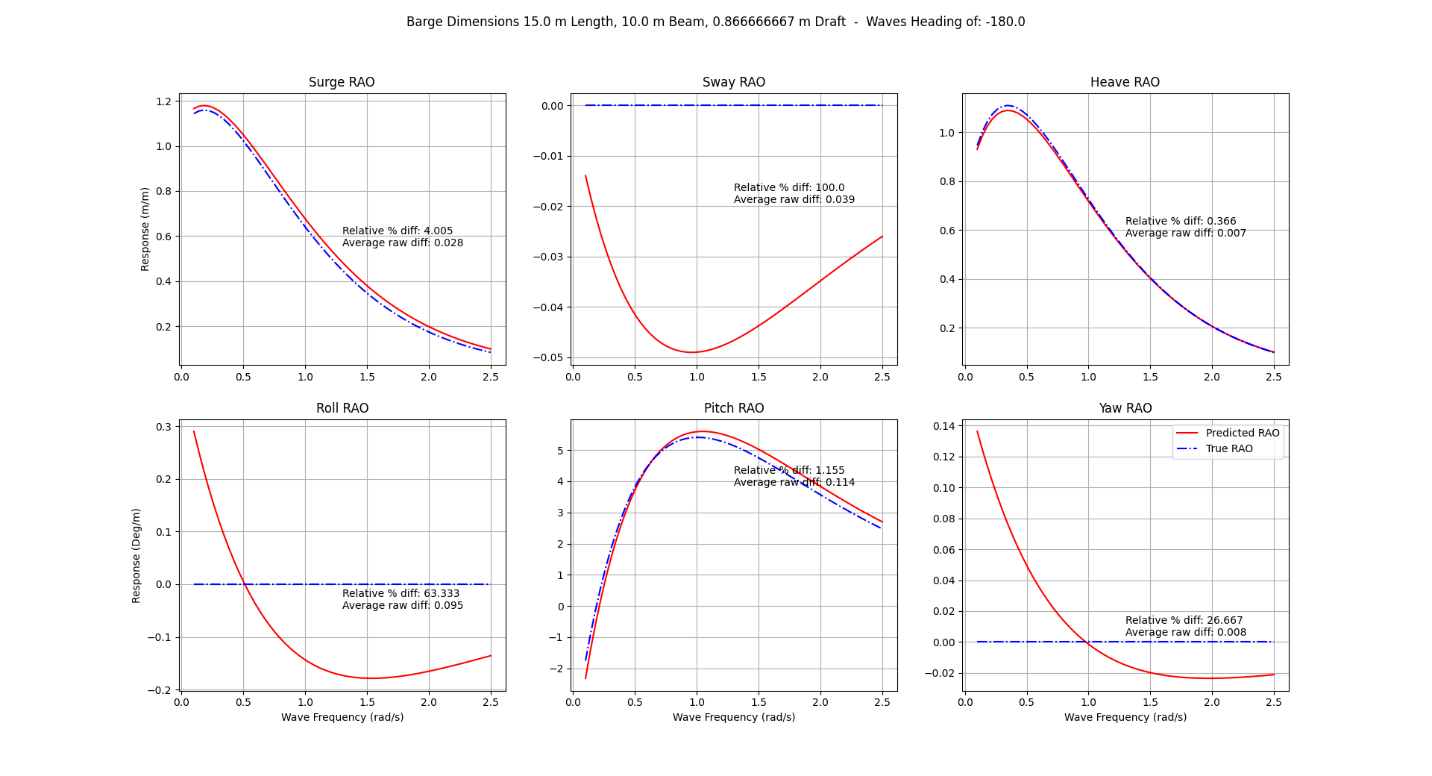


Figure 5: Predicted RAO Values

# Conclusion

With the model functioning properly for box barges, the first phase of this research project is complete. Given the accuracy and robustness of the model so far, the proof-of-concept has shown the feasibility of using a neural network to replace the traditional solution process of solving for the RAOs of a vessel.

The next step to proceed with this study is to expand the dataset to include hullforms beyond simple box barges. Currently planned is the use of Wrigley hulls, which can be parameterized in ANSYS in a similar fashion to what was done with the box barges.

# Acknowledgements

This research could not have been done without the help of Dr. Mirjam Furth who oversaw the progress and gave advice on the approaches taken, and the Texas A&M Ocean Engineering Department who provided workstation computers which were used to collect the dataset.

# References

Please properly cite the paper