Neural Network Meta-Model for FPSO Roll Motion Prediction from Environmental Data

Lucas P. Cotrim[1], Henrique B. Oliveira[2], Asdrubal N. Q. Filho[3], Eduardo A. Tannuri[4], Anna Helena R. Costa[5], Edson S. Gomi[6], Ismael H. F[7]. Santos, Rodrigo A. Barreira[8]

Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, 380, 05508-010, São Paulo-SP, Brazil
[1]*lucas.cotrim@usp.br, orcid 0000-0001-5288-1813,* [2]*henrique.barrosoliveira@usp.br,* [3]*asdrubal.filho@tpn.usp.br,* [4]*eduat@usp.br, orcid 0000-0001-7040-413X,* [5]*anna.reali@usp.br, orcid 0000-0001-7309-4528,* [6]*gomi@usp.br, orcid 0000-0003-1267-9519*

Petrobras, Rio de Janeiro, Brazil
[7]*ismaelh@petrobras.com.br,* [8]*barreira@petrobras.com.br*

Abstract. The current design process of mooring systems for Floating Oil Production and Offloading units (FPSOs) is highly dependent on the availability of the platform's mathematical model and accuracy of dynamic simulations, through which resulting time series motion is evaluated according to design constraints. Out of the six degrees of freedom, roll motion is among the most complex to accurately simulate. We propose a Neural Simulator, a set of neural network surrogate models designed to predict an FPSO's roll motion statistics directly from metocean data when subject to different loads. This approach bypasses the need to perform traditional time series dynamic simulation, as the trained models take measured metocean conditions and directly output the desired roll motion statistics. This allows for Artificial Neural Networks (ANNs) to be trained through simulation and later fine-tuned on real FPSO motion. As a result, our proposal presents higher accuracy and reduced computational time when compared to traditional methods. The ANN surrogate models are trained by real current, wind and wave data measured in 3h periods at the Campos Basin from 2003 to 2010 and the associated roll response of a Spread Moored FPSO subject to different drafts, which is obtained through time-domain simulations using the Dynasim software. Hyperparameter Optimization techniques are performed in order to obtain optimal ANN models specialized in different platform drafts. Finally, the proposed models are shown to correctly capture platform dynamics, providing good results when compared to the statistical analysis of roll motion time series obtained from Dynasim. We conclude that an ANN surrogate model can be trained directly on real measured metocean conditions and platform roll motion to provide increased accuracy and reduced computational time over traditional methods based on dynamic simulation. Moreover, the proposed architecture can be integrated into an automated learning framework: The data-based surrogate models can be continuously fine-tuned and updated with newly measured data, resulting in improved accuracy over time.

Keywords: Floating Offshore Platforms, Artificial Neural Networks, Surrogate Models, Hyperparameter Optimization, Neural Architecture Search

1 Introduction

Current FPSO's Mooring System Design consists in the measurement of local environmental conditions over a representative period of time and subsequent dynamic simulation of the FPSO model subject to combinations of extreme winds, waves and currents expected in the next 10 to 100 years of operation, which are obtained from statistical projections of the measured environmental conditions. Design variables, such as the maximum roll angle, are obtained through time-series analysis and verified to remain within project criteria. This process, however, relies on the numerical simulation of a dynamic model on softwares such as Dynasim [1], which multiplies the approximated wave energy spectrum and the FPSO's RAOs (Response Amplitude Operators) to obtain the expected vessel's movement. This process can be time-consuming and give slightly inaccurate responses when compared to the actual measured movement.

Recently, the increasing performance of data-based machine learning models in a variety of domains in conjunction with the high computational times of traditional models and the unprecedented availability of data have motivated the study and development of alternative models, denominated surrogate or meta-models. The main motivation behind using such models is to directly model complex and computationally expensive dynamics through available data. Meta-models have been successfully implemented as alternatives for Finite Element (FE) [2–4] and Computational Fluid Dynamics (CFD) models [5] for predicting mooring line tensions and submerged riser's vibration responses, respectively. Gumley, Henry and Potts [6] successfully implemented a neural network capable of predicting the hourly mean offset of a turret-moored FPSO from environmental conditions, and compared it to Kriging time series prediction. In this paper, we focus on the prediction of maximum roll amplitude due to its complexity and relevance in mooring system design.

1.1 Objectives

In this article, the main objective is to contribute with a new framework composed of a set of data-based meta-models capable of predicting the maximum roll angle associated with the dynamic response of an FPSO to generic environmental conditions, as well as validating these meta-models. Overall, a data-based approach is expected to present three main advantages in comparison to traditional methods:

- **Increased accuracy:** Training directly on real measured environmental conditions and the corresponding platform responses avoids several approximations and simplifications of physical phenomena, such as varying mooring line damping and second order wave drift, implemented on traditional simulation softwares. Moreover, the availability of a considerable volume of data (over 18 thousand 3h periods) improves the accuracy of trained data-based models.
- **Automated Learning:** The resulting system is designed to be integrated to other design tools and continuously updated with newly measured environmental and platform motion data.
- **Reduced Computational Time:** After training, the computational time associated with the evaluation of a neural network prediction is significantly shorter than that associated with time integration of the system's dynamic equations.

2 Methodology

This section aims to provide an overview of the methodology adopted in the project, briefly describing the different steps followed from the obtention of the measured environmental conditions to the preparation of the neural simulator training dataset.

Upon receiving the environmental data, an exploratory statistical analysis is conducted and a representative, uninterrupted period of 6 years is chosen from the available data in which no missing values were present. The environmental conditions and the FPSO description are then loaded and simulated in Dynasim, which generates the corresponding time series platform motion associated with each environmental condition. A python Post Processor module then analyzes the stored time series and extracts relevant statistical information used in the mooring system design, such as maximum platform offset and roll.

The resulting dataset, comprised of both the measured environmental conditions as well as the associated motion statistics, is used to train and validate each meta-model. Each meta-model's hyperparameters are optimized through 5-fold cross validation and test data is used to evaluate the proposed models' performance. Figure 1 details the adopted framework from environmental data to model training.

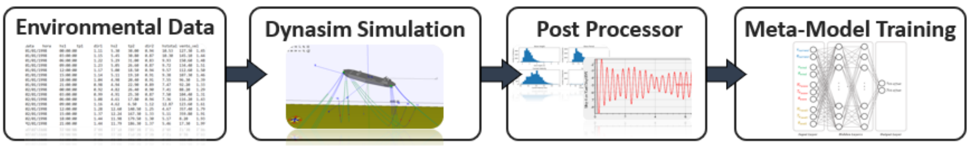


Figure 1. Project Workflow Diagram.

2.1 Environmental Data

The environmental conditions were provided by Petrobras Oceanography Group in 3h periods from November 2003 to December 2009 at the Campos Basin. The data comes from a hindcast model of the area, calibrated by a large number of measurements by wave radars, anemometers, wave buoys, and current meters, that are either installed in the platforms or moored to the seabed. The data can be described by the following variables:

1. **Current velocity**: Mean current velocity v_c (m/s).
2. **Current direction**: Current propagation angle θ_c (°).
3. **Wind velocity**: Mean wind velocity v_w (m/s).
4. **Wind direction**: Wind incidence angle θ_w (°).
5. **First Wave component height**: Significant wave height $H_{s1}(m)$ corresponding to highest energy wave.
6. **First Wave component period**: Peak Period $T_{p1}(s)$ corresponding to highest energy wave.
7. **First Wave component direction**: Incidence angle θ_1 (°) corresponding to highest energy wave.
8. **Second Wave component height**: Significant wave height $H_{s2}(m)$ corresponding to second highest energy wave.
9. **Second Wave component period**: Peak Period $T_{p2}(s)$ corresponding to second highest energy wave.
10. **Second Wave component direction**: Incidence angle θ_2 (°) corresponding to second highest energy wave.

Table 1 shows samples of the measured environmental conditions and the corresponding values of each variable.

Table 1. Samples of measured environmental conditions.

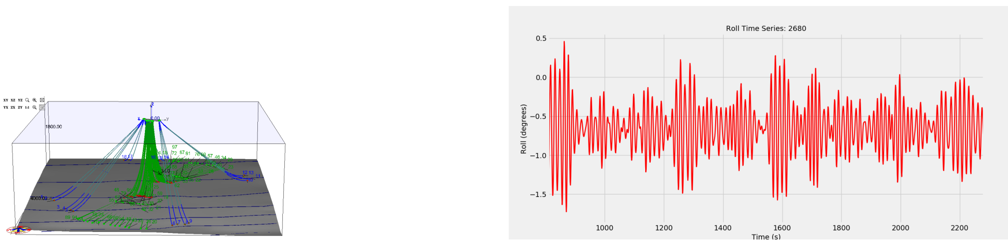
Index	v_c (m/s)	θ_c (°)	v_w (m/s)	θ_w (°)	Hs_1 (m)	Tp_1 (s)	θ_1 (°)	Hs_2 (m)	Tp_2 (s)	θ_2 (°)
1	0.11	118.33	5.47	161.1	1.73	7.10	132.5	0.61	3.74	180.5
2	0.13	133.33	7.46	186.9	1.68	7.61	152.7	1.08	5.62	198.2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18006	0.55	178.10	11.41	1.4	2.56	7.46	9.6	0.85	8.07	55.3
18007	0.54	178.88	10.21	3.9	2.54	7.35	9.1	0.00	0.00	0.0

2.2 Post Processor

Dynasim [1] is a numerical simulation software that combines sea environment conditions i.e. wind, current and waves through their statistical energy spectrum, as well as a platform's Response Amplitude Operators (RAOs) to obtain the resulting 6 DoF position, velocity and acceleration time-series. Figure 2a illustrates a model of the studied spread-moored platform in the Dynasim interface, the 18 mooring lines are shown in blue.

The platform's model, subject to each environmental condition, is simulated for periods of 11400s (approx. 3 hours), with an integration time-step of 0.5s. The resulting 6 DoF position, velocity and acceleration time-series are stored with a time-step of 1s. The platform is moored by 18 lines and has an equilibrium heading direction of 210° measured from north. Simulations are performed for a range of platform drafts from $8m$ to $21m$, corresponding to varying platform loads.

Since the meta-models are designed to predict roll motion statistics, rather than perform time-series prediction, the time-series obtained through dynamic simulation are analyzed and the maximum roll amplitude observed in the 3h period is extracted. Figure 2b illustrates 20 minutes of the roll angle time-series obtained from Dynasim simulation of condition 2680. During the first seconds the resulting motion is highly dependent on initial configuration, while subsequent dynamics are governed by the incident environmental conditions. In order to isolate their effects, a cutoff time t_{cutoff} of 2000s was implemented and time-series analysis was performed from this time forwards.



(a) Dynasim software interface showing model of simulated spread-moored platform.

(b) Roll angle time series corresponding to simulation 2680 obtained from Dynasim.

Figure 2. Dynasim Software Interface (a) and roll angle time-series example (b).

Maximum roll amplitude is measured relative to the platform's steady state equilibrium subject to no environmental conditions, so that a value of zero corresponds to no environment-induced mooring line tensions. Roll amplitude is mathematically given by:

$$\phi_{max} = \max_{t>t_{cutoff}} |\phi(t) - \phi_{eq}|.$$

3 Neural Simulator

The proposed Neural Simulator framework consists of 14 separately trained meta-models, each designed to predict roll amplitude for different FPSO drafts (Fig. 3). This modular framework allows for the application of Neural Architecture Search (NAS) techniques and individual fine-tuning of each ANN model.

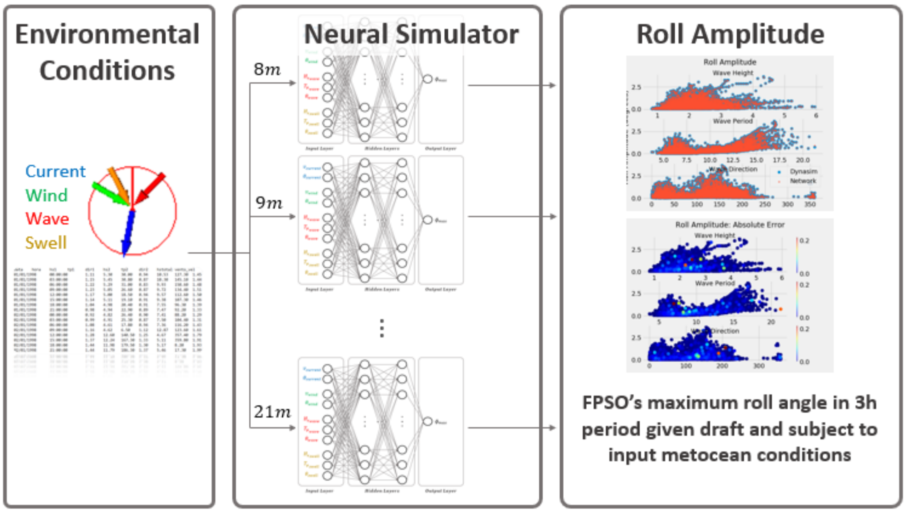


Figure 3. Neural Simulator Framework.

Since the prediction of roll motion statistics directly from environmental conditions is a regression task, any function approximation method can be implemented. In this work, a type of ANN known as MultiLayer Perceptron (MLP) was chosen due to its capability of stochastically learning complex functions from a large amount of data. MLPs are trained with the Back-propagation algorithm.

3.1 Data Preparation

In order to improve the numerical convergence of back-propagation learning algorithms, several techniques can be applied. These range from input transformation and weight initialization methods to batch learning and adaptive learning rates. This section focuses on Data Preparation techniques implemented on the environmental conditions prior to training.

Let $\mathbf{e} = (v_c, \theta_c, v_w, \theta_w, H_{s1}, T_{p1}, \theta_1, H_{s2}, T_{p2}, \theta_2)$ be a set of observed environmental conditions as defined previously. As angular variables are defined in $[0°, 360°]$, their periodic property implies that values such as $0.1°$ and $359.9°$ are functionally close despite being numerically distant. This can cause slow ANN convergence as similar environmental conditions may be far apart in the network input space. As a result, the projections of current velocity, wind velocity and wave height in the N-S and E-W directions were used, rather than their magnitude and incidence angle, so that the same set of environmental conditions can be represented as:

$$\begin{aligned}\mathbf{e}^{\mathbf{proj}} = (&v_c \sin(\theta_c), v_c \cos(\theta_c),\\ &v_w \sin(\theta_w), v_w \cos(\theta_w),\\ &H_{s1} \sin(\theta_1), H_{s1} \cos(\theta_1), T_{p1},\\ &H_{s2} \sin(\theta_2), H_{s2} \cos(\theta_2), T_{p2}).\end{aligned}$$

Since the values of different input variables have different orders of magnitude e.g., local wind velocity can be as high as 20 m/s while current velocity is lower than 1 m/s, a Gaussian Standardization method was applied to transform the projected environmental conditions into the network's input data:

$$x_i = \frac{e_i^{proj} - \mu_i}{\sigma_i},\ i = 1, ..., 10,$$

where μ_i and σ_i are the mean and standard deviation of the i-th variable on the complete dataset. This ensures that each input variable follows a normal distribution with zero mean and unit-variance and improves numerical convergence.

The resulting dataset was then divided into train, validation and test, as illustrated in Fig. 4. In order to accurately evaluate each model's generalization capabilities, 5-fold Cross Validation was performed and the average Mean Absolute Error (MAE) across all folds was used as the objective function to be minimized during Hyperparameter Optimization.

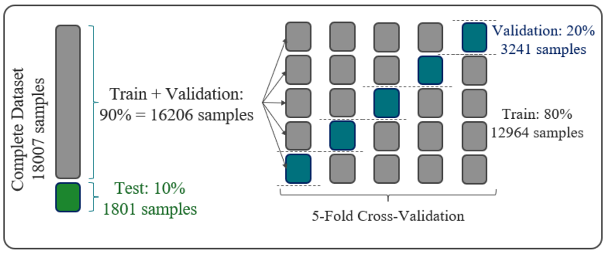


Figure 4. Dataset split for 5-fold Cross Validation.

3.2 Hyperparameter Optimization and Neural Architecture Search

While parameters such as node weights are learned during training, hyperparameters are related to the model selection task or the algorithm itself and can be used to control the learning process. Examples of hyperparameters in ANNs are the number of nodes in each layer, the learning rate and mini-batch size. Hyperparameter optimization consists in determining a model's optimal hyperparameters for a given task, and the process of optimizing ANN architectures is referred to as Neural Architecture Search (NAS).

Since the platform's response to metocean conditions is highly dependent on platform draft, NAS techniques were applied in order to find optimal MLP architectures as draft ranges from 8m to 21m. Specifically, a Bayesian Optimization algorithm known as the Tree-Structured Parzen Estimator (TPE) [7] was used in the Optuna python framework to determine the best number of neurons in each of the three hidden layers used in the MLP architecture. While other algorithms such as Grid Search, Random Search and Simulated Annealing were investigated, we found that Bayesian Optimization yielded better results. The algorithm performs iterative trials in which an MLP candidate architecture is chosen and the objective function (average Cross-Validation MAE) is evaluated. In each trial, TPE fits a Gaussian Mixture Model to the set of hyperparameters associated with the best objective function values and chooses the next MLP architecture in a promising region of the search space by maximizing expected improvement according to the history of previous trials.

For each of the 14 meta-models (one for each draft on the platform), TPE Bayesian Optimization was performed for 500 trials. During each trial, the investigated MLP architecture was trained and evaluated five times (once for each validation fold). In order to avoid intractable computational times, a reduced number of only 100 training epochs was used during Hyperparameter Optimization, as opposed to 5000 training epochs used in the final training of the proposed models. NAS was limited to the number of nodes in each of the three hidden layers, while other hyperparameters such as batch size and optimizer remained fixed as shown in Tab. 2

Table 2. Fixed and optimized MLP hyperparameters.

Optimized Hyperparameters	**Range**
Number of nodes per hidden layer	[20,4500]x[20,4500]x[20,4500]
Fixed Hyperparameters	**Value**
Number of hidden layers	3
Training Epochs	100
Batch Size	800
Optimizer	Adam
Learning Rate	0.001
Activation Function	ReLU

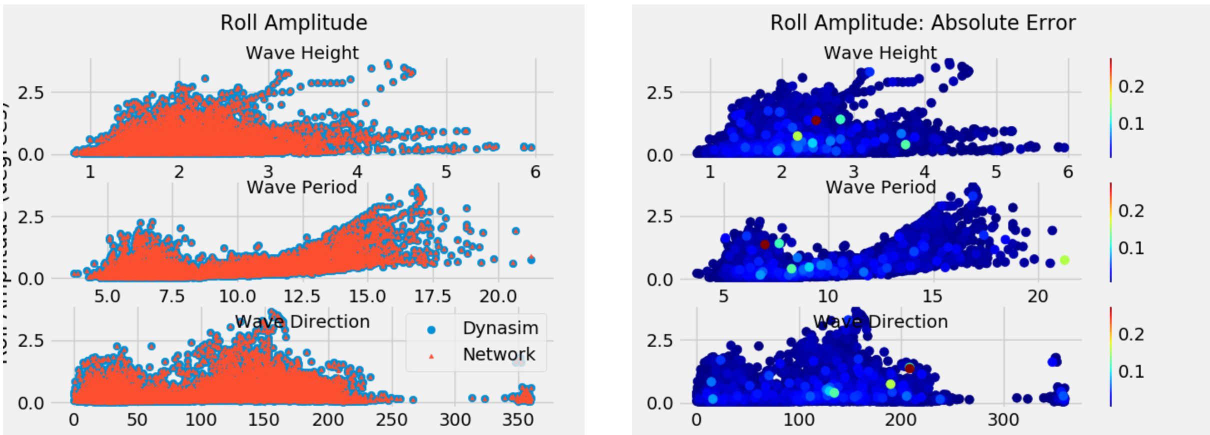
4 Results

Table 3 shows the best performing MLP architectures for each platform draft among the 500 TPE trials. It can be seen that the number of neurons in the first hidden layer ranged from 1311 (16m draft) to 4212 (12m draft), while in the second layer it ranged from 186 (17m draft) to 1214 (12m draft) and in the third, from 278 (11m and 17m drafts) to 4092 (18m drafts).

Table 3. Optimal MLP architectures for each platform draft.

Draft	8m	9m	10m	11m
MLP Architecture	1716-431-869	3896-535-689	2766-326-361	2238-280-278
Draft	12m	13m	14m	15m
MLP Architecture	4212-1214-1181	1555-1096-341	3341-1120-1644	2274-319-536
Draft	16m	17m	18m	19m
MLP Architecture	1311-501-1059	3314-186-278	1878-288-4092	1790-332-704
Draft	20m	21m		
MLP Architecture	2393-230-462	2256-215-1065		

Figure 5 illustrates the results obtained by the 14m draft model in comparison to Dynasim simulation. The proposed meta-model showed good results across all environmental conditions, with a maximum absolute error of 0.28°and Mean Absolute Error (MAE) of 0.002°. The error plot illustrates that errors in critical metocean conditions, where the wave period is close to the FPSO's natural period, are relatively small. Overall, the Neural Simulator presented competitive results when compared to traditional dynamic simulation methods.



(a) Meta-model (orange) and Dynasim (blue).

(b) Error plot.

Figure 5. Comparison between Dynasim and obtained MLP architecture for 14m draft. Roll Amplitude as a function of Wave Height (top), Wave Period (middle) and Wave Direction (bottom).

The proposed meta-models were tested both as individual MLP models and as ensembles of 5 and 10 MLPs, in which the final prediction is given by the average between the 5 and 10 best MLPs obtained through Bayesian Optimization, respectively. Table 4 summarizes the Mean Absolute Error (MAE) and Maximum Error for each platform draft in the three different settings.

5 Conclusions

The results obtained indicate that data-based surrogate models can be successfully used to capture complex roll motion responses of simulated FPSOs exclusively from environmental data, predicting relevant statistics without the limitations associated with traditional dynamic simulation methods. The observed error margins are competitive in comparison to the errors of dynamic models relative to real platform motion. This suggests that a set of meta-models trained directly on measured FPSO responses can provide more accurate results than traditional methods in reduced computational time.

The prediction performance of the MLP architectures obtained through TPE Bayesian Optimization when compared to that of random architectures indicate the value of robust hyperparameter optimization techniques in

Table 4. Proposed Meta-model results.

Draft	MAE			Max Error		
	Single	Ensemble (5)	Ensemble (10)	Single	Ensemble (5)	Ensemble (10)
8m	2.9°e-3	2.4°e-3	1.9°e-3	0.314°	0.389°	0.301°
9m	3.4°e-3	1.9°e-3	1.9°e-3	0.355°	0.293°	0.280°
10m	2.4°e-3	1.8°e-3	1.5°e-3	0.301°	0.265°	0.250°
11m	2.4°e-3	1.7°e-3	1.8°e-3	0.200°	0.216°	0.217°
12m	2.7°e-3	1.9°e-3	1.7°e-3	0.206°	0.242°	0.235°
13m	1.6°e-3	1.5°e-3	1.3°e-3	0.212°	0.199°	0.190°
14m	2.2°e-3	1.5°e-3	1.3°e-3	0.279°	0.228°	0.222°
15m	2.9°e-3	1.7°e-3	1.4°e-3	0.209°	0.225°	0.238°
16m	1.5°e-3	1.4°e-3	1.4°e-3	0.264°	0.249°	0.256°
17m	2.0°e-3	1.6°e-3	1.3°e-3	0.253°	0.220°	0.206°
18m	1.9°e-3	1.6°e-3	1.4°e-3	0.343°	0.211°	0.203°
19m	1.8°e-3	1.2°e-3	1.3°e-3	0.163°	0.220°	0.211°
20m	2.2°e-3	2.0°e-3	1.5°e-3	0.207°	0.210°	0.218°
21m	3.9°e-3	1.8°e-3	1.4°e-3	0.210°	0.233°	0.231°
Avg.	2.4°e-3 (0.63%)	1.7°e-3 (0.44%)	1.5°e-3 (0.39%)	0.251°(5.8%)	0.236°(5.5%)	0.233°(5.4%)

modern machine learning. Additionally, ensembles of 5 models seemed perform best in the trade off between low error metrics and overall training time.

Future work includes training the Neural Simulator architecture to predict other design variables, training with real measured FPSO motion and the optimization of additional hyperparameters such as activation functions. The prediction of other relevant design variables, such as maximum platform offset and fairlead displacements, is currently under study and has shown promising results.

Acknowledgements. This work was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES Finance Code 001), Brazil, and ANP/PETROBRAS, Brazil (project N. 21721-6). We also gratefully acknowledge partial support from CNPq (grants 310085/2020-9, and 310127/2020-3).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] K. Nishimoto, C. H. Fucatu, and I. Q. Masetti. Dynasim—A Time Domain Simulator of Anchored FPSO. *OMAE*, vol. 1, 2002.

[2] de A. C. Pina, C. H. Albrecht, de B. S. Lima, and B. P. Jacob. Wavelet network meta-models for the analysis of slender offshore structures. *Engineering Structures*, vol. 68, pp. 71–84, 2014.

[3] D. Sidarta, J. Kyoung, J. O'Sullivan, and K. Lambrakos. Prediction of offshore platform mooring line tensions using artificial neural network. OMAE2017-61942, 2017.

[4] G. M. Gonzalez, M. Q. Siqueira, M. L. Simão, P. M. Videiro, and L. V. S. Sagrilo. On the Use of Artificial Neural Networks for Estimating the Long-Term Mooring Lines Response Considering Wind Sea and Swell. OMAE2020-18868, 2020.

[5] M. Damasceno, H. R. Neto, T. Costa, A. Cavalini Júnior, L. Aguiar, and M. Martins. Using Kriging Surrogate Models to Predict the Vibration Responses of a Submerged Riser. OMAE2020-18915, 2020.

[6] J. M. Gumley, M. J. Henry, and A. E. Potts. A Novel Method for Predicting the Motion of Moored Floating Bodies. OMAE2016-54674, 2016.

[7] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, eds, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.