Learning nonlinear operators via LSTM neural networks for predicting vessel dynamics in extreme sea states

A Preprint

José del Águila Ferrandis*
Department of Mechanical Engineering
Massachusetts Institute of Technology
jaguila@mit.edu

George Karniadakis
Division of Applied Mathematics,
Brown University
george_karniadakis@brown.edu

Chryssostomos Chryssostomidis
Department of Mechanical Engineering
Massachusetts Institute of Technology
chrys@mit.edu

Michael Triantafyllou
Department of Mechanical Engineering
Massachusetts Institute of Technology
mistetri@mit.edu

February 3, 2021

Abstract

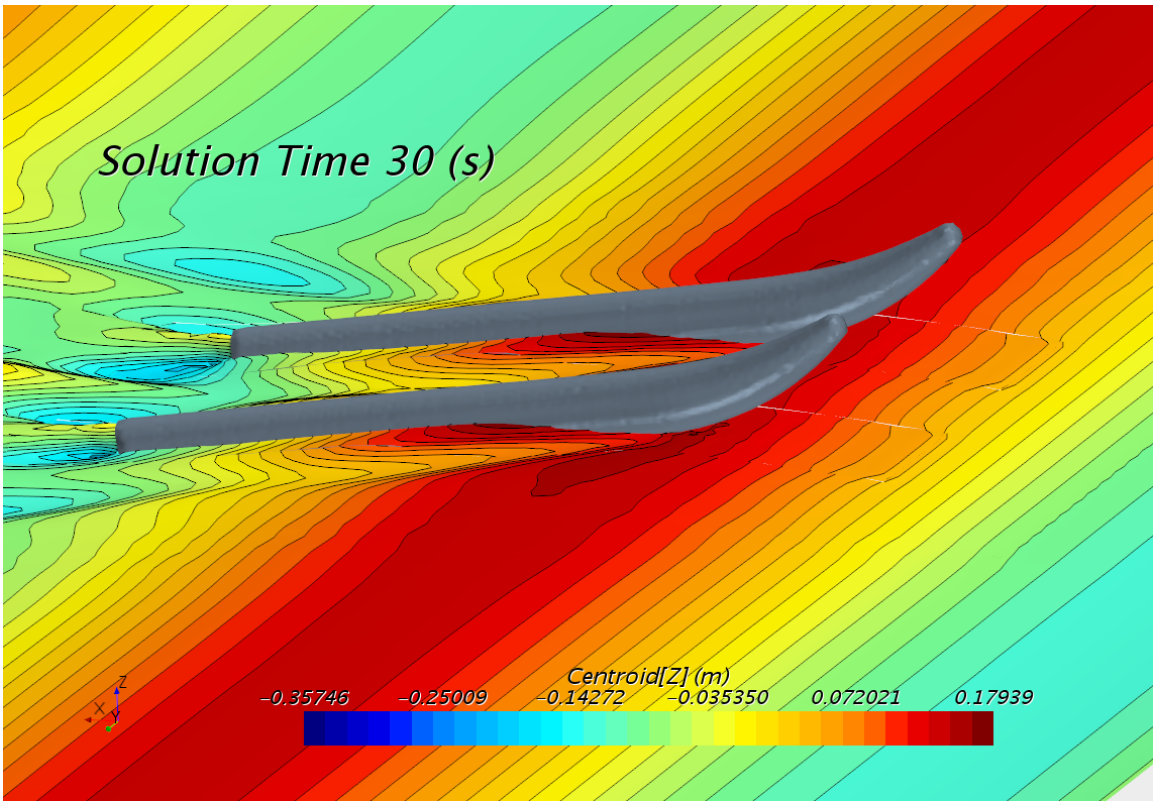
Predicting motions of vessels in extreme sea states represents one of the most challenging problems in naval hydrodynamics. It involves computing complex nonlinear wave-body interactions, hence taxing heavily computational resources. Here, we put forward a new simulation paradigm by training recurrent type neural networks (RNNs) that take as input the stochastic wave elevation at a certain sea state and output the main vessel motions, e.g., pitch, heave and roll. We first compare the performance of standard RNNs versus GRU and LSTM neural networks (NNs) and show that LSTM NNs lead to best performance. We then examine the testing error of two representative vessels, a catamaran in sea state 1 and a battleship in sea state 8. We demonstrate that good accuracy is achieved for both cases in predicting the vessel motions for unseen wave elevations. We train the NNs with expensive CFD simulations offline *but upon training, the prediction of the vessel dynamics* online *can be obtained at a fraction of a second. This work is motivated by the universal approximation theorem for nonlinear functionals and operators [**?**], and it is the first implementation of such theory to realistic engineering problems.*

Keywords LSTM Neural Networks · Extreme sea states · Seakeeping · Nonlinear Operators

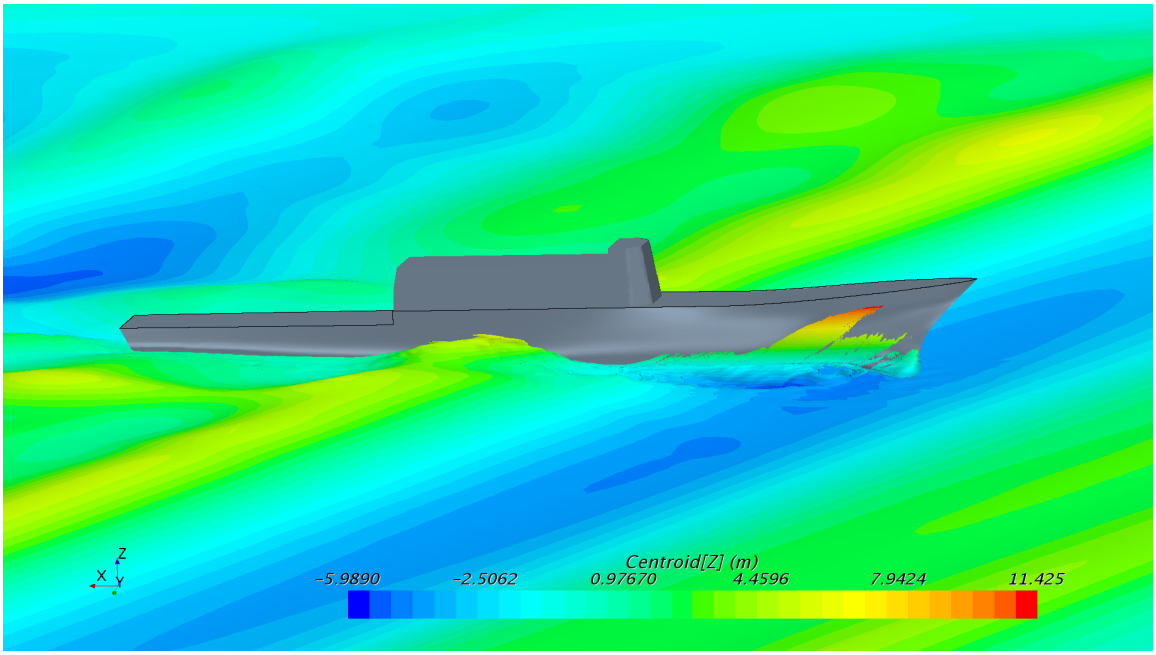
1 Brief Discussion of the Physical Problem

Predicting the motion of vessels in nonlinear waves constitutes one of the most challenging problems in naval hydrodynamics. A complete solution of the seakeeping problem involves resolving complex nonlinear wave-body interactions that results in large computational costs. For this reason, over the years many seakeeping models have been formulated in order to predict ship motions using simplified flow theories [10–14], usually based on potential flow theories solved within the linear assumption, which limits their range of applicability in cases of practical interest. Furthermore, second-order potential-flow solvers [15–25] represent the state-of-the-art numerical methods for wave-induced response of vessels and other large-volume marine structures. Important viscous damping effects such required in modeling the rolling of ships and slow-drift motions of moored structures are accounted for by empirical formulas.

*Personal webpage: jaguila.mit.edu



(a) Catamaran sailing in regular 5th order Stokes Waves. This constitutes a relatively easy condition to approximate. Potential methods can be used for such a problem.



(b) Notional DTMB battleship sailing in WMO sea state 8 at high Froude Number (0.4). This condition is beyond the operability limits of the vessel. However, it is of great interest, since it corresponds to strong nonlinear motions and can be challenging for the neural networks. Viscous CFD solvers have to be used to account for nonlinearities, leading to prohibitive computational costs. Meshes for these cases had more than 20 million elements and required several days to process on a parallel co,mputer with 300 physical cores.

Figure 1: *Snapshots of the URANS simulations.*

Neglecting viscous effects might largely underestimate the energy dissipated by the system while moving at the free surface. This problem is particularly relevant for unconventional floating bodies at resonance, when waves are steep and nonlinear and when the motions of the vessels are large. For this reason we use a fully viscous model URANS solver [26–33]. A drawback is that URANS depends on empirical modelling of turbulence. Motion predictions obtained using fully viscous models are limited by the massive computational costs required. The approach taken here can be aimed at reducing the amount of simulation that is necessary to characterize the operating conditions that the vessel/platform will encounter during its lifespan.

The data set is obtained from a viscous Volume of Fluid URANS solver (STAR-CCM+). The equations solved are the averaged continuity and momentum equations for incompressible fluids where there are no body-forces.

$$\frac{\partial(\rho\overline{u}_i)}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial(\rho\overline{u}_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho\overline{u}_i\overline{u}_j + \rho\overline{u_i'u_j'}) = -\frac{\partial\overline{p}}{\partial x_i} + \frac{\partial\overline{\tau}_{ij}}{\partial x_j} \tag{2}$$

$$\overline{\tau}_{ij} = \mu\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right) \tag{3}$$

where $\overline{\tau}_{ij}$, in equation 2, are the components of the averaged viscous force tensor, $\overline{p}$ is the averaged pressure and $\overline{u}$ are the cartesian components of the averaged velocity. In equation 2, $\overline{u_i'u_j'}$ are the Reynolds stresses, ρ the fluid density, and μ the dynamic viscosity.

To model the free surface, the time-domain fully viscous model uses a Volume of Fluid method (VOF) [32]. This model assumes that the same equations governing the physics of one of the phases can be solved for all phases present in the computational domain (each cell or finite volume). A good reference for the theory behind this type of numerical method can also be found in [26].

In order to simulate the behavior and to obtain realistic platform motions, a Dynamic Fluid Body Interaction (DFBI) model is used. The vessel is allowed to move in two (catamaran vessel) or three (DTMB) degrees of freedom. It is allowed to translate in the vertical direction (heave), to rotate around the transversal direction (pitch) and to rotate around the longitudinal direction (roll).

1.1 Definition of Irregular Long Crested Seas

Ocean waves used to induce motions in the vessels are reconstructed from experimental sea spectra (see Fig. 2) that characterizes the stochastic process of sea surface elevations. At a particular spatial location, let $\xi(t)$ be the sea surface elevation as a function of time. Then, this time signal can be defined as a sum of sinusoidal waves with random phases ε_i taken from a uniform distribution:

$$\xi(t) = \sum_{i=1}^{n} a_i \cos\left(\omega_i t + \varepsilon_i\right) \tag{4}$$

The wave amplitude for a given frequency is obtained from the following relation,

$$\frac{1}{2}a_i^2 = S\left(\omega_i\right)\Delta\omega \tag{5}$$

where $S\left(\omega\right)$ is the modified Pierson-Moskowitz spectrum **[?]** with T_1 as the mean wave period:

$$\frac{S(\omega)}{H_s^2 T_1} = \frac{0.11}{2\pi}\left(\frac{\omega T_1}{2\pi}\right)^{-5}\exp\left[-0.44\left(\frac{\omega T_1}{2\pi}\right)^{-4}\right] \tag{6}$$

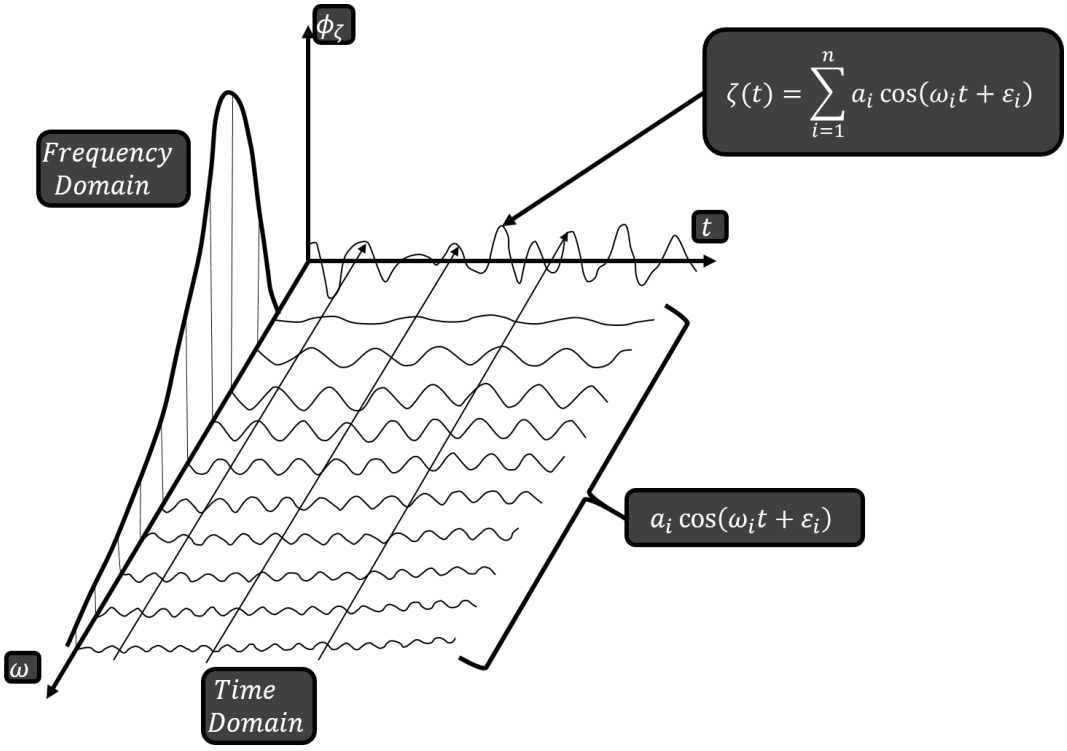


Figure 2: *Graphical description of the spectral analysis performed to characterize sea states.*

where

$$Y = \exp\left[-\left(\frac{0.191\omega T_1 - 1}{2^{1/2}\sigma}\right)^2\right] \tag{7}$$

and

$$\sigma = \begin{cases} 0.07 & \text{if } \omega \leq 5.24/T_1 \\ 0.09 & \text{if } \omega > 5.24/T_1 \end{cases} \tag{8}$$

To impose the initial conditions in the domain the velocity and pressure fields are calculated as a superposition of the individual regular waves (see Table 1). These are integrated in time by the numerical scheme which also tracks nonlinear interactions with the vessels.

The sea states modelled vary for each vessel. In both cases we use the Pierson-Moskowitz spectrum to generate irregular long crested seas. The catamaran's sea state is: $H_s = 0.3m$ and $T_p = 1.48s$. Where H_s is the significant wave height and T_s is the peak wave period. The DTBM's vessel sea state is $H_s = 10.66m$ and $T_s = 13.4s$, which corresponds to a WMO Sea State Code 8.

2 Approximation of Continuous Functionals by Neural Networks & Application to Dynamical Systems

Our approach is inspired by *Approximations of Continuous Functionals by Neural Networks with Application to Dynamical Systems* [1], which proves that NNs can approximate arbitrarily well continuous functionals and operators. Some good references on these results are [34–38]. LSTM-RNNs provide robust modeling for complex dynamic physical systems. Nevertheless, the data analyzed cannot be guaranteed to comply with all the assumptions under which the theorems presented in [1] are proven. Specifically, the continuity of the functions being fed to the functional that models the physical system, giving reason to believe that these results can be further generalized **XXX - what do you mean here? - Actually, In one of our**

Variable	Formula
Wave Profile.	$\xi(x,t) = \sum_{n=1}^{N} a_n cos[k_n(x - c_n t) + \varepsilon_n]$
Horizontal Velocity	$u(x,y,t) = \sum_{n=1}^{N} \frac{a_n \omega_n}{sinh(k_n h)} cosh[k_n(y+h)] cos[k_n(x - c_n t) + \varepsilon_n]$
Vertical Velocity	$v(x,y,t) = \sum_{n=1}^{N} \frac{a_n \omega_n}{sinh(k_n h)} sinh[k_n(y+h)] sin[k_n(x - c_n t) + \varepsilon_n]$
Horizontal Velocity	$\dot{u}(x,y,t) = \sum_{n=1}^{N} \frac{a_n \omega_n}{cosh(k_n h)} cosh[k_n(y+h)] sin[k_n(x - c_n t) + \varepsilon_n]$
Vertical Acceleration	$\dot{v}(x,y,t) = \sum_{n=1}^{N} \frac{a_n \omega_n}{sinh(k_n h)} cosh[k_n(y+h)] cos[k_n(x - c_n t) + \varepsilon_n]$
Dynamic Pressure	$p(x,y,t) = \sum_{n=1}^{N} \frac{a_n \rho g}{sinh(k_n h)} cosh[k_n(y+h)] cos[k_n(x - c_n t) + \varepsilon_n]$

Table 1: Formulation for irregular long crested seas, generalizing the spatial location x.

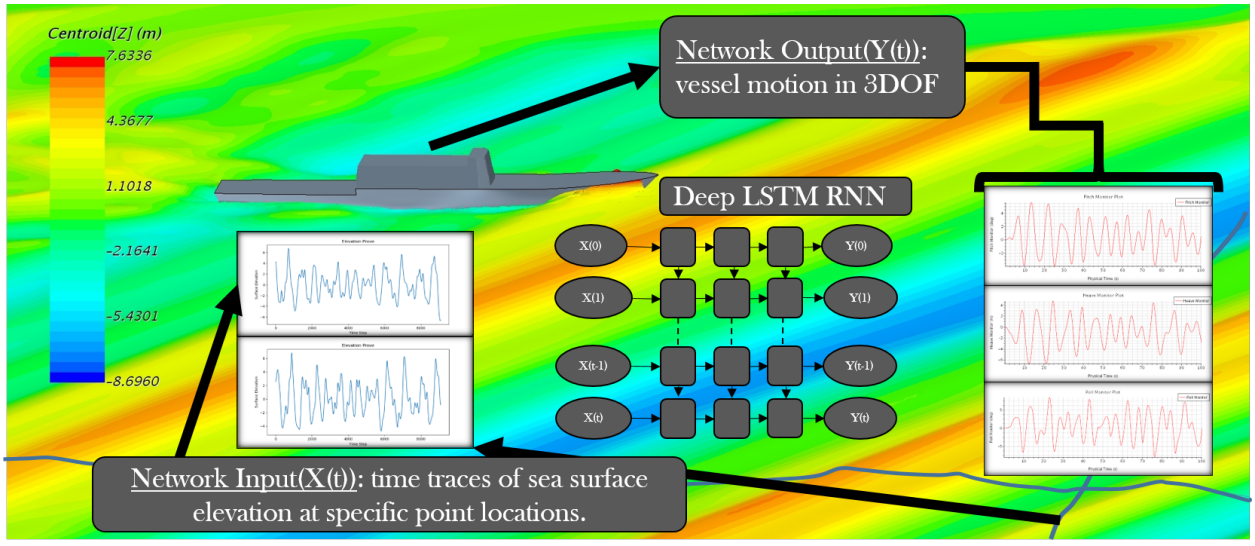


Figure 3: *Physical problem simulated; Inputs and outputs of the deep LSTM RNN.* Throught the paper motions of different vessels are approximated by reading information about phenomena that induce the motions (wave) and training LSTM RNNs to approximate vessel motions given this information. Sea surface elevations are recorded at specific point locations that can be chosen from lines over the free surface. Vessel motions in the training cases are obtained from an unsteady URANS solver. In the test cases only information about the sea surface is available.

discussions you mentioned that NN can also represent discontinuous functions if they can be defined piece-wise (or something similar). I was actually thinking about something similar here, fluid-structure interaction in our problem happens at different flow regimes that I believe could be defined piece-wise. Has this been proven for functionals as it has been proven for functions?.

The results presented in [1] can be summarized as follows. Given very mild conditions, a functional defined on a compact set in $C[a,b]$ or $L^p[a,b]$ (spaces of infinite dimensions) can be approximated arbitrarily well by a neural network with just one hidden layer. Particularly, given U a compact set in $C[a,b]$, σ (a bounded sigmoidal function) and f a continuous functional defined on U, then $\forall u \in U, f(u)$ can be approximated by

$$f(u) = \sum_{i=1}^{N} c_i \sigma \left(\sum_{j=0}^{m} \xi_{i,j} u\left(x_j\right) + \theta_i \right) \tag{9}$$

In the above expression, c_i, ξ_{ij}, θ_i are real numbers and $u(x_j)$ is the value that u takes at x_j.

The stated theorem and the explicit expression provided can have a very wide impact on the foreseeable applications of neural networks to model dynamical systems. We have adopted this framework and tested it on realistic applications in this paper for first time.

In the application, we view the output of the dynamical system (vessels with unsteady motions) as a functional of a forcing term (ocean waves). The results presented suggest that the joint application of dynamic physical models and RNN already lead to large computational savings when the physical models are complex to simulate. For the cases simulated in this paper, given the desired accuracy, we have seen that we could have saved 50% of the computational time, spent on producing the dataset that is designed to characterize the motions of the vessel. Furthermore, the obtained surrogate model can potentially be used for online approximation of real vessel motions to ensure operability and safety in extreme weather conditions.**xxx - what do you mean? - The model can potentially be a digital twin of the vessel and it can be used as such. Should I rephrase?**

The two degree-of-freedom (2DOF) (heave & pitch) motions of a catamaran vessel are modeled, at first, for nonlinear 5th-order regular waves (some results are presented in fig. 4) and, secondly, for irregular stochastic waves that represent real life sea-states (some results presented in fig. 6). Lastly, the three degrees-of-freedom (heave, pitch & roll) motion approximation of a notional DTMB battleship has been successfully performed (figs. 9 to 12). In doing this, we believe that we have achieved a state-of-the -art generalization of motions that are largely governed by the Navier-Stokes equations. A brief discussion of the physics is presented in subsection 1, and more details can be found in [7].

2.1 Representation of Dynamical Systems with Functionals

The extension of expression 2 to dynamical systems is remark (from [1]) here to present the key aspects that guarantee the Learnability of these systems.

The *first* assumption is that the dynamical system subject to study can be modeled with a continuous functional, defined in a compact set.

The *second* assumption is that the map that is used to represent functionals for the dynamical system can be performed with a windowing operator. Stating this in a formal way, we assume that X_1 and X_2 are sets in $\mathbf{R}^{q_1}$ and $\mathbf{R}^{q_2}$ -valued functions defined in $\mathbf{R}^n$. The dynamical system we study is viewed as map from X_1 to X_2, such that $\forall\ u \in X_1$, $Gu = v \in X_2$. We can define an n-dimensional windowing operator W for $x \in X$ centered at α and with width $2a$:

$$\left(W_{\alpha,a} x\right)(\beta) = \begin{cases} x(\beta) & \text{if } \beta \in \Gamma_{\alpha,a} \\ 0 & \text{if } \beta \notin \Gamma_{\alpha,a} \end{cases}$$

Using this window operator, we can restrict a non empty set U of X_1, $U_{\alpha,a} = \left\{ u|_{\Gamma_{\alpha,a}}, u \in U \right\}$. This reads as, $u|_{\Gamma_{\alpha,a}}$ is the restriction of u to $\Gamma_{\alpha,a}$.

We consider that a map G from X_1 to X_2 is of *approximately finite memory* if $\forall \epsilon > 0$ there is $a > 0$ such that:

$$\left| (Gu)_j(\alpha) - \left(GW_{\alpha,a} u\right)_j(\alpha) \right| < \epsilon, \quad j = 1, \ldots, q_2 \quad \forall \alpha \in \mathbf{R}^n, u \in U$$

The above statement has a very deep impact on the learnability of the dynamical process. It limits the extension of the proof of approximation of functional to those dynamical systems where we do not need to know all the history of states in order to approximate it. Furthermore, to give an explicit expression of the functional, the following assumptions are necessary:

1. If $u \in U$, then $u|_{\Gamma_{\alpha,a}} \in U$, $\forall \alpha \in \mathbf{R}^n, a > 0$.
2. $\forall\ \alpha \in \mathbf{R}^n, a > 0, U_{\alpha,a}$ is a compact set in $C_V\left(\prod_{k=1}^n [\alpha_k - a_k, \alpha_k + a_k]\right)$ or a compact set in $L_V^p\left(\prod_{k=1}^n [\alpha_k - a_k, \alpha_k + a_k]\right)$, where V stands for $\mathbf{R}^{q_1}$.
3. Then, if we let $(Gu)(\alpha) = \Big((Gu)_1(\alpha), \ldots, (Gu)_{q_2}(\alpha)\Big)$, consequently each $(Gu)_j(\alpha)$ will be a continuous functional defined over $U_{\alpha,a}$, with the corresponding topology in $C_V\left(\prod_{k=1}^n [\alpha_k - a_k, \alpha_k + a_k]\right.$ or $L_V^p\left(\prod_{k=1}^n [\alpha_k - a_k, \alpha_k + a_k]\right)$.

Given the above conclusions and following the process given in reference [1], we can find the extension of the explicit expression 2 to dynamical systems in the following theorem:

Theorem: If U and G satisfy all the assumptions (1-3) made previously, and G is of *approximately finite memory*, then $\forall \epsilon > 0$, $\exists a > 0$, m a positive integer, $(m+1)^n$ points in $\prod_{k=1}^n [\alpha_k - a_k, \alpha_k + a_k]$, N a positive integer, constants $c_i(G, \alpha, a)$ that only depend on G, α, a, and $q_2 \times (m+1)^n$ - vectors ξ_i, $i = 1, \ldots, N$, such that:

$$\left| (Gu)_j(\alpha) - \sum_{i=1}^{N} c_i(G, \alpha, a)\sigma\left(\overline{\xi}_i \cdot \overline{u}_{q_1,n,m} + \theta_i\right)\right| < \epsilon, \quad j = 1, 2, \ldots, q_2$$

To conclude, we would like to put an emphasis on the assumption of *approximately finite memory*. This assumption provides the blocks to build the functional approximation as a sum of functionals defined in the subsets given by the window operator, previously defined. During the empirical analysis, we benchmark against a case where we hope that *approximately finite memory* will allow representing the functional arbitrarily well from the subsets given by the window operator.

3 A Brief Overview of the Machine Learning Algorithm

RNNs are a generalization of feedforward neural networks that provides a greater flexibility when storing key aspects of sequences. Furthermore, they are able to maintain a notion of *state*. This *state* evolves as a larger fragment of the sequence is 'seen' by the network. Furthermore, we compute and/or predict sequences considering this *state* with some information that becomes available to the network. Current research has proven RNNs to be powerful but difficult to train. The main difference between feed-forward neural networks and RNNs is that RNNs use the same functional mapping to transform the previous state, providing parameter sharing. Since basic RNNs suffer from vanishing gradient problems, and the time series used in this paper are quite long, we have to use gated units, i.e., LSTM and GRU cells. The gates in these units are used to control the flow of information that passes through the state of the cell. For the first part, after a convergence study of the properties of the networks (Fig 4), we have found LSTMs to be more expressive. Feed-forward architectures such as NARX (Non-linear Auto-Regressive with Exogenous input) have been tested. However, they have been discarded because they are unable to 'forget', which results in an ever increasing residual error that propagates and compromises the performance of the NN when predicting long sequences.

The chosen type of gated cell (LSTM) was introduced two decades ago [5] and has now gained popularity in the context of language modeling. However, we believe its potential regarding time series modeling is relaively underappreciated at the momenent.

The formulation of LSTM cells is as follows. Let c_t the internal memory and h_t the visible state. Consequently, the state that evolves is the pair $[c_t, h_t]$. The forget gate f_t, as its name indicates, controls what to forget from the memory cell. The input gate i_t, controls what to read out of the memory cell into the visible state h_t, via the output gate o_t. Finally, the memory cell c_t evolves by the addition of inputs that come from the forget and input gates.

$$f_t = \sigma\left(W^{f,h}h_{t-1} + W^{f,x}x_t\right) \quad \text{(forget gate)}$$

$$i_t = \sigma\left(W^{i,h}h_{t-1} + W^{i,x}x_t\right) \quad \text{(input gate)}$$

$$o_t = \sigma\left(W^{o,h}h_{t-1} + W^{o,x}x_t\right) \quad \text{(output gate)}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_h\left(W^{c,h}h_{t-1} + W^{c,x}x_t\right) \quad \text{(memory cell)}$$

$$h_t = o_t \odot \sigma_h\left(c_t\right) \quad \text{(visible state)}$$

$$p_t = \text{softmax}\left(W^o h_t\right)$$

The objective function to minimize during the training is the mean squared error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{10}$$

where Y is the vector of observed values and $\hat{Y}$ are the predicted values.

4 Nonlinear Functional Approximation for Modeling Nonlinear Motions

4.1 Catamaran Vessel

Regarding the nonlinear functional approximation for predicting the motion dynamics, we have designed a LSTM network that approximates 2DOF or 3DOF motions of vessels advancing in irregular waves in head or oblique seas. First, we start with simpler Stokes 5th-order waves (results in figure 4). At this point of the experiments, we develop benchmark cases to choose between the two most used RNN cells. In figure 4, by comparing plots in subfigures 4a - 4d to subfigures 4e - 4h, we can clearly see that LSTM networks are more expressive for this particular application. Moreover, in figures 4g and 4h we see that the GRU cells are only able to capture a regular trend.

Having chosen the type of RNN cell, we have tested our model capabilities with stochastic waves modeling typical sea states that the catamaran vessel may encounter. We have found that the pitch angular motion is modeled better than the vertical motion. The next step that we take is to optimize the architecture of the network augmenting its expressiveness without inducing data over-fitting.

The results of this study are presented in figures 6 and 7. In sub-figures 6e - 6h the amount of data available to the network is incremented in a parametric analysis. We note that the final size of the training data set has been limited by the computational resources available. It can be seen that the growth of available training data has somehow prevented the RNN from overfitting. In 6h we can see how the training MSE stabilizes around 10^{-12}, contrary to the other cases that march on overfitting the data and not improving results on test cases, very mildly though.

In figure 7, a subset of 64 parametric variations of 5 network architecture parameters is presented. These parameters are: number of hidden units, number of hidden layers, training steps, number of sequences in the training process and fraction of anyone sequence used in the training process. In the **first** 6 plots, sub-figures 7a - 7f, we analyze the evolution of motion predictions when we vary the fraction of sequence used in the training process (7a $\rightarrow$ 7c or 7d $\rightarrow$ 7f) or when we vary the number of hidden layers (7a $\rightarrow$ 7d or 7b $\rightarrow$ 7e or 7c $\rightarrow$ 7f). The **second** set of subplots (7d - 7l) is similar to the **first** set, the only change is the variable in the plots. Instead of the vertical motion (heave) we plot the angular motion (Pitch). The main trends observed are the following:

1. *First*, as we decrease the amount of information given to the network (compare column 7a $\downarrow$ 7j) to 7c $\downarrow$ 7l) we can clearly see that the network is less accurate and is more prone to over-fitting.
2. *Secondly*, over-fitting is emphasized in subplots 7f and 7l, which is natural since they are the cases with greatest number of parameters, because they have 3 hidden layers. So as we increase the expressive power of the network we must also increase the amount of information given for training.

The conclusions of the network architecture convergence are the following:

1. Given the amount of data available, 1 hidden layer with 15 hidden units approximates accurately the functional of the vessels motions.

2. Given enough data these predictions can be improved with more expressive networks, however, these networks are also quite prone to overfitting. This can be concluded since accuracy in figures 7d and 7j is better or equal than in 7a and 7g; however, it is clearly worse when we reduce the amount of information in the training cases. To see this, one can see that accuracy in figures 7f and 7l is worse than in 7c and 7i.
3. The data modeling framework is theoretically feasible and practical. It proves that there are computational savings to be made of at least 75% of the original simulation computational cost. In Subfig. 7c, it can easily be seen that by learning on the first 1/4 of the signal, we would be able to reproduce the remaining part. **XXx what do you mean? - I added some more elaboration.**

4.2 DTBM Vessel

To further test the capabilities of LSTM-NN we introduce a third DOF and we approximate motions of a second vessel. The vessel is allowed to rotate around its longitudinal axis (rolling). This type of motions are affected by the viscous effects, in contrast to heave and pitch motions that can be well approximated given the stiffness and mass matrix of the vessel. Nevertheless, it is also true that in the sea state modeled much of the assumptions of the linear theory that is conventionally used does not hold. The nonlinearities in heave and pitch can be successfully modeled with nonlinear Boundary Element Methods (BEMs) that track he vessels' wet surface [23]. This is not sufficient to model accurately roll motions, since the solver needs to resolve the viscous terms of the Navier-Stokes Equations to have the necessary information to accurately compute the motions. A validated approximation is given by the URANS numerical scheme [27]. Nevertheless, this requires a prohibitive expensive series of simulations from which we obtain data that is difficult to generalize, thus it is better to develop a surrogate model.

In a similar fashion as in Sec. 4.1, using deep recurrent neural networks with LSTM cells, we attempt to construct functionals that given the sea surface time series will approximate the motions of the DTBM vessel that sails in it. A total of 4 examples (each with 3DOF) of the architecture convergence are shown in figs. 9 to 12. In each of these figures we can see two test sequences. These two sequences are unseen sea state realizations for the motions the network approximates, i.e., heave, pitch and roll motions. We use these to later calculate the MSE for each set of architecture parameters.

The architecture convergence shows consistent and accurate approximation of the vessel motions in the extreme sea states. Performance in roll motions is somewhat worse but this can be expected given the higher complexity of the dynamics. Fig. 11 shows the network architecture with the best performance. We see that roll motions with large amplitudes, almost 20 degrees, are accurately approximated both in amplitude and frequency of the time signal. The same can be said about the heave and pitch motions.

The overall conclusions of the architecture convergence are similar to those obtained in Sec. 4.1. Given more data we should be able to train more complex networks to improve the accuracy of the motion prediction. This data could potentially come from several different sources and possibly a real vessel, for which approximating a lifetime or near future motions is interesting. These will be vessels that may be required to operate in very harsh weather conditions. Good examples can be military, coastguard or rescue vessels.

5 Conclusions

Deep recurrent neural networks of LSTM type have been successfully trained to approximate nonlinear motions in irregular long crested head and oblique seas. To the authors' best knowledge, this constitutes a new paradigm in simulating the motion of vessels. The presented approach is not limited to specific application presented here. Furthermore, the CFD simulations used are aimed at modeling real physical systems and when data is available for these systems the RNNs can be trained using this data. This, given the availably wave radars and even satellite images, can provide a way of simulating in a quick and accurate way vessel motions. Under extreme weather conditions this would constitute a potentially powerful predictive tool to avoid associated hazards of these situations.

References

[1] T. Chen and H. Chen, "Approximations of continuous functionals by neural networks with application to dynamic systems," in IEEE Transactions on Neural Networks, vol. 4, no. 6, pp. 910-918, Nov. 1993.

doi: 10.1109/72.286886

[2] Ferziger, Joel H & Peric, Milovan. *Computational methods for fluid dynamics*. 2014. Springer Science & Business Media.

[3] Menter, Florian R. *Two-equation eddy-viscosity turbulence models for engineering applications*. AIAA journal. Vol. 32. Num. 8. 1598–1605. 1994.

[4] Aurélien Géron. Hands-On Machine Learning with Scikit-Learn & Tensorflow. O'Reilly Media, Inc. 2017. ISBN: 978-1-491-96229-9.

[5] Hochreiter, Sepp & Schmidhuber, Jürgen, *Long Short-Term Memory*, Neural Comput., November 15 1997, volume 9, number 8, pages 1735–1780.

[6] T. Jaakkola and R. Barzilay, 6.036 class notes, February 5, 2018.

[7] del Águila Ferrandis, José and Brizzolara, Stefano and Chryssostomidis, Chryssostomos. Influence of large hull deformations on the motion response of a fast catamaran craft with varying stiffness. Ocean Engineering, volume 163, pages 207–222, year 2018, publisher Elsevier.

[8] Kring, DC & Milewski, WM & Fine, NE. *Validation of a NURBS-based BEM for multihull ship seakeeping*. 25th Symposium on Naval Hydrodynamics, St. John's. 2004.

[9] Tezdogan, Tahsin & Demirel, Yigit Kemal & Kellett, Paula & Khorasanchi, Mahdi & Incecik, Atilla & Turan, Osman. *Full-scale unsteady RANS CFD simulations of ship behaviour and performance in head seas due to slow steaming*. Ocean Engineering. 97. 186–206. 2015. Elsevier.

[10] Lee, C-H & Newman, JN. *Computation of wave effects using the panel method*. WIT Transactions on State-of-the-art in Science and Engineering. Volume 18. 2005. WIT Press.

[11] Lee, C-H & Newman, JN. *The computation of wave loads on large offshore structures*. Proc. of Int. Conf. on Behaviour of Offshore Structures (BOSS'88). pages 605–622. 1988.

[12] Newman, John Nicholas. *Marine hydrodynamics*. 1977. MIT press.

[13] Newman, JN. *Algorithms for the free-surface Green function*. Journal of engineering mathematics. Vol. 19. Num. 1. 57–67. 1985. Springer.

[14] Newman, JN. *Distributions of sources and normal dipoles over a quadrilateral panel*. Journal of Engineering Mathematics. Vol. 20. Num. 2. 113–126. 1986. Springer.

[15] Faltinsen, Odd. *Sea loads on ships and offshore structures*. Cambridge university press. Vol. 1. 1993.

[16] Faltinsen O.M. *Second Order Nonlinear Interactions Between Waves and Low Frequency Body Motion*. Nonlinear Water Waves. International Union of Theoretical and Applied Mechanics. 1988. Springer, Berlin, Heidelberg.

[17] You, J. Faltinsen, O.M. *A numerical investigation of second-order difference-frequency forces and motions of a moored ship in shallow water*. J. Ocean Eng. Mar. Energy (2015). `https://doi.org/10.1007/s40722-015-0014-6`.

[18] Faltinsen, O.M. *Wave Loads on Offshore Structures*. Annual review of fluid mechanics. 1990.

[19] Faltinsen, Odd M and Sortland, Bjørn. *Slow drift eddy making damping of a ship*. Vol. 9, num. 1, pages 37–46. 1987. Applied Ocean Research. Elsevier.

[20] Kring, David Charles. *Time domain ship motions by a three-dimensional Rankine panel method*. Massachusetts Institute of Technology. 1994.

[21] Kring,D.C., Korsmeyer, F.T., Singer, J., Danmeier, D., & White, J.. *Accelerated nonlinear wave simulations for large structures*. 7th Int'l Conf on Numerical Ship Hydrodynamics. Nantes. France. 1999.

[22] Kring. D.C., Korsemeyer, F.T., Singer, J., & White, J.. *Analyzing mobile offshore bases using accelerated boundary element methods*. J. of Marine Structures. 13. pp 301-313. 2000.

[23] Kring, DC & Milewski, WM & Fine, NE. *Validation of a NURBS-based BEM for multihull ship seakeeping*. 25th Symposium on Naval Hydrodynamics, St. John's. 2004.

[24] Beck, Robert F & Scorpio, Stephen M. *A desingularized boundary integral method for fully nonlinear water wave problems*. Twelfth Australasian Fluid Mechanics Conference (Sydney, Australia). Vol. 6. 1995.

[25] Cao, Yusong & Beck, Robert F & Schultz, William W. *Nonlinear computation of wave loads and motions of floating bodies in incident waves*. Proceedings of the 9th International Workshop on Water Waves and Floating Bodies (IWWWFB). 1994.

[26] Ferziger, Joel H & Peric, Milovan. *Computational methods for fluid dynamics*. 2014. Springer Science & Business Media.

[27] Tezdogan, Tahsin & Demirel, Yigit Kemal & Kellett, Paula & Khorasanchi, Mahdi & Incecik, Atilla & Turan, Osman. *Full-scale unsteady RANS CFD simulations of ship behaviour and performance in head seas due to slow steaming*. Ocean Engineering. 97. 186–206. 2015. Elsevier.

[28] Ruth, Eivind & Berge, Bjørn Ola & Borgen, Henning. *Simulation of Added Resistance in High Waves*. ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering. V011T12A018–V011T12A018. 2015. American Society of Mechanical Engineers.

[29] Shen, Zhirong & Wan, Decheng, et al. *RANS computations of added resistance and motions of a ship in head waves*. International Journal of Offshore and Polar Engineering. Vol. 23. Num. 04. 264–271. 2013. International Society of Offshore and Polar Engineers.

[30] Mousaviraad, Maysam & Conger, Michael & Stern, Frederick & Peterson, Andrew & Ahmadian, Mehdi. *Validation of CFD-MBD FSI for High-Fidelity Simulations of Full-Scale WAM-V Sea-Trials with Suspended Payload*.

[31] Mousaviraad, SM & Bhushan, S & Stern, F. *URANS studies of wam-v multi-body dynamics in calm water and waves*. Third International Conference on Ship Maneuvering in Shallow and Confined Water, Ghent, Belgium. 3–5. 2013.

[32] Menter, Florian R. *Two-equation eddy-viscosity turbulence models for engineering applications*. AIAA journal. Vol. 32. Num. 8. 1598–1605. 1994.

[33] Hirt, Cyril W & Nichols, Billy D. *Volume of fluid (VOF) method for the dynamics of free boundaries*. Journal of computational physics. Vol. 39. Num. 1. 201–225. 1981. Elsevier.

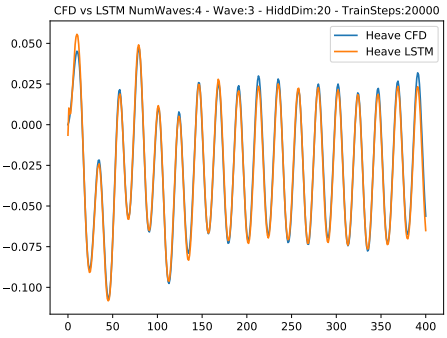
[34] Hornik, Kurt and Stinchcombe, Maxwell and White, Halbert. *Multilayer feedforward networks are universal approximators*. Neural networks. Vol. 2. Num. 5. 359–366. 1989. Elsevier.

[35] Hornik, Kurt. *Approximation capabilities of multilayer feedforward networks*. Neural networks. Vol. 4. Num. 2. 251–257. 1991. Elsevier.

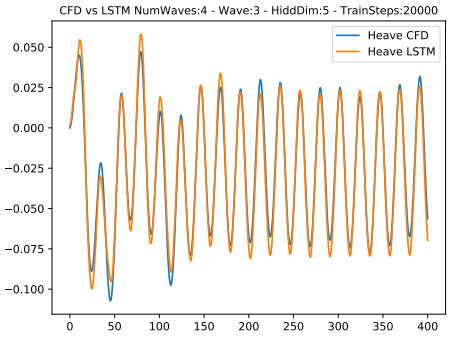
[36] Cybenko, George. *Approximation by superpositions of a sigmoidal function*. Mathematics of control, signals and systems. Vol. 2. Num. 4. 303–314. 1989. Springer.

[37] Sandberg, Irwin W. *Approximation theorems for discrete-time systems*. [1991] Proceedings of the 34th Midwest Symposium on Circuits and Systems. 6–7. 1992. IEEE.

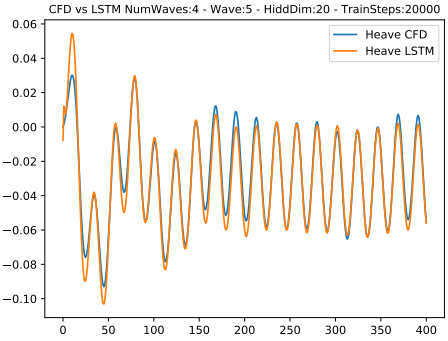
[38] Kreinovich, Vladik Ya. *Arbitrary nonlinearity is sufficient to represent all functions by neural networks: a theorem*. Neural networks. Vol. 4. Num. 3. 381–383. 1991. Elsevier.



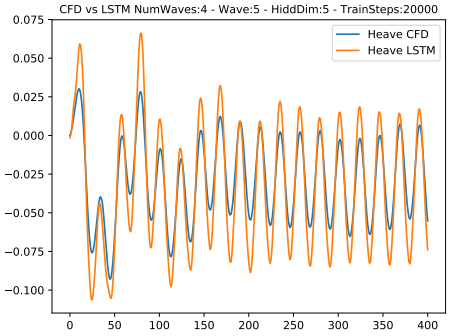
(a) Train Case. 20 hidden cells, 1 hidden layer & 20000 train steps.



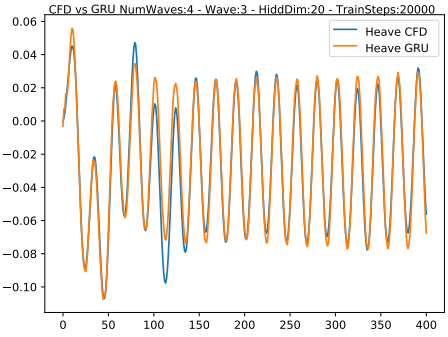
(b) Train Case. 5 hidden cells, 1 hidden layer & 20000 train steps.



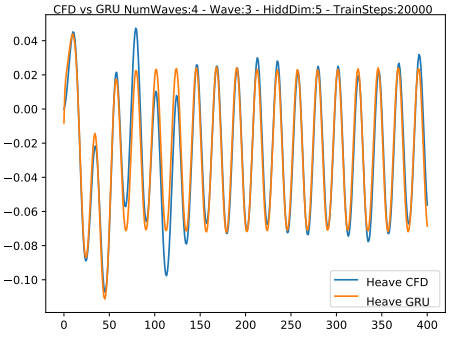
(c) Test Case. 20 hidden cells, 1 hidden layer & 20000 train steps.



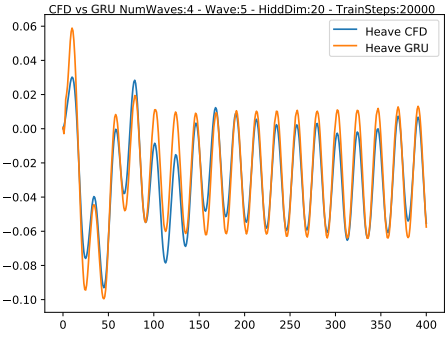
(d) Test Case. 5 hidden GRU cells, 1 hidden layer & 20000 train steps.



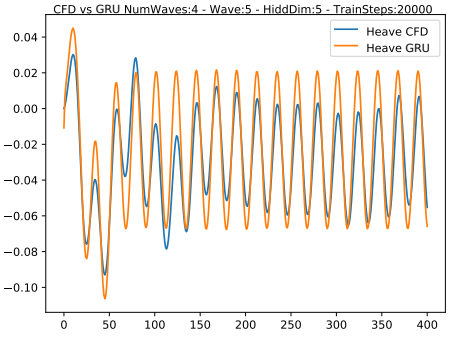
(e) Train Case. 20 hidden GRU cells, 1 hidden layer & 20000 train steps.



(f) Train Case. 5 hidden GRU cells, 1 hidden layer & 20000 train steps.

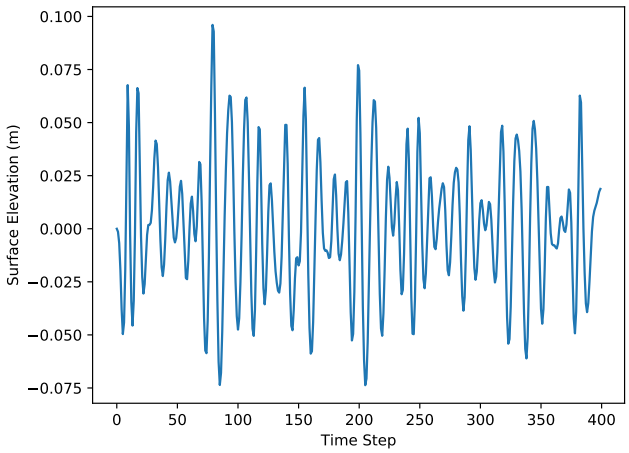


(g) Test Case. 20 hidden GRU cells, 1 hidden layer & 20000 train steps.

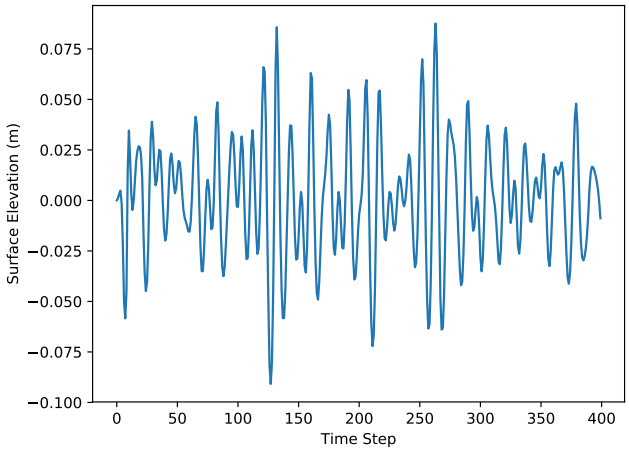


(h) Test Case. 5 hidden GRU cells, 1 hidden layer & 20000 train steps.

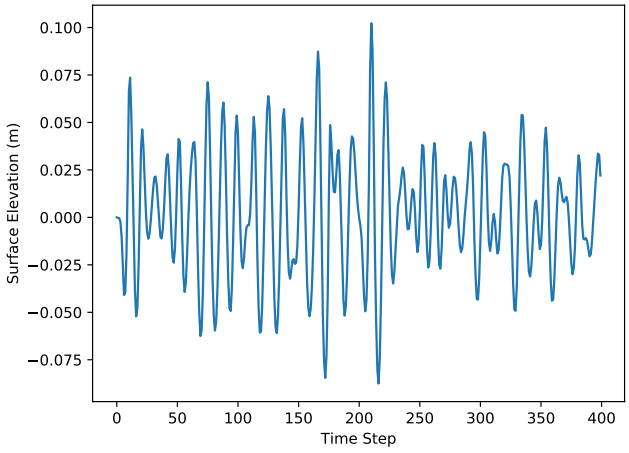
Figure 4: *Examples of the hyper-parameter convergence study.* The data-set is composed of 5 waves of varying amplitude. Because of the limitation of computational resources we limit the parameter variations done in the study. The first 4 waves are used as train cases and the last wave is used for testing. In the plots provided above, going from left to right & top to bottom we have a decreasing level of accuracy in the solution. In the study parametric variations where performed for 5, 10, 15& 20 hidden cells and 2000, 5000, 10000 & 20000 training steps. As it can be seen in the figures the length of the signal is of 400 steps. LSTM & GRU cells are used for the examples provided above.



(a) Network input Subfigs. 6a and 6c.

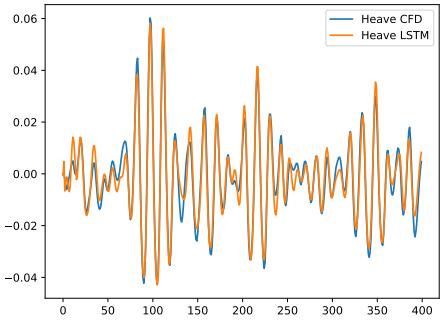


(b) Network input Subfigs. 6b and 6d.

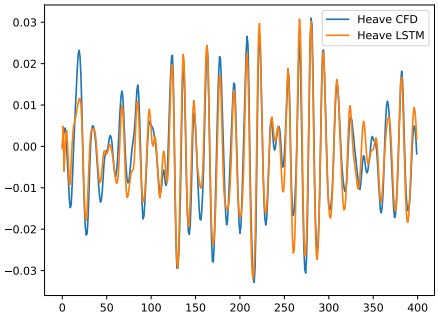


(c) Network input Fig. 7.

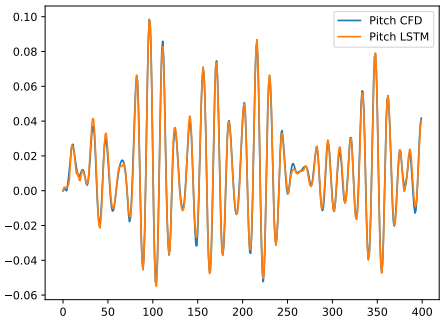
Figure 5: *Network Inputs for 2DOF catamaran vessel in irregular head seas.*



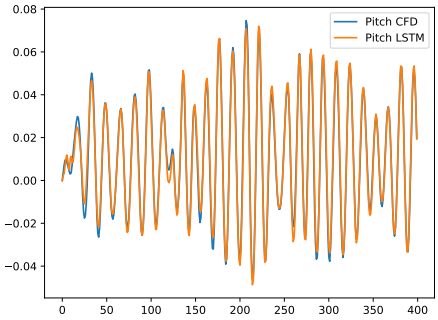
(a) Vertical vessel motion. **Train Case** case having trained on 3 different sequences.



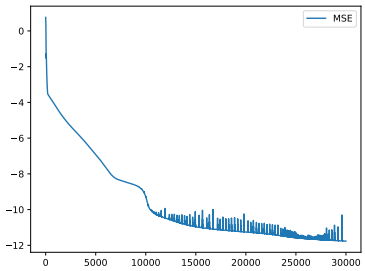
(b) Vertical vessel motion. **Test Case** having trained on 3 different sequences.



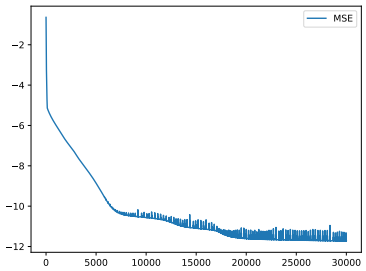
(c) Angular vessel motion. **Train Case** having trained on 3 different sequences.



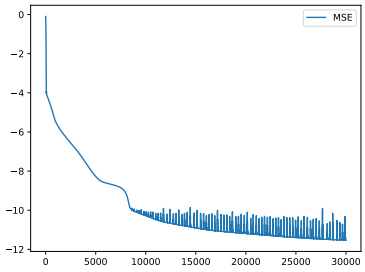
(d) Angular vessel motion. **Test Case** having trained on 3 different sequences.



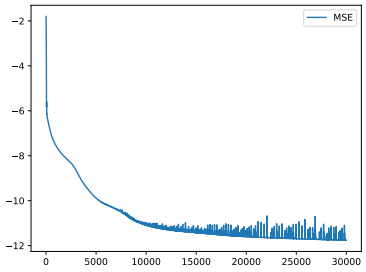
(e) Train MSE, having trained on 3 different sequences.



(f) Train MSE, having trained on 5 different sequences.

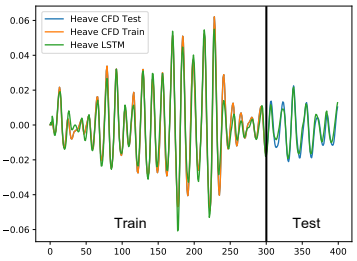


(g) Train MSE, having trained on 7 different sequences.

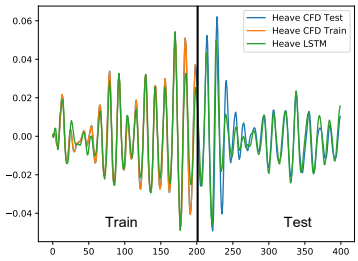


(h) Train MSE, having trained on 9 different sequences.

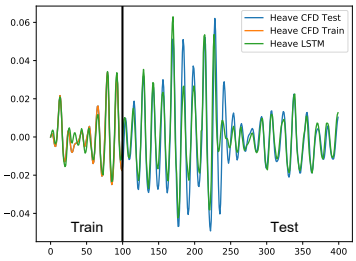
Figure 6: *Dependence of training MSE behaviour on information available.* Regarding non-linear sequence modeling for motion quantification, we have initially increased the capabilities of the network in two ways. First, we have extended our model capabilities to stochastic waves that model typical sea states that the vessel may encounter. Second, we are modeling not just one but the two motions of the vessel. Namely vertical displacement and pitch angular motion. We have found that pitch angular motion, the added motion, is modeled better than the previous. Increasing the amount of data available to the network has somehow prevented it from overfitting, in 6h we can see how training MSE stabilizes nicely around 10^{-12}, contrary to the other cases that march on overfitting the data and not improving results on test cases.



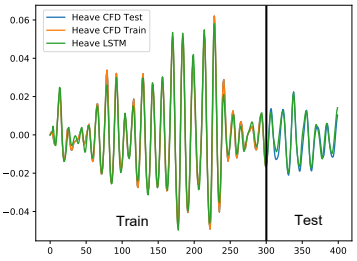
(a) Vertical Motion, **Train on 2/3 of sequence**. 15 hidden units, **1 hidden layer** and 30000 train steps. MSE= 1.7020e-05.



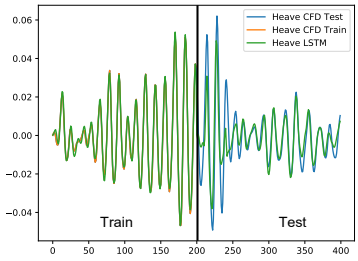
(b) Vertical Motion, **Train on 1/2 of sequence**. 15 hidden units, **1 hidden layer** and 30000 train steps. MSE = 3.7029e-05.



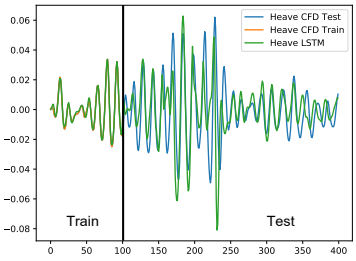
(c) Vertical Motion, **Train on 1/4 of sequence**. 15 hidden units, **1 hidden layer** and 30000 train steps. MSE = 5.7327e-05.



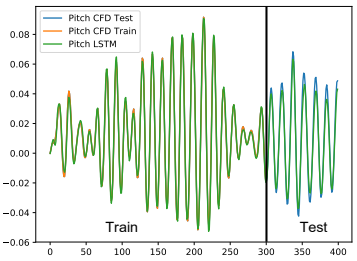
(d) Vertical Motion, **Train on 2/3 of sequence**. 15 hidden units, **3 hidden layers** and 30000 train steps. MSE = 8.4781e-06.



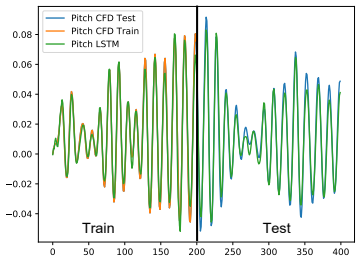
(e) Vertical Motion, **Train on 1/2 of sequence**. 15 hidden units, **3 hidden layers** and 30000 train steps. MSE = 4.5099e-05.



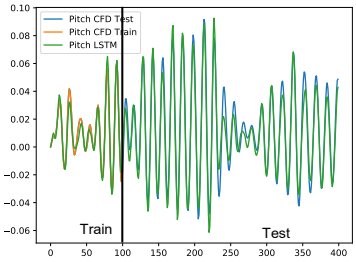
(f) Vertical Motion, **Train on 1/4 of sequence**. 15 hidden units, **3 hidden layers** and 30000 train steps. MSE = 2.5656e-04.



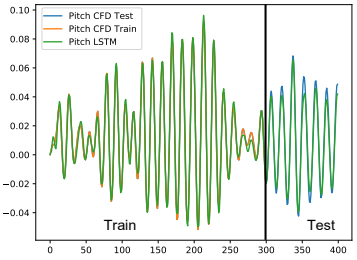
(g) Angular Motion, **Train on 2/3 of sequence**. 15 hidden units, **1 hidden layer** and 30000 train steps. MSE =1.0249e-05.



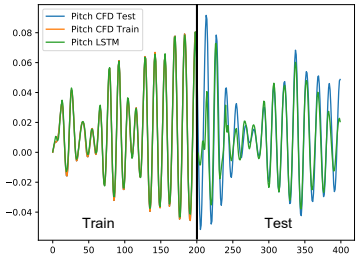
(h) Angular Motion, **Train on 1/2 of sequence**. 15 hidden units, **1 hidden layer** and 30000 train steps. MSE =2.3605e-05.



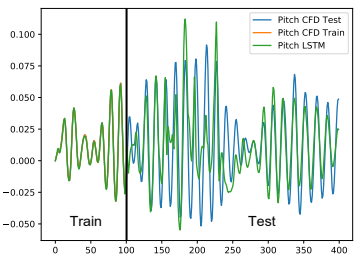
(i) Angular Motion, **Train on 1/4 of sequence**. 15 hidden units, **1 hidden layer** and 30000 train steps. MSE =4.0008e-05.



(j) Angular Motion, **Train on 2/3 of sequence**. 15 hidden units, **3 hidden layers** and 30000 train steps. MSE = 1.2446e-05.

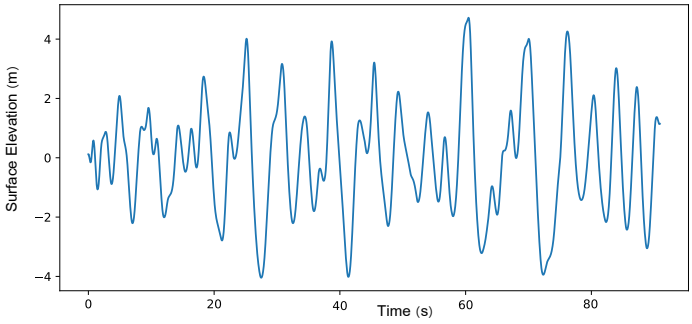


(k) Angular Motion, **Train on 1/2 of sequence**. 15 hidden units, **3 hidden layers** and 30000 train steps. MSE = 1.1840e-04.

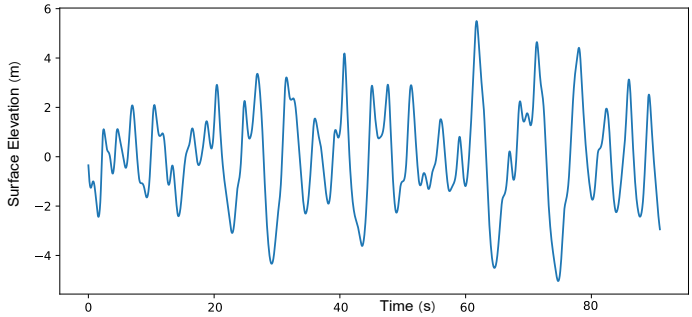


(l) Angular Motion, **Train on 1/4 of sequence**. 15 hidden units, **3 hidden layers** and 30000 train steps. MSE = 4.3936e-04.

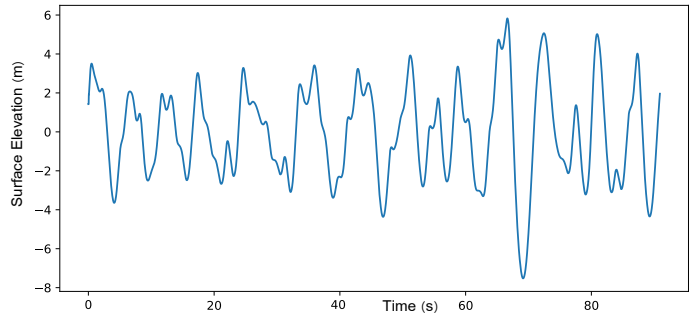
Figure 7: *Network performance dependence on training data and network parameters*. The plots presented above are a representative subset of 64 parametric variations of 5 network parameters. These parameters are: number of hidden units, number of hidden layer, training steps, number of sequences in the training process, fraction of anyone sequence used in the training process. The **first** 6 plots, sub-figures 7a - 7f, analyze the evolution of motion predictions when we vary fraction of sequence used in the training process (7a $\rightarrow$ 7c or 7d $\rightarrow$ 7f) or when we vary the number of hidden layers (7a $\rightarrow$ 7d or 7b $\rightarrow$ 7e or 7c $\rightarrow$ 7f). The **second** set of subplots (7d - 7l) is similar to the **first** set, the only thing that changes is the variable in the plots. Instead of vertical motion (heave) we plot angular motion (Pitch).



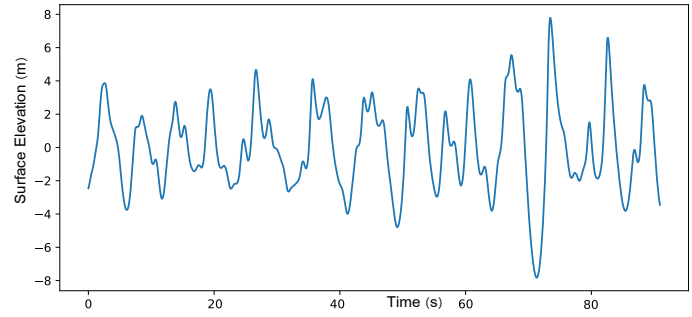
(a) Wave cut in the X-direction corresponding to 1st **test** case.



(b) Wave cut in the Y-direction corresponding to 1st **test** case.

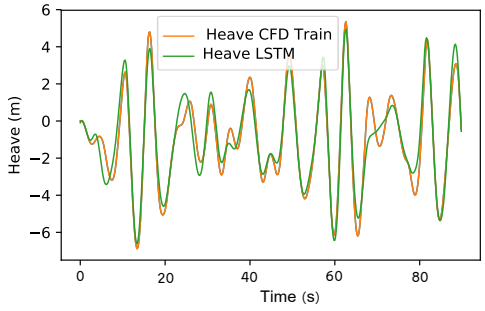


(c) Wave cut in the X-direction corresponding to 2nd **test** case.

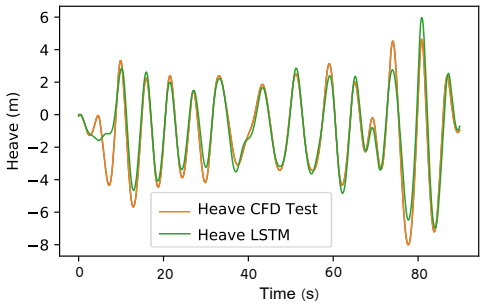


(d) Wave cut in the Y-direction corresponding to 2nd **test** Case.

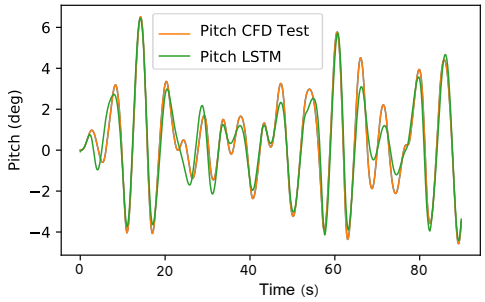
Figure 8: *Network inputs*. In the subfigures above we can see the network inputs. It is 2D and characterizes a long crested irregular sea.



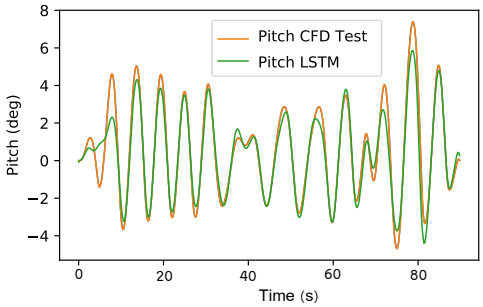
(a) Test 1 - Relative MSE = 0.0770



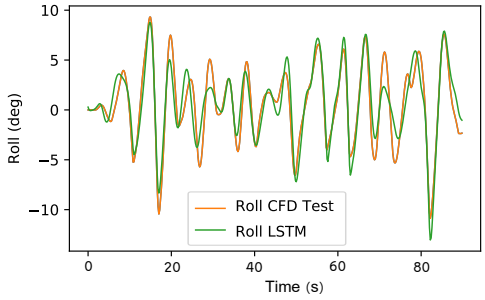
(b) Test 2 - Relative MSE = 0.0910



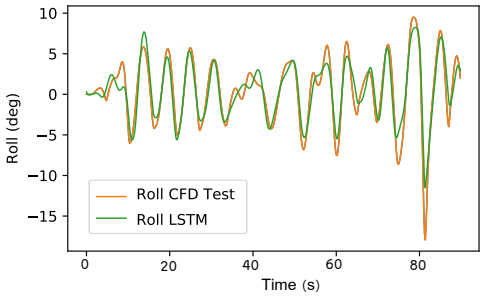
(c) Test 1 - Relative MSE = 0.0734



(d) Test 2 - Relative MSE = 0.0747

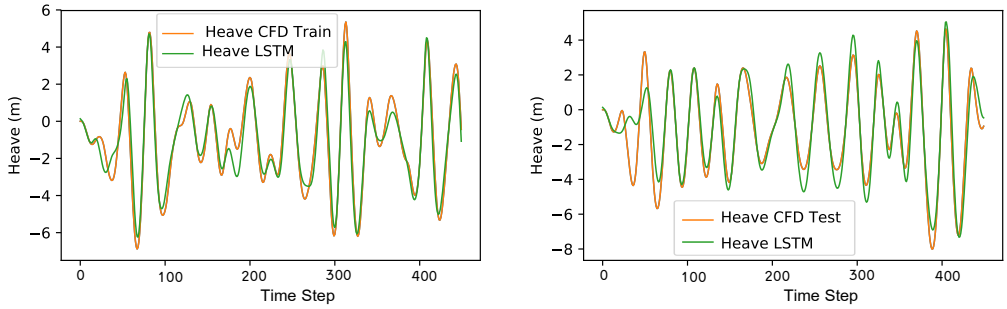


(e) Test 1 - Relative MSE = 0.1595



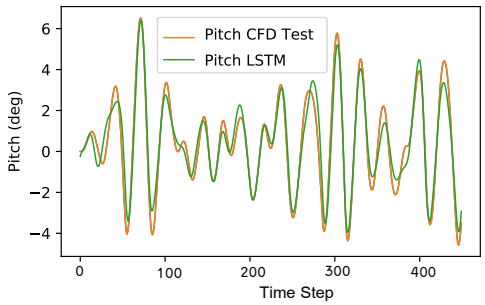
(f) Test 2 - Relative MSE = 0.1257

Figure 9: *Test results network architecture 1, MSE=0.192.* **Heave, Pitch and Roll** motions for 3DOF motion approximation of a notional DTBM battleship sailing in WMO sea state at high Froude Number (0.4). The motion approximations provided in the unseen realizations of the sea state (All of the above) obtain a good performance both in amplitude and frequency approximation of the vessel response. It is interesting to notice that, at simple sight, we can see that roll spectra is very different to that of heave and pitch motions, that have more linear characteristics and are similar to each other. It is not surprising to realize that the performance of the network approximating roll motions is somewhat worse compared to heave and pitch motions. The network parameters are: {*Hidden Units*, *Number of Layers*, *Number of Train Steps*} = {90,8,5000}. Parameters that are not relevant remain constant. The inputs provided to the network are shown in Fig. 8.

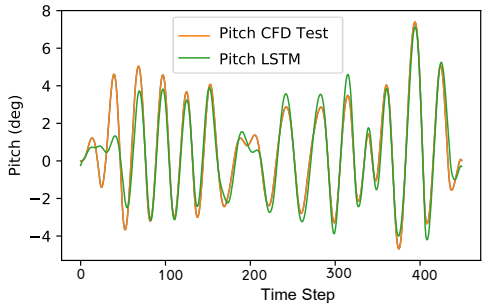


(a) Test - Relative MSE = 0.0674

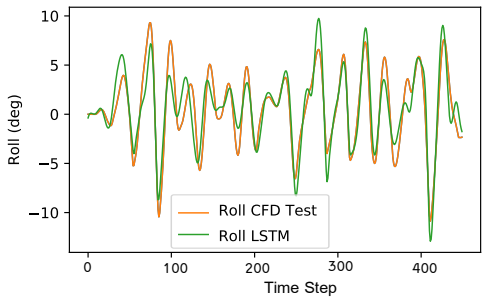
(b) Test - Relative MSE = 0.1245



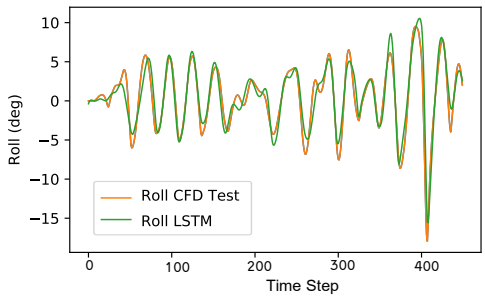
(c) Test - Relative MSE = 0.0685



(d) Test - Relative MSE = 0.0962

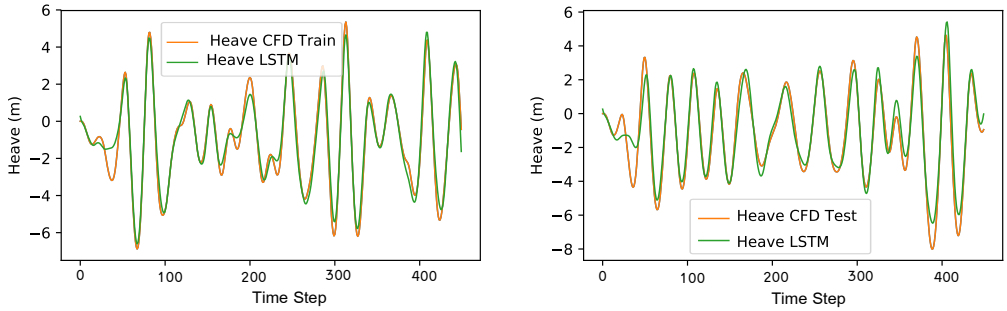


(e) Test - Relative MSE = 0.1910



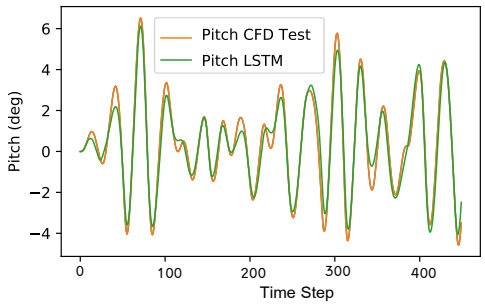
(f) Test - Relative MSE = 0.1210

Figure 10: *Test results network architecture 2, MSE=0.220.* **Heave, Pitch and Roll** motions for 3DOF motion approximation of a notional DTBM battleship sailing in WMO sea state at high Froude Number (0.4). The motion approximations provided in the unseen realizations of the sea state obtain a good performance both in amplitude and frequency approximation of the vessel response. It is interesting to notice that, at simple sight, we can see that roll spectra is very different to that of heave and pitch motions, that have more linear characteristics and are similar to each other. It is not surprising to realize that the performance of the network approximating roll motions is somewhat worse compared to heave and pitch motions. The network parameters are: {*Hidden Units*, *Number of Layers*, *Number of Train Steps*} = {70,8,5000}. Parameters that are not relevant remain constant. The inputs provided to the network are shown in Fig. 8.

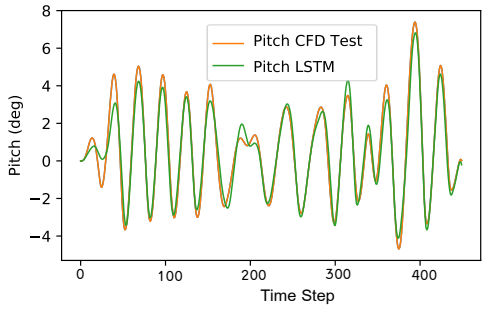


(a) Test - Relative MSE = 0.0380

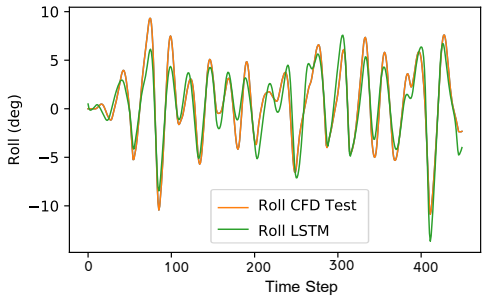
(b) Test - Relative MSE = 0.0773



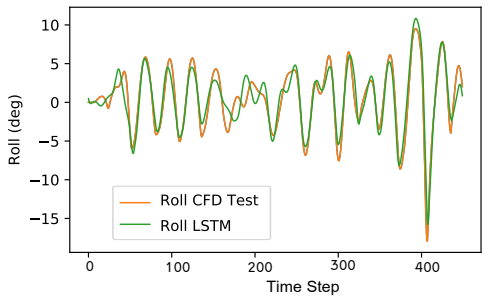
(c) Test - Relative MSE = 0.0444



(d) Test - Relative MSE = 0.0463

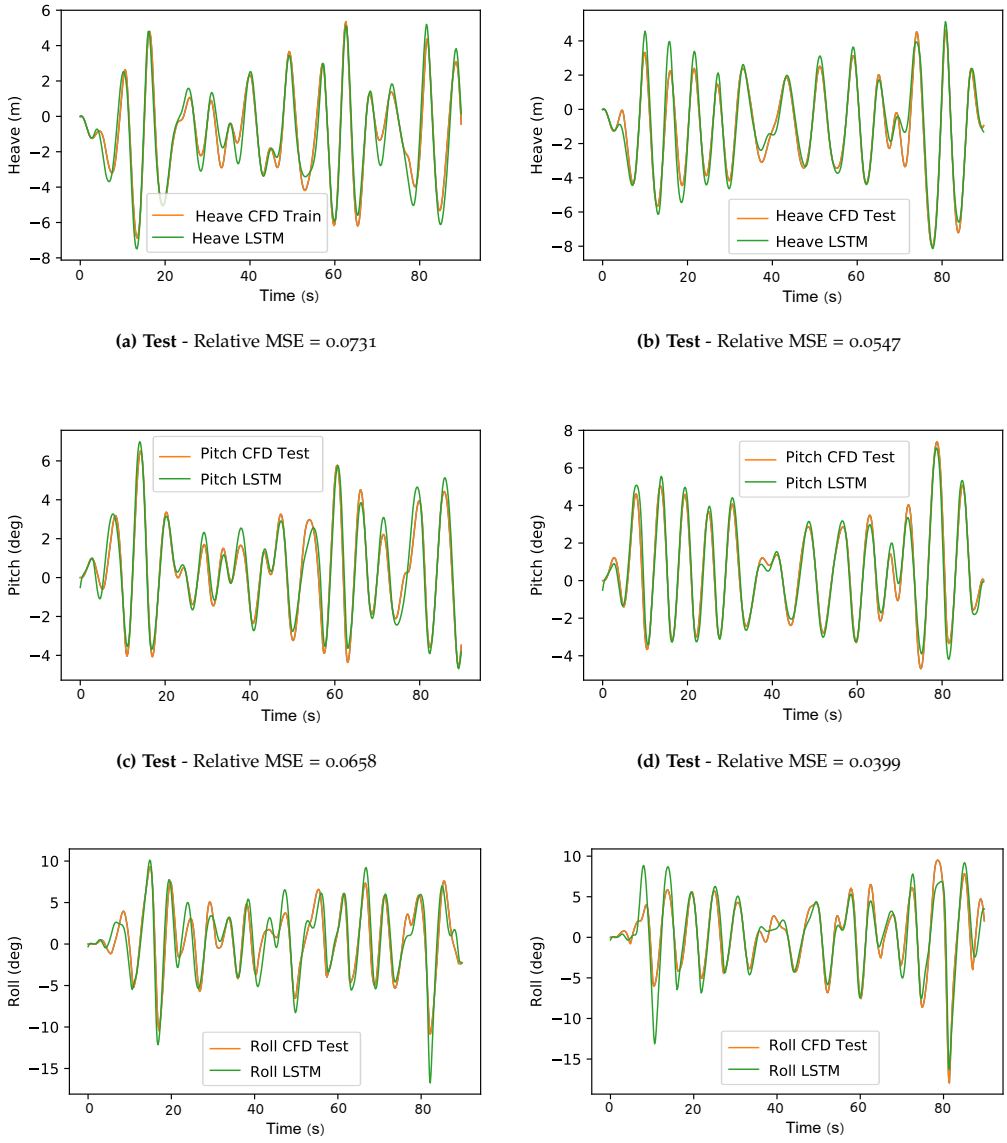


(e) Test - Relative MSE = 0.1490



(f) Test - Relative MSE = 0.1215

Figure 11: *Test results network architecture 3, MSE=0.165.* **Heave, Pitch and Roll** motions for 3DOF motion approximation of a notional DTBM battleship sailing in WMO sea state at high Froude Number (0.4). The motion approximations provided in the unseen realizations of the sea state obtain a good performance both in amplitude and frequency approximation of the vessel response. It is interesting to notice that, at simple sight, we can see that roll spectra is very different to that of heave and pitch motions, that have more linear characteristics and are similar to each other. It is not surprising to realize that the performance of the network approximating roll motions is somewhat worse compared to heave and pitch motions. The network parameters are: {*Hidden Units*, *Number of Layers*, *Number of Train Steps*} = {90,4,5000}. Parameters that are not relevant remain constant. The inputs provided to the network are shown in Fig. 8.



(a) Test - Relative MSE = 0.0731

(b) Test - Relative MSE = 0.0547

(c) Test - Relative MSE = 0.0658

(d) Test - Relative MSE = 0.0399

(e) Test - Relative MSE = 0.1919

(f) Test - Relative MSE = 0.1634

Figure 12: *Test results network architecture 4, MSE=0.230.* **Heave, Pitch and Roll** motions for 3DOF motion approximation of a notional DTBM battleship sailing in WMO sea state at high Froude Number (0.4). The motion approximations provided in the unseen realizations of the sea state (Fig. 12e & Fig. 12f) obtain a good performance both in amplitude and frequency approximation of the vessel response. It is interesting to notice that, at simple sight, we can see that roll spectra is very different to that of heave and pitch motions, that have more linear characteristics and are similar to each other. It is not surprising to realize that the performance of the network approximating roll motions is somewhat worse compared to heave and pitch motions. The network parameters are: {*Hidden Units*, *Number of Layers*, *Number of Train Steps*} = {70,16,5000}. Parameters that are not relevant remain constant. The inputs provided to the network are shown in Fig. 8.