X International Conference on Computational Methods in Marine Engineering
MARINE 2023
J. García-Espinosa, L. González, J.E. Gutiérrez & B. Serván-Camas (Eds)

Investigating Nonlinear Forces in Ship Dynamics using Machine Learning

Kyle E. Marlantes[1]*, Piotr J. Bandyk† and Kevin J. Maki[1]

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

† MARIN USA, Inc., Houston, TX, USA

* Corresponding author: Kyle Marlantes, kylemarl@umich.edu

ABSTRACT

In this paper a hybrid data-driven method is applied to model the nonlinear heave responses of a two-dimensional wedge in irregular waves. The method uses a machine learning model to learn nonlinear components of the total force. The data-driven force model is embedded in a linear ordinary differential equation which is solved to yield position, velocity, and acceleration. Training data for the wedge are generated for different levels of nonlinear forces, including nonlinear hydrostatics, Froude-Krylov forces, and hydrodynamic memory effects, to investigate how the different forces affect the performance and configuration of the data-driven model. It is found that the data-driven model is most effective when linear forces are not learned, but modeled analytically within the governing equation, and the presence of memory effects requires longer input sequences in the machine learning model.

Keywords: nonlinear; dynamics; seakeeping; wave-induced; motions; data-driven; neural network.

NOMENCLATURE

α	Nonlinear part of a solution [m]
A_∞	Two-dimensional linear infinite-frequency added mass coefficient [kg/m]
B_0	Two-dimensional linear damping coefficient [kg/s/m]
C_0	Two-dimensional linear hydrostatic stiffness coefficient [kg/s^2/m]
d	Instantaneous depth of wedge [m]
d_0	Equilibrium depth of wedge [m]
δ	Nonlinear force correction [N]
η	Incident wave elevation [m]
F	Force [N]
g	Acceleration due to gravity [m/s^2]
Γ	Deadrise angle of wedge [rad]
(h)	Superscript denoting a high-fidelity quantity [-]
k	Stencil size of input sequence to machine learning model [-]
k_i	Wave number of i'th Fourier component [m]
K	Retardation function in heave [kg/s^2/m]
L	Length of submerged half-girth of wedge [m]

(l)	Superscript denoting a low-fidelity quantity [-]
m	Two-dimensional physical mass [kg/m]
ω_i	Wave frequency of i'th Fourier component [rad/s]
ϕ_i	Wave phase angle of i'th Fourier component [rad]
ρ	Density of water [kg/m^3]
t	Time [s]
w	Width of wedge waterplane [m]
ξ, $\dot{\xi}$, $\ddot{\xi}$	Heave position, velocity, acceleration [m], [m/s], [m/s^2]
x, y, z	Longitudinal, transverse, vertical coordinate direction [m]
ζ_i	Wave amplitude of i'th Fourier component [m]

1. INTRODUCTION

The motivation behind most mathematical models of ship dynamics is to predict the motions and loads of a vessel in waves. Much of the difficulty is the computation of the fluid forces acting on the vessel. Computational fluid dynamics (CFD) predicts the total force directly at high-accuracy, though this comes at a large computational cost. Most workhorse methods are based on a phenomenological subdivision of the forces, so that they may be modeled in a component-wise manner. Linear methods are common in engineering practice due to their low-cost and robust solutions, though they do not perform well when evaluating large amplitude responses in high-energy seaways (Smith & Silva, 2017). For this reason, it is also common to use weakly-nonlinear methods, where some force components which are inexpensive to compute are modeled in a nonlinear manner (Weems & Wundrow, 2013). However, these methods still omit important physics. Due to the cost of predicting hydrodynamic forces at high-fidelity, it is impossible to analyze the motions of a vessel in a large number of wave conditions, or to generate long time series to identify extreme values, without a reduction in accuracy. This compromise between computational cost and accuracy precludes comprehensive high-fidelity evaluations of a ship's performance in waves.

Some work has been done to bring high-fidelity, yet low cost, simulations into the state-of-the-art. Most of the effort focuses on data-driven methods, such as Long Short-Term Memory (LSTM) neural networks (Hochreiter & Schmidhuber, 1997), to develop time-domain surrogate models of the problem. These methods rely on training datasets and machine learning models to learn the entire physical process, taking waves as input and returning vessel motions as output. Some examples of LSTM-based data-only surrogate models in ship motions include the work by Xu (2020), Xu, Maki, and Silva (2021), Guo, Zhang, Tian, Lu, and Li (2022), and Silva and Maki (2022). Some authors have explored alternative architectures, such as the feed-forward networks of (Liong & Chua, 2022), the recent work by M. Zhang, Taimuri, Zhang, and Hirdaris (2023), and the Temporal Convolutional Network (TCN) with attention by (B. Zhang, Wang, Deng, Jia, & Xu, 2023). However, data-only methods typically require a large training data set and if a design parameter is changed, such as rudder size or loading condition, the entire data set must be updated and the surrogate model retrained. Since training data is ideally developed using high-fidelity, yet high-cost, simulation tools, the practicality of frequent retraining is limited.

Hybrid methods, which combine physics-based models with data-driven techniques, are gaining traction in recent years as they have been shown to reduce the required size of the training data set and to better generalize across input conditions (Willard, Jia, Xu, Steinbach, & Kumar, 2020). Some examples include the physics-based learning models (PBLMs) proposed by Weymouth and Yue (2014), where one input in the data-driven model is a "physics basis". Wan, Vlachas, Koumoutsakos, and Sapsis (2018) expressed a "complementary dynamics" where an LSTM network corrects the overall dynamics and applied the technique to example problems in dynamics and fluids. Raissi, Perdikaris, and Karniadakis (2019) proposed the Physics Informed Neural Network (PINN) which considers physics in the development of the network architecture. More recently, Diez, Serani, Gaggero, and Campana (2022) proposed a unique method where RNNs and DMD work in tandem as a hybrid multi-fidelity framework. Contributions such as Schirmann, Collette, and Gose (2022) continue to point out the value of physics-based knowledge, further exposing the limitations of data-only methods.

One hybrid method, the neural-corrector method proposed by Marlantes and Maki (2022), focuses on modeling only the nonlinear component of the total force which cannot be represented using linear terms. The method takes the same phenomenological approach to the subdivision of forces as classical ordinary differential equation (ODE) methods, but corrects the low-order physics by adding data-driven force terms. The idea is to use data-driven modeling only when absolutely necessary, and to use robust low-order physics whenever possible. Focusing on the force means the model does not necessarily need to be retrained if a design parameter is changed, especially if the change is captured by the low-order physics. Furthermore, Marlantes and Maki (2022) show that the method can accurately predict responses in wave conditions that are different from the original training data set.

The work in this paper uses a canonical problem–a two-dimensional heaving wedge–to explore how different nonlinear force components acting on a ship in waves affect the performance and configuration of the neural-corrector method when predicting nonlinear motions. The goal is to determine how best to learn the nonlinear force components and how the learned forces affect solutions to the governing equations. The remainder of this paper is divided into three sections. Section 2 outlines the neural-corrector method, the two-dimensional wedge model, the governing equations, and the different hydrodynamic force terms which will be used as the subject of the study. Section 3 gives the results of the study, with particular attention to the effects of different nonlinear forces and optimal model configurations. Section 4 gives a brief summary of the findings and provides conclusions.

2. METHODOLOGY

The neural-corrector method requires a high-fidelity (h) and low-fidelity (l) model, as shown by Eqs. (1) and (2), respectively, and indicated by superscripts. Eqs. (1) and (2) collect all force components into the force vector on the right-hand-side, which may include fictitious forces if using a non-inertial reference frame.

$$m\ddot{\vec{\xi}}^{(h)} = \vec{\mathbf{F}}^{(h)} \tag{1}$$

$$m\ddot{\vec{\xi}}^{(l)} = \vec{\mathbf{F}}^{(l)} \tag{2}$$

$$m\ddot{\vec{\xi}}^{*} = \vec{\mathbf{F}}^{(l)} + \vec{\delta}(\vec{\xi}^{(l)}, \eta) \tag{3}$$

The h- and l-models are brought together in Eq. (3), where a data-driven term $\vec{\delta}$ is included on the right-hand-side to correct the low-fidelity forces. This decomposition of the forces can be shown analytically using expansions of the high-fidelity force components, as shown in Marlantes and Maki (2022). The solution to Eq. (3) is the state $\vec{\xi}^{*}$ which should approach $\vec{\xi}^{(h)}$ if $\vec{\delta}$ is accurately modeled. If $\vec{\delta}$ is zero, $\vec{\xi}^{*}$ will equal the low-fidelity solution. Solutions to Eq. (3) are sought via numerical integration, so that $\ddot{\vec{\xi}}^{*}$, $\dot{\vec{\xi}}^{*}$, and $\vec{\xi}^{*}$ are obtained simultaneously.

In this work, $\vec{\delta}$ is modeled as a function of the low-fidelity state $\vec{\xi}^{(l)}$, its derivatives, and the wave elevation η. This is done to uncouple the force model from the solution $\vec{\xi}^{*}$ when solving Eq. (3)

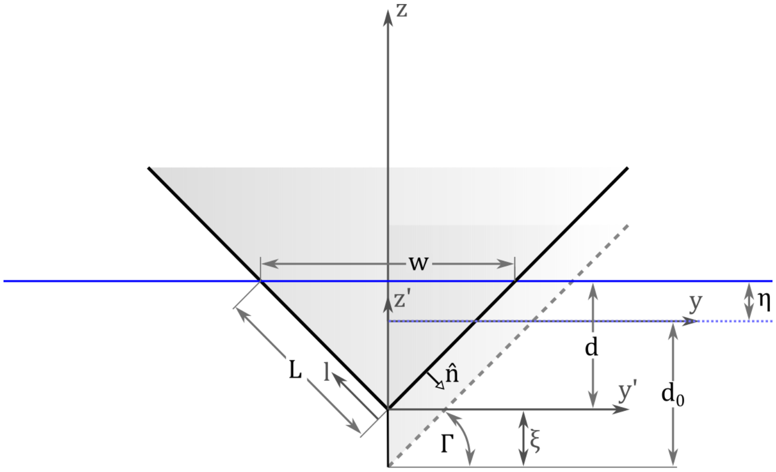

Figure 1: Two-dimensional heaving wedge.

numerically. To accommodate numerical integration of Eq. (3), $\vec{\delta}$ receives k-length discrete sequences of $\vec{\xi}^{(l)}$ and η as inputs at each time step. The optimal value of k and how it is influenced by the underlying physics in the h- and l-models will be investigated.

Figure 1 gives the geometric details of the heaving wedge which serves as a prototypical example of a ship in waves. The reference frame (y, z) is inertial and the motion is restricted to a single degree-of-freedom, so that the governing equations from this point onward are scalar and fictitious forces are not present. The initial static equilibrium depth is d_0 and the instantaneous depth is d. Note that incident waves are assumed to travel into the page, i.e. head seas if the wedge is viewed as a ship section. The unsteady free-surface is then defined by η, which appears as a vertical change in surface elevation.

Eq. (4) gives the high-fidelity model that will be used in this work, where $\xi^{(h)}$ is the high-fidelity heave position of the wedge.

$$(m + A_\infty)\ddot{\xi}^{(h)}(t) + B_\infty\dot{\xi}^{(h)}(t) + \underbrace{C(\xi^{(h)})\xi^{(h)}(t)}_{\text{hydrostatics}} + \underbrace{\int_0^t K(\tau)\dot{\xi}^{(h)}(t-\tau)\, d\tau}_{\text{memory}} = \underbrace{F_w(t, \eta, \xi^{(h)})}_{\text{wave excitation}} \tag{4}$$

The model includes linear infinite-frequency added mass A_∞ and damping B_∞, nonlinear hydrostatic restoring forces $C(\xi^{(h)})$, memory effects in the radiation forces, where $K(\tau)$ is the retardation function, and nonlinear wave excitation forces $F_w(t, \eta, \xi^{(h)})$. m is the physical mass of the wedge.

The irregular wave elevation η is given by Eq. (5), where N is the number of Fourier components and x is the longitudinal location of the wedge in the wave, which is always zero. The wave components are sampled from a wave energy spectrum and the phase angles ϕ_i are randomly selected.

$$\eta(t) = \sum_i^N \zeta_i \cos(k_i x - \omega_i t + \phi_i) \tag{5}$$

The low-fidelity model is given by Eq. (6), which is simply the linearized version of Eq. (4). A linear damping coefficient B_0, determined using an energy method or system identification, replaces the memory terms and the hydrostatic stiffness coefficient C_0 corresponds to the equilibrium condition when $d = d_0$. The wave excitation forces also no longer depend on the state and are instead linearized around the equilibrium condition.

$$(m + A_0)\ddot{\xi}^{(l)}(t) + B_0\dot{\xi}^{(l)} + C_0\xi^{(l)}(t) = F_w^{(l)}(t, \eta) \tag{6}$$

In this work, the neural-corrector form of the governing equation can take two forms, Eq. (7) or Eq .(8), depending on how the hydrodynamic forces are included in δ.

$$(m + A_0)\ddot{\xi}^*(t) + B_0\dot{\xi}^*(t) + C_0\xi^*(t) = F_w^{(l)} + \delta_-(\xi^{(l)}, \eta) \tag{7}$$

$$m\ddot{\xi}^*(t) = F_w^{(l)} + \delta_+(\xi^{(l)}, \eta) \tag{8}$$

The first form excludes the linear added mass, damping, and restoring force terms from δ_-. The second form includes the linear force terms in the force correction δ_+, so that they are learned by the data-driven model and not computed analytically. Of particular interest is whether the linear force terms should be included in δ and learned by the data-driven model or modeled directly within the governing equation.

2.1 Wave Excitation Force

The excitation force due to incident waves is the body-exact Froude-Krylov force, given by Eq. (9) for a harmonic wave, where ζ is the wave amplitude, $\hat{\boldsymbol{n}} = \sin(\Gamma)\hat{\boldsymbol{j}} - \cos(\Gamma)\hat{\boldsymbol{k}}$ is the surface normal vector pointing into the fluid, as shown in Fig. 1. For simplicity, diffraction forces are ignored.

$$\vec{\mathbf{F}}_{FK} = -2\rho g\zeta\cos(kx - \omega t + \phi)\int_L \exp(kz)\cdot\hat{\boldsymbol{n}}\,dl \tag{9}$$

The wetted surface L is related to the instantaneous depth of the wedge d and the deadrise angle Γ, given by Eq. (10). The instantaneous depth of the wedge is a function of the incident wave elevation and the heave position of the wedge as given by Eq. (11). The assumption that $d \geq 0$ for all t is enforced.

$$L = d\csc(\Gamma) \tag{10}$$

$$d = d_0 + \eta - \xi \tag{11}$$

Since the wedge is free in heave, only the vertical component of the force $\vec{\mathbf{F}}_{FK}$ is needed and $\hat{\boldsymbol{n}}$ is replaced by its vertical component, $-\cos(\Gamma)\hat{\boldsymbol{k}}$. By making a coordinate transformation $z = l\sin(\Gamma) - d$, the integral in Eq. (9) is evaluated to yield Eq. (12), where subscript i denotes the i'th Fourier component of the irregular wave train. The total wave excitation force for an irregular wave train composed of N Fourier components is given by Eq. (13), where ϕ_i is the component phase angle.

$$F_{w,i}(t, \eta, \xi) = 2\rho g\zeta_i\cos(k_i x - \omega_i t + \phi_i)\frac{k_i\tan(\Gamma)}{\exp(k_i d)}\left(\exp(k_i L\sin(\Gamma)) - 1\right) \tag{12}$$

$$F_w(t, \eta, \xi) = \sum_i^N F_{w,i}(t, \xi) \tag{13}$$

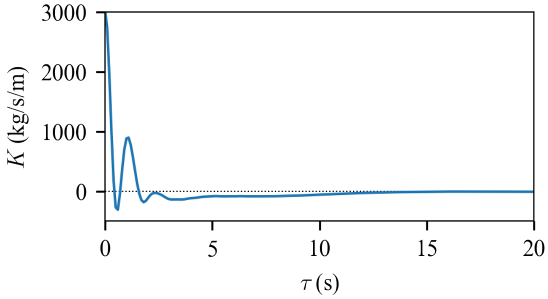

Figure 2: Retardation function for a two-dimensional wedge ($\Gamma = 45°$) oscillating in heave. The function is computed from the frequency dependent damping coefficient for the equilibrium condition $d_0 = 0.5$ m.

2.2 Memory Effects

Hydrodynamic memory effects are included using a convolution term, as shown in Eq. (4). The kernel is the impulse response function $K(\tau)$ in heave computed from the frequency dependent hydrodynamic damping coefficient $B(\omega)$, given by Eq. (14). In this work $B(\omega)$ is computed using a 2D boundary element potential flow method over a range of wave frequencies.

$$K(\tau) = \frac{2}{\pi}\int_0^\infty \left(B(\omega) - B_\infty\right)\cos\left(\omega t\right)\, d\omega \tag{14}$$

2.3 Hydrostatic Restoring Force

The body-exact hydrostatic restoring force acting on the wedge is modeled using a nonlinear stiffness coefficient $C(\xi)$, given by Eq. (15), where w is the instantaneous width of the waterplane of the wedge. The coefficient is derived exactly for the wedge geometry shown in Fig. 1, yielding the expression for w given by Eq. (16).

$$C(\xi) = \rho g w(\xi) \tag{15}$$

$$w(\xi) = 2\cot(\Gamma) d(\xi) \tag{16}$$

Substituting the instantaneous depth d from Eq. (11) into Eq. (16), simplifying and expanding, the nonlinear hydrostatic restoring force is given by Eq. (17), where the linear and nonlinear components of the force are evident.

$$F_{HS} = C(\xi)\xi = 2\rho g \cot\left(\Gamma\right)\left(d_0\xi + \eta\xi - \xi^2\right) \tag{17}$$

2.4 Machine Learning Model for δ

The neural-corrector method may be configured using any type of machine learning architecture suitable for the time-marching problem. In Marlantes and Maki (2022), LSTM recurrent neural networks are utilized. However, given the relatively small values of k used in that study, the need for recurrent architectures may be questioned. In this work, feed-forward dense neural networks are utilized due to their simplicity, rapid training, and low evaluation time. Fig. 3 illustrates the neural network architecture used in this study. A total of three hidden layers with ten cells each are used. ReLu activation functions are specified in the input and hidden cells, and a single cell linear activation

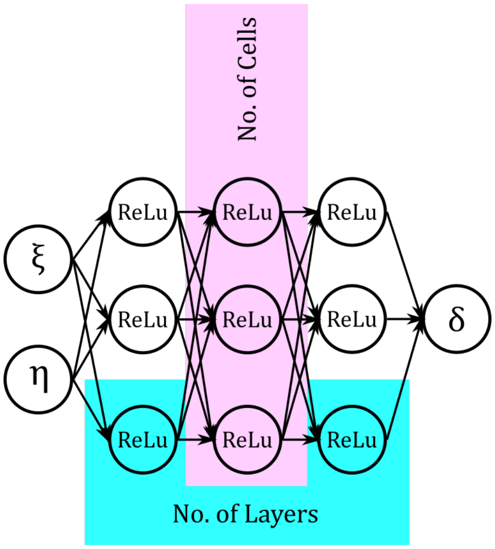

Figure 3: Feed-forward dense neural network architecture used to model δ.

function is used for the output layer. The model is configured according to a many-to-one paradigm, so that it will receive k-length sequences of $\xi^{(l)}$ and η in the input layer and output a single value of δ at the next time step.

The dense networks are trained using the Adam optimizer (Kingma & Ba, 2014) using a mean-squared-error loss function. The training process is halted when convergence is observed in the loss on successive epochs. This stopping policy avoids over-fitting and ensures adequate training when the size or quality of training data may vary between comparable models.

3. RESULTS

The particulars of the wedge considered in this study are shown in Table 1.

Table 1: Particulars of wedge.

Γ (deg)	45.0
d_0 (m)	0.5
m (kg/m)	256.3
ρ (kg/m^3)	1000.0

Irregular waves are generated using a JONSWAP wave spectrum with a significant wave height H_s of 1 meter and a peak period T_p of 4 seconds. The energy spectrum is discretized using $N = 100$ harmonic components using a uniform random sampling over the frequency range $[1/2T_p, 4/T_p]$ to prevent a repeat period in the resulting irregular wave elevation. Two different wave realizations η_{train} and η_{test} are developed–the first is used as training data and the second is reserved for testing data. Each wave realization is 400 s in length with a time step of 0.01 s and has a different set of random phase angles ϕ_i and harmonic components.

Vertical motions of the wedge are computed numerically for the two irregular wave realizations using four different levels of nonlinearity, as shown in Table 2. Level I corresponds to a numerical solution of the linearized low-fidelity model Eq. (6). Level II includes only memory effects–the hydrostatic

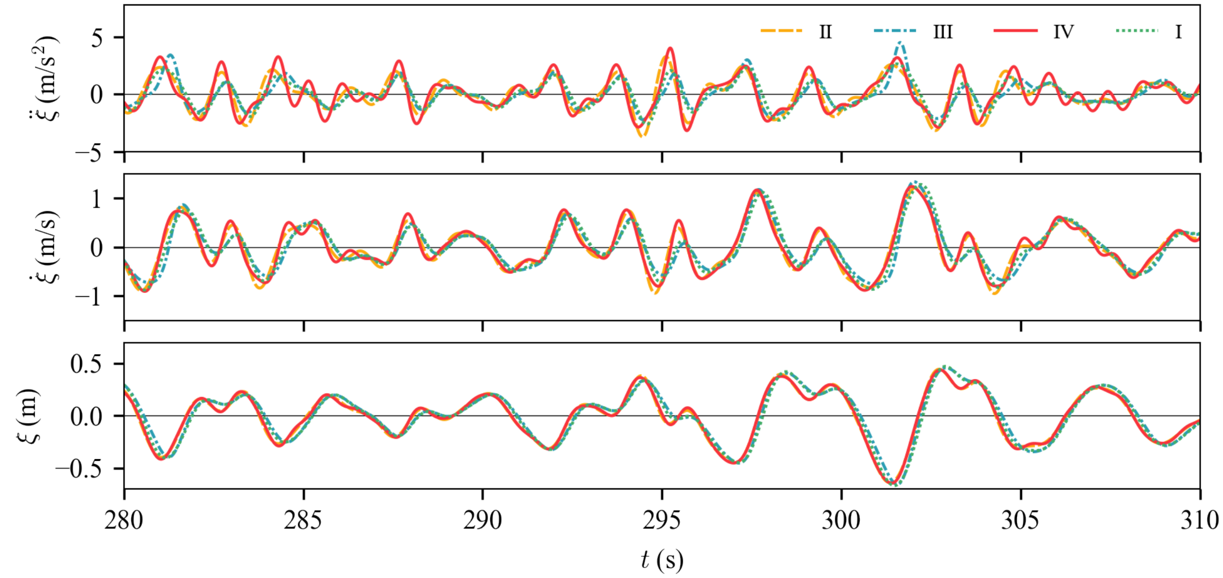

Figure 4: Acceleration, velocity, and position training data predictions for levels I though IV.

restoring force and Froude-Krylov forces remain linear. Level III includes only nonlinear hydrostatic and Froude-Krylov forces, which must be considered together. Level IV includes all nonlinear force components. Where nonlinearity is present, a system identification is used to determine the linear coefficients, see Appendix B. Figure 4 shows a sample of the acceleration, velocity, and position time series training data for each level of nonlinearity.

Table 2: Levels of nonlinearity and linear coefficients for wedge motion predictions.

Level, i	Nonlinear HS	Nonlinear FK	Memory Effects	A_0 (kg/m)	B_0 (kg/s/m)	C_0 (kg/s^2/m)
I	False	False	False	22.8	1500.0	9810.0
II	False	False	**True**	44.5	527.0	10719.2
III	**True**	**True**	False	-85.7	1712.3	9156.2
IV	**True**	**True**	**True**	29.5	515.0	10714.4

For each level in Table 2, training data for δ are computed by solving Eqs. (7) and (8) for δ_- and δ_+, respectively. To ensure that ξ^* approaches $\xi^{(h)}$, the responses are replaced with the high-fidelity time series from each level of nonlinearity to develop the training data. Figure 5 shows a snippet of δ_- and δ_+ time series for levels II through IV, with $F^{(l)}$ also shown for reference.

The three levels of nonlinearity in Table 2 and the two different force corrections, δ_- and δ_+ in Eqs. (7) and (8), form a matrix of six model configurations. A separate neural network model is trained for each configuration using responses from the training waves η_{train} for k values between 1 and 200. The trained models are embedded in Eqs. (7) or (8), as appropriate, and the equations are solved numerically to yield responses in η_{test}. The predicted position ξ^*, velocity $\dot{\xi}^*$, acceleration $\ddot{\xi}^*$ are compared to direct solutions of Eq. (4) and average L_2 and L_∞ error norms are computed for the time series. Table 3 gives the minimum average L_2 error in the state predictions for each model configuration.

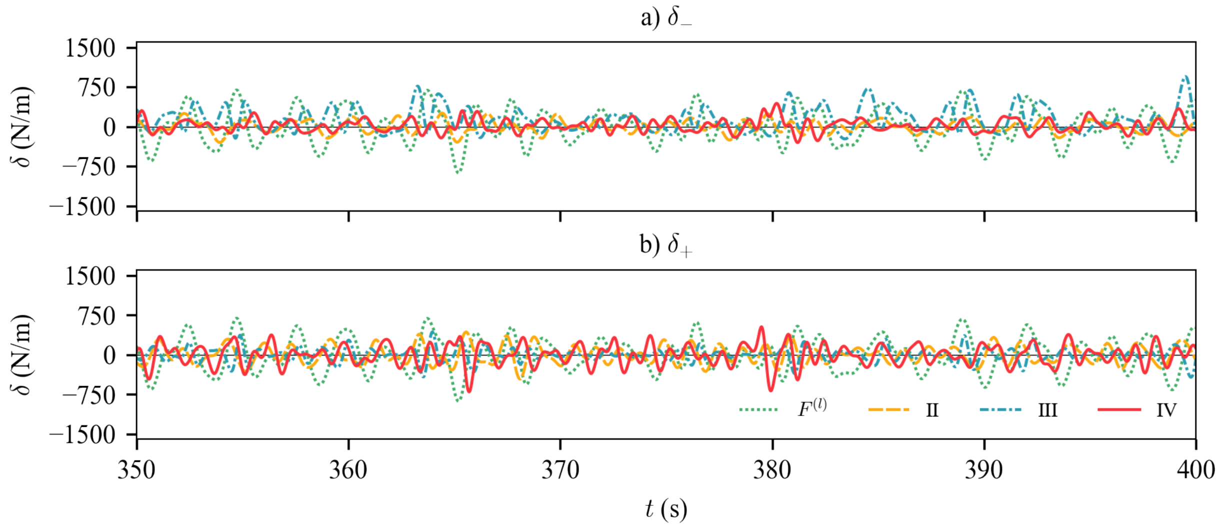

Figure 5: δ for the different levels of nonlinearity, excluding or including linear forcing terms: a) gives δ_- where the linear force terms are modeled analytically. b) gives δ_+ where the linear force terms are included in the force correction.

Table 3: Minimum average L_2 errors and associated k values for different configurations of the neural-corrector method when compared to test data.

		Linear Force Terms	
Level	Response	Exclude, δ_-	Include, δ_+
II	acc (m/s^2)	0.371, k=150	**0.289**, k=200
	vel (m/s)	**0.068**, k=150	0.301, k=25
	pos (m)	**0.022**, k=100	17.257, k=1
III	acc (m/s^2)	0.372, k=150	**0.284**, k=200
	vel (m/s)	**0.058**, k=150	0.212, k=50
	pos (m)	**0.014**, k=200	10.697, k=200
IV	acc (m/s^2)	0.370, k=150	**0.281**, k=200
	vel (m/s)	**0.069**, k=100	0.278, k=100
	pos (m)	**0.023**, k=200	9.769, k=100

Nonlinear effects are often most evident in the accelerations, as shown in Figure 4. For this reason, it may be advantageous to optimize k based on the acceleration (and therefore total force) given that integration of an accurate force model should also result in accurate velocity and position predictions, but a model optimized for position may not yield a good prediction of acceleration and force. The values of k that yield the lowest errors in acceleration for most configurations in Table 3 are >50, however, the value of k that corresponds to the minimum possible error is not necessarily the optimal value in practice.

Figure 6 shows the L_2 and L_∞ errors for the acceleration, velocity, and position predictions for levels II, III, and IV using δ_-. The errors for levels II and IV drop rapidly after $k \gtrsim 40$ and then plateau. This suggest that the minimum k value when memory effects are present should be about 50, or a time period of about 0.5 s, which is rather brief. Though this may not yield the absolute minimum error, it is computationally more efficient, as shorter sequences are typically less costly to evaluate.

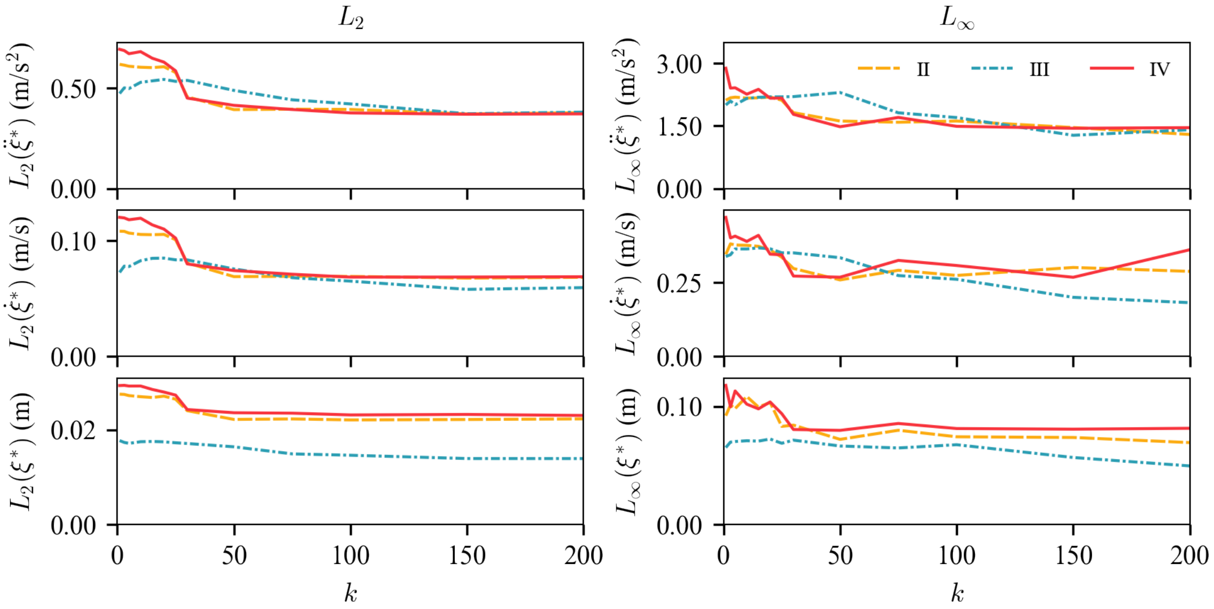

Figure 6: L_2 and L_∞ prediction errors using δ_- for increasing values of k.

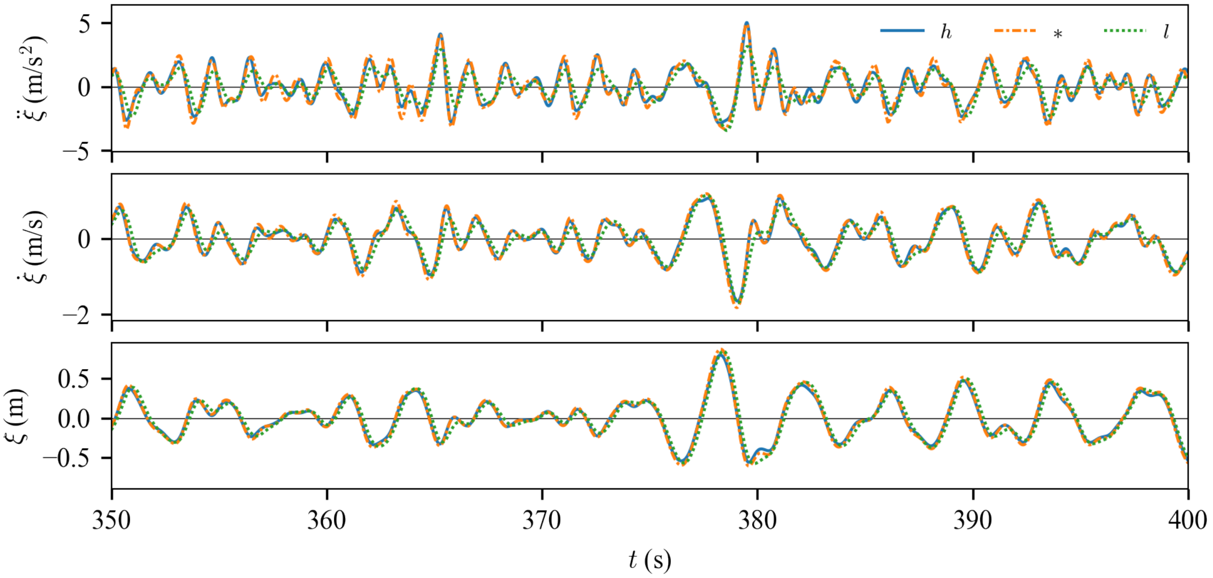

Figure 7: Acceleration, velocity, and position predictions using δ_- and k=100 in test waves η_{test} for level IV.

For level III, which includes nonlinear hydrostatic and Froude-Krylov forces, the prediction errors for acceleration and velocity initially start off smaller, then increase near k=25, before decreasing and plateauing. For predictions of position, the errors remain similar across changes in k. The prediction errors for level III are also much smaller than the errors for II and IV for small values of k. This is consistent with the underlying physics, where the nonlinear hydrostatic and Froude-Krylov forces depend only on the instantaneous position of the wedge. As a result, small values of k yield relatively good performance. In the presence of memory effects, however, small values of k lead to larger errors.

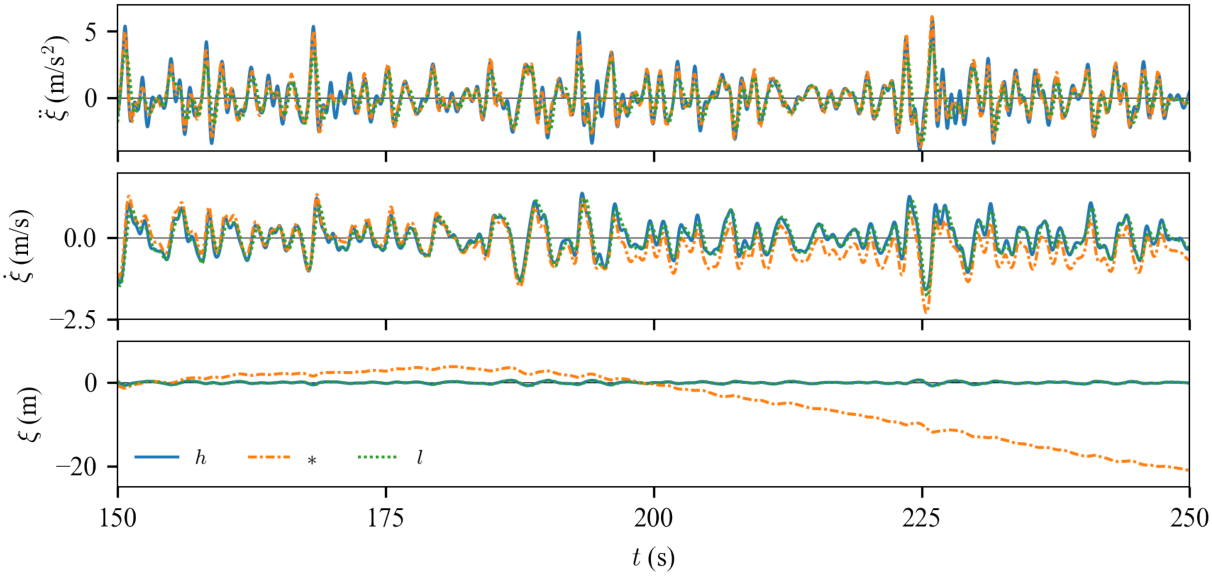

Figure 8: Position, velocity, and acceleration predictions for δ_+ using k=75 for level IV. Notice the drift in the velocity and position predictions.

Figure 7 gives the level IV acceleration, velocity, and position predictions in η_{test} for δ_- and a k value of 100. The neural-corrector predictions follow the high-fidelity test data closely. Solutions to the low-fidelity model, Eq. (6), are also shown for reference.

3.1 Effect of Linear Force Terms

Table 3 shows that errors for δ_-, which excludes linear force terms, are lower for velocity and position predictions. For δ_+, which includes the linear forces, errors in the acceleration predictions are lower, however, very large errors are observed for velocity and position. Figure 8 gives the acceleration, velocity, and position time series at level IV using the δ_+ configuration. Notice the drift in the velocity and position predictions which grows with time.

The reason for this drift can be explained analytically. Consider a simplified equation of motion as given by Eq. (18), where $F^{(l)}$ is the linear part of the force. Assuming δ_- is exact, the solution to Eq. (18) yields the nonlinear state $\xi^{(h)}$.

$$m\ddot{\xi}^{(h)} + C_0\xi^{(h)} = F^{(l)} + \delta_- \tag{18}$$

The neural-corrector method makes the assumption that the total force can be decomposed into a linear and nonlinear part. Under this assumption, the state can also be decomposed into a linear and nonlinear part, such that $\xi^{(h)} = \xi^{(l)} + \alpha$ as given in Eq. (19), where $F^{(l)}$ and δ_- are also brought to the left-hand-side of the equation. Because the linear equation satisfies the linear terms, the terms cancel and only the nonlinear component of the solution α remains, as shown by Eq. (20).

$$m(\ddot{\xi}^{(l)} + \ddot{\alpha}) + C_0(\xi^{(l)} + \alpha) - F^{(l)} - \delta_- = 0 \tag{19}$$
$$m\ddot{\alpha} + C_0\alpha - \delta_- = 0 \tag{20}$$

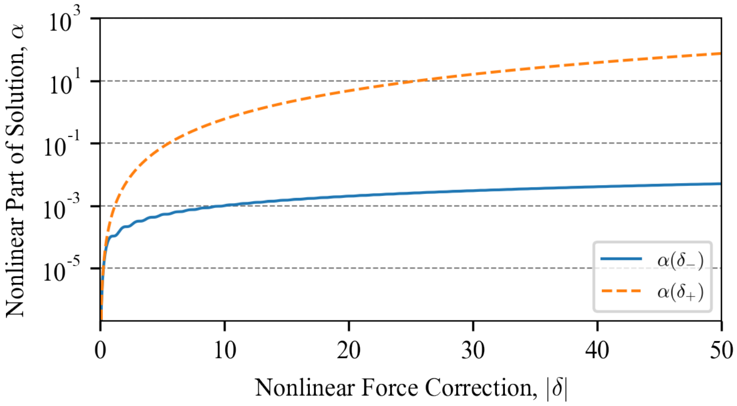

Figure 9: Nonlinear solution component α as a function of nonlinear force correction δ_- or δ_+ for C_0=9810.0 kg/s^2/m and m=256.3 kg/m.

δ_- is treated as a perturbation parameter, where a solution to Eq. (20) is expressed as a power series in terms of δ_- and the derivatives of the solution are simply derivatives of the expansion.

$$\alpha(\delta_-) = a_0 + a_1\delta_- + a_2\delta_-^2 + a_3\delta_-^3 + a_4\delta_-^4 + a_5\delta_-^5 + a_6\delta_-^6 + ... \tag{21}$$

Substituting Eq. (21) and its derivatives into Eq. (20), expressions for the coefficients a_n are found. Note that a_0 and a_1 correspond to the initial conditions for $\alpha(\delta_-(t = t_0)) = \alpha_0$ and $\dot{\alpha}(\delta_-(t = t_0)) = \dot{\alpha}_0$, respectively. Constructing the solution leads to Eq. (22), which is a closed-form function that describes the nonlinear part of the solution α as a function of the nonlinear force correction δ_-. Note that $\sqrt{C_0/m} = \omega_n$ is the natural frequency of the system and the function is only valid for $C_0 > 0$. Details of the solution process are given in Appendix A.

$$\alpha(\delta_-) = \alpha_0\left(1 + (\cos(\omega_n\delta_-) - 1)\right) + \frac{\delta_-}{C_0} + \omega_n m\left(\frac{\dot{\alpha}_0 C_0 - 1}{C_0^2}\right)\sin(\omega_n\delta) \tag{22}$$

To derive an expression for α when $C_0\alpha$ is included in δ_+ and not modeled separately, C_0 is set to zero in Eq. (20). The result is given by Eq. (23), where δ_+ includes the linear restoring force.

$$\alpha(\delta_+) = \alpha_0 + \dot{\alpha}_0\delta_+ + \frac{\delta_+^3}{3!}\frac{1}{m} \tag{23}$$

Selecting values for C_0 and m to match the values for the wedge and assuming $\alpha_0 = \dot{\alpha}_0 = 0$, the solution α for both Eq. (22) and Eq. (23) is shown in Fig. 9. The vertical axis is log-scaled to better show the difference between the two solutions. The results suggest that α will evolve much faster as a function of δ when restoring forces are included in the force correction. This means a numerical solution to Eq. (8) is more sensitive to errors in δ_+, as the solution may vary quickly and potentially lead to solution drift or possible instability.

4. CONCLUSIONS

The data-driven neural-corrector method of Marlantes and Maki (2022) is applied to predict nonlinear heave responses of a two-dimensional wedge in irregular waves. A machine learning model is used to learn a nonlinear force correction δ which is then included in a low-fidelity ordinary differential equation. The force correction δ is modeled as a function of k-length sequences of a low-fidelity state solution $\xi^{(l)}$ and the wave elevation η. The resulting hybrid equation of motion, which includes both analytical and data-driven terms, is solved numerically to yield improved-fidelity predictions of position ξ^*, velocity $\dot{\xi}^*$, and acceleration $\ddot{\xi}^*$. Training data are generated for different levels of nonlinear forces, including nonlinear hydrostatics and Froude-Krylov forces, and hydrodynamic memory effects, to explore how different physics present in ship hydrodynamics affect the performance and configuration of the neural-corrector method.

Several conclusions can be drawn from the work:

1. The presence of hydrodynamic memory effects requires a larger stencil size k. When only nonlinear hydrostatic and Froude-Krylov forces are present, smaller values of k are more accurate as the nonlinearity depends only on the instantaneous state.
2. Numerical and analytical results show that including the linear force terms, especially the hydrostatic restoring force $C_0\xi$, in the nonlinear force correction δ often results in large prediction errors in velocity and position. Therefore, linear forces should not be learned by the machine learning model, and instead modeled analytically.
3. The neural-corrector method is able to learn all four levels of nonlinear forces effectively, suggesting that the method could be used with more complex high-fidelity simulation data.

ACKNOWLEDGEMENTS

KEM would like to recognize the support of the University of Michigan Rackham Merit Fellowship (RMF) and the Rackham Travel Grant program.

AUTHOR'S CONTRIBUTION

KEM: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization. KJM: Supervision, Project administration, Funding acquisition, Writing - Review & Editing. PJB: Conceptualization, Methodology, Resources, Writing - Review & Editing, Supervision.

REFERENCES

Diez, M., Serani, A., Gaggero, M., & Campana, E. (2022, June). Improving knowledge and forecasting of ship performance in waves via hybrid machine learning methods. In *Proceedings of the 34th symposium on naval hydrodynamics, washington, dc, usa.*

Guo, X., Zhang, X., Tian, X., Lu, W., & Li, X. (2022). Probabilistic prediction of the heave motions of a semi-submersible by a deep learning model. *Ocean Engineering*, *247*, 110578.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(08), 1735-1780.

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization.* arXiv.

Liong, C., & Chua, K. (2022). Data assimilation for deterministic prediction of vessel motion in real-time. *Ocean Engineering*, *244*, 110356.

Marlantes, K. E., & Maki, K. J. (2022). A neural-corrector method for prediction of the vertical motions of a high-speed craft. *Ocean Engineering*, *262*, 112300.

Raissi, M., Perdikaris, P., & Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686-707.

Schirmann, M. L., Collette, M. D., & Gose, J. W. (2022). Data-driven models for vessel motion prediction and the benefits of physics-based information. *Applied Ocean Research*, *120*, 102916.

Silva, K. M., & Maki, K. J. (2022). Data-driven system identification of 6-dof ship motion in waves with neural networks. *Applied Ocean Research*, *125*, 103222.

Smith, T., & Silva, K. (2017, June). Linear seakeeping high sea state applicability. In *Proceedings of the 16th international ship stability workshop, belgrade, serbia.*

Wan, Z. Y., Vlachas, P., Koumoutsakos, P., & Sapsis, T. (2018, 05). Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLOS ONE*, *13*(5), 1-22.

Weems, K. M., & Wundrow, D. (2013). Hybrid models for the fast time-domain simulation of stability failures in irregular waves with volume based calculations for froude-krylov and hydrostatic forces. In *Proceedings of the 13th international ship stability workshop, brest, france.* ISSW.

Weymouth, G., & Yue, D. K. P. (2014, 1). Physics-based learning models for ship hydrodynamics. *Journal of Ship Research*, *57*.

Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2020). *Integrating physics-based modeling with machine learning: A survey.*

Xu, W. (2020). *A machine learning framework to model extreme events for nonlinear marine dynamics* (Unpublished doctoral dissertation). University of Michigan, Ann Arbor, MI.

Xu, W., Maki, K. J., & Silva, K. M. (2021). A data-driven model for nonlinear marine dynamics. *Ocean Engineering*, *236*, 109469.

Zhang, B., Wang, S., Deng, L., Jia, M., & Xu, J. (2023). Ship motion attitude prediction model based on iwoa-tcn-attention. *Ocean Engineering*, *272*, 113911.

Zhang, M., Taimuri, G., Zhang, J., & Hirdaris, S. (2023). A deep learning method for the prediction of 6-dof ship motions in real conditions. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 14750902231157852.

A. APPENDIX

Closed form expressions for the nonlinear part of the solution α as a function of the nonlinear force correction δ can be derived following a perturbation approach. The expressions show how excluding linear restoring force terms from δ leads to solutions which grow much slower for increasing values of δ than solutions obtained when including the terms. Consider a simplified equation of motion as given by Eq. (A1), where $F^{(l)}$ is the linear part of the force. Assuming δ_- is exact, the solution to Eq. (A1) yields the nonlinear state $\xi^{(h)}$.

$$m\ddot{\xi}^{(h)} + C_0\xi^{(h)} = F^{(l)} + \delta_- \tag{A1}$$

It is assumed that the total force can be decomposed into a linear and nonlinear part. Under this assumption, the state can also be decomposed into a linear $\xi^{(l)}$ and nonlinear α part as shown, where $F^{(l)}$ and δ_- are also brought to the left-hand-side of the equation. Because the linear equation satisfies the linear terms, the terms cancel and only the nonlinear component of the solution α remains, as shown by Eq. (A3).

$$m(\ddot{\xi}^{(l)} + \ddot{\alpha}) + C_0(\xi^{(l)} + \alpha) - F^{(l)} - \delta_- = 0 \tag{A2}$$

$$m\ddot{\alpha} + C_0\alpha - \delta_- = 0 \tag{A3}$$

δ_- is treated as a perturbation parameter, where a solution to Eq. (A3) is expressed as a power series in terms of δ_- and the derivatives of the solution are simply derivatives of the expansion.

$$\alpha(\delta_-) = a_0 + a_1\delta_- + a_2\delta_-^2 + a_3\delta_-^3 + a_4\delta_-^4 + a_5\delta_-^5 + a_6\delta_-^6 + ... \tag{A4}$$

$$\dot{\alpha}(\delta_-) = a_1 + 2a_2\delta_- + 3a_3\delta_-^2 + 4a_4\delta_-^3 + 5a_5\delta_-^4 + 6a_6\delta_-^5 + ... \tag{A5}$$

$$\ddot{\alpha}(\delta_-) = 2a_2 + 6a_3\delta_- + 12a_4\delta_-^2 + 20a_5\delta_-^3 + 30a_6\delta_-^4 + ... \tag{A6}$$

Substituting Eqs. (A4) though (A6) into Eq. (A3) and collecting like terms, expressions for the coefficients a_n are found. Note that a_0 and a_1 correspond to the initial conditions for $\alpha(\delta(t = t_0)) = \alpha_0$ and $\dot{\alpha}(\delta(t = t_0)) = \dot{\alpha}_0$, respectively. Constructing the solution leads to Eq. (A8), which can be expressed using the summation notation shown in Eq. (A9).

$$\alpha(\delta_-) = \alpha_0 - \frac{\delta_-^2}{2!}\frac{C_0\alpha_0}{m} + \frac{\delta_-^4}{4!}\frac{C_0^2\alpha_0}{m^2} - \frac{\delta_-^6}{6!}\frac{C_0^3\alpha_0}{m^3} + \frac{\delta_-^8}{8!}\frac{C_0^4\alpha_0}{m^4} - ... \tag{A7}$$

$$+ \dot{\alpha}_0\delta_- + \frac{\delta_-^3}{3!}\frac{1 - C_0\dot{\alpha}_0}{m} - \frac{\delta_-^5}{5!}\frac{C_0(1 - C_0\dot{\alpha}_0)}{m^2} + \frac{\delta_-^7}{7!}\frac{C_0^2(1 - C_0\dot{\alpha}_0)}{m^3} - \frac{\delta_-^9}{9!}\frac{C_0^3(1 - C_0\dot{\alpha}_0)}{m^4} + ... \tag{A8}$$

$$= \alpha_0 + \sum_{n=1}^{\infty}\frac{(-1)^n\delta_-^{2n}}{(2n)!}\frac{C_0^n\alpha_0}{m^n} + \dot{\alpha}_0\delta_- - \sum_{n=1}^{\infty}\frac{(-1)^n\delta_-^{2n+1}}{(2n+1)!}\frac{C_0^{n-1}(1 - C_0\dot{\alpha}_0)}{m^n} \tag{A9}$$

The first line corresponds closely with the power series representation of cosine and the second line corresponds closely to that of sine. Also note that $\sqrt{C_0/m} = \omega_n$, which is the natural frequency of the system. Making these substitutions and simplifying results in Eq. (A10), which is a closed-form function that describes the nonlinear part of the solution α as a function of the nonlinear force correction δ_-. Note that this function is only valid for $C_0 > 0$.

$$\alpha(\delta_-) = \alpha_0 \left(1 + (\cos(\omega_n \delta_-) - 1)\right) + \frac{\delta_-}{C_0} + \omega_n m \left(\frac{\dot{\alpha}_0 C_0 - 1}{C_0^2}\right) \sin(\omega_n \delta_-) \tag{A10}$$

If the body starts from rest, $\alpha_0 = \dot{\alpha}_0 = 0$, and the first term in Eq. (A10) vanishes. The solution simplifies to Eq. (A11).

$$\alpha(\delta_-) = \frac{\delta_-}{C_0} - \frac{\omega_n m}{C_0^2} \sin(\omega_n \delta_-) \tag{A11}$$

To derive an expression for α when $C_0\alpha$ is included in δ_+ and not modeled separately, C_0 is set to zero in Eq. (A3) and the process repeated such that only three non-zero terms remain, as shown in Eq. (A12).

$$\alpha(\delta_+) = \alpha_0 + \dot{\alpha}_0 \delta_+ + \frac{\delta_+^3}{3!} \frac{1}{m} \tag{A12}$$

Again assuming $\alpha_0 = \dot{\alpha}_0 = 0$, the expression simplifies to a cubic expression in terms of δ_+.

B. APPENDIX

The original intent of the neural-corrector method is to use a data-driven model to learn the component of the total force which cannot be represented using linear coefficients. This leads to the following minimization problem: minimize $\|\delta\|_2$ for a given high-fidelity solution $\ddot{\xi}^{(h)}$, $\dot{\xi}^{(h)}$, $\xi^{(h)}$, and low-fidelity force model $F^{(l)}$, by choosing optimal values for A_0^*, B_0^* and C_0^*. Here, $\|\cdot\|_2$ is the 2-norm defined by Eq. (B1), which also relates to the total energy in the signal δ. The vector notation here indicates a vector in time. The cost function is given by Eq. (B3).

$$\|\delta\|_2 = \sqrt{\vec{\delta}\cdot\vec{\delta}} \tag{B1}$$

$$f(A_0, B_0, C_0) \rightarrow \ \min \tag{B2}$$

$$f(A_0, B_0, C_0) = \|\delta\|_2 \tag{B3}$$

Substituting Eq. (7) into Eq. (B1), the cost function can be expanded to yield Eq. (B4), where the expression under the radical will be called Ω and is given by Eq. (B5). The superscript (h) is omitted from the responses $\ddot{\xi}$, $\dot{\xi}$, and ξ for brevity.

$$f(A_0, B_0, C_0) = \Omega^{1/2} \tag{B4}$$

$$\begin{aligned}\Omega = A_0^2\vec{\ddot{\xi}}\cdot\vec{\ddot{\xi}} + 2A_0B_0\vec{\dot{\xi}}\cdot\vec{\ddot{\xi}} + 2A_0C_0\vec{\xi}\cdot\vec{\ddot{\xi}} - 2A_0\vec{\mathbf{F}}^{(l)}\cdot\vec{\ddot{\xi}} \\ + B_0^2\vec{\dot{\xi}}\cdot\vec{\dot{\xi}} + 2B_0C_0\vec{\xi}\cdot\vec{\dot{\xi}} - 2B_0\vec{\mathbf{F}}^{(l)}\cdot\vec{\dot{\xi}} \\ + C_0^2\vec{\xi}\cdot\vec{\xi} - 2C_0\vec{\mathbf{F}}^{(l)}\cdot\vec{\xi} + \vec{\mathbf{F}}^{(l)}\cdot\vec{\mathbf{F}}^{(l)}\end{aligned} \tag{B5}$$

The 2-norm is strictly convex, which means a gradient-based method such as Newton's iteration is appropriate. The update equation is given by Eq. (B6) where $\vec{\mathbf{X}}^k$ is a vector of the coefficients A_0^k, B_0^k, and C_0^k for the k'th iteration, $\boldsymbol{\nabla} f$ is the gradient of the cost function, $\mathbf{H}$ is the Hessian matrix of f, and α^k is an over-relaxation parameter.

$$\vec{\mathbf{X}}^{k+1} = \vec{\mathbf{X}}^k - \alpha^k\mathbf{H}(\vec{\mathbf{X}}^k)^{-1}\cdot\boldsymbol{\nabla} f(\vec{\mathbf{X}}^k) \tag{B6}$$

Expressions for the gradient and Hessian can be evaluated analytically by taking partial derivatives of Eq. (B4) and are given as follows:

$$\boldsymbol{\nabla} f = \begin{Bmatrix} \frac{\partial f}{\partial A_0} \\ \frac{\partial f}{\partial B_0} \\ \frac{\partial f}{\partial C_0} \end{Bmatrix} = \frac{1}{2}\Omega^{-1/2}\begin{Bmatrix} \frac{\partial \Omega}{\partial A_0} \\ \frac{\partial \Omega}{\partial B_0} \\ \frac{\partial \Omega}{\partial C_0} \end{Bmatrix}, \ \mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial A_0^2} & \frac{\partial^2 f}{\partial A_0 \partial B_0} & \frac{\partial^2 f}{\partial A_0 \partial C_0} \\ \frac{\partial^2 f}{\partial B_0 \partial A_0} & \frac{\partial^2 f}{\partial B_0^2} & \frac{\partial^2 f}{\partial B_0 \partial C_0} \\ \frac{\partial^2 f}{\partial C_0 \partial A_0} & \frac{\partial^2 f}{\partial C_0 \partial B_0} & \frac{\partial^2 f}{\partial C_0^2} \end{bmatrix} \tag{B7}$$

$$\frac{\partial^2 f}{\partial A_0^2} = \frac{1}{2}\Omega^{-1/2}\frac{\partial^2 \Omega}{\partial A_0^2} - \frac{1}{4}\Omega^{-3/2}\frac{\partial \Omega}{\partial A_0}\frac{\partial \Omega}{\partial A_0} \tag{B8}$$

$$\frac{\partial^2 f}{\partial A_0 \partial B_0} = \frac{1}{2}\Omega^{-1/2}\frac{\partial^2 \Omega}{\partial A_0 \partial B_0} - \frac{1}{4}\Omega^{-3/2}\frac{\partial \Omega}{\partial B_0}\frac{\partial \Omega}{\partial A_0} \tag{B9}$$

$$\frac{\partial^2 f}{\partial A_0 \partial C_0} = \frac{1}{2}\Omega^{-1/2}\frac{\partial^2 \Omega}{\partial A_0 \partial C_0} - \frac{1}{4}\Omega^{-3/2}\frac{\partial \Omega}{\partial C_0}\frac{\partial \Omega}{\partial A_0} \tag{B10}$$

$$\frac{\partial^2 f}{\partial B_0 \partial A_0} = \frac{\partial^2 f}{\partial A_0 \partial B_0} \tag{B11}$$

$$\frac{\partial^2 f}{\partial B_0^2} = \frac{1}{2}\Omega^{-1/2}\frac{\partial^2 \Omega}{\partial B_0^2} - \frac{1}{4}\Omega^{-3/2}\frac{\partial \Omega}{\partial B_0}\frac{\partial \Omega}{\partial B_0} \tag{B12}$$

$$\frac{\partial^2 f}{\partial B_0 \partial C_0} = \frac{1}{2}\Omega^{-1/2}\frac{\partial^2 \Omega}{\partial B_0 \partial C_0} - \frac{1}{4}\Omega^{-3/2}\frac{\partial \Omega}{\partial C_0}\frac{\partial \Omega}{\partial B_0} \tag{B13}$$

$$\frac{\partial^2 f}{\partial C_0 \partial A_0} = \frac{\partial^2 f}{\partial A_0 \partial C_0} \tag{B14}$$

$$\frac{\partial^2 f}{\partial C_0 \partial B_0} = \frac{\partial^2 f}{\partial B_0 \partial C_0} \tag{B15}$$

$$\frac{\partial^2 f}{\partial C_0^2} = \frac{1}{2}\Omega^{-1/2}\frac{\partial^2 \Omega}{\partial C_0^2} - \frac{1}{4}\Omega^{-3/2}\frac{\partial \Omega}{\partial C_0}\frac{\partial \Omega}{\partial C_0} \tag{B16}$$

$$\frac{\partial \Omega}{\partial A_0} = 2A_0\vec{\ddot{\xi}} \cdot \vec{\ddot{\xi}} + 2B_0\vec{\dot{\xi}} \cdot \vec{\ddot{\xi}} + 2C_0\vec{\xi} \cdot \vec{\ddot{\xi}} - 2\vec{\mathbf{F}} \cdot \vec{\ddot{\xi}} \tag{B17}$$

$$\frac{\partial \Omega}{\partial B_0} = 2A_0\vec{\dot{\xi}} \cdot \vec{\ddot{\xi}} + 2B_0\vec{\dot{\xi}} \cdot \vec{\dot{\xi}} + 2C_0\vec{\xi} \cdot \vec{\dot{\xi}} - 2\vec{\mathbf{F}} \cdot \vec{\dot{\xi}} \tag{B18}$$

$$\frac{\partial \Omega}{\partial C_0} = 2A_0\vec{\xi} \cdot \vec{\ddot{\xi}} + 2B_0\vec{\xi} \cdot \vec{\dot{\xi}} + 2C_0\vec{\xi} \cdot \vec{\xi} - 2\vec{\mathbf{F}} \cdot \vec{\xi} \tag{B19}$$

$$\frac{\partial^2 \Omega}{\partial A_0^2} = 2\vec{\ddot{\xi}} \cdot \vec{\ddot{\xi}} \tag{B20}$$

$$\frac{\partial^2 \Omega}{\partial A_0 \partial B_0} = 2\vec{\dot{\xi}} \cdot \vec{\ddot{\xi}} \tag{B21}$$

$$\frac{\partial^2 \Omega}{\partial A_0 \partial C_0} = 2\vec{\xi} \cdot \vec{\ddot{\xi}} \tag{B22}$$

$$\frac{\partial^2 \Omega}{\partial B_0^2} = 2\vec{\dot{\xi}} \cdot \vec{\dot{\xi}} \tag{B23}$$

$$\frac{\partial^2 \Omega}{\partial B_0 \partial C_0} = 2\vec{\xi} \cdot \vec{\dot{\xi}} \tag{B24}$$

$$\frac{\partial^2 \Omega}{\partial C_0^2} = 2\vec{\xi} \cdot \vec{\xi} \tag{B25}$$