Sixth International Symposium on Marine Propulsors
smp'19, Rome, Italy, May 2019

A Data-Driven Multi-Degree of Freedom Body Force Propeller Model for Maneuvering

Bradford Knight[1], **Kevin Maki**[1]

[1]Department of Naval Architecture and Marine Engineering, University of Michigan, MI, USA

ABSTRACT

Simulation of a maneuvering vessel in waves is expensive using CFD tools. Accurately and efficiently predicting the forces of the propeller is especially challenging. To model the propeller using viscous numerical tools, like RANS CFD, requires not only extra mesh discretization, but also a very small time-step size. One way to avoid this cost and reduce the overall cost of modelling a maneuvering vessel in waves is to implement the propeller as a body force in the Navier-Stokes equations and add the propeller forces to the six degree of freedom solver used to calculate the motions of the vessel. However, the six degree of freedom body force of the propeller must be known with a high level of precision. This paper examines using a neural network to determine the unsteady propeller forces. The objective of this paper is to train a single hidden layer neural network using snapshots of an unsteady propeller CFD simulations to accurately predict the forces on the propeller. Both open water and the behind condition are examined. Once trained, the neural network is tested on other unsteady propeller simulations which have different time varying dimensionless speeds and oblique flow angles.

Keywords

Neural network, Body force, Propeller-hull interaction.

1 INTRODUCTION

Accurately and efficiently determining the multi-degree of freedom forces on a propeller for a maneuvering vessel is challenging. Potential flow methods are inexpensive but can be less accurate than viscous methods when the propeller operates off design. Viscous flow methods, like RANS, DES, or LES can be more accurate than potential flow methods but require a very small time step and require extra mesh discretization for the propeller. This extra cost can be prohibitive for CFD of a maneuvering vessel, especially in waves. Maintaining the accuracy of a viscous CFD method while decreasing the cost is desirable. One way to accomplish this goal is to determine the body force using a data-driven model. The body force can be applied to the flow and to the vessel equations of motion. A data-driven model that is trained with a viscous CFD method can be as accurate as that CFD method, but when implemented as a body force in a maneuvering analysis has negligible cost. There are various methods of creating data-driven models for dynamical systems ranging from regression to neural networks. Hornik et al (1989) showed that neural networks can predict any function to any requisite level of accuracy, given adequate training and inputs. Neural networks have been applied to fluid dynamics problems ranging from improved turbulence modeling (Singh et al 2017) to determining propeller forces for different advance coefficients (Abramowski 2005).

This paper examines the use of a neural network to predict unsteady propeller forces due to prescribed motions. The thrust T, torque Q, and side force S of a propeller with unsteady forward velocity U, side velocity V, and propeller revolution rate n are examined in this paper. Both the open water and behind condition are examined. The objective of this paper is to train and test neural networks that are capable of predicting the unsteady three degree of freedom body force of a propeller undergoing unsteady motions.

The propeller examined is the NMRI scale KCS propeller. Details of the CFD setup as well as the derivation for a semi-empirical propeller force method can be found in the authors' previous work (Knight & Maki 2018, 2019). A neural network is trained and tested on unsteady RANS CFD results to determine the three degree of freedom force of the propeller in open water and the behind condition.

2 METHODOLOGY

The methodology is comprised of four parts. The first part discusses the open water CFD setup, while the second part describes the behind condition CFD setup. The third part describes the neural network used to predict the unsteady open water propeller forces. The fourth part discusses how the neural network used for the open water predictions is extended to a propeller operating in the behind condition.

2.1 Unsteady Open Water Propeller CFD Setup

The NMRI scale KCS propeller is examined. The CFD used is very similar to that used by Knight & Maki (2018, 2019), with the only difference of using an inlet condition on the sides of the domain when there is oblique flow. The mesh used in this study correlates to the coarse grid used by Knight & Maki (2018, 2019). While a coarse grid with RANS CFD may not be as accurate off-design as finer grids or LES, the purpose of this paper is to illustrate how a neural network could be trained and tested to predict unsteady propeller forces. The goal is for the neural network to be

*

as accurate as the method that it is trained with. Therefore, the CFD used to train and test the neural net is treated as the truth.

The motions of the propeller are prescribed using a customized OpenFOAM six degree of freedom solver. The velocity is initially accelerated from rest to the average velocity over a time t_r. To note, in all cases the t_r used is 0.0628 seconds. The general form of the velocity of the propeller after a time t_r can be given by Equation (1). The unsteady motions are prescribed to be harmonic about the average velocity for each respective degree of freedom. Thus, the motions depend upon the average velocity $\dot{x}_{j,avg}$, the amplitude A_j, the frequency of oscillation $\omega_{osc,j}$, and time t. In this form the general motions are given with subscript j. The specific degrees of motion of forward speed U, side velocity V and revolution rate ω can be given by Equations (2)-(4) respectively.

$$\dot{x}_j = \dot{x}_{j,avg} + A_j \omega_{osc,j} \cos\left(\omega_{osc,j}(t - t_r)\right) \tag{1}$$

$$U(t) = \dot{x}_1 \tag{2}$$

$$V(t) = \dot{x}_2 \tag{3}$$

$$\omega(t) = 2\pi n = \dot{x}_4 \tag{4}$$

2.2 Unsteady Behind Condition Propeller CFD Setup

The behind condition CFD is also an extension of the work by Knight & Maki (2018, 2019). The propeller in the behind condition is analyzed with a symmetry plane at the waterline. Therefore, a double body approximation is used. Particulars for the CFD setup can be found in (Knight & Maki 2018, 2019).

2.3 Setup of the Neural Network for Unsteady Open Water Propeller Forces

A single hidden layer neural network is used with a sigmoid activation function. The inputs to the neural network are snapshots of the advance ratio, J, and the oblique flow angle, α. Thus, the inputs are instantaneous feature vectors, where the feature vector includes polynomials of J and α. The forces on the propeller depend upon the diameter of the propeller D, the density of water ρ, n, U, and α. The diameter of the propeller examined is 0.105 m and the density of the water is 997.66 kg/m^3. Therefore, the output of the neural network is the instantaneous thrust coefficient, K_T, the instantaneous side force coefficient, K_S, and the instantaneous torque coefficient, K_Q. The equations for K_T, K_S, K_Q, J, and α are shown in Equation (5) to Equation (9).

$$K_T = \frac{T}{\rho n^2 D^4} \tag{5}$$

$$K_S = \frac{S}{\rho n^2 D^4} \tag{6}$$

$$K_Q = \frac{Q}{\rho n^2 D^5} \tag{7}$$

$$J = \frac{U}{nD} \tag{8}$$

$$\alpha = \tan^{-1}\left(\frac{V}{U}\right) \tag{9}$$

A feedforward neural network with a single hidden layer is programmed in MATLAB. The input to the neural network is a feature vector, $\vec{x}$, for each snapshot in time. For each snapshot in time the input feature vector, $\vec{x}$, is shown by Equation (10). The feature vector is comprised of permutations of the instantaneous J, up to second order, and the absolute value of α, up to sixth order. A single hidden layer with k=10 hidden neurons is used. The output layer has n=3 neurons correlating to K_T, K_S, and K_Q which may be scaled and shifted to improve the scaling of the problem.

$$\begin{aligned}\vec{x} = [1, J, J^2, |\alpha|, ..., |\alpha|^6, J|\alpha|, ..., J|\alpha|^6,\\ J^2|\alpha|, ..., J^2|\alpha|^6]^T\end{aligned} \tag{10}$$

The snapshots for the training data are taken by incrementally sampling the time history of the observables of CFD simulations of an open water propeller with unsteady motions, U, V, and n. The unsteady motions are prescribed, and thus are known. Therefore, the feature vector comprised of J and α can be determined. The observables from the CFD simulations include the thrust, side force, and torque. These are processed to give instantaneous values of K_T, K_S, and K_Q. Similarly, J, α, K_T, K_S, and K_Q can all be preprocessed for the test data from CFD simulations. The force coefficients from the CFD are treated as the truth.

Based upon Equation (10), the input matrix has m=21 rows and the number of columns equal to the number of snapshots examined. A schematic of a single hidden layer neural network is shown in Figure 1. The left hand column of neurons is the input, the middle column of neurons is the hidden layer, and the right hand column of neurons is the output layer. A sigmoid activation function is used which is defined by Equation (11). Two weight matrices and two bias vectors are used. The first weight matrix $\mathbf{W_1}$ and bias vector $\vec{b}_1$ is used between the input layer and the hidden layer. The second weight matrix $\mathbf{W_2}$ and bias vector $\vec{b}_2$ is used between the hidden layer and the output layer. The values of the weight matrices and bias vectors are determined using back propagation. The accuracy for a single snapshot i can be evaluated by the loss function L_i defined in Equation (12), where $\hat{\vec{y}_i}$ is the prediction and $\vec{y}_i$ is the truth for that snapshot. $\hat{\vec{y}_i}$ is calculated from Equation (13) which depends on the output of Equation (14).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{11}$$

$$L_i = \frac{1}{2}\|\vec{y}_i - \hat{\vec{y}_i}\|_2^2 \tag{12}$$

$$\hat{\vec{y}_i} = \sigma(\mathbf{W_2}\vec{\eta}_i + \vec{b}_2) \tag{13}$$

$$\vec{\eta}_i = \sigma(\mathbf{W_1}\vec{x}_i + \vec{b}_1) \tag{14}$$

The average loss function L across all snapshots is most indicative of the error. Therefore, the loss function of each snapshot in time is calculated. The loss functions for each snapshot are summed and divided by the number of snapshots N as shown by Equation (15). The program iterates

until the loss function is below a user defined tolerance. Stochastic gradient descent is used to improve the speed of convergence. The stochastic gradient descent algorithm works by randomly selecting one of the snapshots and calculating the gradients of the weight matrices based on that snapshot. Once the neural network is trained, it is tested with data from other unsteady propeller CFD simulations. The accuracy of the neural network is evaluated by how accurate it is compared to the force coefficients calculated from the CFD.

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i \tag{15}$$

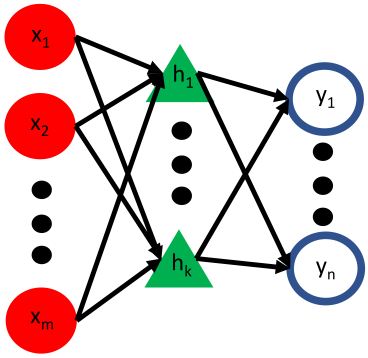

Figure 1: Schematic of neural network with single hidden layer.

2.4 Extension of Neural Network to Predict Unsteady Behind Condition Propeller Forces

The neural network for the behind condition propeller forces is similar to the open water propeller neural network. For the behind condition neural network the feature vector defined by Equation (10), the $|\alpha|$ is changed to α. Therefore, the oblique flow angle maintains its sign. In the open water propeller case the direction of motion did not have an effect except to determine the sign of the side force. However, in the behind condition the hull interacts with the propeller. The direction of α affects the propeller forces since the propeller is rotating in the wake of the hull. Equation (16) shows the feature vector used as input to the neural network for the behind condition for each snapshot in time. To note, J and α are calculated based upon the prescribed vessel velocities and n, just like the open water method.

$$\vec{x} = [1, J, J^2, \alpha, ..., \alpha^6, J\alpha, ..., J\alpha^6, J^2\alpha, ..., J^2\alpha^6]^T \tag{16}$$

The second difference between the behind conditional neural network and the open water neural network is the manner in which they are trained. As aforementioned, the open water neural network uses stochastic gradient descent. The open water neural network is also trained with the results of one unsteady open water propeller CFD simulation. However, the behind condition neural network uses gradient descent instead. Therefore the average gradient of the weights and bias vector is used to correct the current weights. The behind condition propeller neural network is trained with two different types of simulations. It uses one unsteady CFD simulation of the propeller operating in the behind condition. It also uses steady state open water propeller coefficients which are extended into the behind condition in accordance with (Knight & Maki 2018, 2019). Knight & Maki showed a semi-empirical method for how steady state open water K_T and K_Q values can be extended into K_T and K_Q coefficients for the propeller operating in the behind condition if the same propeller is run in the behind condition at a single U and n. The KCS propeller has been simulated using a Moving Reference Frame approach at different J values in open water and a rotating propeller simulation was performed for the propeller operating in the behind condition with constant U and n at one speed to determine the w parameter from (Knight & Maki 2019). Therefore, the behind condition case is trained with the results of one unsteady behind condition propeller simulation and the open water K_T and K_Q values extended into the behind condition using the semi-empirical relation.

3 TRAINING AND TESTING THE NEURAL NETWORK TO PREDICT UNSTEADY OPEN WATER PROPELLER FORCES

The neural network to predict unsteady open water propeller forces is first trained with the results of an open water unsteady propeller CFD simulation. Discrete snapshots are examined for the training data. Before testing the neural network on a second unsteady propeller motion a validation step is taken. The validation step examines the full time history of the unsteady propeller data that was used for training the algorithm. Once the neural network is trained and validated, it is tested on an unsteady propeller which undergoes a different unsteady motion. Table 1 shows the parameters that define the unsteady motion for each of the cases.

Table 1: Open water parameters for unsteady motion.

Parameter	Train	Test
$\dot{x}_{1,avg}$ (m/s)	1.68	1.68
$\dot{x}_{2,avg}$ (m/s)	0.00	0.00
$\dot{x}_{4,avg}$ (rad/s)	201	201
A_1 (m)	0.20	0.11
A_2 (m)	0.21	-0.20
A_4 (rad)	31.4	31.4
$\omega_{osc,1}$ (rad/s)	4.00	4.00
$\omega_{osc,2}$ (rad/s)	4.00	4.00
$\omega_{osc,4}$ (rad/s)	1.00	1.00

3.1 Training and Validating the Neural Network for the Unsteady Open Water Propeller Force

The neural network is trained on a case that has unsteady velocity in all three degrees of freedom. The average forward velocity is $\dot{x}_{1,avg} = 1.68$ m/s, the average side velocity is $\dot{x}_{2,avg} = 0$ m/s, and the average propeller revolution rate is $\dot{x}_{4,avg} = 201.06$ rad/s. The amplitudes for each

degree of freedom are $A_1 = 0.2$ m, $A_2 = 0.213$ m, $A_4 = 31.4$ radians. The frequencies of oscillation are $\omega_{osc,1} = 4$ rad/s, $\omega_{osc,2} = 4$ rad/s, $\omega_{osc,4} = 1$ rad/s. The motions of the propeller in surge, sway, and rotational degrees of freedom were imposed in the customized OpenFOAM multi-degree of freedom solver. Similarly, since the motion is prescribed, Equations (1)-(4) can be used to extract the velocities for each degree of freedom. The first 0.1 seconds are neglected to account for the ramp time and the initial transient effects. After this time, the CFD forces are uniformly sampled to generate 70 snapshots. J and α for each of the snapshots are shown in Figure 2. The feature vector is generated as a function of these J and α values.

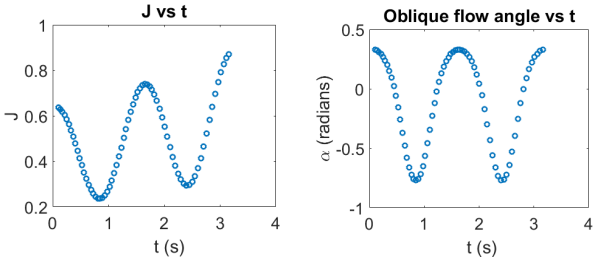

Figure 2: Discrete samples of J (left) and α (right) as a function of time for the unsteady open water propeller training case.

At each of these snapshots the output matrix is extracted from the CFD results. As discussed earlier each column of output from the neural network represents the predicted $[K_T K_S K_Q]^T$ of each snapshot. K_Q is much smaller than K_T. Additionally, since K_S is zero at $\alpha = 0$ radians, K_S can be very small. Furthermore, K_S oscillates about zero depending upon the direction of the sway velocity. Therefore, for both the training and testing of the open water unsteady propeller neural network, scaling is applied to the K_Q and K_S extracted from the forces calculated by the CFD. The value of K_S is also shifted to improve the scaling. K_S and K_Q are recast in Equations (17) and (18), where the subscript u denotes unscaled, correlating to the original K_S and K_Q defined in Equation (6) and Equation (7) respectively. The absolute value of $K_{S,u}$ is used since the input matrix contains functions of the absolute value of the oblique flow angle, thus, directionality is not accounted for. It was found that the neural network was more robust when directionality was omitted in the neural network. The sign of the side force is corrected after the neural network calculates K_S. This is demonstrated when the neural network is tested.

$$K_S = 10|K_{S,u}| + 0.3 \tag{17}$$

$$K_Q = 10K_{Q,u} \tag{18}$$

Figure 3 shows L versus the training iteration. Since stochastic gradient descent is used on some iterations the loss function may increase, but the loss function tends to decrease until reaching a floor. The training loss function is reduced to 3.1×10^{-5}.

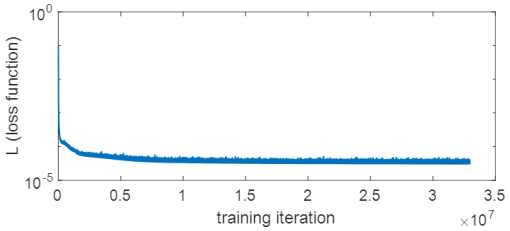

Figure 3: L as a function of training iteration for the open water neural network.

Figure 4 shows the neural network prediction for the training data. This demonstrates that the neural network is trained sufficiently since it does a good job predicting the data that it was trained with. The neural network can further be validated by examining the neural network prediction of the coefficients versus the CFD predictions for the whole time series, instead of the 70 discrete snapshots used to train the neural network. Figure 5 shows that the neural network does a good job of predicting the force coefficients from the CFD for the whole time series. One thing to note is that the side force predicted by the CFD has a high frequency oscillation due to the side force varying as a function of the rotation angle of the propeller. However, this oscillation is not modelled by the neural network since it was not trained with the rotational position of the propeller as a feature, nor were enough snapshots used to train the neural network to discern this characteristic. This oscillation is small and since the intent of the neural network is for use in vessel maneuvering this small oscillation is not necessary to capture.

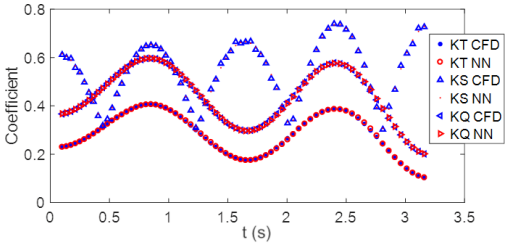

Figure 4: Coefficients as a function of time for open water training.

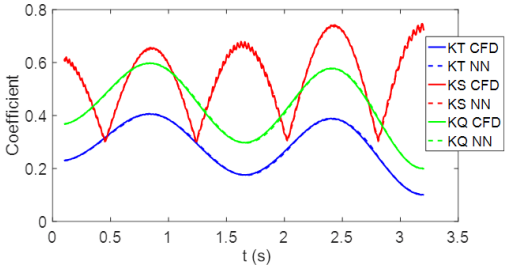

Figure 5: Coefficients as a function of time for the entire time series used for open water training. 70 snapshots of this series were used to train the neural network.

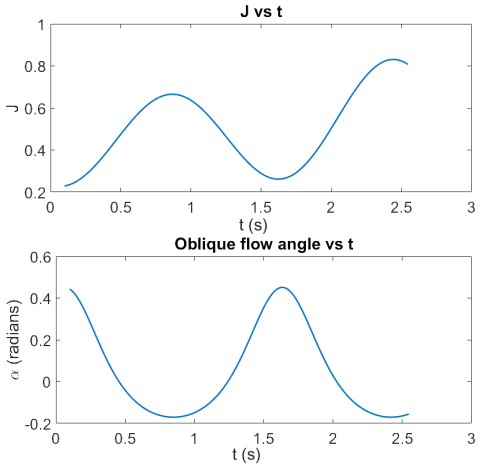

Figure 6: J (top) and α (bottom) as a function of time for the unsteady open water propeller test case.

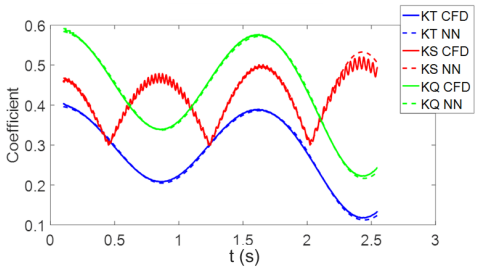

Figure 7: Open water coefficients for the test data set.

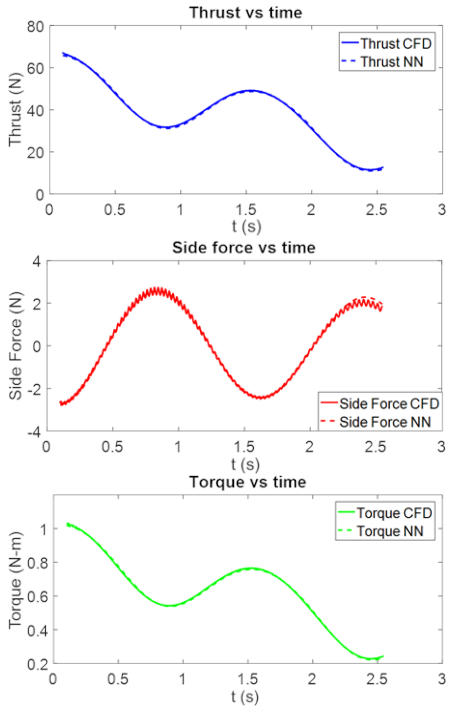

Figure 8: Open water thrust, side force, and torque for the test data set as a function of time.

3.2 Testing the Neural Network for the Unsteady Open Water Propeller Force

The neural network is tested on a case that has unsteady velocity in all three degrees of freedom. The average forward velocity is $\dot{x}_{1,avg} = 1.68$ m/s, the average side velocity is $\dot{x}_{2,avg} = 0$ m/s, and the average propeller revolution rate is $\dot{x}_{4,avg} = 201.06$ rad/s. The amplitudes for each degree of freedom are $A_1 = 0.107$ m, $A_2 = -0.2$ m, $A_4 = 31.4$ radians. The frequencies of oscillation are $\omega_{osc,1} = 4$ rad/s, $\omega_{osc,2} = 4$ rad/s, $\omega_{osc,1} = 1$ rad/s. The resulting J and α as a function of time are shown in Figure 6.

Figure 7 shows the coefficients predicted by the neural network compared to those computed from the CFD. The average loss function is 6.56×10^{-5}, the average L2 norm error of the K_T is 0.0035, the average L2 norm error of the K_S is 0.01015, and the average L2 norm error of the K_Q is 0.0040. Thus, the neural network does a good job of predicting the coefficients for the case that it was not trained with. The coefficients can be expanded back into the forces using Equations (5)-(7), (17), and (18). The sign of the side force is determined with the knowledge that the force acts in the opposite direction that the vessel sways. The thrust, side force, and torque are plotted in Figure 8. Good agreement is seen between the forces predicted by the CFD and the neural network.

4 TRAINING AND TESTING THE NEURAL NETWORK TO PREDICT UNSTEADY BEHIND CONDITION PROPELLER FORCES

The neural network to predict unsteady behind condition propeller forces is first trained with the results of a behind condition unsteady propeller CFD simulation and the steady state open water coefficients extended to the behind condition. Discrete snapshots are examined for the training data. Once the neural network is trained it is tested with several different unsteady behind condition propeller CFD cases that have various motions. These three different cases are denoted 'Test 1', 'Test 2', and 'Test 3'. Table 2 shows the parameters that define the unsteady motion for each of the cases.

Table 2: Behind condition parameters for unsteady motion.

Parameter	Train	Test 1	Test 2	Test 3
$\dot{x}_{1,avg}$ (m/s)	1.68	1.68	1.10	1.68
$\dot{x}_{2,avg}$ (m/s)	0.00	0.00	0.00	0.00
A_1 (m)	0.00	0.00	0.05	0.80
A_2 (m)	0.42	0.11	0.00	0.50
n (rev/s)	32.00	32.00	20.95	32.00
$\omega_{osc,j}$ (rad/s)	1.00	1.00	6.72	1.00

4.1 Training the Neural Network for the Unsteady Behind Condition Propeller Force

The training data is split into two parts. The first part is the unsteady behind condition propeller training case. The average forward velocity is $\dot{x}_{1,avg} = 1.68$ m/s, the average side velocity is $\dot{x}_{2,avg} = 0$ m/s, and the average propeller

revolution rate is $n = 32$ rev/s or $\dot{x}_{4,avg} = 201.06$ rad/s. The amplitudes for each degree of freedom are $A_1 = 0$ m, $A_2 = 0.4265$ m, $A_4 = 0$ radians. The amplitude of sway is equal to the maximum waterline beam (BWL). Therefore, only the sway degree of freedom is unsteady. The frequency of oscillation is $\omega_{osc,2} = 1$ rad/s. The second half of the training data is the steady state open water propeller coefficients which are converted to the behind condition coefficients using the semi-empirical algorithm described earlier (Knight & Maki 2019). Therefore, the K_T and K_Q for steady state open water J values of [0.1, 0.3, 0.5, 0.7 and 0.9] are expanded to the behind condition if the vessel has the same U and n used for those open water coefficients. The K_S is assumed to be zero. This series is repeated twenty times so that the unsteady data and the steady data have similar data sizes for the determination of the weights of the neural network.

To improve the scaling of the training case the coefficients are scaled differently than in the open water neural network. K_T is shifted up by 0.2 as shown by Equation (19). K_S is multiplied by five and is shifted up 0.5 as shown by Equation (20). K_Q follows the same scaling as in the open water neural network, shown by Equation (18). The tolerance of L was set to 8.3×10^{-5} since the convergence met a floor at this level and over 15 million training iterations were used. Figure 9 shows the coefficients predicted by the neural network compared to those extracted from the CFD for this training case. The top plot shows the neural network prediction of the unsteady training data. The bottom plot shows the neural network prediction of the steady training data.

$$K_T = K_{T,u} + 0.2 \tag{19}$$

$$K_S = 5|K_{S,u}| + 0.5 \tag{20}$$

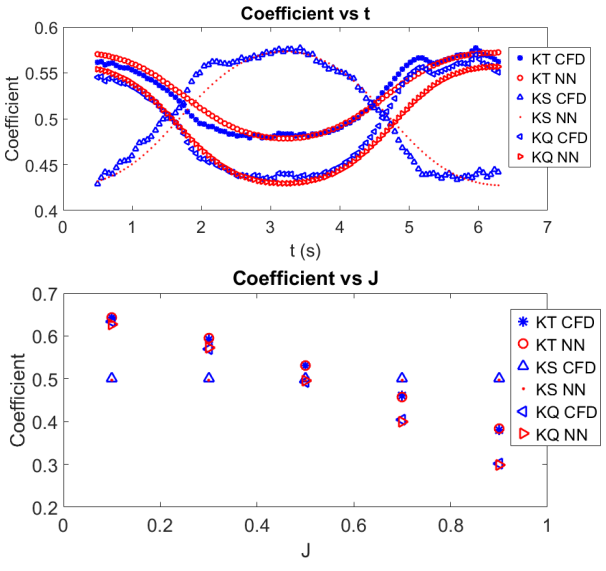

Figure 9: Coefficients for the behind condition training cases. Top: Unsteady training. Bottom: Steady training.

4.2 Testing the Neural Network for the Unsteady Behind Condition Propeller Force

The behind condition neural network is tested against three different CFD simulations, each of which have different prescribed motions as shown by Table 2. Test 1 examines a lower amplitude sway motion than the training case with J held constant at 0.5. Test 2 examines a case where the oblique flow angle is held constant at 0 radians but U, and thus J, varies with time. Test 3 examines a case in which both α and J vary in time. In each of these cases the first 0.5 seconds are neglected to allow the flow to develop.

4.2.1 Behind condition Test 1

The average forward velocity for Test 1 is $\dot{x}_{1,avg} = 1.68$ m/s, the average side velocity is $\dot{x}_{2,avg} = 0$ m/s, and the average propeller revolution rate is $n = 32$ rev/s. Only the sway degree of freedom is unsteady, with an amplitude of $A_2 = 0.1066$ m and a frequency of oscillation of $\omega_{osc,2} = 1$ rad/s. This case has a sway amplitude that is a quarter of the amplitude of the training case. The J is constant at 0.5.

Figure 10 shows the coefficients predicted by the neural network compared to those computed from the CFD for Test 1. The average loss function is 3.43×10^{-5}, the average L2 norm error of the K_T is 0.0032, the average L2 norm error of the K_S is 0.003, and the average L2 norm error of the K_Q is 0.0067. Figure 11 shows the thrust, side force and torque as a function of time. Since the error in the coefficients was low, the predictions of the force by the neural network is also close to that of the CFD. This demonstrates that the neural network can correctly predict the forces when the amplitude of sway is less than it was trained with.

4.2.2 Behind condition Test 2

The average forward velocity for Test 2 is reduced to $\dot{x}_{1,avg} = 1.1$ m/s, the average side velocity is $\dot{x}_{2,avg} = 0$ m/s, and the average propeller revolution rate is $n = 20.95$ rev/s. Only the surge degree of freedom is unsteady, with an amplitude of $A_1 = 0.0487$ m and a frequency of oscillation of $\omega_{osc,1} = 6.72$ rad/s. This frequency of oscillation was used in (Knight & Maki 2018) and represents the frequency of encounter if the vessel were to be in head seas with the wavelength equal to the length of the waterline. The α is constant at 0 radians. J as a function of time is shown in Figure 12.

The top of Figure 13 shows the comparison between the neural network coefficient predictions and the CFD predictions. The average loss function is 1.18×10^{-4}, the average L2 norm error of the K_T is 7.25×10^{-3}, the average L2 norm error of the K_S is 4.58×10^{-3}, and the average L2 norm error of the K_Q is 1.27×10^{-2}. The oscillation in side force is caused by the azimuthal position of the propeller and the resulting force is around zero. The error for K_T and K_Q is higher than Test 1, but still quite accurate. The bottom two plots of Figure 13 show the thrust and torque as a function of time. This demonstrates that the neural network can predict the forces well when J oscillates.

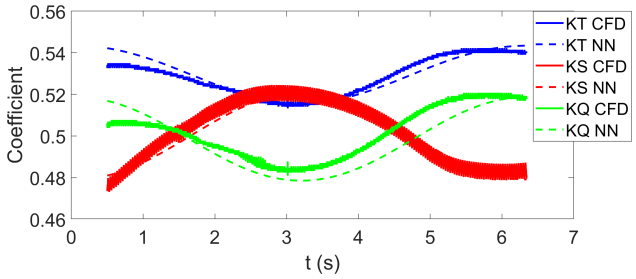

Figure 10: Behind condition Test 1 coefficients as a function of time.

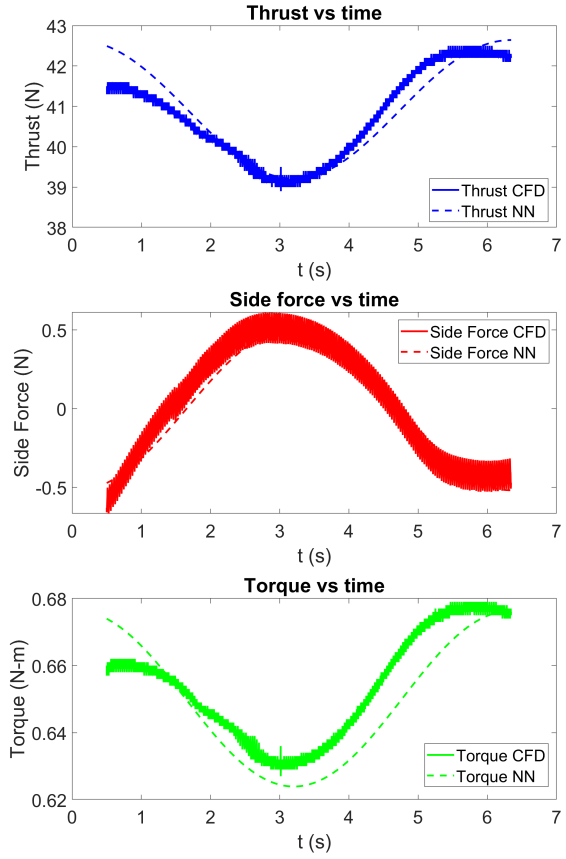

Figure 11: Behind condition thrust, side force, and torque for Test 1 as a function of time.

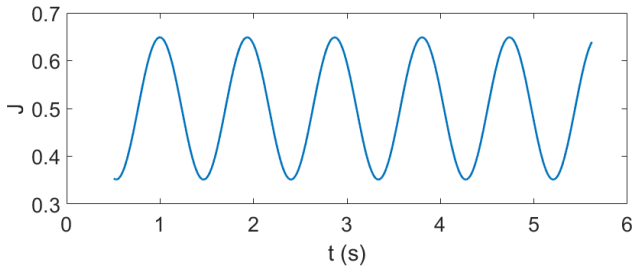

Figure 12: J as a function of time for behind condition Test 2.

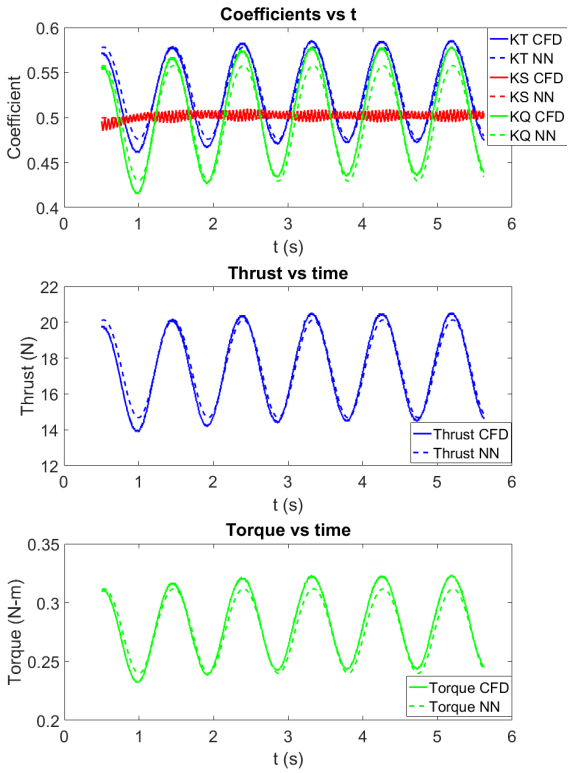

Figure 13: Behind condition coefficients, thrust, and torque for Test 2 as a function of time.

4.2.3 Behind condition Test 3

Test three has both unsteady forward velocity as well as unsteady sway velocity. The average forward is $\dot{x}_{1,avg} = 1.68$ m/s, the average side velocity is $\dot{x}_{2,avg} = 0$ m/s, and the average propeller revolution rate is $n = 32$ rev/s. Frequency of oscillation for both surge and sway is $\omega_{osc,1} = \omega_{osc,2} = 1$ rad/s. The amplitude of unsteady motion is $A_1 = 0.8$ m and $A_2 = 0.5$ m. Thus, the sway amplitude is larger than the training case. Figure 14 shows how J and α vary with time for Test 3.

Figure 15 shows the neural network coefficient predictions compared to the CFD predictions. The average loss function is 2.58×10^{-4}, the average L2 norm error of the K_T is 9.55×10^{-3}, the average L2 norm error of the K_S is 1.86×10^{-2}, and the average L2 norm error of the K_Q is 8.84×10^{-3}. Thus, there is larger L and L2 norm error compared to Test 1; however, the trends are quite good as shown by Figure 15 and Figure 16. Figure 16 shows the thrust, side force and torque as a function of time. This demonstrates that the neural network can predict the forces well when the amplitude of sway larger than it was trained with and when there is simultaneously unsteady forward speed. To note, the results may be improved if velocity probe data was added as a feature in the input to the neural network. Due to the large amplitude sway motion, vortices are shed from the bow of the vessel and the propeller plane passes through these vortices. When the propeller passes through the vortices it creates a perturbation in the propeller forces and this effect is not modelled by this neural network.

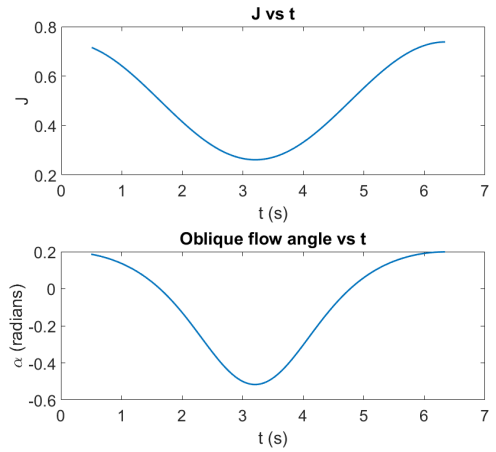

Figure 14: J (top) and α (bottom) as a function of time for Test 3.

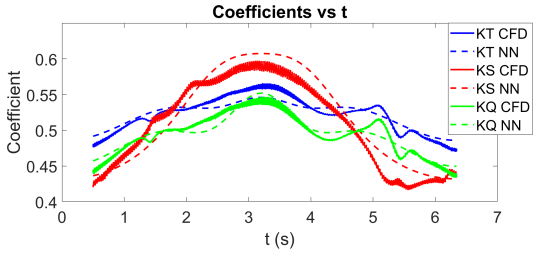

Figure 15: Behind condition Test 3 coefficients as a function of time.

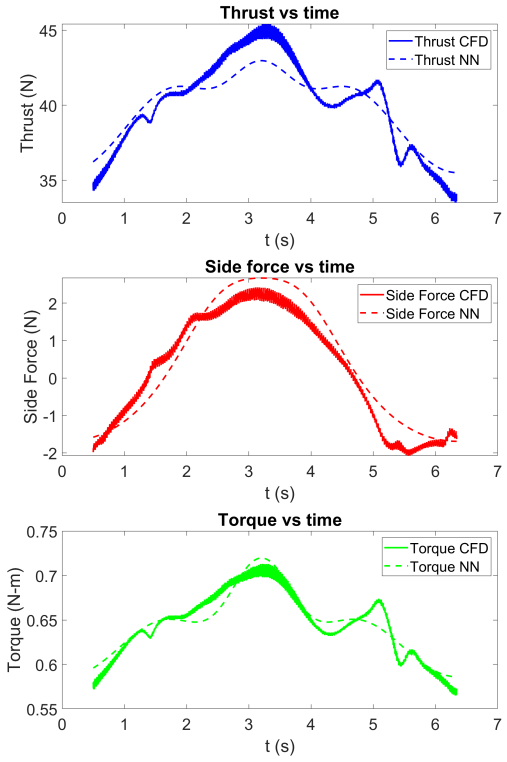

Figure 16: Behind condition thrust, side force, and torque for Test 3 as a function of time.

CONCLUSIONS

Two neural networks have been trained and tested to determine the unsteady propeller forces due to unsteady motions. The first neural network examined is for open water and it takes a feature vector comprised of various permutations of the instantaneous J and $|\alpha|$ for each instant in time. The second neural network examined is for the behind condition and it uses a similar feature vector, but with the difference of using α instead of $|\alpha|$. Each neural network is able to predict the unsteady propeller forces for unsteady motions. This study is a demonstration of how a neural network can be trained and tested to predict the correct propeller forces for unsteady motion.

The behind condition neural network could be further improved by incorporating other features in the input. For example, the feature vector could be expanded to incorporate history terms to account for memory effects in the flow. Furthermore, probes could be placed upstream of the propeller plane and the effects of shed vortices from the hull or incident velocities could be accounted for.

The neural networks, especially the behind condition neural network, could be very useful for analyzing a maneuvering vessel. Instead of having to discretize the propeller and use a rotating mesh to simulate the viscous effects on the propeller, the neural network could be used. This could dramatically reduce the cost of a maneuvering calculation using CFD. Furthermore, the neural network could be trained with any type of training data whether it be experiments, potential flow calculations, RANS CFD, LES, or any combination of these. For example, LES could be used to train the neural network in off design circumstances and potential flow methods could be used to train the on design points.

REFERENCES

Abramowski, T. (2005). 'Prediction of Propeller Forces During Ship Maneuvering'. Journal of Theoretical and Applied Mechanics **43**(1), pp. 157-178.

Hornik, K., Stinchcombe, M., & White, H. (1989). 'Multilayer feedforward networks are universal approximators'. Neural Networks **2**(5), pp. 359-366.

Knight, B. & Maki, K. (2018). 'Body Force Propeller Model for Unsteady Surge Motion'. ASME 37th International Conference on Ocean, Offshore and Arctic Engineering (OMAE), Madrid, Spain.

Knight, B. & Maki, K. (2019). 'A Semi-Empirical Multi-Degree of Freedom Body Force Propeller Model'. Ocean Engineering **178**, pp. 270-282.

Singh, A.P., Medida, S., & Duraisamy, K. (2017). 'Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils'. AIAA Journal **55**(7), pp. 2215-2227.