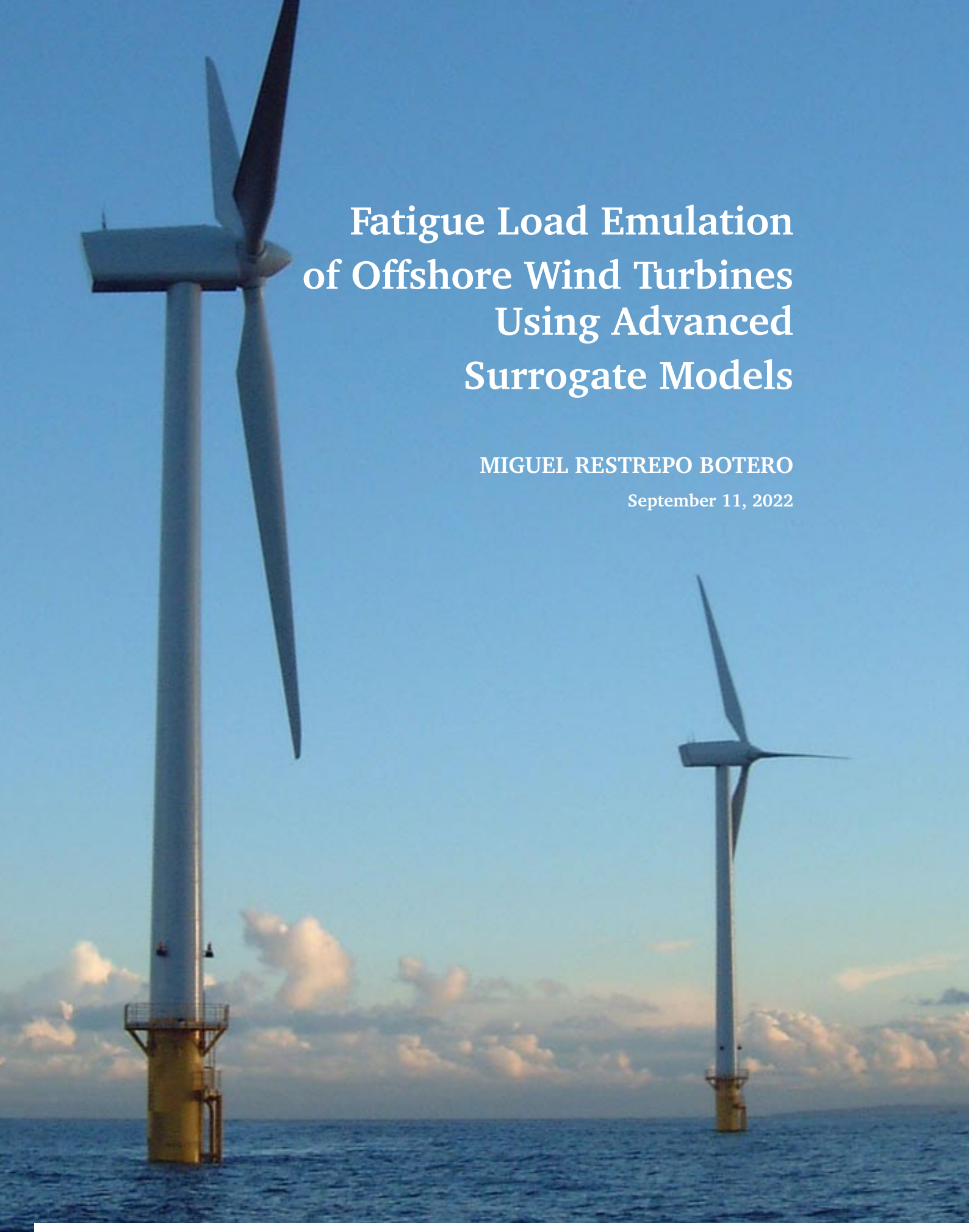


Fatigue Load Emulation of Offshore Wind Turbines Using Advanced Surrogate Models

MIGUEL RESTREPO BOTERO

September 11, 2022

TU Delft Delft University of Technology

SIEMENS Gamesa RENEWABLE ENERGY

Fatigue Load Emulation of Offshore Wind Turbines Using Advanced Surrogate Models

Master of Science Thesis

Author:	Miguel Restrepo Botero (5239486)	
Committee:	Dr. Ir. Carlos Simão Ferreira	TU Delft - Chairman
	Dr. Ir. Wei Yu	TU Delft
	Dr. Ir. Richard P. Dwight	TU Delft
	Dr. Ir. Anh Khoa Doan	TU Delft
	Dr. Ir. Alexandros Iliopoulos	Siemens Gamesa Renewable Energy
	Ir. Tobias Stinenbosch	Siemens Gamesa Renewable Energy

September 11, 2022

TU Delft Delft University of Technology

SIEMENS Gamesa RENEWABLE ENERGY

Summary

Understanding the fatigue load history of wind turbines in operation can be critical when taking decisions regarding the lifetime of a project. However, direct measurement of fatigue loads at each turbine in a wind farm is unfeasible, especially at critical locations such as the mudline level where placing a sensor is more expensive. For this reason, surrogate models offer a useful alternative. In this thesis, a methodology for creating surrogate models for emulating fatigue loads of offshore wind turbines is presented. The methodology is unique in that it accounts for the variability of site-specific conditions that may be present between wind turbines of the same power rating and pitch control behavior. First, a method for creating simplified structural models which depends only on a few degrees of freedom is derived. A sensitivity study is conducted on this simplified model to further understand the relationship between the degrees of freedom and the dynamics, and then the simplified model is validated against a corresponding detailed model of a real offshore site to ensure the accuracy of the load estimation of the simplified model.

After this, a database of simulation data is assembled by varying the geometric, dynamic, and environmental degrees of freedom selected from a range which captures a high degree of variability of site-specific conditions. Two different strategies are proposed for selecting the degrees of freedom of each simulation, one using a Monte Carlo method and one using the Sobol low-discrepancy sequence. Neural networks are trained on these data sets and their results are compared. It was found that using low-discrepancy sequences resulted in lower errors as measured by MAPE, MPE, RMSE, and the total DEL error than when using Monte Carlo sampling. In general, the load emulators were capable of predicting fatigue loads with a MAPE of 6% and the total DEL load is estimated with less than a 3% error in the majority of cases. A global sensitivity study using Sobol indices revealed that the standard deviation of the nacelle acceleration, the rotor speed, the wind direction, and the tower frequency were critical inputs for all the load emulators. Wave-related parameters proved to be important inputs for mudline and interface load estimation but not for tower top and blade loads.

Acknowledgements

I would like to first thank my supervisors at Siemens Gamesa Renewable Energy, Dr. Ir. Alexandros Iliopoulos and Ir. Tobias Stinenbosch for their guidance and support throughout the course of the thesis project. It was thanks to your dedication and encouragement that this project was made possible. I also express my gratitude to Dr. Ir. Wei Yu for her flexibility in supervising the thesis project and providing timely and valuable feedback that helped improve the quality of the thesis. I also thank Sachin and Stavros at SGRE for bringing this thesis opportunity to my attention, and Colinda for helping make the arrangements to make this a reality. Additionally, I extend a general thanks to all the employees at SGRE, with whom I always felt welcome.

I thank my friends at TU Delft for making this graduate program an enjoyable experience even through difficult times. I'd like to specially thank Jackson, Santiago, Maria, Irene, and Tadek with whom we shared the highs and lows of student life and formed friendships that will last a lifetime.

Most importantly I thank my parents and my sister Laura for their love and support. Without them I would not be who I am today. Their encouragement for me to pursue my passions are what made this program a possibility.

Contents

List of Figures

List of Tables

Nomenclature

Latin symbols

$\bar{u}$	Average wind speed
$a_{nacelle}$	Nacelle acceleration
d_{int}	External diameter at the interface level
d_{wl}	External diameter at the water level
f_1	First eigen-frequency in the fore-aft direction
f_2	First eigen-frequency in the side-side direction
i_{model}	Index of structural model
i_{tower}	Index of tower geometry
k	Stiffness
m	Wöhler slope that describes the fatigue properties of a material
n	Number of samples
n_{inputs}	Number of input units to the neural network
n_{models}	Number of structural models
t_{int}	Wall thickness at the interface level
t_{MG}	Marine growth thickness
t_{wl}	Wall thickness at the water level
u	Wind speed
x_{dof}	Vector which encodes the values of all degrees of freedom
z	Height above the water level (coordinate) OR z-score
$z_{\Delta d}$	Level to which the additional hydrodynamic diameter extends
$DEL_{predict}$	Damage equivalent load predicted by the load emulator
$DEL_{test,avg}$	Average damage equivalent load from the test data set
DEL_{test}	Damage equivalent load from the test data set
EF_i	i-th engineered feature
H_1	Structural transfer function of the first eigenmode
H_s	Significant wave height
M	Mass

$M_{b,edge}$	Damage equivalent bending moment at the blade root in the edgewise direction
$M_{b,flap}$	Damage equivalent bending moment at the blade root in the flapwise direction
$M_{b,tors}$	Damage equivalent bending moment at the blade root in the torsional direction
M_{int}	Damage equivalent bending moment at the interface level
M_{mud}	Damage equivalent bending moment at the mudline level
$M_{towertop}$	Damage equivalent bending moment at the tower top
N_{hidden}	Number of hidden layers in a neural network (depth of the neural network)
P_{ele}	Electrical power generated
R^2	Coefficient of determination
$S_{1,i}$	First order Sobol index of the i-th parameter
$S_{T,i}$	Total Sobol index of the i-th parameter
T_p	Peak spectral wave period
V_{ij}	Variance contribution of the interaction of the i-th and j-th parameter towards the total variance of the model
V_i	Variance contribution of the i-th parameter towards the total variance of the model
Z_{int}	Reference level of the interface
Z_{mud}	Reference level of the mudline

Greek symbols

α	Normalized deflection at the water level
α_{NN}	Learning rate
$\sigma(x)$	Standard deviation of x
$\theta_{misalign}$	Wind-wave misalignment
θ_{pitch}	Blade pitch angle
θ_{wave}	Wave direction
θ_{wind}	Wind direction
$\theta_{yaw,bearing}$	Yaw bearing angle of the rotor-nacelle assembly
ε	Finite difference
ζ_1	Damping ratio
Δd	Additional hydrodynamic diameter
Δz_{PPD}	Additional pile penetration depth
Ω	The domain of all degrees of freedom OR rotor speed

Others

$\mathcal{I}$	Integration of the loss function across the domain
$\mathcal{L}$	Loss or cost function

Abbreviations

BHawC	Bonus Horizontal Axis Windturbine Code
DEL	Damage Equivalent Load
DLA	Design Load Assessment
DLC	Design Load Case
DoF	Degree of Freedom
FA	Fore-Aft direction
FLM	Fatigue Load Monitoring
LAT	Lowest Astronomical Tide
LCOE	Levelized Cost of Energy
MAF	Modified Apparent Fixity
MAPE	Mean Absolute Percentage Error
MC	Monte Carlo
MPE	Mean Percentage Error
MSL	Mean Sea Level
NN	Neural Network
NSS	Normal Sea State
PCE	Polynomial Chaos Expansion
PPD	Pile Penetration Depth
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RMSE	Root Mean Square Error
RNA	Rotor-Nacelle Assembly
RS	Response Surface
SCADA	Supervisory Control and Data Acquisition
SGRE	Siemens Gamesa Renewable Energy
SM	Surrogate Model
SS	Soil Springs
tDELE	Total Damage Equivalent Load Error
TI	Turbulence Intensity

Chapter 1

Introduction

In this chapter, a general introduction of the research topic is provided. The motivation behind the topic of the thesis is described in section 1.1. Followed this, a summary of the literature review is included in section 1.2. Having presented the literature review, the key knowledge gaps and their respective research questions are described in section 1.3. In section 1.4, the general methodology used throughout the thesis is presented. Finally, section 1.5 describes the general outline that is followed for the thesis.

1.1 Motivation

The goals placed by nations and organizations for achieving a timely energy transition as a means to combat climate change place a large importance on the continued growth of the wind energy industry. The IEA roadmap for Net Zero Emissions by 2050 calls for a 11 fold increase of installed capacity from wind energy by 2050 [1]. To achieve this, annual additions of 390 GW by 2030 are required. Of these annual additions, 80 GW are budgeted to be sourced from offshore wind (Figure 1.1). This represents a 400% increase when compared to the added offshore wind capacity in 2021 (21.1 GW), which itself was 3 times that of 2020 [2]. In order achieve the planned growth for 2030 and 2050, the wind energy industry must continue to develop new technologies that will lower the Levelized Cost of Energy (LCOE), currently estimated at $77/MWh [2], in order to compete with other energy projects.

For this purpose, having a better understanding of the fatigue loads of an offshore wind turbine can lead to more accurate estimation of the damage accumulation of a turbine and aid decision-making of existing projects, lowering the levelized costs of projects. By being able to reconstruct the fatigue load history of a wind turbine, decisions regarding lifetime extension or controller upgrades can be made. Obtaining the fatigue loading history of a wind turbine is difficult, however. Direct measurement of loads is unfeasible due to the high costs and logistical challenges of installing strain gauges, particularly in the foundation.

As an alternative, Surrogate Models (SMs) can be used to estimate the loads

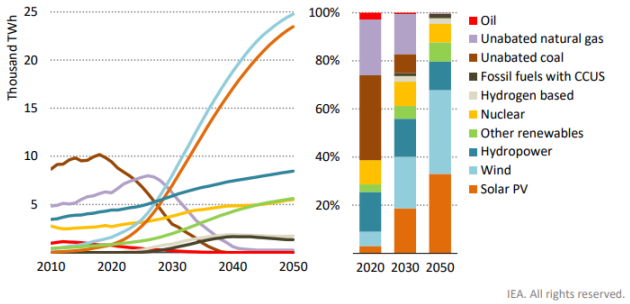


Figure 1.1: Global electricity generation by source in the IEA Net Zero Emissions by 2050 roadmap [1].

through a process of load emulation. The SM is provided with measurement data that is readily available at offshore wind turbines and the model predicts the fatigue loads that would be measured at the fatigue hotspots. By doing so, the fatigue loading history of a wind turbine can be recreated without the need of additional sensors.

Nevertheless, the SM approach is limited. Turbines of the same class may differ greatly in their site-specific conditions. Within the same windfarm, wind turbines can have different foundation designs, water depths, soil characteristics, among other differences, as illustrated in Figure 1.2. These differences affect their dynamics and therefore the expected behavior of the fatigue loads. A model derived on a single site-specific design will not perform well when tested on a separate design.

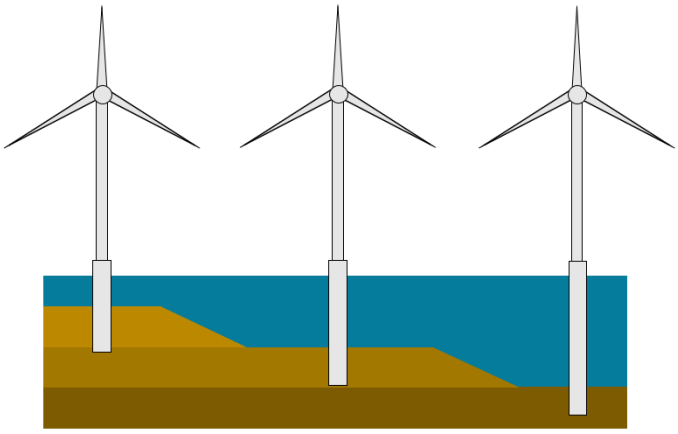


Figure 1.2: Differences in site-specific conditions among turbines of the same class.

A way to resolve this is by ensuring that the process for obtaining the SM accounts for the differences in design that may arise. Instead of building a SM for the specific site, a SM for the entire class is built. With this approach it is potentially possible to construct a single SM applicable to any turbine of the same operating behavior (power, rotor speed, and pitch control).

With this in mind, the goal of this thesis is to present a methodology for creating a generic SM (i.e. one that can be applied to any wind turbine of the same class) that emulates the fatigue loads during normal operation and idling conditions, at several fatigue hotspots along the foundation, tower, and blades. After this, the generic SM is tested against site-specific hydro-aeroelastic models in order to assess its performance in terms of various error metrics.

1.2 Literature review

In previous literature, SMs for load emulation have been applied successfully to offshore wind turbines. These models can be classified in two categories according to their underlying principles. These categories are (1) physics informed models and (2) data-driven models.

Physics informed models, also referred to as white- or grey-box models, emulate the fatigue loads using a physical relationship between the measurable inputs and the target outputs. For example, a fully white-box model can be a hydro-aeroelastic model used to run simulations using measured input wind and wave conditions. Grey-box models may use physics along with other supplemental data to reconstruct the fatigue load history. These methods include Kalman filter techniques [3, 4] and modal expansion methods [5–8].

On the other hand, data-driven models (i.e. black box models) use available data of known pairs of measurable inputs and target outputs to derive an empirical relationship between input and output. Examples of models that have been applied to load emulation in offshore wind turbines include Neural Networks (NNs) [9–17], Polynomial Chaos Expansion (PCE)[17–19], Response Surfaces (RS) [16, 17], and Kriging [19].

In addition to the different types of models, literature has also been divided in the target application of these models. Some load emulators have been used with environmental data as input, with the objective of performing Design Load Assessment (DLA), e.g. as part of an initial load calculation of a new project. Other load emulators have been used for Fatigue Load Monitoring of existing projects (FLM). Although the target variables in both cases are the same, the input variables for the two applications differ. In a design load assessment application, only inputs related to the wind-wave climate are used, while the application for load monitoring utilizes only measured signals such as SCADA data or accelerometer data. A summary of the load emulators that have been used in literature, together with their target application, is included in Table 1.1.

A comparison of the SMs that have been used in literature for load emulation is provided in Table 1.2. When comparing the SMs, the potential of NNs for use in load emulation is particularly appealing. Once trained, a NN has a very low computational cost and has been shown to achieve low errors in load prediction. Additionally, it has the advantage that it requires no calibration of physics-sensitive parameters that the physics-informed models require and, most importantly, does not explicitly require a

Table 1.1: Summary of literature on surrogate models for fatigue load emulation.

Method	Study	Data	Offshore?	Case
Kalman filter	Maes et al. [3]	Real	Yes	FLM
	Branlard et al. [4]	Sim.	No	FLM
Modal expansion	Baqersad et al. [5]	Real	No	FLM
	Iliopoulos et al. [8]	Real	Yes	FLM
	Noppe et al. [7]	Real	Yes	FLM
	Iliopoulos et al. [6]	Real	Yes	FLM
PCE	Murcia et al. [18]	Sim.	No	DLA
	Dimitrov [17]	Sim.	Yes	DLA
	Dimitrov et al. [19]	Sim.	No	DLA
Neural networks	Cosack [9]	Sim.	No	FLM
	Obdam et al. [10]	Real	Yes	FLM
	Smolka et al. [11]	Real	Yes	FLM
	Souliotis [12]	Sim.	Yes	FLM
	Lee [13]	Real	Yes	FLM
	Venu et al. [14]	Real	Yes	FLM
	De Nolasco et al. [15]	Real	Yes	FLM
	Schröder et al. [20]	Sim.	No	DLA
	Dimitrov [17]	Sim.	Yes	DLA
Quadratic RS	Schröder et al. [20]	Sim.	No	DLA
	Dimitrov et al. [19]	Sim.	No	DLA
Kriging	Dimitrov et al. [19]	Sim.	No	DLA

new SM to be constructed for each new application.

With that said, previous studies on the applicability of SMs for load emulation have focused solely on site-specific models. This means only data from a single site is used to train a SM. Although the resulting model performs well on data from the same site, they have failed to generalize well to different sites [13]. Different sites of the same class may differ in foundation design, reference levels, dynamic properties, environmental conditions, and other variables. This poses a challenge when trying to use a SM that has been built using a site-specific concept. A white or grey-box model requires a new physical model for the new site, and the black-box model cannot extrapolate beyond the domain of data it was trained on [21]. To be able to apply a black-box model on multiple sites, the model must be built using a generic approach. SMs that use a generic concept for gathering training data have yet to be studied in literature.

Table 1.2: Advantages and disadvantages of surrogate models for fatigue load assessment.

Model	Advantages	Disadvantages
Kalman filter	+ Based on physics (solution space is unbounded)	- Memory intensive - Requires a system model
Modal expansion	+ Based on physics (solution space is unbounded) + Higher accuracy than Kalman filter	- Memory intensive - Requires a structural model - Quasi-static loads difficult to model
Polynomial chaos expansion	+ Highly interpretable + Propagation of uncertainty easier to model + No physics required	- Depends on training data - Bounded solution space - Needs the distribution of input variables
Kriging	+ High accuracy + Interpretable + No physics required	- Depends on training data - Bounded solution space - High computational cost
Artificial neural networks	+ Low computational cost + High accuracy + Can be used with SCADA, accelerometer data + No physics required	- Depends on training data - Bounded solution space - Not easily interpreted

1.3 Research questions

As was discussed in the summary of the literature review, there has yet to be a SM developed for load emulation that is applicable for any wind turbine of the same class. Previous studies of data-driven models have used data from only one site. Therefore the performance of the SM failed to generalize to other sites. This is the principle knowledge gap that will drive the objective of the thesis. To address this knowledge gap, NNs are chosen as the SM for the reasons presented in section 1.2. The training data will be obtained via hydro-aeroelastic simulations as this data is more readily available than measurement data, is less noisy, and the samples can be carefully selected in order to ensure a large domain of inputs and thus a wide solution space. If measurement data were to be used, the solution space would be limited by the choice of sites where the measurements are obtained from, and the model would have difficulties generalizing beyond those cases.

In order to achieve a SM capable of generalizing to different sites, the data used to train this model must also be generic. This is to allow for the performance of the

SM on the training data to transfer well when testing on different sites. To obtain the data, simulations are to be performed on many structural models that represent various possible geometric configurations and foundation designs. This process requires having a structural model that can be easily modified to reflect differences between potential sites. The structural models are usually very detailed and are the result of a refined design process. Obtaining the models in this way is unfeasible as this would be extremely demanding and time consuming, at time requiring site-specific information that is not available in a generic concept (e.g. soil characteristics). For this reason, simplified models are preferred. In these models, reference detailed models are reduced to their most important characteristics, reducing the number of degrees of freedom required to describe the model. In this way, new models can be easily generated by selecting different combinations of degrees of freedom. Using this approach, the main research question is as follows:

> How accurately can artificial neural networks emulate the fatigue loads in fatigue-critical locations when trained with data from simulations of multiple simplified structural models?

In order to answer this question, a series of sub-questions are used to guide the research. Firstly, it is necessary to study the process with which the simplified models are created, and determine their accuracy with respect to detailed models. It is critical for the generic SM that the simplified structural models accurately represent their corresponding detailed model. Otherwise, the accuracy of the NN would not transfer well when testing on site-specific, detailed models. Therefore, the first sub-question is:

> Can a model with simplified tower and foundation geometry be used to accurately estimate fatigue loads, when compared to a model with corresponding detailed geometry?

When creating simplified models, choices are made regarding the modeling assumptions and the hyperparameters that control the generation of the simplified models. These choices will have an effect on the geometry, dynamics, and resulting fatigue loads of the simplified model. Therefore, understanding the effects of these choices plays a pivotal role in being able to create a generic model that can be used accurately as a substitute for the detailed model. To do this, a series of sensitivity studies on the simplified models are to be conducted. The sub-question related to this is:

> What is the sensitivity of the dynamics and fatigue loads of a model with simplified geometry, with respect to the hyperparameters and modeling assumptions?

As a product of the first two sub-questions, a methodology for generating generic simplified models has been developed. This allows the focus to be shifted towards the SM itself. To create a database that is generic (applicable to many possible sites), a

strategy for constructing a database that encompasses many geometric combinations and environmental conditions at a low computational cost is required. This leads to the sub-question:

> How should the environmental and geometric degrees of freedom be selected to create a training database that can be used to train a surrogate model with a broad solution space?

By answering this sub-question, a methodology for building a generic training database is developed. A load emulator can then be trained on this data. However, to optimize the training of the NN, it is beneficial to identify the key features that are relevant for fatigue load emulation. To do this, a final sub-question is formalized:

> Which SCADA and accelerometer variables and features should be used as inputs to the surrogate model for estimating fatigue loads?

Once the set of sub-questions are answered, the main research question can be answered fully.

1.4 Methodology

In this section, the methodology used in the thesis will be briefly described. Throughout the thesis, various SGRE in-house tools are used. Structural models are created using SGRE's Matlab-based tools. Various in-house tools are available for performing eigenvalue analysis or wave load generation based on linear wave theory on these models. These models are then transferred to input files that are compatible with BHawC (Bonus Horizontal Axis Windturbine Code), the in-house hydro-aeroelastic simulation engine. To perform the analysis, 10-minute simulations are used. Fatigue loads are calculated as Damage Equivalent Loads (DELs) using the in-house post-processing tools. These tools count the number of loading cycles and output the corresponding DEL load using Dirlik's method [22].

For training the NN, the results are transferred to Python to make use of the TensorFlow library. This library was chosen because it allows for fast implementation of a feed-forward neural network. Results are quantified in terms of their error using several metrics, namely Mean Absolute Percentage Error (MAPE), Mean Percentage Error (MPE), Root Mean Square Error (RMSE), total DEL error (tDELE) and the coefficient of determination (R^2).

1.5 Thesis framework

In this section, the outline of the thesis is presented. The research questions identified in section 1.3 are used to organize the structure of the thesis.

In chapter 1, the thesis topic is introduced. A motivation for the topic is first presented, followed by a summary of a comprehensive literature review on the subject

of fatigue load emulation of wind turbines. The knowledge gaps are identified and the research questions are presented.

Followed this, chapter 2 briefly presents the theoretical framework of NNs. This is done here because many of the choices performed throughout the thesis are driven by the strength, limitations, and requirements of NNs.

Then, chapter 3 describes the process in which simplified structural models are created, the choices in modeling assumptions and degrees of freedom, as well as sensitivity studies regarding the influence of the hyperparameters on the dynamic response of the model and on the wave loads generated based on this model.

In chapter 4 the methodology for creating simplified structural models is validated by comparing the fatigue loads of the simplified model against a corresponding detailed model. This is done for cases in which the wind turbine is in normal operation and in standstill.

Afterwards, chapter 5 presents the methods by which the different environmental conditions, geometric properties, and dynamic properties are selected in order to create a generic database of simulation data.

In chapter 6 shows the process of training and testing of the load emulators. The load emulators are tested against simulation data from detailed structural models that lie within the solution space of the training data. This chapter is key in that it tests the accuracy of the SM and its ability to generalize to any wind turbine geometry and environmental condition.

Lastly, chapter 7 contains an analysis of the relative importance of the input features using a global sensitivity analysis. Some permutations of the load emulators are considered in this section, namely to test the accuracy of the surrogate if some of the input data used (such as wave statistics) were not available.

Finally, chapter 8 provides a conclusion on the applicability of a generic SM for load emulation, along with recommendations for future work.

Chapter 2

Neural Networks as Surrogate Models

A neural network is a type of regression and/or classification algorithm that utilises a series of simple successive regressions to estimate outputs from a given set of inputs. The concept of a neural network model is loosely inspired by the firing mechanisms of neurons in the brain. A neural network contains many *units*, or neurons, arranged in l layers. Each unit holds a value a denoted the unit's *activation*. The first layer is referred to as the input layer, while the last layer is referred to as the output layer. All the layers in between are hidden layers. A schematic representation of a neural network is shown in Figure 2.1.

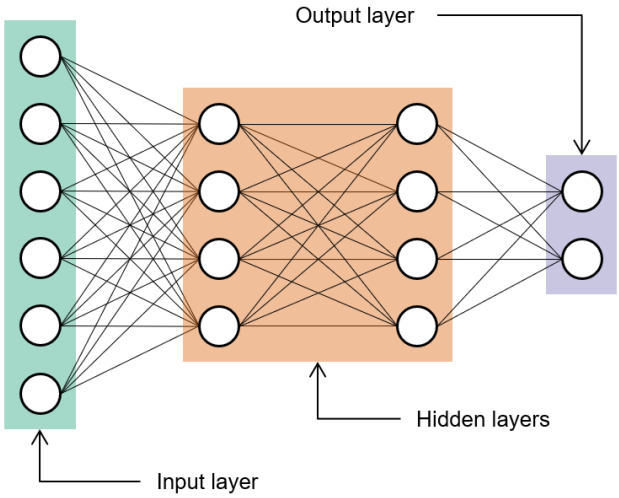


Figure 2.1: Diagram of a typical neural network with two hidden layers.

In a simple feed-forward neural network, the activation of any given unit is a nonlinear function of the activations of all the units in the previous layer. To compute the nonlinear function, first a linear combination of the previous layer's activations is

calculated. This intermediate value is referred to as z. To calculate z, each activation of the previous layer's units is assigned a weight value w and the combination of all the units is summed, together with a bias term b. Then, a nonlinear function is applied to z, in the form $a = g(z)$, where g is a function such as the sigmoid function, hyperbolic tangent, or a rectified linear unit (ReLU). Using these nonlinear activation function in layers allows for neural networks to achieve high levels of nonlinearity in their output. When this process is applied sequentially, starting from the input units until the last layer L is reached, the process is called forward propagation.

For each unit, each weight and bias term used to calculate its activation is a parameter of the model. Therefore, a neural network with many layers and units can have a high number of parameters to tune so that the model has the desired outputs. For this, supervised learning is used. In supervised learning, a large group of known inputs and outputs are used to calibrate the model by comparing the predicted outputs and the known outputs to a given set of inputs, and then adjusting the parameters in such a way that the errors of the predictions are minimized. The method for determining how to adjust each parameter follows a workflow process that starts in the output layer and works its way backward all the way up to the first layer. This process is called backward propagation.

2.1 Forward and backward propagation

In this section, forward and backward propagation is described with mathematical rigor. The following theoretical background is adapted from [23]. The neural network takes in an input x and produces an output y. Here, x is a column vector of the values of each input feature of a single data point. In a generalized form, for m examples, $Y = f(X)$, where X and Y are the set of inputs and outputs in matrix form:

$$X = \left[x^{(1)}, x^{(2)}, ..., x^{(m)}\right] \tag{2.1}$$

$$Y = \left[y^{(1)}, y^{(2)}, ..., y^{(m)}\right] \tag{2.2}$$

Where $x^{(i)}$ and $y^{(i)}$ are the column vectors of inputs and outputs, respectively, of the i-th data point.

In a feed-forward neural network a layered structure is used to map the inputs X to the output Y. In this way, the activations of each layer are determined by the activations of the previous layer. If X is considered as the activations of the 0-th layer and Y the activations of the L-th layer, the activation $A^{\{l\}}$ of any layer l is given by:

$$A^{\{l\}} = g\left(W^{\{l\}} A^{\{l-1\}} + b^{\{l\}}\right) \tag{2.3}$$

Where $W^{\{l\}}$ is a matrix of weights that tune the sensitivity of each unit in layer l to the values of the previous layer. In the weight matrix, w_{ij} contains the weight applied to the j-th unit in the previous layer for computing the i-th unit of the next

layer. The linear combination $Z^{\{l\}} = W^{\{l\}}A^{\{l-1\}} + b^{\{l\}}$ can be seen as a simple linear regression. The power of neural networks comes from subjecting this linear combination to a nonlinear activation function, denoted g. This function, typically a ReLU function, sigmoid, or hyperbolic tangent, is used to introduce a degree of nonlinearity which can allow the neural network to model more complex functions. The process of calculating the activations of each layer from the activations of the previous is known as forward propagation. The procedure is outlined in pseudocode in Algorithm 1.

Algorithm 1 Forward Propagation

procedure FORWARDPROP($X, W^{\{1\}}, \ldots, W^{\{L\}}, b^{\{1\}}, \ldots, b^{\{L\}}$)
  $A^{\{0\}} = X$
  for $l = 1$ `to` L **do**
    $Z^{\{l\}} = W^{\{l\}}A^{\{l-1\}} + b^{\{l\}}$
    $A^{\{l\}} = g^{\{l\}}\left(Z^{\{l\}}\right)$
  end for
  return $A^{\{1\}}, \ldots, A^{\{L\}}$
end procedure

In order to achieve the desired response of the model, the weights and biases need to be calibrated in such a way that the predicted outputs are close to the known outputs. For this, a large set of training examples are used. These are data points where the input and outputs are already known. The inputs are fed through the model and the predicted outputs $A^{\{l\}}$ are compared to the known outputs Y through a loss function $\mathcal{L}$. Tuning the weight and bias terms is treated as an optimization problem in which the objective is to minimize the total losses of the training data. It is typical for these optimization algorithms to use the gradient of the loss function to search for a local minimum. To determine the derivatives with respect to each weight and bias term, chain rule is applied from the output layer to the input layer in a method called backward propagation. For the last layer:

$$\frac{\partial \mathcal{L}}{\partial W^{\{L\}}} = \frac{\partial \mathcal{L}}{\partial Z^{\{L\}}}\frac{\partial Z^{\{L\}}}{\partial W^{\{L\}}} = \left(\frac{\partial \mathcal{L}}{\partial A^{\{L\}}}\frac{\partial A^{\{L\}}}{\partial Z^{L}}\right)\frac{\partial Z^{\{L\}}}{\partial W^{\{L\}}} = \frac{\partial \mathcal{L}}{\partial A^{\{L\}}}g'(Z^{\{L\}})A^{\{L-1\}} \tag{2.4}$$

For the second to last layer:

$$\frac{\partial \mathcal{L}}{\partial W^{\{L-1\}}} = \frac{\partial \mathcal{L}}{\partial Z^{\{L\}}}\frac{\partial Z^{\{L\}}}{\partial A^{\{L-1\}}}\frac{\partial A^{\{L-1\}}}{\partial Z^{\{L-1\}}}\frac{\partial Z^{\{L-1\}}}{\partial W^{\{L-1\}}} \tag{2.5}$$

$$\frac{\partial \mathcal{L}}{\partial W^{\{L-1\}}} = \left(\frac{\partial \mathcal{L}}{\partial Z^{\{L\}}}\right)W^{\{L\}}g'(Z^{\{L-1\}})A^{\{L-1\}} \tag{2.6}$$

And more generally:

$$\frac{\partial \mathcal{L}}{\partial W^{\{l-1\}}} = \left(\frac{\partial \mathcal{L}}{\partial Z^{\{l\}}}\right)W^{\{l\}}g'(Z^{\{l-1\}})A^{\{l-1\}} \tag{2.7}$$

$$\frac{\partial \mathcal{L}}{\partial Z^{\{l-1\}}} = \left(\frac{\partial \mathcal{L}}{\partial Z^{\{l\}}} \right) W^{\{l\}} g'(Z^{\{l-1\}}) \tag{2.8}$$

This procedure is applied starting from the last layer step-by-step until the first layer is reached. A similar method is used to determine the derivatives with respect to the bias terms $b^{\{l\}}$. The procedure is outlined in pseudocode in Algorithm 2. Once the gradient is calculated with backward propagation, the parameters can be updated. A detailed explanation is provided in section 2.4.

Algorithm 2 Backward Propagation

procedure BACKPROP($Y, X, Z^{\{1\}}, A^{\{1\}}, W^{\{1\}}, b^{\{1\}}, \ldots, Z^{\{L\}}, A^{\{L\}}, W^{\{L\}}, b^{\{L\}}$)
  $m =$ number of columns of Y
  $A^{\{0\}} = X$
  $dZ^{\{L\}} = \left(A^{\{L\}} - Y\right) \cdot g'^{\{L\}}$ ▷ If the loss function is Mean Squared Error
  $dW^{\{L\}} = 1/m \cdot dZ^{\{L\}} \cdot A^{\{L-1\}^T}$
  $db^{\{L\}} = 1/m \cdot dZ^{\{L\}} \cdot \text{ones}(m, 1)$
  for $l = L - 1$ `to` 1 **do**
    $dZ^{\{l\}} = W^{\{l+1\}^T} \cdot dZ^{\{l+1\}} * g'^{\{l\}}\left(Z^{\{l\}}\right)$
    $dW^{\{l\}} = 1/m \cdot dZ^{\{l\}} \cdot A^{\{l-1\}^T}$
    $db^{\{l\}} = 1/m \cdot dZ^{\{l\}} \cdot \text{ones}(m, 1)$
  end for
  return $dW^{\{1\}}, db^{\{1\}}, \ldots, dW^{\{L\}}, db^{\{L\}}$
end procedure

2.2 Bias and variance

Two important concepts when assessing the performance of a neural network are bias and variance. To understand the concepts of bias and variance, it is useful to separate the data into a training set used during backward propagation to determine the weights and biases, and test set which the model has never seen before.

A model with high bias is unable to fit the training data well and will therefore also perform poorly on the test data. The loss function $\mathcal{L}$ will be large in both cases. When this happens, it is said that the model underfits the data. A model with high variance will perform poorly on test data when compared to the training data. Therefore $\mathcal{L}_{Test} >> \mathcal{L}_{Train}$. In this case, it is said that the model overfits the training data.

To diagnose if a model has high bias and/or high variance, it is useful to plot learning curves. In these plots, the loss function of the training and test sets are plotted as a function of m, the number of training examples used. If the loss function converges to a high value for both training and test sets, then the model has high bias. If there's a large gap between the converged values of the loss function of the training set and the test set, then the model has high variance. Additionally, if the

loss function does not converge asymptotically to a value, then it is indicative that not enough training examples were used.

The following strategies can be used for solving problems of bias and variance. To reduce bias, more layers and/or units per layer can be added to the neural network in order to increase the complexity of the model and allow for more nonlinearities. Alternatively, it's possible that better results can be achieved when adding more input features. On the other hand, to reduce variance, using more data in the training phase can be useful, as well as introducing regularization to the training of the model. More details on regularization methods are given in section 2.3.

For large neural networks, the amount of parameters that are tuned during the training phase makes these models more prone to overfit than to underfit. This holds especially true for deep neural networks with many hidden layers. As a consequence, a large focus is placed on acquiring more data. For more shallow neural networks, there is a higher risk of underfitting.

2.3 Regularization

Regularization is a strategy that can be employed to reduce the risk of overfitting the data during training. The intuition behind regularization is that high variance comes from weight terms that are very high. High weight terms means that small changes in the input will have large changes in the output thanks to being scaled by a large weight. Regularization methods try to keep weight terms low while still minimizing the loss function as much as possible.

One of these strategies is L2 regularization. To implement this regularization, an additional term is added to the loss function, denoted $\mathcal{L}_{Reg}$. This term scales with the magnitude of the coefficients in the weight matrices W. By applying this to the loss function, the algorithm will be penalized if coefficients are too large, and therefore will be incentivized to keep them small. By minimizing the combined loss function $\mathcal{L}_{Total} = \mathcal{L} + \mathcal{L}_{Reg}$, the optimization algorithm will not only try to minimize the error of the model, but will also try to keep the parameters small. The loss due to regularization is expressed as the sum of the Frobenius norms of all weight matrices:

$$\mathcal{L}_{Reg} = \frac{\lambda}{2m} \sum_{l=1}^{L} \left|\left| W^{\{l\}} \right|\right|_F^2 \tag{2.9}$$

where λ is the regularization parameter and controls the degree to which L2 regularization is applied. Higher values of λ will increase bias and decrease variance. Note that when L2 regularization is used, the backward propagation algorithm shown in Algorithm 2 should be slightly modified to include the loss introduced by regularization in the derivative terms.

A second strategy that can be used to prevent overfitting is dropout regularization. In this method, the model is taught not to rely too heavily on any given connection between units of consecutive layers by randomly eliminating units during training.

This forces the magnitudes of weight terms to be similar and small, and thus has a regularizing effect. For each layer a probability p of eliminating a unit on the layer is chosen. This probability is used to determine if each activation value should be kept or eliminated (set to zero). After this, the activations are re-scaled by dividing by the probability that an activation is kept $(1 - p)$. This is done so the activations of susequent layers keep the same magnitude. A description on how this is applied in practice is shown in pseudocode in Algorithm 3. When dropout regularization is used, dropout should only be applied with the training set. During testing, the full, densely connected neural network should be used. Like for L2 regularization, increasing the parameter p will increase the regularization effect, thus increasing bias and lowering variance.

Algorithm 3 Dropout regularization

procedure APPLYDROPOUT(A, p)
  $\varepsilon = 1 - p$
  $d =$rand(shape(A)) $< \varepsilon$ ▷ d is a matrix of logicals
  $A = A * d$ ▷ Element-wise product
  $A = A/\varepsilon$
  return A
end procedure

2.4 Optimization algorithms

When the gradient is computed using a backward propagation algorithm like the one shown in Algorithm 2, the newly computed gradient can be used to update each parameter in an effort to minimize the loss function. In its simplest form, this process is known as gradient descent. Other complexities can be included to make gradient descent more efficient. Each iteration of gradient descent takes a step in the direction opposite of the gradient (since the objective is to minimize the loss function) proportional to the gradient itself, scaled by a learning rate parameter α_{NN}. Weight and bias parameters are updated as follows:

$$W^{\{l\}} := W^{\{l\}} - \alpha_{NN} \left(\frac{\partial \mathcal{L}}{\partial W^{\{l\}}} \right) \tag{2.10}$$

$$b^{\{l\}} := b^{\{l\}} - \alpha_{NN} \left(\frac{\partial \mathcal{L}}{\partial b\{l\}} \right) \tag{2.11}$$

This is a hyperparameter of the model and can be tuned to improve the performance of the neural network. Increasing α_{NN} generally leads to faster convergence to the local minimum, however, it may cause the gradient descent algorithm to be unstable and the solution will diverge.

When training with large datasets, it can be inefficient to wait until all the training examples have gone through forward propagation and backward propagation to take a single gradient descent step. A small subset of the entire training set can already provide gradient descent with a general direction that it should search in. Therefore, it can be advantageous to divide the training set in smaller batches of data and apply forward propagation, backward propagation, and update the parameters for each batch sequentially. When gradient descent has been calculated for all batches, it is said that one epoch has been completed. Using gradient descent with batches can sometimes lead to the gradient descent steps being taken in many different directions. To reduce some of the noise encountered in the gradient, it can be beneficial to implement gradient descent with momentum. This algorithm using an exponentially weighted moving average to "remember" the gradient of previous batches and weighs it with the gradient of the new batch.

Further acceleration of convergence of gradient descent can be achieved by dividing the gradient by the magnitude of the exponentially weighted moving average of the gradient. This procedure is known as RMSProp. The Adam algorithm uses a combination of momentum and RMSProp to speed up gradient descent. This algorithm is one of the more commonly used optimization algorithms for training neural networks, and will be the optimization scheme that will be used throughout this thesis.

Chapter 3

Simplified Support-Structures

In order to create a load emulator that can predict fatigue loads at many possible sites, a method for creating generic structural models that accurately approximate the response of a wind turbine is required. The generic models should be easy to modify in order to create many permutations of possible wind turbines. In this chapter, a methodology for creating and fine-tuning generic, simplified models is presented. In section 3.1, the modeling assumptions and degrees of freedom selected for the model are presented. This is followed by a sensitivity analysis in section 3.2, whereby the dynamic response and hydrodynamic properties of the simplified models are studied with respect to the degrees of freedom and hyperparameters. Finally, a methodology for selecting the hyperparameters to fine-tune the simplified model is presented in section 3.3.

3.1 Modeling assumptions and degrees of freedom

A first step towards creating a load emulator that can represent loads of many possible sites of the same turbine type is to build a model which is generic and adaptable to fit the properties of any given site. To obtain the generic model, the approach taken is to start with a detailed model and simplify it, stripping away details in the model that are of low importance. The goal of the simplification process is to reduce the number of degrees of freedom (DoFs) of the detailed model as much as possible, such that the resulting model conserves only the essential parameters useful to describe the turbine's dynamics, operation, loading, and response. This simplification process is essential for limiting the computational expense of creating permutations of the model. Nevertheless, the simplification must preserve a high degree of versatility, whereby many possible configurations can be represented by few DoFs. For this purpose the following degrees of freedom related to the structural model are selected:

- Reference levels: Mudline level (Z_{mud}) and interface level (Z_{int}).
- Geometric properties: External diameter at the interface level (d_{int}), wall thickness at the interface (t_{int}) and external diameter at the water level (d_{wl}).

- Dynamic properties: Fore-aft frequency (f_1), damping ratio (ζ_1), and normalized displacement at water level (α).

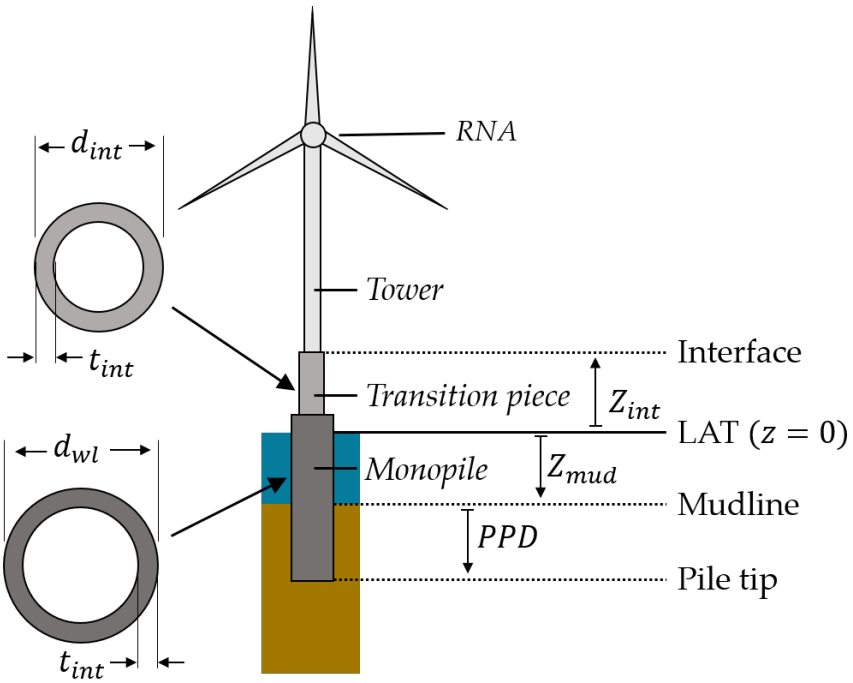


Figure 3.1: Schematic representation of the geometry of an offshore wind turbine and the selected DoFs for simplified modeling of the support structure.

Figure 3.1 shows a schematic diagram of an offshore wind turbine where the selected DoFs are illustrated. The mudline and interface levels (Z_{mud} and Z_{int} respectively) are chosen as DoFs as these are fundamental to describing the geometry and dynamics of the support structure. Dynamic properties such as the first mode frequency in the fore-aft direction (f_1) and the normalized deflection at water level (α) are strongly related to the mudline level, while wave loading can be highly dependent on the water depth and mudline level [24, 25]. The interface level is preserved as this is a hotspot for fatigue loading and loads at this location can be important drivers in the design of offshore wind turbines. For this same reason it was chosen to include the diameter and wall thickness at the interface level as DoFs. The diameter at the water level (d_{wl}) is chosen as a DoF because, as dictated by the Morison equation, the wave loading is highly dependent on the external diameter and the loading occurs primarily near the water level. This diameter is then extended to the mudline and pile tip, therefore the pile diameter and other possible changes to diameter below the water level are not considered. This choice was motivated by the desire to limit the computational cost of the generation of generic models. Including these parameters would increase the number of DoFs and therefore the computational cost of the type-generic process. Furthermore, as it will be discussed in section 3.3, the differences

in the dynamics of the simplified model arising from omitting these parameters are expected to be bridged by a fine-tuning process.

Finally, three key dynamic properties of the first fore-aft mode are selected as DoFs: the frequency (f_1), damping ratio (ζ_1), and the normalized deflection at water level (α). The frequency and damping are key parameters that describe the response of the model to dynamic loads [26]. They are tied to the amplification of loads occurring at the resonant frequencies and therefore have a high impact on the fatigue loads [27]. Treating them as DoFs adds versatility to the simplified model as the same geometry can be independently adjusted to represent different possible dynamic responses, therefore representing variabilities of different offshore sites. This is not only helpful to capture differences between sites due to site-specific conditions such as soil stiffness, pile penetration depth, or support structure design, but also potentially could be used to close the gap between measured and simulated dynamic properties. While the scope of the thesis is using simulated data, this is a key outcome for the practical applicability of the method.

Similarly, the α value is also treated as a DoF as this parameter is crucial in understanding the relationship between the stiffness of the support structure relative to the stiffness of the tower. As such, it is expected that the parameter will play an important role in the model in terms of hydrodynamic loading and the interaction between the hydrodynamic loads and the foundation displacements [28]. Therefore this parameter is important for modeling wave induced fatigue loads.

3.2 Sensitivity analysis

In order to understand the impact of the different DoFs that were selected for the simplified support structure model, a sensitivity analysis is conducted. For this, a reference detailed model provided by SGRE, representative of a 7.0MW wind turbine in the German North Sea, is simplified. First, the DoFs related to the structure dynamics are modified within a range of values in a one-at-a-time approach and the dynamic response is studied in order to further understand the impact of the selected DoFs. After this, the DoFs and other hyperparameters which relate to the hydrodynamics (and therefore the wave loading) are modified one-at-a-time and the resulting wave loads (generated with SGRE in-house software) are studied in terms of accuracy with respect to wave loads generated using the detailed model.

3.2.1 Dynamic properties

First, the dynamic properties are studied when perturbations are applied to the geometric parameters of the simplified model. Four hyperparameters which are expected to play a significant role in the dynamics of the wind turbine are selected. The chosen parameters are: t_{int} (which extends from interface level to pile tip), d_{int}, d_{wl}, and the pile penetration depth (PPD).

The analysis consists of varying each parameter independently within a range of -25% and $+25\%$ of the value of the detailed model. For most of the parameters, a variance of $\pm 25\%$ is considerably high and can be considered nonphysical for a detailed model. It is unlikely for turbines of the same type to exhibit such large variations for thickness and diameter. Nonetheless, the objective of the sensitivity study is to describe the impact of the modeling assumptions that are used. While the d_{wl} for detailed models may vary only slightly, the simplified model uses this value for the diameter of the foundation, from water level to pile tip. In the simplified model, the d_{wl} is used to define both the diameter at water level as well as the pile diameter. With this in mind, the the range of $\pm 25\%$ can be physically realistic, as the typical variations of pile diameters are expected to fall within this range. Similarly, an increase of 25% in t_{int} can capture the variations in foundation thicknesses along the monopile. On the other hand, decreasing the thickness 25%, is however largely unrealistic, as such a large decrease is likely to result in a structure that does not satisfy the design criteria for extreme and fatigue loading. This is not problematic as the scope of the sensitivity study is to study the interaction between the DoFs and the dynamic response, regardless of the physical interpretation of the perturbations that are applied.

For each DoF that is varied, the dynamic response of the resulting model is evaluated. The dynamic properties of interest are f_1, α, the frequency of the second mode in the fore-aft direction (f_2), and ζ_1. These four responses are intrinsically linked to the response of the model under dynamic loads, playing a role in the fatigue loads produced from wind and wave conditions. By analyzing these dynamic responses, the effects of the simplification process and the sensitivity of the model to these changes are shown.

Table 3.1: Summary of the geometric degrees of freedom of the detailed and simplified models.

DoF	Detailed	Model 1	Model 2
Z_{int}	✓	✓	✓
Z_{mud}	✓	✓	✓
Z_{pile}	✓	✓	✓
t_{int}	✓	✓	✓
t_{mud}	✓		
t_{pile}	✓		
d_{int}	✓	✓	✓
d_{wl}	✓	✓	✓
d_{mud}	✓		✓
d_{pile}	✓		✓
Coning angle	✓		✓

For the sake of comparison and for assessing the impact of the simplification strategies, a simplified model with an alternate geometric definition which requires

more DoFs is considered at this stage. By using more DoFs, this geometry is a more accurate approximation of the geometry of the detailed model. As such, it is expected that it will also show a more similar response to the detailed model. The two geometries are compared to the detailed model in Figure 3.2. In addition to the DoFs of the simplified geometry derived in section 3.1, this model introduces a geometric coning near the water level ($z = 0$) and the pile diameter as additional DoFs. A summary of the DoFs of the two models is presented in Table 3.1. While the practical applications of this model are limited, the results of the sensitivity analysis on this model can be indicative of overall trends in the detailed model as hyperparameters are changed between sites.

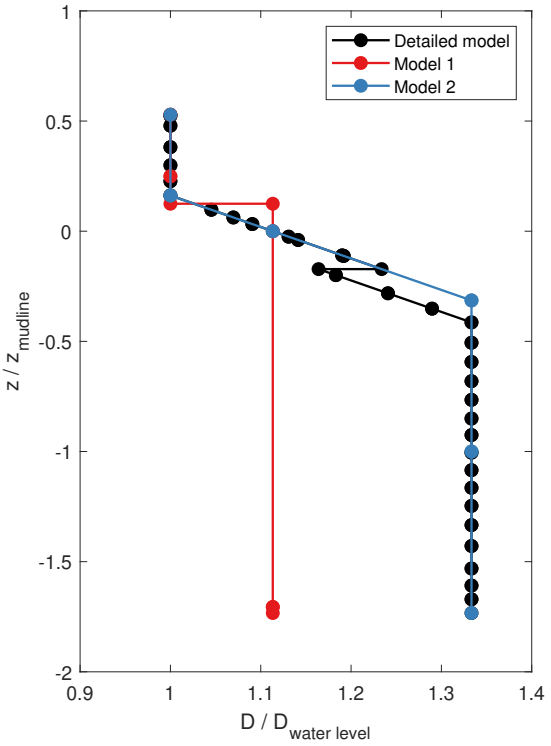


Figure 3.2: Comparison of the geometry of the two simplified models to the detailed model.

The analysis is carried out for the two modeling strategies, and for each modeling strategy two soil models are adopted: the modified apparent fixity model (MAF) [29] and the traditional soil spring model. For the cases which use a soil spring model, the soil springs are taken from the reference detailed model which also uses this soil model.

It should be noted that the sensitivity analysis on the simplified models are carried out using a different SGRE in-house tool (referred to herein as “Tool 2”) than what

was used to gather the values of the detailed model which are used to normalize the results (referred as "Tool 1"). Because of this the dynamic properties of the detailed model as evaluated by Tool 2 are also reported.

In Figure 3.4 the sensitivity of the first mode frequency in fore-aft direction is presented. The results are compared to the values of the reference model. The trends of all the models are consistent with our understanding of the underlying physics. For example, it is shown that increasing t_{int}, d_{int}, and d_{wl} has a stiffening effect, increasing f_1. To illustrate this effect, the wind turbine can be approximated by a cantilever beam with a lumped mass M and a stiffness k. The frequency of this system is given by,

$$f = \frac{1}{2\pi}\sqrt{\frac{k}{M}} \tag{3.1}$$

The stiffness is proportional to the second moment of area of the cross section I, which for a hollow, thin-walled cylinder approximately scales with the cube of the external diameter (d^3) and linearly with the wall thickness (t). Because d_{int} is only extended between the interface level and water level, its overall effect on the frequency is less than that of d_{wl}.

A similar analysis is applied for understanding the effects of modifying the PPD. It is seen that increasing PPD has the tendency to decrease f_1 for the MAF models, while having little effect for the Soil Spring (SS) models (Figure 3.4(d)). Internally, the MAF model applies a rigid constraint to the pile tip and replaces the beams below mudline with beams with modified stiffness. Increasing the PPD has the effect of increasing the overall length of the tower. This is illustrated in Figure 3.3.

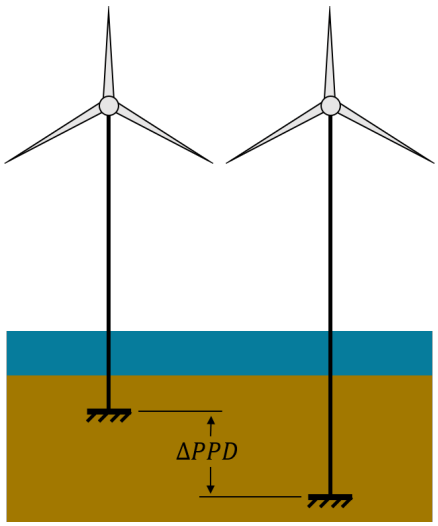
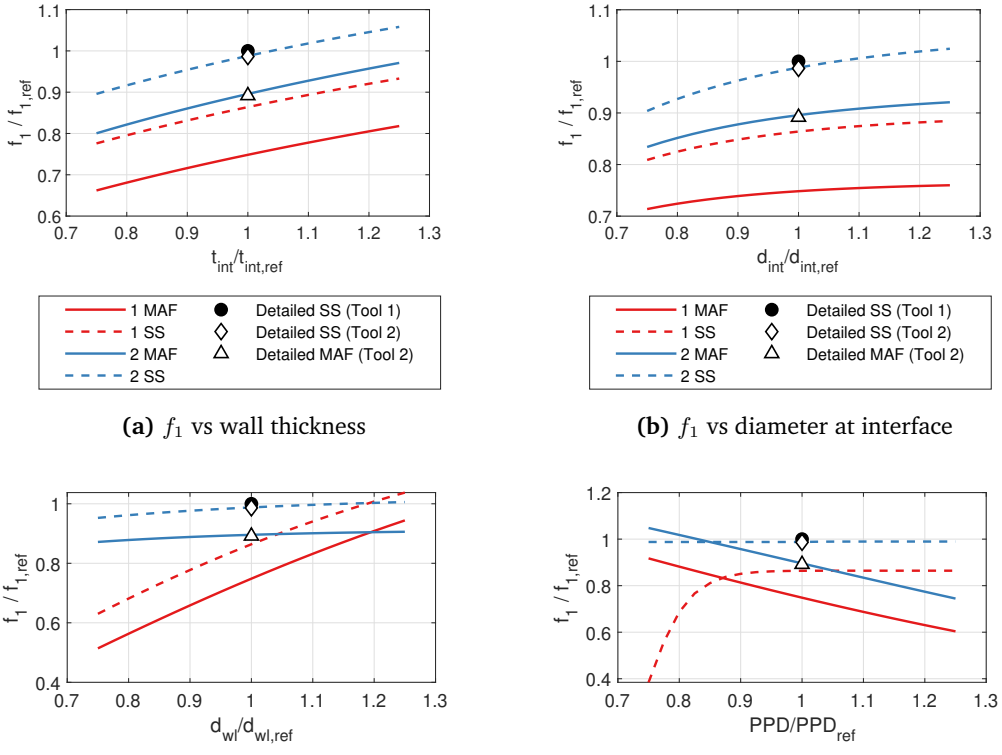


Figure 3.3: Effect of increased pile penetration depth on a MAF soil model.

Using the cantilever beam analogy, increasing the length of the beam has the effect of decreasing the stiffness and therefore the frequency of the beam. This means that increasing the pile penetration makes the MAF models more flexible. For soil spring

models, the effect is different. Because the soil spring models add stiffness via $p-y$ springs, increasing the pile penetration does have a stiffening effect. However, a large contribution of the stiffness of the soil is due to the first few layers, with the deeper layers having a lesser effect on the stiffness. This is why the soil spring models were largely unaffected by the changes in PPD. The exception is that the decrease in PPD for Model 1 had a large effect on f_1. The interpretation of this is that Model 1, having lower stiffness, is more sensitive to the added stiffness of the top layers of soil. Model 2, having a more rigid foundation, is less sensitive to this effect.

From Figure 3.4 it's also noted that Model 2 with a spring soil model ("2 Soil Springs") most accurately resembles the detailed model when the base values of the DoFs are used. Additionally, Model 1 with a modified apparent fixity (MAF) soil model shows the largest error in f_1, exhibiting a 25% underprediction. This difference can be explained when comparing the geometry of the simplified model with that of the detailed model. The simplified model does not account for the larger diameter of the monopile thus having a softening effect.



(a) f_1 vs wall thickness

(b) f_1 vs diameter at interface

(c) f_1 vs diameter at water level

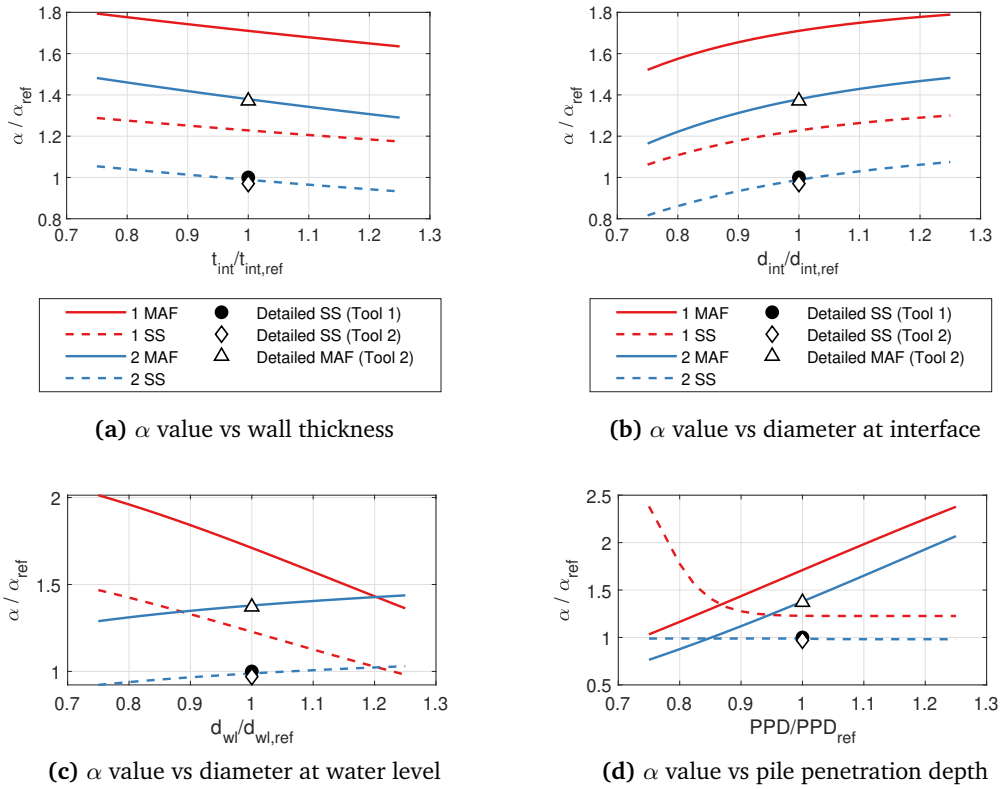
(d) f_1 vs pile penetration depth

Figure 3.4: Sensitivity of the first fore-aft frequency with respect to geometric degrees of freedom.

Similarly, the sensitivity of the α value is shown in Figure 3.5. As with f_1, increasing the wall thickness or the diameters has a stiffening effect, resulting in lower

values for α. This is because the foundation becomes stiffer while the tower remains the same, meaning that the normalized deflection of the foundation is lower.

However when it comes to interpreting the effects of increasing the PPD, the analysis is more involved. Increasing PPD has the effect of increasing α (Figure 3.5(d)). If the turbine is again viewed as a cantilever beam, increasing the pile penetration increases the total length of the beam. Because the tower height is kept constant, the water level–where the α value is extracted–is kept in the same position relative to the free end of the beam. Relative to the fixed end, the water level is further away from the fixed end. Since the mode shape as a function of height is approximately hyperbolic, the value α increases with increased PPD in the MAF models despite softening the structure in terms of frequency. In other words, the increased α is a result of a more flexible foundation on MAF models, as the fixed end is placed further away from the water level. This same analysis is consistent with the results found for the frequency dependence on the PPD, as increasing the PPD had a softening effect on the MAF models resulting in a lower frequency. This phenomenon is illustrated in Figure 3.6.



(a) α value vs wall thickness

(b) α value vs diameter at interface

(c) α value vs diameter at water level

(d) α value vs pile penetration depth

Figure 3.5: Sensitivity of the α value with respect to geometric degrees of freedom.

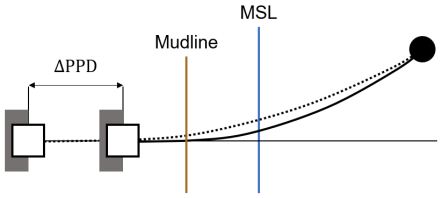
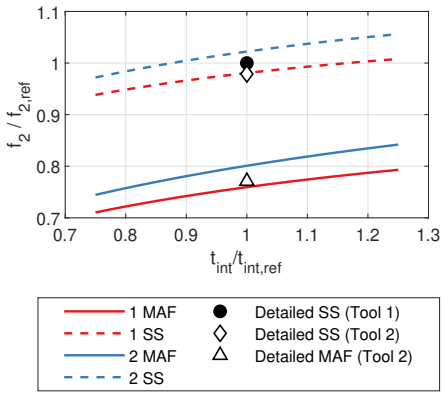
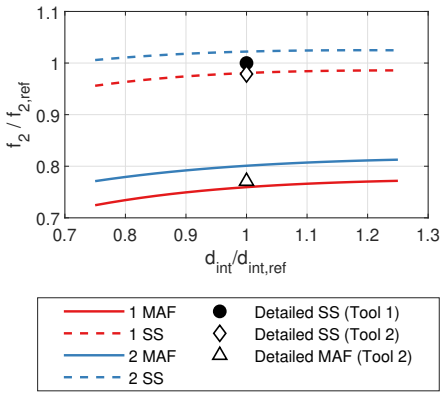


Figure 3.6: Effect of increased pile penetration depth for MAF models on the α value.

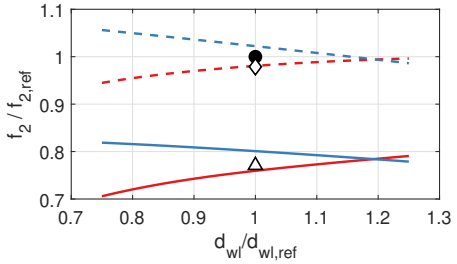
Additional checks on the second mode frequency (Figure 3.7) and the damping ratio of the first mode in the fore-aft direction (Figure 3.8 show similar trends to the ones observed for frequency and α value. Overall, the models that use a MAF approach tend to be softer than models with soil springs. This reduction in stiffness is then reflected in a lower damping ratio, as damping is defined for these models proportional to stiffness.



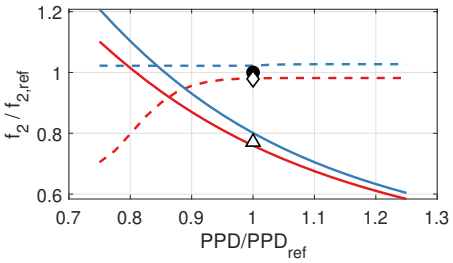
(a) f_2 vs wall thickness



(b) f_2 vs diameter at interface

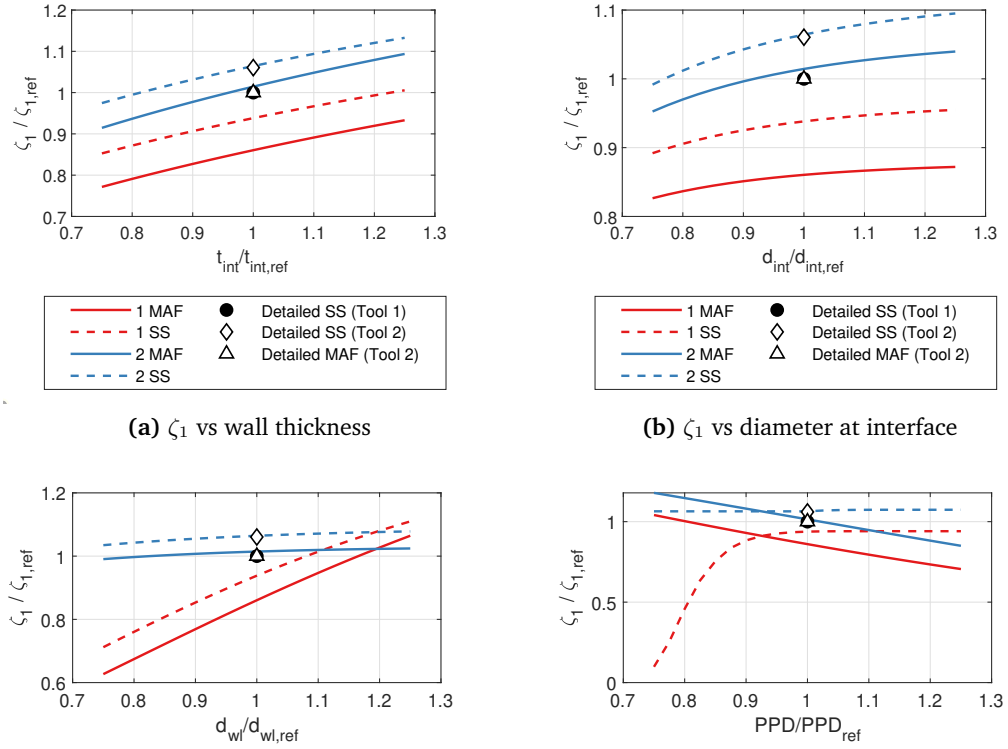


(c) f_2 vs diameter at water level



(d) f_2 vs pile penetration depth

Figure 3.7: Sensitivity of the second fore-aft frequency with respect to geometric degrees of freedom.



(a) ζ_1 vs wall thickness

(b) ζ_1 vs diameter at interface

(c) ζ_1 vs diameter at water level

(d) ζ_1 vs pile penetration depth

Figure 3.8: Sensitivity of the damping ratio with respect to geometric degrees of freedom.

The analysis presented in this section points towards a method with which the simplified model can be modified to more closely match the detailed model. In terms of first fore-aft frequency, mode shape value, second fore-aft frequency, and first fore-aft damping ratio, decreasing the PPD for models with a MAF soil model almost entirely eliminates the difference between the simplified model and the detailed model. It is proposed that by modifying the depth of the pile penetration, when combined with minor modifications to Rayleigh damping parameters and beam stiffness factors, a simplified model can be altered to match a target first mode frequency, α value, and damping ratio ζ_1. It is expected that matching these three dynamic properties will result in a simplified model with a response very similar to a corresponding detailed model. This process is presented in section 3.3.

Evaluating the four modeling approaches that were analyzed, Model 1 with a MAF soil model is preferred due to its simplicity and versatility. Although Model 2 had more similar dynamics to the detailed model, the geometry of this model is not easy to generalize. The diameters used to define the foundation are highly site-specific, dependent on the type of connection between tower and foundation, among other variables. This introduces many several more DoFs that would require a higher

computational cost and ultimately detract from the overall objective of creating a generic, simplified model that is versatile and robust. With the same reasoning, the MAF model is chosen over the soil-spring model as changes in the soil stiffness between sites are easily applied to the MAF model by changing the pile penetration depth. If the soil-spring model where to be used, a soil study at each (potential) site to construct the $p-y$ curves would be needed. Because the generic models do not correspond to a specific site, it is impossible to assign generic $p-y$ curves for each generic model.

3.2.2 Wave loading

Having studied the effect of various hyperparameters on the dynamic response, a similar study is conducted related to the hydrodynamics of the simplified model. This is an important step, as the wave loads are dependent on the geometry of the foundation. Therefore, when a new simplified foundation is created, appropriate wave loads must be generated using the geometry of the simplified foundation. To conserve accuracy in the generated wave loads, it is crucial to understand the effects of the relevant DoFs and the hyperparameters that govern the generation of wave loads.

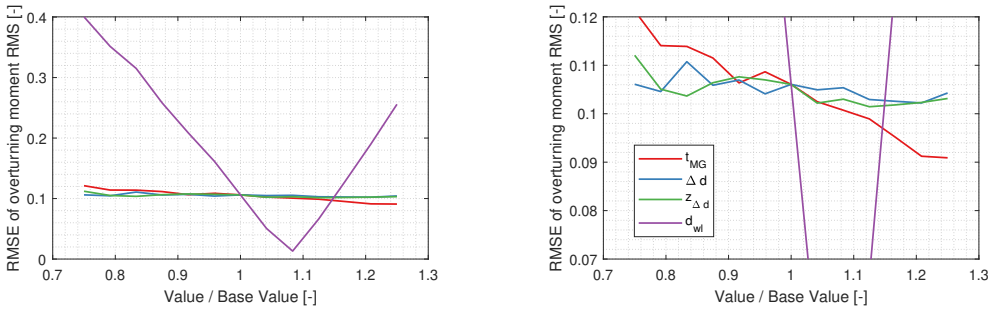
The SGRE wave generation engine, which is based on linear wave theory, is used to generate the wave loads. The wave loads are analyzed in terms of accuracy when compared to waves generated on a detailed model. The waves generated from the simplified model and the detailed model are compared using the relative Root Mean Squared Error (relative RMSE). Because the wave loads are represented as a time series of lateral loads at the fluid nodes of the model, it is convenient to summarize the loads in terms of a single value. It was chosen to use the time series RMS of the overturning moment for this purpose. The calculation of the relative RMSE is shown in Equation 3.2, where n is the number of wave load cases, y is the overturning moment RMS of the detailed model, and $\hat{y}$ is the overturning moment RMS of the simplified model.

$$\text{Relative RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(\frac{\hat{y}_i - y_i}{y_i}\right)^2} \tag{3.2}$$

For this analysis, four key parameters are studied. These parameters are the marine growth thickness t_{MG}, an additional hydrodynamic diameter Δd, the level to which the additional hydrodynamic diameter extends $z_{\Delta d}$, and the diameter at water level d_{wl}. These parameters have been shown in literature to have a large effect on the fatigue loading of offshore wind turbines [30]. The additional hydrodynamic diameter is a modification to the external diameter within the range where it applies that is intended to capture additional hydrodynamic loads from appurtenances (ladder, boat landings, etc.) which are not modeled in the simplified structure. The additional hydrodynamic diameter can also serve as a means to reduce the differences in the external diameter between the simplified and detailed model that can have a large effect on the generation of wave loads. The model used to determine the sensitivity

of the wave loads is the “1 MAF” model, which was chosen in the previous subsection due to its versatility.

Similar to the methodology used to assess the sensitivity of the dynamic response, the parameters are varied within a range of $\pm 25\%$ around a base value using a one-at-a-time approach. The base values for t_{MG} and d_{wl} are extracted directly from the detailed model. The additional hydrodynamic diameter is not present in the detailed model because the appurtenances are explicitly included. For this reason, generic values for Δd and $z_{\Delta d}$ are used. For each case, 125 wave load files are selected at random from a pool of 6264 sea states that correspond to a normal sea state (NSS) in the German North Sea. This was done so that the trends across all sea states could be captured while also reducing the computational cost of the sensitivity analysis. The stochastic nature of the process means some fluctuations in the relative RMSE are expected due to random error. These should not affect the overall trends that are observed as a result of the analysis.



(a) Full range of variations **(b)** Detailed view

Figure 3.9: Sensitivity of the overturning moment RMS with respect to the wave-related hyperparameters and degrees of freedom.

As is shown in Figure 3.9, the diameter at water level plays a predominant role in the accuracy of the wave loads derived from the simplified model. This is expected as the hydrodynamic loading is strongly dependent on the external diameter, as is described by the Morison Equation [31]. Also to note is that the hydrodynamic parameters (marine growth and additional diameter) play a lesser but still important role as their modification can lead to better accuracy between the detailed and simplified models. The clear trend that is observed for these parameters is that increasing the effective diameter will result in more accurate wave loads. When comparing with Figure 3.2, increasing the values of the hydrodynamic hyperparameters serves as a means to increase the effective diameter of the Morison Equation, reducing the differences in diameters between the detailed model and the simplified model.

3.3 Selection of hyperparameters

As discussed in the previous section, the simplified model presents some differences in dynamic and hydrodynamic response with respect to the detailed model. These differences can be reduced by modifying the pile penetration depth, the Rayleigh damping parameters, and introducing a stiffness factor for matching the dynamics, and selecting optimal hyperparameters that govern the wave load generation to match the wave loading. This section presents the methods used to systematically modify the dynamics and hydrodynamic properties to achieve the best possible match between a simplified model and a corresponding detailed model.

3.3.1 Tuning of dynamic properties

The sensitivity analysis on the structural dynamics of the simplified model indicates that the PPD, damping parameters, and stiffness factors can be used to modify its dynamics. These three variables work together to tune the simplified model to a target value in terms of f_1, α value, and ζ_1. It should be noted, however, that the interaction between pile penetration depth and stiffness factors (roughly equivalent to modifying the wall thickness) with frequency and α is a coupled system. For this reason, the tuned parameters must be selected simultaneously.

The approach taken to select the tuned parameters is to combine changes to the pile penetration depth with a SGRE in-house tool that can apply changes to the stiffness matrix via stiffness factors and select Rayleigh damping parameters to target a specific fore-aft frequency and damping ratio. To resolve the coupling problem, several simplified models are created which are identical to each other with the exception of the pile penetration depth. For each model, a different pile penetration depth is selected such that a broad range of values are considered. Each of these models is then tuned to the same frequency and damping ratio as the detailed model, and the mode shape value for these models is extracted.

After this process, a curve of α value versus PPD can be constructed similar to the one shown in Figure 3.10. Since all the models have been tuned in frequency and damping, the resulting curve represents all models with the same frequency and damping as the detailed model, but with different pile penetration and α. Through interpolation, the PPD which results in the same mode shape value as the detailed model can be extracted. Using this approach, the tuning problem is reduced from having three tunable inputs and three output targets to a single input, single output problem.

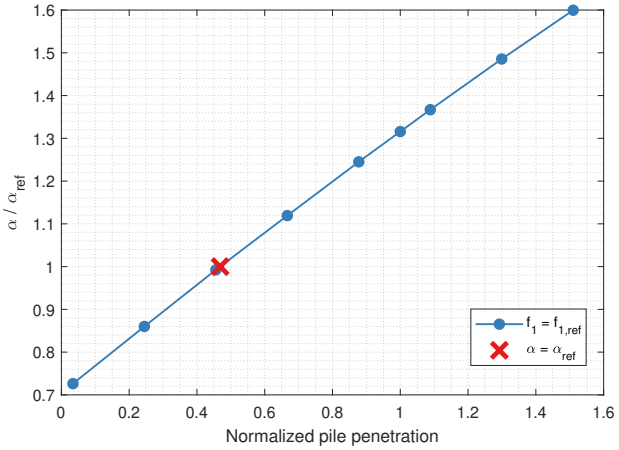


Figure 3.10: Differences in site-specific conditions among turbines of the same class.

3.3.2 Optimization of hydrodynamic parameters

Additionally, for determining the best combination of hydrodynamic parameters such that the waves resulting from the simplified model most accurately match those of a detailed model an optimization problem is proposed. For this, the relative root mean square error of the overturning moment RMS is chosen as the objective function and is minimized via gradient descent. The root mean squared error is computed across a sample of 360 wind-wave environments representative of a NSS in the German North Sea. The variables considered for optimization are t_{MG}, Δd, and $z_{\Delta d}$. These are chosen among the four parameters studied in the sensitivity analysis of the previous subjection. The diameter at water level d_{wl} was not modified as the sensitivity studied proved that the accuracy of the wave loads are highly sensitive to this value. Therefore the optimized values may not generalize well between sites which have different diameters around the water level.

The optimization problem is presented as

$$\min_{t_{MG},\Delta d,z_{\Delta d}} \mathcal{L}\left(t_{MG},\Delta d,z_{\Delta d}\right) \tag{3.3}$$

Where the objective, loss, or cost function $\mathcal{L}$ is defined as

$$\mathcal{L}\left(t_{MG},\Delta d,z_{\Delta d}\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y}_i\left(t_{MG},\Delta d,z_{\Delta d}\right)-y_i}{y_i}\right)^2} \tag{3.4}$$

With n indicating the number of samples in the batch and y representing the overturning moment RMS, which is a function of the geometry of the model and the wave generation engine:

$$\hat{y}_i\left(t_{MG},\Delta d,z_{\Delta d}\right) = f\left(t_{MG},\Delta d,z_{\Delta d}\right) \tag{3.5}$$

The values of these parameters are initialized at their generic value used for the sensitivity analysis in Figure 3.9. It is not possible to obtain the gradient of the objective function analytically, therefore the gradient is estimated via centered finite differences. These choices mean that the computational cost of this process is high, as one step in the gradient descent can involve thousands of wave load generations. To mitigate the computational cost, an approach using mini-batch gradient descent is preferred, where the total sample is split in groups of 20 sea state conditions and each group is used to take a step in the gradient descent.

For the mini-batch gradient descent, the gradient is estimated as follows [32]:

$$\frac{\partial \mathcal{L}}{\partial x_i} \approx \frac{\mathcal{L}(x_1, ..., x_i + \varepsilon, ..., x_n) - \mathcal{L}(x_1, ..., x_i - \varepsilon, ..., x_n)}{2\varepsilon} \tag{3.6}$$

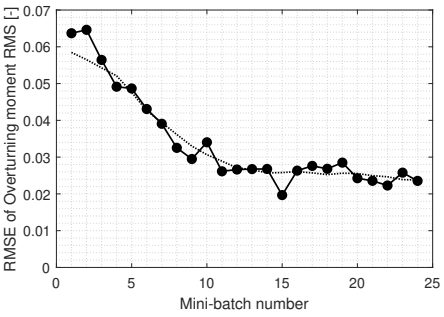
with

$$x = \{t_{MG}, \Delta d, z_{\Delta d}\} \tag{3.7}$$

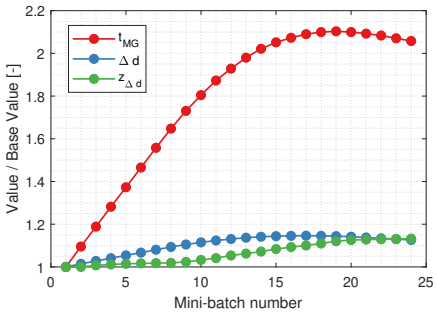
After computing the gradient of the objective function, a step is taken in the direction opposite to the gradient proportional to the size of the gradient to minimize the error:

$$x_{j+1} = x_j - \alpha \left(\frac{\partial \mathcal{L}}{\partial x} \right) \tag{3.8}$$

where j denotes the mini-batch number.



(a) Convergence of the loss function.



(b) Convergence of the wave-related hyperparameters.

Figure 3.11: Convergence history of the optimization algorithm for determining the optimal hyperparameters for wave load generation.

Applying this process for one full epoch (a full run through the entire dataset) yields the results shown in Figure 3.11(a). It is shown that in terms of the loss function, the root mean squared error is approximately converged around 2.4%, reduced from a starting 6.4% error. This shows that with only a few mini-batches it is possible to already optimize the parameters of the hydrodynamics of the model and reduce the error to one third of what was obtained with generic wave parameters. The adopted values show similar trends in convergence as shown in Figure 3.11(b). It can be seen

that the overall trend of the optimization process was to increase the parameters, increasing the effective diameter of the foundation. This is consistent with what was found in the sensitivity analysis presented in the previous section, and the comparison of the simplified model with the detailed model (Figure 3.2).

Chapter 4

Validation of the Simplified Model

In this chapter, the methodology for creating simplified models of offshore wind turbines is validated. A simplified model is generated based on a corresponding detailed model and their response in terms of fatigue loads are compared. For this, time domain hydro-aeroelastic simulations are performed on both models and the damage equivalent loads (DELs) at fatigue critical locations are compared. The simulations for this study are pre-processed, run, and post-processed using the in-house tools of SGRE for fatigue load calculation.

To further understand the effects of the methods used to match the dynamics response (i.e. f_1, α value, and ζ_1) and wave loading, additional simplified models in which these methods are omitted are compared to the detailed model. The models included in the comparison are then (1) a model tuned to the target frequency, damping ratio, and α value with generic wave load hyperparameters, (2) a model tuned to the target frequency, damping ratio, and α value with optimized wave load hyperparameters, (3) a model with tuned frequency and α value with an un-tuned damping ratio and using optimized wave load hyperparameters, (4) a model with tuned frequency and damping ratio but with an un-tuned α value, and using optimized wave load hyperparameters, and (5) a model that utilizes optimized wave load parameters but is otherwise not tuned to the target frequency, damping, or α value. In all cases, the target values correspond to the values of the detailed model.

In section 4.1, a description of the Design Load Cases (DLCs) used for the validation of the models is presented. Followed this, the results for a normal power production case are shown in section 4.2. In section 4.3 the responses for an idling case are shown. Finally, section 4.4 summarizes the conclusions drawn from the validation process.

4.1 Description of load analysis cases

The load cases studied for the validation analysis are the DLC 1.2 and DLC 7.2, as defined by the IEC 61400-3:2019 [33]. These two cases were chosen because they capture the two operating conditions that represent the majority of fatigue damage

accumulation in wind turbines: normal power production and idling/standstill. The normal power production case represents the conditions in which the wind turbine is operating normally, generating power as intended. Typically, upwards of 95% of the lifetime of a wind turbine is spent during normal operation. When the turbine is not operating, e.g. due to maintenance operations, the wind turbine is set to idle. The loading during this condition still contributes to the fatigue loading of the wind turbine and therefore is accounted for in DLC 7.2. The description of the simulation conditions used to analyze both DLCs are presented in Table 4.1.

In DLC 1.2, the power production case, the wind environment is composed by wind speeds every 1 m/s from 3 m/s to 31 m/s and a normal turbulence model. Wind-wave misalignments from -60 to 90 degrees are considered in sectors of 30 degrees. These are typical ranges of wind-wave misalignment in the North Sea [34]. While a typical analysis for certification would include wind directions according to the site-specific wind-rose, for validation purposes it is sufficient to only consider a single wind direction. The comparison between the simplified model and the detailed model is conducted in terms of the RMSE of the DELs for the individual wind-wave combinations of each DLC. For each wind speed and wave direction combination, three turbulence seeds are considered to capture the statistical variability at each wind speed. For each, simulation, a matching wave environment (defined by H_s and T_p) is assigned, consistent with the site-specific joint-probability distribution for the site in the German North Sea. This process yields 522 total simulations for this DLC case.

Table 4.1: Description of the design load cases used for validating the fatigue loads of the simplified model.

	DLC 1.2	DLC 7.2
Description	Normal operation	Idling/Standstill
Turbulence	Normal Turbulence Model	Normal Turbulence Model
Sea state	Normal Sea State	Normal Sea State
Wind speeds	3 to 31 m/s, every 1 m/s	2 to 32 m/s, every 2 m/s
Wind directions	0 deg	0 to 360 deg, every 30 deg
Wave misalignments	-60 to 90 deg, every 30 deg	0 deg
Yaw misalignments	-5.6, 0 and 5.6 deg	0 deg
Turbulence seeds	3, one per yaw misalignment	6
Simulation duration	10 minutes	10 minutes
Total simulations	$29 \times 6 \times 3 = 522$	$16 \times 12 \times 6 = 1152$

For DLC 7.2, the idling case, wind speeds every 2 m/s are considered from 2 m/s up to 32 m/s. In this case, no wind-wave misalignment is considered according to the IEC 61400-3 [33], meaning that wind and waves are co-directional. Wind and wave directions are included in sectors of 30 degrees, from 0 degrees to 330 degrees for a total of 12 possible directions. For each wind speed and wind direction pair, six turbulence seeds are simulated, each with its own wave load defined by the

appropriate wave height and period. This process in total yields 1152 simulations for this load case. Although similar reasoning could be applied to consider only one wind direction, doing so would result in only 96 simulations. This is too low of a low sample number, therefore all 12 wind directions are considered.

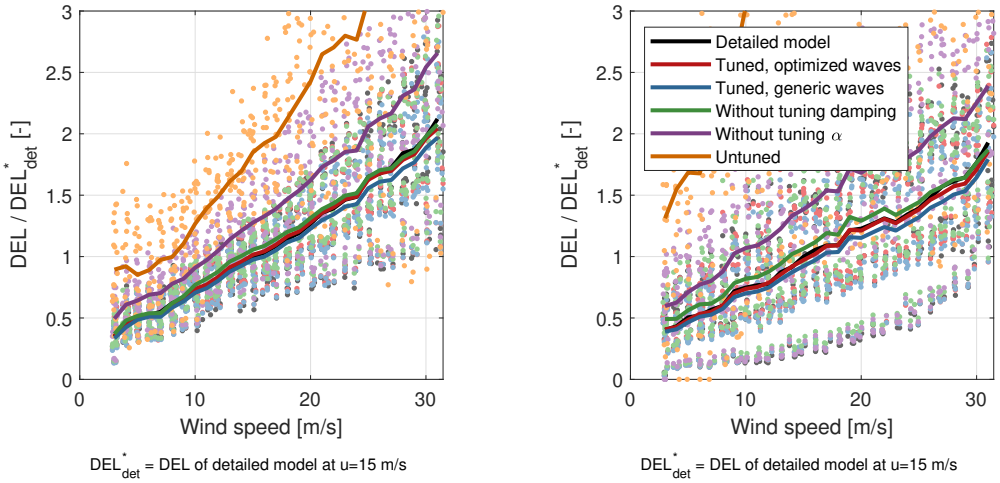
The DELs compared in the validation exercise are the bending moments at the mudline level (M_{mud}), at interface level (M_{int}), and tower top ($M_{towertop}$). The bending moments at each location are composed of a bending moment about the global X-axis and about the global Y-axis. In the case of DLC 1.2, because only a wind direction of 0 degrees is considered, the moment about the X-axis corresponds to the moment in the fore-aft direction, and the moment about the Y-axis corresponds to the moment in the side-side direction. For DLC 7.2, since all directions are considered, they do not correspond to moments in fore-aft or side-side directions. The DELs selected are summarized in Table 4.2.

Table 4.2: Summary of the fatigue DELs used for validation of the simplified model.

Load	Description	Wöhler slope
M_{mud}	Bending moment at the mudline (DEL)	3.5
M_{int}	Bending moment at the interface level (DEL)	3.5
$M_{towertop}$	Bending moment at the tower top (DEL)	3.5

4.2 DLC 1.2: Normal power production

Figure 4.1 shows the mudline DELs for DLC 1.2. The results of each simulation are plotted against wind speed in a scatter plot, and the binned average according to wind speed is plotted as a continuous line. It is apparent from Figure 4.1 that an un-tuned model is not viable to accurately represent the loads of the detailed model. The dynamic properties of the model that results from the simplification process is too different from those of the detailed model to be able to make any meaningful conclusions about the detailed model using the un-tuned simplified model. Moreover, the importance of tuning for target values in the dynamics is magnified by the close accuracy achieved by the two fully tuned models that use either generic or optimized wave load parameters; while the difference between the two wave parameter strategies is small, Figure 4.1 indicates that better accuracy is found when using the optimized wave parameters. This validates the optimization process of these hyperparameters.

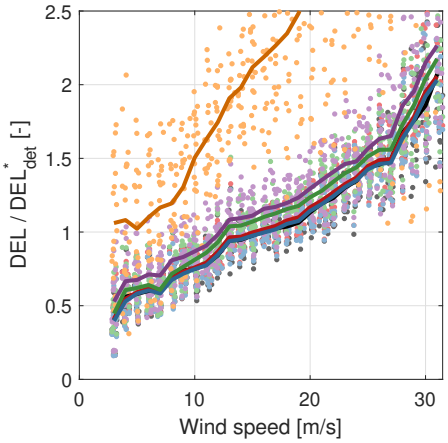


(a) Bending moment in fore-aft direction.

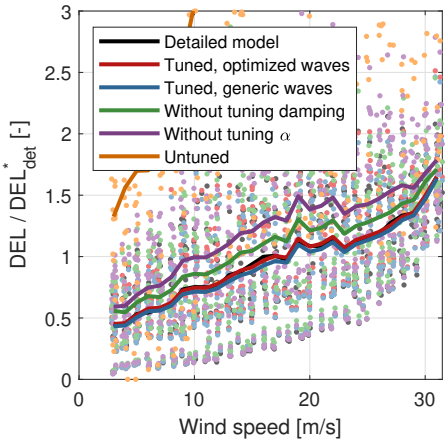
(b) Bending moment in side-side direction.

Figure 4.1: Bending moment at mudline (DEL) for DLC 1.2 as a function of average wind speed.

In addition to this, the cases with an un-tuned damping ratio or mode shape value can shed some light into the importance of the two tuning processes. By comparing the tuned simplified model that is properly tuned to the model without a tuned α value (shown in purple), it is seen that tuning the α value is of great importance for producing accurate estimations of the mudline loads for DLC 1.2. The loads at this hotspot are expected to have contributions from wind-related loading and wave-related loading. Similarly, comparing the tuned model to a model where the damping ratio has not been tuned (depicted in green), shows that the importance of tuning this parameter is less than that of the α value, but not negligible. This can be attributed to the presence of aerodynamic damping which is predominant over the contribution of the structural damping. Additional evidence for this claim is obtained when comparing the behavior of the un-tuned damping model in the FA and SS directions. The differences between the tuned and untuned model are greater in the SS direction than in the FA direction. Aerodynamic damping acts largely in the FA direction rather than the SS direction, therefore the structural damping is more important in the SS direction.

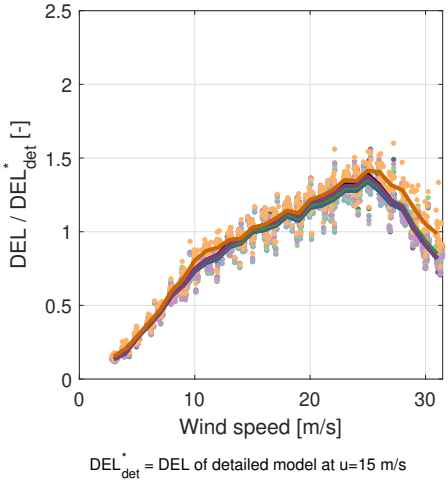


(a) Bending moment in fore-aft direction.

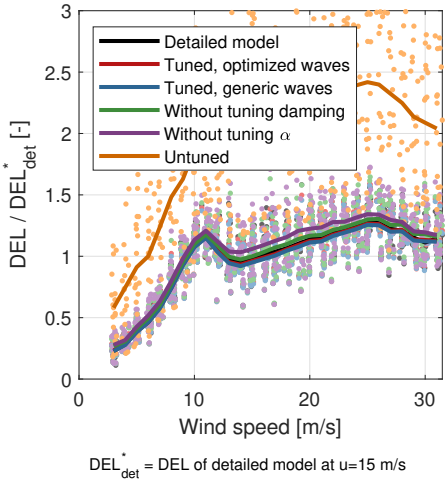


(b) Bending moment in side-side direction.

Figure 4.2: Bending moment at the interface (DEL) for DLC 1.2 as a function of average wind speed.



(a) Bending moment in fore-aft direction.

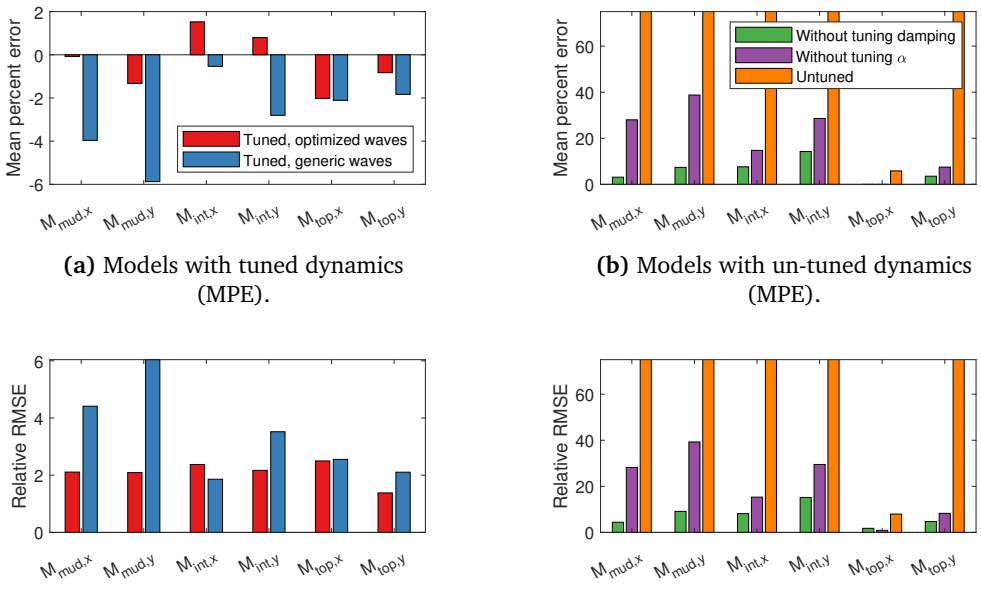


(b) Bending moment in side-side direction.

Figure 4.3: Bending moment at the tower top (DEL) for DLC 1.2 as a function of average wind speed.

Similar trends are found for the interface loads (Figure 4.2) and tower top loads (Figure 4.3). It is noteworthy that as the importance of the wave-related hyperparameters, α value, and damping ratio decreases as the location of the DEL moves further up the tower. This is because the loads near the tower top are driven by aero-

dynamic loads while loads near the tower bottom and at the foundation are a result of the combination of wind and wave loading. To illustrate this, Figure 4.3(a) shows that the FA bending moment at the tower top is almost completely independent to the choices made in the dynamic tuning process or the support structure model as a whole. This is due to the fact the loading at the tower top for the normal power production case is dominated by the aerodynamic loading of the rotor (i.e. the thrust loading produced by power generation). This makes this load channel insensitive to differences in wave loading and dynamics for this DLC. Contrary to this, the side-side loads in Figure 4.3(b) show some sensitivity to the frequency, but not the other parameters. This is because, with zero yaw misalignment, the side-side loads in normal operation are a product of free vibration of the tower. This is supported by the fact that side-side loads in this location are approximately 1/5th of the fore-aft loads. This was not the case for the previous load channels because loading was a product of the interaction of wind and wave conditions at those locations.



(a) Models with tuned dynamics (MPE).

(b) Models with un-tuned dynamics (MPE).

(c) Models with tuned dynamics (RMSE).

(d) Models with un-tuned dynamics (RMSE).

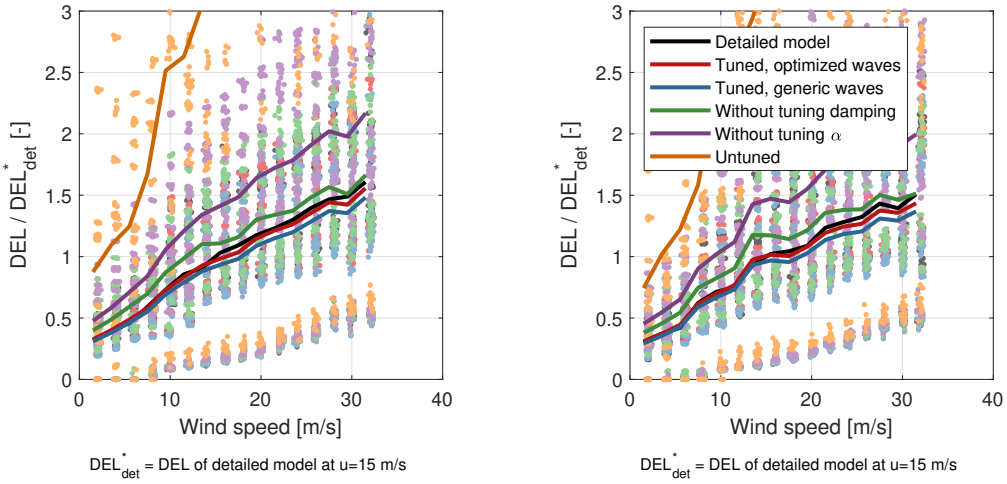
Figure 4.4: MPE and RMSE of the DELs for DLC 1.2.

Quantitative comparisons in terms of the mean percentage error (MPE) and relative root mean square error (RMSE) are presented in Figure 4.4. The MPE is chosen as a measure of whether the simplified model underestimates or overestimates the loads when compared to the detailed model, while the relative RMSE is a measure of the dispersion in the results of the simplified model when compared to the detailed model. The axes for the models with un-tuned dynamics have been truncated in order to make the data more readable. From Figure 4.4, similar conclusions can be

drawn regarding the importance of the different tuning strategies for the load locations studied. The resulting simplified model is shown to estimate the loads with an accuracy of $\pm 2\%$ at mudline, tower bottom, and tower top, while having a dispersion of approximately 2% across all load locations for DLC 1.2.

4.3 DLC 7.2: Idling

The same comparisons are carried out for DLC 7.2. As this is a standstill case, loading conditions are mostly dependent on wave loads and thus the effects of the simplified support structure are expected to be amplified. As was the case for DLC 1.2, the un-tuned model completely overestimates the loads when compared to the detailed model and the other simplified models (Figure 4.5). The differences between the un-tuned simplified model and the detailed model are reduced by adequately tuning the dynamics of the model (frequency, damping, and α value). Because DLC 7.2 is influenced more by the wave loading than DLC 1.2, the models un-tuned α value shows a larger difference with respect to the detailed model for this DLC. Similarly, as aerodynamic damping from the rotor operation is no longer present, the importance of tuning the damping ratio is amplified. Without aerodynamic damping and a MAF soil model, the only source of damping is the structural damping.



(a) Bending moment about the global X-direction.

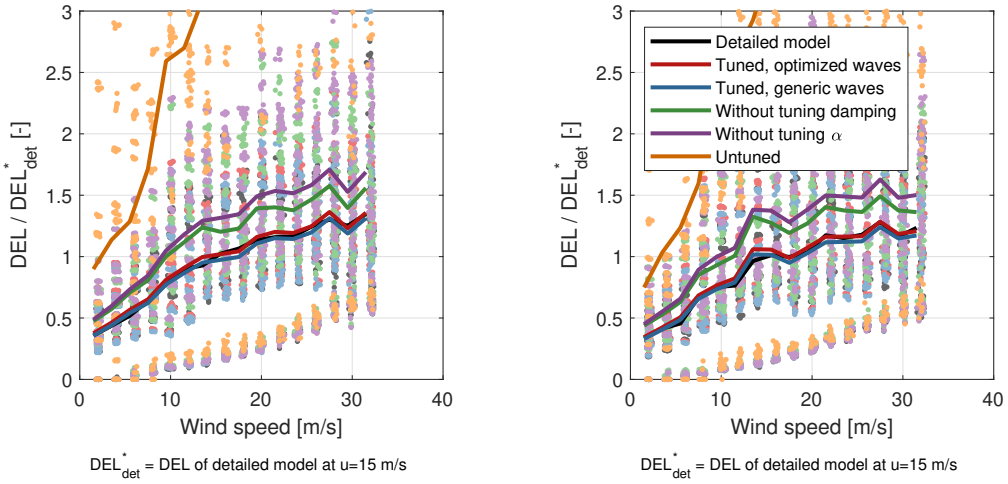
(b) Bending moment about the global Y-direction.

Figure 4.5: Bending moment at mudline (DEL) for DLC 7.2 as a function of average wind speed.

Additionally, in DLC 7.2 – as opposed to DLC 1.2 – since wind directions ranging from 0 to 360 degrees are considered, the global X and Y directions do not necessarily correspond to FA and SS directions. However, because the rotor is not operational,

there is little differences between the the FA and SS directions in terms of loading and dynamics. On an operational rotor (e.g. DLC 1.2) aerodynamic loading will be primarily concentrated in the fore-aft direction. Because the loads reported are in the global reference frame and no transformations are applied in post-processing, the moment about the X-axis can sometimes represent the load in FA direction, SS direction, or a direction in between the two, depending on the wind direction.

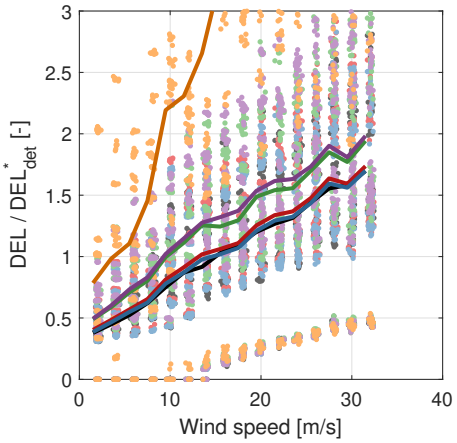
As is seen in Figure 4.6 and Figure 4.7, the importance of tuning the α value is greater across all load channels than tuning the damping ratio, and its importance is greatest near the bottom of the tower and diminishes towards the top of the tower. These results are in-line with the results found for DLC 1.2. Between the optimized wave parameters and the generic wave parameters, it is seen that the optimized wave parameters increase the accuracy of the model, especially for mudline level loads. The tower top loads shown in Figure 4.7 for DLC 7.2 are comparable in magnitude to the side-side loads for DLC 1.2. This is because these loads are primarily a result of the vibration induced by wave-loading.



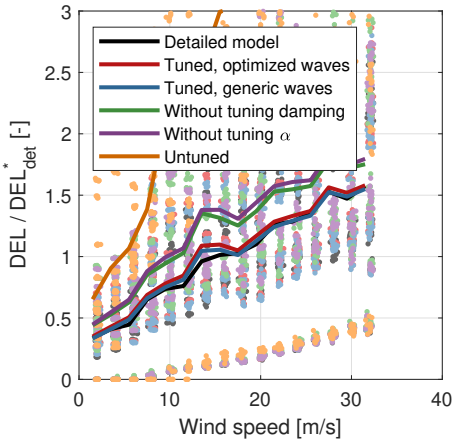
(a) Bending moment about the global X-direction.

(b) Bending moment about the global Y-direction.

Figure 4.6: Bending moment at the interface (DEL) for DLC 7.2 as a function of average wind speed.

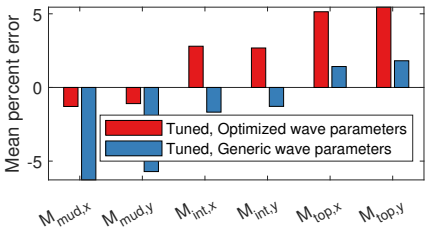


(a) Bending moment about the global X-direction.

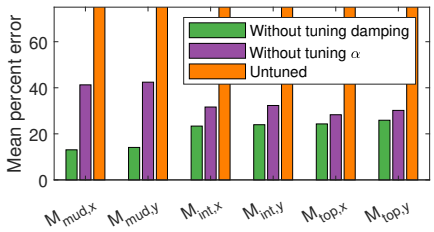


(b) Bending moment about the global Y-direction.

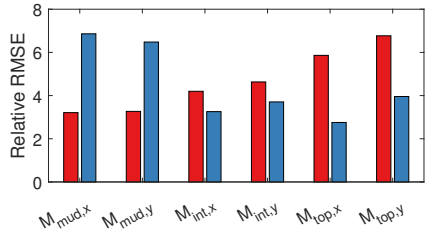
Figure 4.7: Bending moment at the tower top (DEL) for DLC 7.2 as a function of average wind speed.



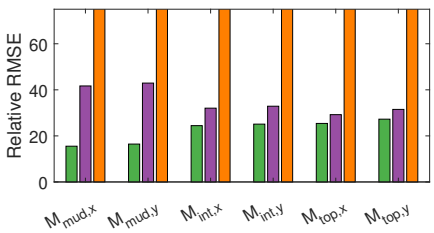
(a) Models with tuned dynamics (MPE).



(b) Models with un-tuned dynamics (MPE).



(c) Models with tuned dynamics (RMSE).



(d) Models with un-tuned dynamics (RMSE).

Figure 4.8: MPE and RMSE of the DELs for DLC 7.2.

In terms of the MPE and RMSE (Figure 4.8), the observations made in the previous figures are corroborated. The accuracy of the simplified model is greatly increased via

adequate tuning of the model, and the use of optimized wave parameters as opposed to generic parameters increases the accuracy of the model, particularly at mudline level. At tower top, the error actually increases when using the optimized parameters however it should be pointed out that the absolute value of the moment at tower top for this DLC case is much smaller than for DLC 1.2.

Compared to DLC 1.2, the importance of tuning to the target damping ratio on the first mode is highlighted in DLC 7.2. While the errors in DLC 1.2 for the case without tuning the damping ratio remained largely below 10%, in DLC 7.2 these rise above 20%. This is because in the idling case the rotor is not providing aerodynamic damping (which is much higher than the structural damping), and thus the only source of damping is the structural damping.

Thus, the simplified model with all the tuning measures in place can achieve the same DELs as the detailed model within $\pm 6\%$, while having a dispersion between 3% and 6% depending on the DEL location for DLC 7.2.

4.4 Discussion

The complete exercise of validation of the simplified model for DLCs 1.2 and 7.2 shows that the simplified model, when adequately tuned to the dynamics of a corresponding detailed model, can accurately simulate the DELs at mudline level, tower bottom, and tower top within about a 2% error for DLC 1.2 and 6% for DLC 7.2. The higher error for the idling case is expected as the wave loading has a higher influence on the DELs during idling than in normal operation. The simplified model primarily modifies the support structure, which mainly affects the generated wave loads and the response of the wind turbine to wave loading.

Although the simplified model has been tuned to match the dynamics of the first mode of the detailed model, there responses are not identical. This is because the behavior of the wind turbine is not only described by first mode dynamics but also the interaction of the higher modes which have not been specifically tuned. Nevertheless, it is shown that their effect is relatively minor and high accuracy can be achieved through tuning the first mode only.

These results show that the simplified model can be used to estimate the damage equivalent loads in lieu of the detailed model within a tolerance for error of approximately 2-6%. The simplified model achieves this while conserving a small number of DoFs and eliminating the need for excessive modeling detail. This exercise shows the potential of the simplified support structure model to be used in a generic load emulator concept, where only a few parameters need to be changed on the simplified model to represent many potential site-specific designs.

Chapter 5

Database Generation

To be able to train a machine learning algorithm to predict and emulate fatigue loads through a supervised learning approach it is necessary to have a robust training database that covers the solution space desired for the load emulator. As black-box models, neural networks and other machine learning algorithms are unable to extrapolate beyond the domain of the data they were trained on [21]. This is the core reason why site-specific load emulators trained on a given offshore site cannot be generally applied to different sites without domain generalization [13]. Creating a broad solution space for a single emulator is the central objective of this thesis, and therefore special care must be applied when creating the training database.

The process of creating the database is divided in two stages. In the first stage, a variety of simplified models is created based on permutations of the DoFs identified for the simplified model in chapter 3. In the second stage, the environmental conditions related to the specific design load cases (representative of DLC 1.2 i.e. normal production or DLC 7.2 i.e. idling) are created based on a distribution and a sampling method, and these conditions are assigned to the various simplified models. The training database consists of n_{models} simplified models that capture a range of geometric and dynamic properties, each with several possible wind-wave climatic conditions, for a total of n simulations.

In section 5.1 the distributions for the geometric and dynamic degrees of freedom are defined. Later, in section 5.2 the distributions for the environmental DoFs are specified. A discussion of the sampling methods used to create the databases is included in section 5.3. Finally, the catalog of simplified models generated is described in section 5.4.

5.1 Geometric and dynamic degrees of freedom

In this section, the process of selecting the ranges and distributions for the geometric and dynamic degrees of freedom is described. The simplified model was extensively studied in chapter 3 and the degrees of freedom for this model were identified. As was detailed in chapter 3, the process of tuning a simplified model to a target combination

of f_1, ζ_1, and α is an iterative process. When creating dozens of simplified models that each need to be tuned to different targets, this can represent a sizable computational cost. The main reason for the computational cost is the interpolation method for determining the required PPD that achieves a desired α value, which involves several calls to the frequency tuning algorithm provided by SGRE.

To mitigate the computational cost of this process, it was chosen to treat the PPD as a DoF rather than the α value. Compared to determining the PPD that corresponds to the target α value, treating the PPD as the DoF involves only a single call to the frequency tuning algorithm rather than 8 calls of this function (8 points used for interpolation). This means that the computational cost is reduced by 87.5%. The trade off is that there is less control over the resulting α values of the created models. Careful selection of possible PPDs is required and a final sanity check that the achieved α value is realistic for the wind turbine is performed.

5.1.1 Ranges and distributions of each degree of freedom

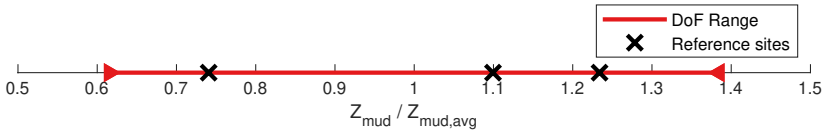
To determine which values are to be used for each DoF for a specific sample, each DoF is defined by a distribution and its parameters. For all DoFs, the distribution selected was a uniform distribution, with parameters specifying the range of the DoF. The ranges were determined using the values of 3 site-specific, detailed models provided by SGRE. These models are composed by one offshore site at the German North Sea and one offshore site at the Belgian North Sea. The minimum and maximum values of the 3 locations are used to define an initial range, and then the range is padded at both ends to augment the solution space. In the case of the PPD, a range is first defined using the values of the three reference cases. An additional DoF for the adjustment of the PPD is then defined (Δz_{PPD}) that covers a range from increasing the depth by 10 meters to decreasing the depth to 1 meter below the surface level. This method of using two DoFs to define the PPD is chosen because it preserves a physical meaning of the "real" PPD of a model and then applies via the Δz_{PPD} a possible tuning adjustment to an α value.

Finally, the reference models contain minor differences in the geometry of the tower (i.e. the diameter and thickness distribution above the interface level). To address this, the tower geometries of the 3 reference models are used "off the shelf" through a DoF that specifies which of the 3 geometries to assign to the simplified model, referred to as i_{tower}. With this the set of DoFs are:

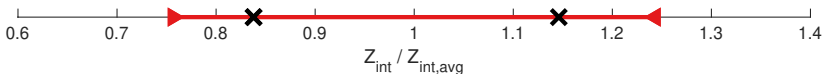
- Reference levels: Z_{mud}, Z_{int}, PPD, and Δz_{PPD}.
- Geometric properties: d_{wl} and i_{tower}
- Dynamic properties: f_1 and ζ_1

The resulting ranges of the DoFs are shown in Figure 5.1, along with a comparison with the values of the three reference sites. The values are normalized with respect to the center (mean) of the resulting range. For some DoFs, the two models at the

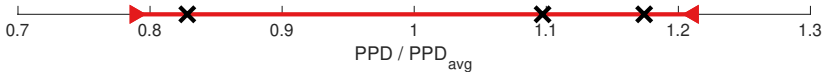
Belgian North Sea location share the same value as they are part of the same project. This is the case for Z_{int}, d_{wl}, and ζ_1 (Figure 5.1(b), Figure 5.1(d), and Figure 5.1(f)). Six of the eight DoFs selected are shown in this figure. The tower index i_{tower} is omitted since this is a discrete variable that can only adopt $i_{tower} \in \{1, 2, 3\}$. The PPD adjustment ΔZ_{PPD} is also omitted as its range is dependent on the PPD value.



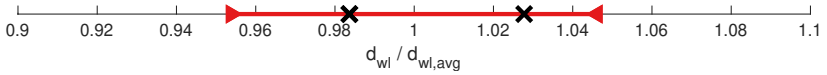
(a) Mudline level



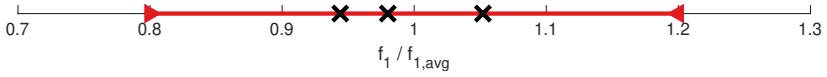
(b) Interface level



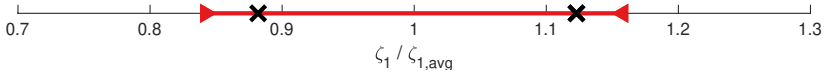
(c) Pile penetration depth



(d) Diameter at water level



(e) Fore-aft frequency



(f) Damping ratio

Figure 5.1: Distribution ranges and reference values for the geometric and dynamic degrees of freedom.

In all the cases, the reference sites are comfortably within the range. This shows that the reference sites are well covered in the solution space. Beyond this, the ranges defined allow for deviations from the values observed in the reference sites. Of all the DoFs, the frequency f_1 (Figure 5.1(e)) was given the widest range relative to the

reference sites. This was chosen intentionally to cover a very broad range of tower and foundation stiffness, and therefore increase the robustness of the SM trained on this data. In practical applications, allowing a wide range of frequencies may allow the SM to cover for discrepancies between measured and modeled natural frequencies. The diameter at the water level also stands out (Figure 5.1(d)) as the reference values show little variation ($\pm 5\%$) compared to the other DoFs. This is largely due to the standardization of this turbine type, which results very similar exterior diameters for the tower and the foundation near the water level.

5.1.2 Discussion of the independence assumption

It should be noted that in the process for assigning values to each DoF, the DoFs are treated as independent variables. This is a strong assumption of the methodology which warrants highlighting. In reality, the dynamics are a result of the geometric definitions. This means that the dynamic DoFs are highly coupled to the reference levels and geometric properties. However, dynamics are also a result of many factors that are omitted from the simplified model, such as more detailed geometric definitions and unknown soil properties. By treating the dynamics and the geometry independently, it is possible to capture the effects of the uncertain factors which are not explicitly modeled. If a strict relationship between the geometric and dynamic degrees of freedom had been used, the resulting models would be limited to the assumptions made for soil stiffness and foundation stiffness.

Typically, the dynamic properties are obtained as a consequence of the geometric definition, which in turn results from an iterative design process for a specific site. Therefore, the dynamics of each wind turbine are coupled to the specific site conditions. Attempting to account for all the variabilities in dynamics possible due to differences in site-specific conditions is a more complex process than treating the two groups independently. More DoFs would be required to describe each model and each model would need to be constructed specifically for each site, incurring in a higher computational cost.

Nevertheless, a limitation of this assumption is that some of the sampled combinations of DoFs can be un-physical or unrealistic. For example a hypothetical model may be assigned to have a deep mudline level and PPD, small diameters d_{int} and d_{wl}, and at the same time determined to have a high frequency. Such a model may be physically unrealistic as the specified reference levels and geometric properties suggest a soft model, while the target frequency implies a stiffer model. Additionally, it is possible that the ranges considered are too broad, meaning computational cost is expended in models that are not as helpful for training the load emulator to predict loads for the realistic applications. Improving on the independence assumption is a topic for future work, as will be discussed in chapter 8.

5.1.3 Discussion on the generality of the database

It is noted that although 3 reference sites are used to define the ranges of the DoFs, the resulting domain is not limited in any way to application on only those 3 sites. The reference sites are only used to provide insights on the degree of variability that is present in the DoFs for wind turbines of the same class. These ranges are then used to define a domain which is expected to encompass most (if not all) wind turbines of this type. A solution space generated by sampling this domain is as applicable to any site within the domain (i.e. within the ranges of each DoF) as it is for the 3 reference sites. In this way, the generic nature of the solution space is ensured.

With the distribution and ranges of each of the geometric and dynamic degrees of freedom defined, generic, simplified models can be created by sampling these distributions and building the models. In section 5.3, the sampling methods are discussed in detail.

5.2 Environmental degrees of freedom

After the simplified models are created, the environmental conditions are to be defined. This is done through the environmental degrees of freedom which encode the wind and wave loading conditions of the DLC, and assigns one of the m created models to the analysis case. The chosen degrees of freedom are:

- Wind parameters: Average wind speed ($\bar{u}$), turbulence intensity (TI), wind direction (θ_{wind}), and seed number.
- Wave parameters: Wave direction (θ_{wave}), significant wave height (H_s), peak spectral wave period (T_p), and seed number.
- Model: Structural model index (i_{model}).

The air density and wind shear exponent were set to constant values across all simulations as they are not expected to play a large role in the support structure loads, which are the main focus of the thesis. The effect of these two parameters is expected to be more apparent for blade loads [18]. In future applications and for deployment of a load emulator tool based on this methodology, it may be convenient to include them as DoFs and their values derived from measurement data across representative sites. This exercise is however beyond the scope of the thesis.

The definition of the distributions and their parameters for the environmental degrees of freedom requires special attention as they will shape the resulting load emulator. It is desirable that the distributions cover a broad range of environmental conditions, such that the solution space derived from the training database is generic and robust. On top of this, combinations of environmental conditions should be weighed according to their likelihood, so that the machine learning algorithm is incentivized to minimize the error of the more likely wind-wave conditions. The way

to accomplish this is to define suitable probability distributions to each DoF without loss of generality across sites.

The mean wind speed $\bar{u}$ is defined using a Weibull distribution representative of the wind climate in the North Sea. The shape parameter of this distribution is 2.3 and the scale parameter is 11.5. These distribution parameters were were obtained from data provided by SGRE. The distribution is shown in Figure 5.2.

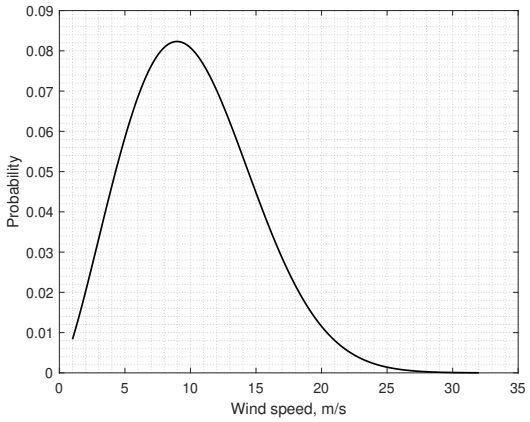


Figure 5.2: Weibull distribution representative of the North Sea.

For the turbulence intensity, many approaches are possible to define its values depending on the desired application of the load emulator. For the sake of the thesis that shows the potential of the methodology, the turbulence intensity is chosen as the ambient turbulence of IEC Class C [33]. Therefore, the turbulence intensity is determined by the wind speed as follows:

$$TI = \frac{0.12\,(0.75\bar{u} + 5.6)}{\bar{u}} \tag{5.1}$$

Future potential use of this methodology can adopt a more complex definition of the turbulence intensity that allows the consideration of wake turbulence. For example, a a lower and upper envelope can be defined for the turbulence intensity according to the wind speed, and the TI can be obtained by sampling uniformly between the two envelopes.

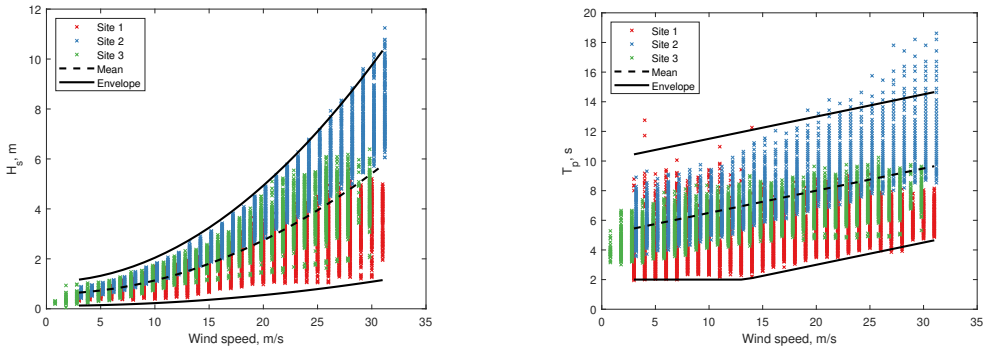
Along with the turbulence intensity, a seed number for the turbulence model is assigned using a discrete uniform distribution, assigning a seed number between 1 and 45. Looping through the seed numbers helps guarantee that the DoFs and the individual samples are not auto-correlated.

Finally, the wind direction is defined by a uniform distribution from 0 to 360 degrees to conserve the generality of the data base.

For defining the wave parameters, some dependence on the wind parameters is required. This is to reflect the coupled nature of the wind-wave climate. For the wave direction, this was defined in terms of the wind direction and a wind wave

misalignment term of the form $\theta_{wave} = \theta_{wind} + \theta_{misalign}$. The misalignment term $\theta_{misalign}$ is modeled by a uniform distribution between -90 and 90 degrees. This ensures that unrealistic or highly unlikely misalginment cases are not included in the database.

The wave statistics H_s and T_p are known to be highly site-specific, as their values can vary greatly even within the same region (e.g. the North Sea). Normally measurement data is required to define the wave climate at a particular site, however this is unavailable for the database generation as the database is composed of hypothetical sites. To resolve this, wind-wave correlation equations and envelopes are proposed based on the climate conditions of three sites in the North Sea. This wind-wave climate data was provided by SGRE. An upper and lower bound is defined for each as a function of the wind speed such that the normal sea states of the three sites lie inside the region enclosed by the envelopes. The significant wave height is approximately quadratic with the wind speed, while the peak wave period is approximately linear. A minimum wave period is assigned to guarantee convergence of the SGRE in-house wave generation engine without loss of generality. The envelopes defined capture more than 99% of the sea states of the 3 sites. This is illustrated in Figure 5.3. Note that the wave statistics of the three sites correspond to whole number wind speeds but they are plotted slightly off-center to make the data points visible.



(a) Significant wave height

(b) Wave period

Figure 5.3: Wind-wave correlation envelopes and comparison with wind-wave climate at three sites in the North Sea.

To sample an H_s and T_p value, the wind speed is used to determine the maximum and minimum values for each according to the envelope and a uniform distribution is used. Additionally, each sample is assigned looping seed numbers between 1 and 99 to be used in the wave generation engine.

Finally, for each of the n climate conditions created during this process, it is necessary to assign it to one of the m previously created models. For this, a discrete uniform distribution is used with parameters $\{1, m\}$.

This section defined a total of 9 DoFs related to the environmental conditions

and their distribution. Although the climate conditions are dictated by site-specific parameters, it is expected that the broad definition used for these DoFs will yield a training database that includes a wide variety of scenarios and possible sites.

5.3 Sampling methods

In the previous sections, the distributions for the geometric, dynamic, and environmental DoFs were defined. However, there is still the question of how to sample points within the distributions. Because of the high number of DoFs, systematically sampling every combination of the DoFs is impractical and inefficient, as the cost of this method scales with $n_{\Delta}^{n_{dof}}$, with n_{Δ} representing the number of values used to discretize each DoF and n_{dof} the number of degrees of freedom. For example using 3 values for each of the geometric/dynamic degrees of freedom would require $3^8 = 6561$ simplified models.

An alternative approach is to utilize random sampling in a Monte Carlo (MC) approach. This has been shown to outperform grid sampling in terms of cost and efficiency. In this method, a total number of samples is selected n, and each DoF for a specific sample is determined stochastically by sampling the probability distribution.

Fundamentally, the MC method is an integration method. In the context of this application, the function that is being integrated is the loss function of the machine learning model. The true loss of the model is

$$\mathcal{I} = \int_{\Omega} \mathcal{L}(\mathbf{x_{dof}}) d\mathbf{x_{dof}} \tag{5.2}$$

where $\mathcal{L}$ is the loss function and $\mathbf{x_{dof}}$ is a vector $\mathbf{x_{dof}} \in \mathbb{R}^{n_{dof}}$ that encodes the values of the DoFs that describe each simulation. The domain Ω is n_{dof}-dimensional, with n_{dof} being the number of DoFs. The MC method approximates the true loss by sampling the domain n times, in the form

$$\mathcal{I} \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(\mathbf{x'_{dof,i}}\right) \tag{5.3}$$

where $\mathbf{x'_{dof,i}}$ is a random sample of the domain Ω. As n increases, the MC method converges to the true loss value. The error of the estimation scales with $1/\sqrt{n}$ [35].

The advantage of the MC method is that it is easy to implement and provides superior performance when compared to a regular grid. However, because of the stochastic nature of the sampling method, it is not guaranteed that the domain will be sampled evenly across all sub-regions. Therefore not all features of the loss function may be captured and some combinations of DoFs may not be represented at all. For machine learning applications the resulting model may be forced to extrapolate to cover the unwanted gaps in the domain, leading to a decrease in performance.

The problem of sampling across the domain “evenly” is known in mathematics as a problem of discrepancy. A sequence of numbers (e.g. a sample) is said to have

high discrepancy if there are regions in the domain with high density of samples and regions with low density of samples. The lowest possible discrepancy is achieved by using an equidistributed regular grid. Random sampling is known to have relatively high discrepancy. Low-discrepancy sequences are quasi-random sequences generated deterministically that are designed to have a low discrepancy and therefore guarantee a more even sampling of the domain. It is possible to use these sequences in lieu of a random sampler in a MC method. This approach is known as a quasi-Monte Carlo (quasi-MC) method . For quasi-MC, the integral is approximated in the same way as in the MC method (Equation 5.3) with the exception that the samples $\mathbf{x}_i$ are obtained from a low discrepancy sequence.

By using a low discrepancy sequence, the error of the quasi-MC method is approximately proportional to $1/n$ [35]. This is a faster convergence rate than the traditional MC approach and means that for a given error tolerance in the estimation, quasi-MC will require less sample points. This has the potential of reducing the computational cost needed for generating the samples and may improve the accuracy of the resulting model during testing, as there will be smaller gaps in the domain where the emulator needs to generalize.

Although there are several low discrepancy sequences available, the Sobol sequence was preferred for this application because of its property that the sample size can be increased if needed without requiring the generation of a completely new batch of samples; the additional samples are appended to the existing ones. In other words, the first n_1 samples of the Sobol sequence with n_2 samples are identical to the Sobol sequence of n_1 samples. This is a notable advantage as it is unknown a priori how many samples are needed for the training database. Therefore if it is seen that more samples are required to improve the performance of the model, the Sobol sequence allows these to be easily added to the existing set. Other low discrepancy sequences such as the Halton sequence do not share this property and therefore increasing the sample size would represent a large computational cost.

For the study, two training databases will be generated. One database will use a traditional MC method to sample all the DoFs (geometric, dynamic, and environmental), while the other database will use a quasi-MC to sample the DoFs. The two databases will be used to train separate load emulators and the performance of the two will be compared in terms of the trade-offs of database size, computational cost, training time, and testing accuracy.

5.4 Database of simplified models

Having defined the distributions for the geometric and dynamic DoFs in section 5.1 and the two sampling methods to be used in section 5.3, the m simplified models can be generated. It is noted that the process of tuning the simplified models to the target values is not always successful. The reasons why some models are not successfully tuned to the targeted dynamics are discussed briefly in subsection 5.4.1. Then, the samples obtained by the two sampling methods are compared in subsection 5.4.2.

5.4.1 Discussion of the non-convergence of the dynamic tuning process

For both sampling methods, a sample size of $n_{models} = 100$ was used for generating the database of simplified models. In the case of MC sampling, 17 of the models generated failed to tune correctly to the target frequency and damping combination. For the quasi-MC sample, there were 18 such cases. Initially it was suspected that the reason some models failed to converge was linked to high discrepancies between the target and initial frequency, however no clear trend was found between the two. Thorough investigation on the failed cases revealed that the causes of failed models fall into two categories, (1) instabilities in the convergence algorithm for frequency and damping tuning leading to unresolved exceptions in the tool provided by SGRE and (2) problems with achieving convergence for the target damping ratio in the torsional vibration mode. It was found that for the quasi-MC method 13 models fall into the first category and the remaining 5 into the second.

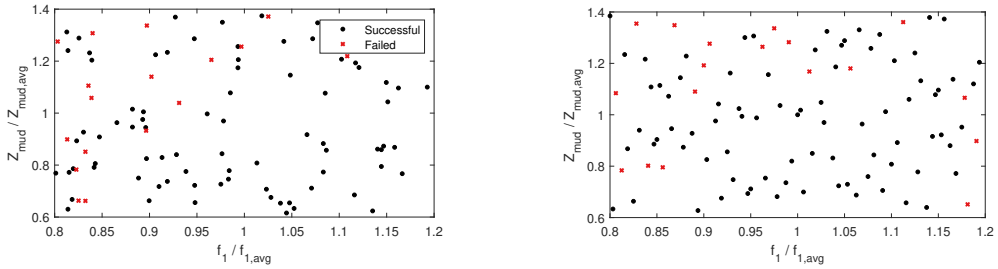
While it is difficult to trace the cause of the instability produced in some of the models it is likely that attempting to tune models that are un-physical or on the fringe of being physically possible plays a hidden role. The instability arises when negative values are proposed by the internal solver for the stiffness proportional damping parameters. This could potentially be resolved by modifying the algorithm used to select the parameters to make it more robust and improve its stability. However, this lies outside of the scope of the thesis.

Similarly, the failure to achieve convergence for the torsional damping could be linked to the robustness of the algorithm. The models are tuned to the target damping ratio for the first mode in the FA and side-side directions, and to a fixed ratio in the torsional direction. Models falling into the second group achieved convergence in frequency and damping ratio in the two translational directions but failed to converge in the torsional direction. This may be linked to a wrong identification of the torsional mode during the eigenvalue analysis of the iteration or to a scenario where the value of the attempted torsional stiffness proportional damping parameter falls below the minimum amount that this parameter can be updated by, as dictated by the SGRE algorithm. In the latter case, the stiffness proportional damping in the torsional direction cannot be decreased further and leads to non-convergence. Modifying this algorithm to resolve these issues fell outside the scope of the thesis and therefore was not attempted. Nevertheless, the database generation process is continued only using the models that were tuned successfully to the target dynamics.

5.4.2 Comparison between sampling methods

Comparing the resulting samples from the database is difficult as the samples cannot be viewed in the hyperspace that they belong to. Projections of the hyperspace in two dimensions can be helpful to observe the behavior of two DoFs at a time. For example, the mudline level can be plotted against the target frequency, as shown in Figure 5.4. Visual inspection of this plot highlights the advantages of the low discrepancy sequence. The samples obtained from a pure MC method have regions

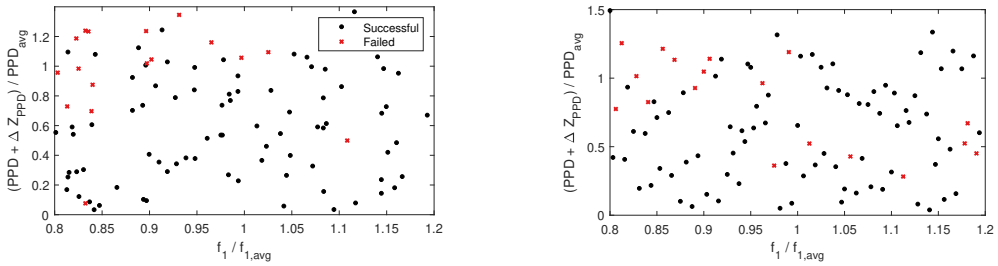
of high density of sampled points and regions of low density of points, whereas the quasi-MC samples have an approximately uniform density.



(a) Monte Carlo **(b)** Quasi-Monte Carlo

Figure 5.4: Comparison of Monte Carlo and quasi-Monte Carlo sampling of frequency and mudline level.

In Figure 5.5, similar trends are observed across other DoFs, such as the pile tip level $PPD^* = PPD + \Delta Z_{PPD}$.



(a) Monte Carlo **(b)** Quasi-Monte Carlo

Figure 5.5: Comparison of Monte Carlo and quasi-Monte Carlo sampling of frequency and pile tip level.

While the qualitative comparison is helpful in understanding the difference between the two methods, a quantitative comparison is also desirable to measure the effect the differences may have on the sample. To do this, an additional 10^5 samples are taken randomly within the ranges of the DoFs and the minimum Euclidean distance (normalized) to one of the obtained samples is determined. This is effectively the distance to the nearest neighbor. Their average across the 10^5 gives an indication of the size of the gaps in hyperspace within the domain. This exercise is performed for both databases, for all the original samples and for only the samples which were successfully tuned.

Table 5.1: Normalized average Euclidean distance to the Nearest Neighbor in the sampled database.

	All	Successful
Monte Carlo	0.3840	0.3973
Quasi-Monte Carlo	0.3674	0.3827

To better understand the behavior of the nearest neighbor distance, a similar exercise is performed on the sets with the full samples in which only subsets of the database are used. In this way, the average nearest neighbor distance is determined for the hypothetical case of using $m \in \{10, 20, ..., 100\}$ simplified models. The results are compared in Figure 5.6.

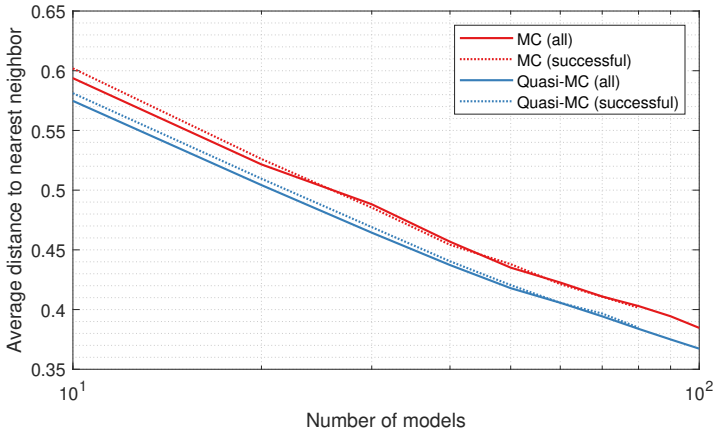


Figure 5.6: Dependence of the average nearest neighbor distance on the number of samples used for the database of simplified models.

It can be seen that the average distance to the nearest neighbor for the quasi-MC method is consistently lower than that of the MC. Moreover, the distance to the nearest neighbor is shown to follow a power law, whereby the average distance is proportional to $n_{models}^{-0.2}$, with n_{models} representing the number of structural models generated. Here it is also shown that even though some models are discarded during the process due to the non-convergence of the algorithm for frequency tuning, the effect of the missing models is minimal when more than 40 models are sampled. Finally, the nearest neighbor distance achieved by the MC method when using 100 samples (0.38) is equivalent to that of the quasi-MC when using 80 samples. This result shows that the quasi-MC can lead to a reduction in computational cost in the order of 20%. This illustrates the advantage of using a low-discrepancy, quasi-MC method over a MC method for the generation of the simplified models to ensure that the database of simplified models cover the domain more evenly.

Chapter 6

Load Emulation Using Artificial Neural Networks

Having constructed a database of simulation data, a machine learning algorithm can now be trained to predict fatigue loads at selected locations. The process of selecting the input channels, features, and encoding strategies is outlined in section 6.1. Successively, section 6.2 explains the process in which the architecture and hyperparameters of the NN are selected. In section 6.3, the selected hyperparameters are used to train the load emulators and their performance is evaluated by testing on data from simulations of detailed structural models.

6.1 Input channels and features

The selection of inputs to the NN is a vital step towards building an accurate load emulator. The predictive power of the NN is intrinsically tied to the inputs that are made available to it. In other words, the input variables provided to the model should be able to capture the variance of the output variables in order for the NN to find an accurate model. First, the basic features selected are described. Then, a series of engineered features are derived from the basic features.

6.1.1 Basic features

Because simulation data is used for the training of the model, many inputs channels are available. However the selection of input variables should be done thoughtfully, as not all of these channels can be obtained from real measurement data. Since the objective of this work is to show the potential of this methodology for practical applications, it is preferred to only consider load channels and features that could be available from measurement data. The chosen inputs consist of the standard SCADA measurements, accelerometer data at the nacelle, geometric and dynamic properties, and wave statistics. These are summarized in Table 6.1.

Table 6.1: Input variables and features selected for fatigue load emulation.

Channel or variable	Type	Features
P_{el}	SCADA	$\bar{x}$, $\sigma(x)$, $\min(x)$, $\max(x)$
θ_{pitch}	SCADA	$\bar{x}$, $\sigma(x)$, $\min(x)$, $\max(x)$
Ω	SCADA	$\bar{x}$, $\sigma(x)$, $\min(x)$, $\max(x)$
$\theta_{yaw,bearing}$	SCADA	$\sin(x)$, $\cos(x)$
$a_{nacelle,x}$	Accelerometer	$\bar{x}$, $\sigma(x)$, $\min(x)$, $\max(x)$
$a_{nacelle,y}$	Accelerometer	$\bar{x}$, $\sigma(x)$, $\min(x)$, $\max(x)$
f_1	Dynamic property	x
ζ_1	Dynamic property	x
d_{wl}	Geometric property	x
Z_{mud}	Geometric property	x
Z_{int}	Geometric property	x
α	Geometric property	x
θ_{wave}	Wave statistics	$\sin(x)$, $\cos(x)$
H_s	Wave statistics	x
T_p	Wave statistics	x

The yaw bearing is encoded in two inputs, as the sine and cosine of the angle. This is chosen because the transformation through trigonometric functions encode more physical meaning in the mathematical operations within the neural connections. This improves the performance of the model for two reasons. First, the NN is not forced to model internally the non-linearities of the trigonometric function. Second, decoupling in sine and cosine eliminates the discontinuity in the angle around 360 or 0 degrees, and directly encodes the symmetries present. This encoding strategy allows the model to learn faster and improves it accuracy of the model.

Finally, wave properties are included at this stage as loading conditions at the mudline and interface level are expected to have a high dependence on wave loads and the wave environment is not directly reflected in the other inputs. The effect of wave loads is only indirectly visible in the nacelle acceleration, which combines wind and wave conditions. These wave properties are not part of the standard measurements of wind turbines, but are included at this stage nonetheless, since they can be practically measured with a wave buoy or a wave radar. The chosen input variables and features consist of a total of 32 basic inputs to the NN.

6.1.2 Engineered features

In addition to these basic inputs, additional inputs are derived from these 32 inputs through a process of feature engineering. The engineered features are a result of mathematical operations between the basic inputs. Knowledge of the underlying physics involved in the fatigue loading of wind turbines is helpful in guiding the process with which the engineered features are created. When done successfully, the en-

gineered features can help reduce the depth (number of layers) of a NN, the training time, and can even help improve the accuracy of the NN. All this without increasing the amount of input data collected.

The first set of engineered features originate from to the relationship between the nacelle acceleration and the coordinate systems. The nacelle acceleration is recorded in a local frame of reference, with the axes aligned with the wind direction. However, tower and foundation loads are in a global coordinate system. To aid the NN in relating the direction of the acceleration with respect to the global coordinate system, the following features are included:

$$EF_1 = \sigma\left(a_{nacelle,x}\right) \sin\theta_{yaw,bearing} \tag{6.1}$$

$$EF_2 = \sigma\left(a_{nacelle,x}\right) \cos\theta_{yaw,bearing} \tag{6.2}$$

$$EF_3 = \sigma\left(a_{nacelle,y}\right) \sin\theta_{yaw,bearing} \tag{6.3}$$

$$EF_4 = \sigma\left(a_{nacelle,y}\right) \cos\theta_{yaw,bearing} \tag{6.4}$$

Additionally, the ranges of nacelle accelerations are included as engineered features:

$$EF_5 = \max a_{nacelle,x} - \min a_{nacelle,x} \tag{6.5}$$

$$EF_6 = \max a_{nacelle,y} - \min a_{nacelle,y} \tag{6.6}$$

The next group of engineered features target the relationship between the wave direction and the global reference system. Projections of various parameters related to wave intensity are considered as engineered features. These parameters are H_s, T_p, and d_{wl}^2. The engineered features then include:

$$EF_7 = H_s \sin\theta_{wave} \tag{6.7}$$

$$EF_8 = H_s \cos\theta_{wave} \tag{6.8}$$

$$EF_9 = T_p \sin\theta_{wave} \tag{6.9}$$

$$EF_{10} = T_p \cos\theta_{wave} \tag{6.10}$$

$$EF_{11} = d_{wl}^2 \sin\theta_{wave} \tag{6.11}$$

$$EF_{12} = d_{wl}^2 \cos\theta_{wave} \tag{6.12}$$

After this, a set of engineered features is constructed using powers of the frequency and diameter. These powers have been shown to appear in simplified calculations of fatigue loads [28].

$$EF_{13} = f_1^{0.5} \tag{6.13}$$

$$EF_{14} = f_1^2 \tag{6.14}$$

$$EF_{15} = f_1^3 \tag{6.15}$$

$$EF_{16} = d_{wl}^2 \tag{6.16}$$

$$EF_{17} = d_{wl}^3 \tag{6.17}$$

Finally, knowledge of the response of a structure when subjected to periodic loading is useful for defining the last batch of engineered features. The periodic loading and the response (in the form of displacements) can be related via a transfer function. A structure with natural frequency f_1, damping ζ, and stiffness K_1 has a transfer function of the first mode of the form [28]:

$$H_1(f) = \frac{1}{1 - \left(\frac{f}{f_1}\right)^2 + i2\zeta\frac{f}{f_1}} \frac{1}{K_1} \tag{6.18}$$

The absolute value of this transfer function of the first mode is:

$$H_1(f) = \frac{1}{\sqrt{\left(1 - \left(\frac{f}{f_1}\right)^2\right)^2 + 4\zeta^2\left(\frac{f}{f_1}\right)^2}} \frac{1}{K_1} \tag{6.19}$$

Using this knowledge, a set of engineered features relating to the absolute value of the transfer function for different excitation frequencies is defined. Namely, the peak wave frequency $1/T_p$ and the rotor frequencies of 1P and 3P. In all these features, the total damping ζ is approximated as the aerodynamic damping for normal operation and the structural damping for idling.

$$EF_{18} = \left[\left(1 - \left(\frac{1/T_p}{f_1}\right)^2\right)^2 + 4\zeta^2\left(\frac{1/T_p}{f_1}\right)^2\right]^{-1/2} \tag{6.20}$$

$$EF_{19} = \left[\left(1 - \left(\frac{\bar{\Omega}/60}{f_1}\right)^2\right)^2 + 4\zeta^2\left(\frac{\bar{\Omega}/60}{f_1}\right)^2\right]^{-1/2} \tag{6.21}$$

$$EF_{20} = \left[\left(1 - \left(\frac{\min\Omega/60}{f_1}\right)^2\right)^2 + 4\zeta^2\left(\frac{\min\Omega/60}{f_1}\right)^2\right]^{-1/2} \tag{6.22}$$

$$EF_{21} = \left[\left(1 - \left(\frac{\max\Omega/60}{f_1}\right)^2\right)^2 + 4\zeta^2\left(\frac{\max\Omega/60}{f_1}\right)^2\right]^{-1/2} \tag{6.23}$$

$$EF_{22} = \left[\left(1 - \left(\frac{3\bar{\Omega}/60}{f_1}\right)^2\right)^2 + 4\zeta^2\left(\frac{3\bar{\Omega}/60}{f_1}\right)^2\right]^{-1/2} \tag{6.24}$$

$$EF_{23} = \left[\left(1 - \left(\frac{3\min\Omega/60}{f_1}\right)^2\right)^2 + 4\zeta^2\left(\frac{3\min\Omega/60}{f_1}\right)^2\right]^{-1/2} \tag{6.25}$$

$$EF_{24} = \left[\left(1 - \left(\frac{3\max\Omega/60}{f_1}\right)^2\right)^2 + 4\zeta^2\left(\frac{3\max\Omega/60}{f_1}\right)^2\right]^{-1/2} \tag{6.26}$$

As such, a total of 24 engineered features are appended to the list of 32 basic features for a total of 56 inputs.

A final step in the feature engineering is the normalization or transformation of the input and output variables. To ensure that all inputs are approximately in the same scale, the z-score of the input variable is used. Even though the inputs may not be normally distributed, the z-score is still effective in normalizing the scale of all inputs. For the outputs, skewed distributions of the target DEL loads motivated the use of transformations for some load channels (mudline, interface, and tower top). The applied transformation was taking the logarithm of the 10-minute DEL load. Doing so transforms the target variable to a symmetric, centered distribution which facilitates the training of the NN. This is possible because tower and foundation loads are approximately Weibull distributed. Independent of whether the transformation is applied, scaling of the outputs is not necessary as the activation function of the last layer is a linear activation.

$$z_i = \frac{x_i - \bar{x}}{\sigma} \tag{6.27}$$

6.2 Selection of architecture and hyperparameters

To maximize the performance of the SM, it was chosen to create separate models for each of loading hotspot. This increases the overall accuracy of the load predictions as the learning algorithm can minimize the error across each output channel independently with no trade-offs between output channels. In a one-for-all architecture, the learning algorithm must balance improving the accuracy of, for example, mudline level loads at the expense of decreasing accuracy at other locations. In total, 6 load emulators are created: (1) mudline level moment in x and y, (2) interface level moment in x and y, (3) tower top moment in x and y, (4) flapwise moment at the

root of the blade, (5) edge-wise moment at the root of the blade, and (6) torsion at the root of the blade. In all cases, the target channel is the DEL over a 10-minute simulation period. Of these 6, primary focus is given to the foundation and tower loads as the modeling assumptions to simplify the support structure will mainly affect these load channels. The blade root load emulators are developed to serve as a point of comparison as it is expected that these load channels will be largely unaffected by the modeling assumptions made during the modeling process.

For each, a suitable NN architecture must be selected. As described in section 6.1, the input layer consists of 56 units. The output layer, depending on the load channel, consists of 1 or 2 units. In all cases, hidden layers are implemented using ReLU activation functions, and linear activation is used for the output layer. To guide the optimization algorithm, the Mean Absolute Percentage Error (MAPE) loss function is used with an Adam optimization algorithm.

When training a NN, there are many possible choices for the hyperparameters that define the model. These choices include selecting a NN architecture and selecting a learning rate and mini-batch size. It is impossible to know a priori which parameters will yield the best performance. To determine this, a cross-validation set is used. In this case, 80% of the available data is used for training while 20% of the data is set aside for cross-validation. However, conducting an exhaustive grid search for the optimal hyperparameters is computationally expensive. To alleviate the computational cost, random sampling is used instead.

While there is no definite rules for selecting a NN architecture, a common rule of thumb is to employ somewhere between $n_{inputs}/2$ and $2n_{inputs}$ units in each hidden layer. In this exercise, hidden units are considered in the range of 30 to 60 per layer. This is on the lower end of the range as it is expected that the engineered features will provide sufficient information to the NN such that less hidden units are required. For the depth of the NN, NNs with 2 to 5 hidden layers are considered. Both the number of hidden layers (depth) and number of hidden units per layer are sampled using discreet, uniform distributions.

On the other hand, the learning rate α_{NN} and the mini-batch size are sampled using logarithmic distributions to cover several orders of magnitude. For α_{NN}, the value is defined as 10^{x_1} with x_1 sampled uniformly between -4 and -1. For the mini-batch size a similar technique is employed, however the mini-batch is constrained to exact powers of 2. This choice is to take full advantage of the numerical methods inherent to the TensorFlow library, which are more efficient when using mini-batch sizes which are multiples of 2. Therefore, the mini-batch size is defined as 2^{x_2}, where x_2 is sampled from a discreet uniform distribution between 6 and 10.

For each load emulator, 60 random combinations of these parameters are chosen and 60 models are trained based on the same training data for 400 epochs. The models are ranked in terms of their performance on the cross-validation set and the best performing parameters are chosen for each load emulator.

At this stage, only the results for the load emulators created using a Monte-Carlo sampling method are presented. In chapter 7 the differences between load emulators created using the quasi-random sampling strategies and the Monte-Carlo sampling

will be compared.

Table 6.2: Neural network architectures and hyperparameters selected for each load emulator.

Load emulator	N_{hidden}	n_{units}	Parameters	α_{NN}	Batch size
$M_{mud,X}$, $M_{mud,Y}$	4	53	11715	2.44×10^{-3}	128
$M_{int,X}$, $M_{int,Y}$	4	46	9202	2.39×10^{-3}	64
$M_{towertop,X}$, $M_{towertop,Y}$	3	42	6092	1.89×10^{-3}	64
$M_{b,flap}$	4	54	12043	5.37×10^{-2}	64
$M_{b,edge}$	4	48	9841	3.22×10^{-2}	64
$M_{b,tors}$	3	39	5383	4.24×10^{-2}	128

A summary of the best performing hyperparameters for each load emulator is presented in Table 6.2. In general, the complexity as measured by the number of parameters is comparable across the different load channels. The most complex load emulators are the mudline and flapwise load emulators, while the tower-top load emulator and the torsional blade load emulator have the least trainable parameters. It is seen that the optimal batch size for the load emulators is around 64 and 128. It is also noted that the order of magnitude of the learning rate α_{NN} is also similar across the load emulators. As a reminder, it is noted that blade load emulators are trained on the DEL load itself, while the other three load emulators are trained on the logarithm of the DEL load. This explains the differences in α_{NN} between these two groups.

Because a training dataset of approximately 22000 samples is used, there is some risk of overfitting the most complex NNs. In most cases, there is only 2 or 3 samples per trainable parameter. This risk is managed by analyzing the convergence of training and cross-validation losses. Later analysis in chapter 7 will determine how much training data is actually required.

6.3 Testing on detailed models

Once the hyperparameters are chosen using the mentioned process, the resulting load emulator is then used on the testing data set. Testing is performed on a dataset separate from the training and cross-validation to detect possible overfitting of the hyperparameters to the data. This data set was obtained by performing simulations on 2 site-specific, detailed models provided by SGRE representing real sites in the North Sea. Approximately 9000 simulations between the two models are used to build the test set. The results for each load emulator when tested on this dataset are presented in this section.

6.3.1 Mudline load emulator

The hyperparameters selected in the previous subsection are used for a load emulator for predicting mudline-level bending DELs. In Figure 6.1, the convergence of the loss function (MAPE) for the training and cross-validation data is shown. Because the target values are the natural logarithm of the load, the value of the loss function is relatively low. Since two output units are present in this load emulator, the loss function is the average of both load channels (X and Y axes). Some noise is present in the cross-validation loss which may indicate that the size of the cross-validation set is slightly small. Future implementations of the methodology may see increased reliability in the cross-validation set by using a larger cross-validation set.

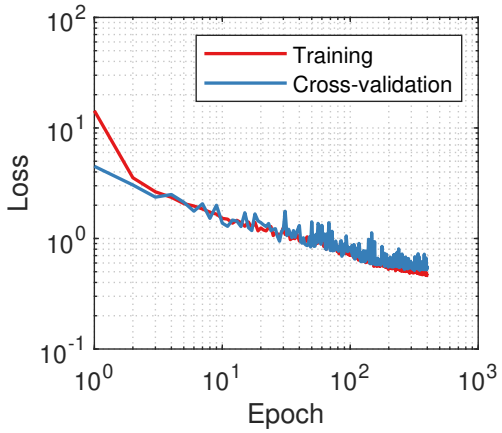
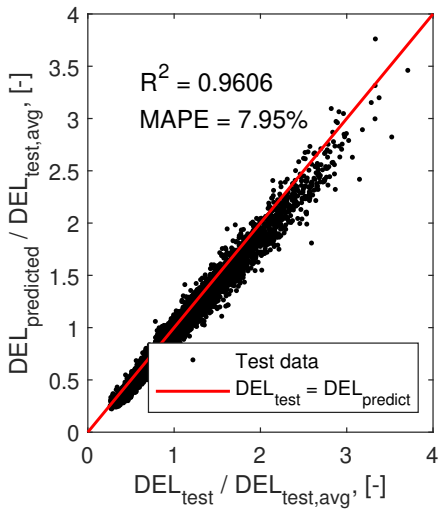


Figure 6.1: Convergence of loss function of training and cross-validation sets for the mudline load emulator.

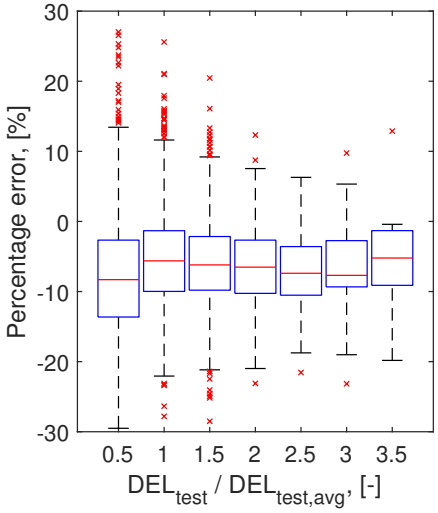
The results for each output unit are also analyzed separately in Figure 6.2 and Figure 6.3 for the moments about the X-axis and Y-axis respectively. In Figure 6.2(a) the predicted loads are compared directly with the known test loads. It is seen that the cloud of points closely follows the line $DEL_{test} = DEL_{predict}$, which indicates a good performance of the load emulator. The quantitative metrics support this, as the coefficient of determination was found to be 0.9606 and the MAPE for this load channel was 7.95%. One downside of the data representation of the scatter plot in Figure 6.2(a) is that the large number of scatter points makes it difficult to see the density of the data points. For this reason, the distribution of known test values is included in Figure 6.2(c), and binned box-plots are presented in Figure 6.2(b).

In Figure 6.2(c) it is seen that the mudline loads follow a skewed distribution in which the majority of the values are concentrated around $DEL_{test}/DEL_{test,avg} = 1$. Because of this, it is reasonable to expect that the algorithm will perform best at this load level and exhibit higher percentage errors for other load levels since they are less common. This is visible in the box-plots of Figure 6.2(b). The boxes are narrowest and closer to a percentage error of 0% for loads between 0.5 and 1.5 times the average load. It is also seen that very low DELs (less than 50% of the

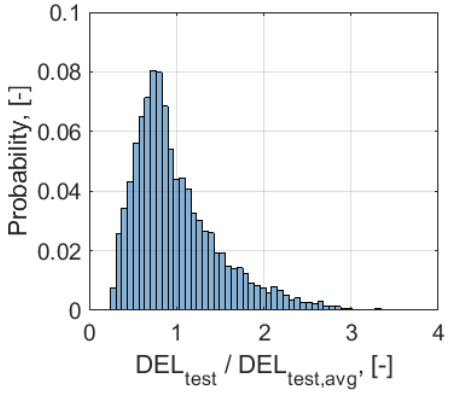
mean DEL) have higher variations in the tails of the box plots. This is due to the nature of MAPE being a relative performance metric: small absolute errors become higher relative errors when the load is smaller. A few outliers are present in the fitted data, as some predictions overestimate the loads by more than 20% (shown in red crosses). However, Figure 6.2(d) shows that these represent a very small fraction of the test data. The percentage errors are shown to follow an approximately Gaussian distribution, with a mean of -6.63% and a standard deviation of 6.76%. Additionally, the RMSE for this load channel is 9.74%. The percentage error of the total DEL for this location is -5.91%.



(a) Predicted vs test DELs



(b) Binned percent error distribution of DEL prediction

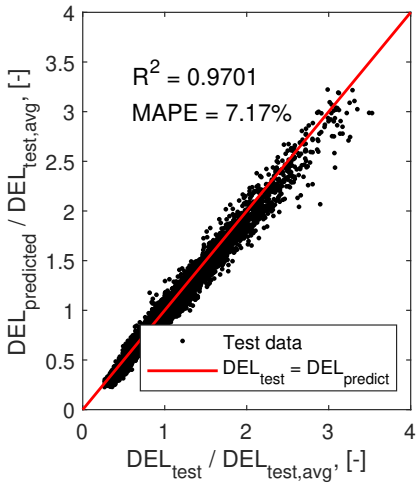


(c) Distribution of test DELs

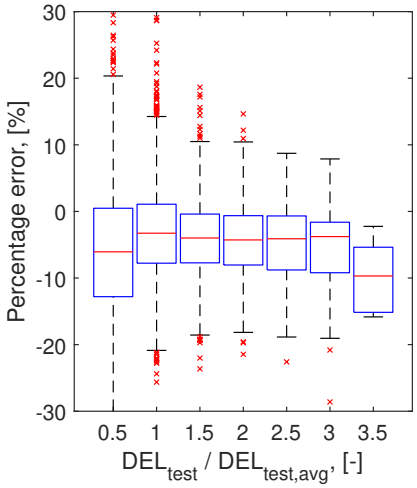
(d) Total percent error distribution of DEL prediction

Figure 6.2: Performance of the load emulator for mudline level bending loads (X-axis) on the test data.

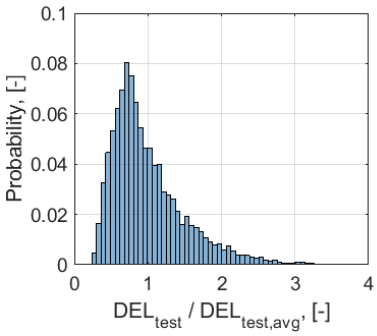
The results for the mudline level bending load about the Y-axis are very similar. This is expected as the full 360 degrees were considered for the wind and wave directions. Therefore, the distributions of inputs and outputs are analogous. This is confirmed by comparing the target distributions in Figure 6.2(c) and Figure 6.3(c). The behavior of the load emulator is very similar as for the X-axis, as the metrics of R^2 and MAPE are found to be very similar to those of the X-axis load (Figure 6.3(a)). A similar analysis of Figure 6.3(b) shows that the NN under-predicts the DEL by about 4% throughout the range of DELs. The error in the Y-axis load follows a Gaussian distribution. The mean of the Gaussian distribution is -4.41% and the standard deviation is 7.59%. The RMSE for this channel is 9.15% and the error in the total DEL is -3.86%.



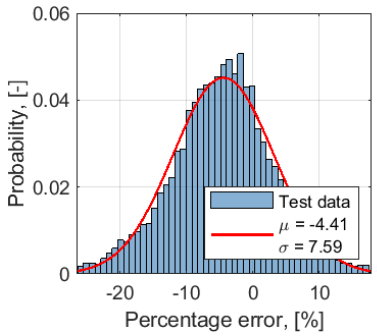
(a) Predicted vs test DELs



(b) Binned percent error distribution of DEL prediction



(c) Distribution of test DELs



(d) Total percent error distribution of DEL prediction

Figure 6.3: Performance of the load emulator for mudline level bending loads (Y-axis) on the test data.

6.3.2 Interface load emulator

In this section, the optimal hyperparameters for the interface load emulator were used to train a SM, which was then evaluated on the test data set. It is shown in Figure 6.4 that the cross-validation loss is approximately converged after 400 epochs.

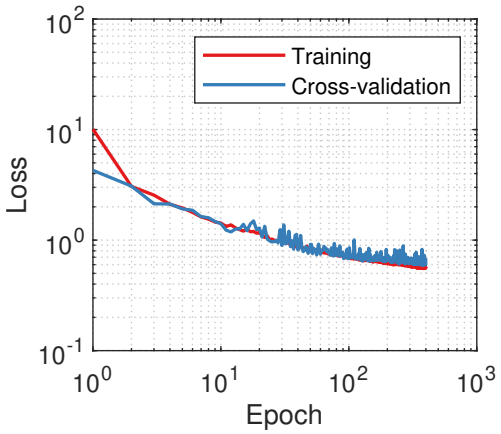
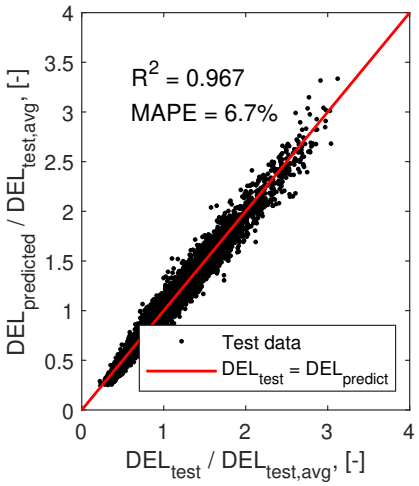
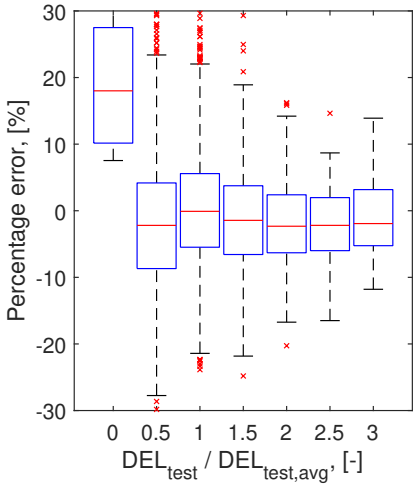


Figure 6.4: Convergence of loss function of training and cross-validation sets for the interface load emulator.

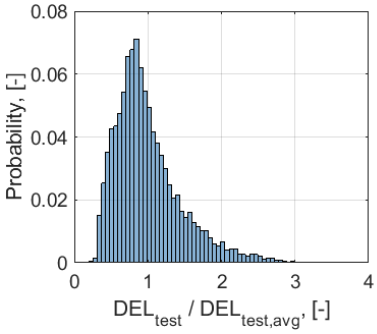
The results for emulating the interface bending load about the X-axis are shown in Figure 6.5. Figure 6.5(a) shows that the predicted loads closely match the known loads from the test data set. Quantitatively, the coefficient of determination is 0.9670 and the MAPE is 6.70%. As was the case for mudline loads, the distribution of the test DELs follow a skewed distribution with the majority of the data between 0.2 and 1.5 times the mean of the test DELs (Figure 6.5(c)). The box-plots in Figure 6.5(b) show that the error is kept around 0% throughout most of the range of DELs. The exception is for low DELs, the NN tends to overestimate these loads by around 20-30%. In Figure 6.5(d) the error is shown to follow a Gaussian distribution with a mean of -0.90% and a standard deviation of 7.99%. The RMSE for this load channel is 8.46% and the total DEL is estimated with an error of -0.74%.



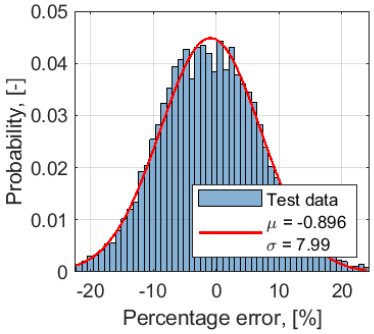
(a) Predicted vs test DELs



(b) Binned percent error distribution of DEL prediction



(c) Distribution of test DELs

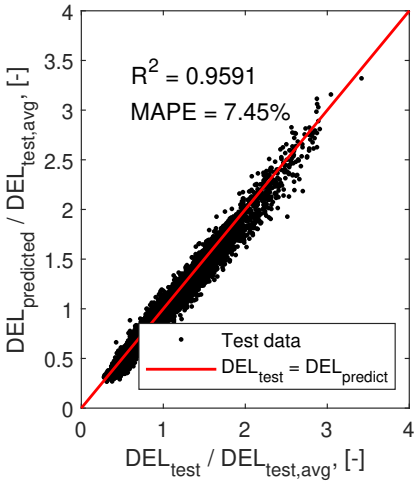


(d) Total percent error distribution of DEL prediction

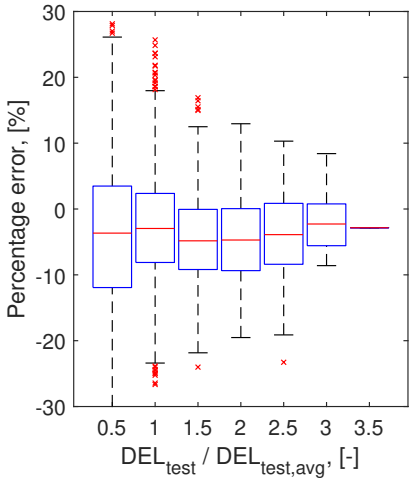
Figure 6.5: Performance of the load emulator for interface level bending loads (X-axis) on the test data.

Similarly, Figure 6.6(a) shows that the metrics of the load emulator for the bending loads about the Y-axis are similar to those of the X-axis. The reason for this is the same as was described when discussing the mudline loads: the input and output distributions for the two loads are essentially identical given that all wind directions and wave directions are considered with equal probability and the loads are emulated in a global coordinate system. Because of this, the distribution of the test data (Figure 6.6(c)) is very similar to that of load about the X-axis. In Figure 6.6(b) the NN is shown to maintain low errors across the entire range of DELs, with a slight tendency to under-predict the DELs. The distribution of the error shown in Figure 6.6(d) is approximately Gaussian with a mean of 3.65% and a standard deviation of 8.30%. The RMSE is 9.46% and the total DEL is estimate within a margin of 3.46%. Compared to the X-axis load emulator, the $Y - axis$ load emulator presents slightly more

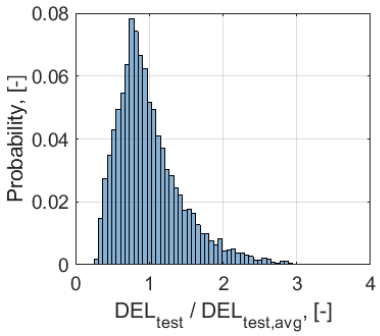
dispersion and bias.



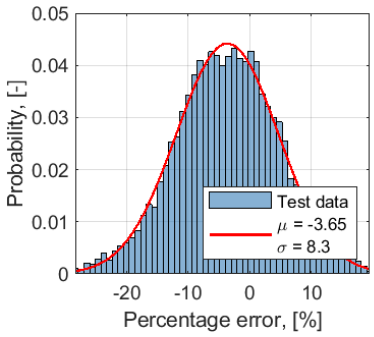
(a) Predicted vs test DELs



(b) Binned percent error distribution of DEL prediction



(c) Distribution of test DELs



(d) Total percent error distribution of DEL prediction

Figure 6.6: Performance of the load emulator for interface level bending loads (Y-axis) on the test data.

6.3.3 Tower-top load emulator

The load emulator for tower-top loads is used on the test data in order to assess its accuracy. The convergence history of the training process is shown in Figure 6.7. In this figure, the tower-top DEL emulator is shown to reach approximate convergence after 400 epochs. The training and cross-validation losses are very similar, suggesting that overfitting of the data was successfully avoided.

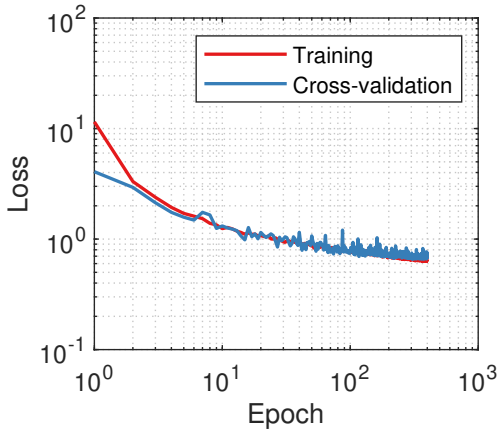
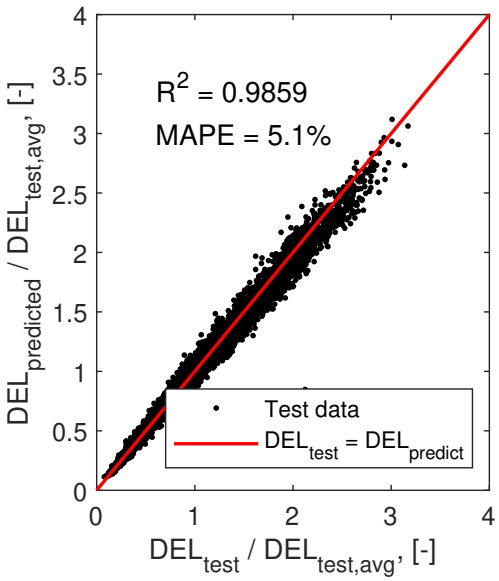
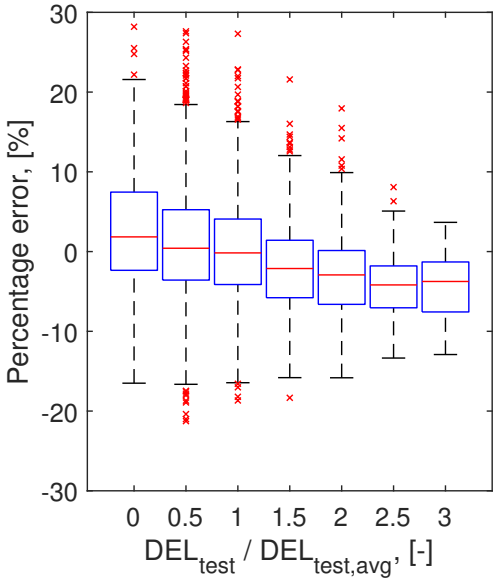


Figure 6.7: Convergence of loss function of training and cross-validation sets for the tower-top load emulator.

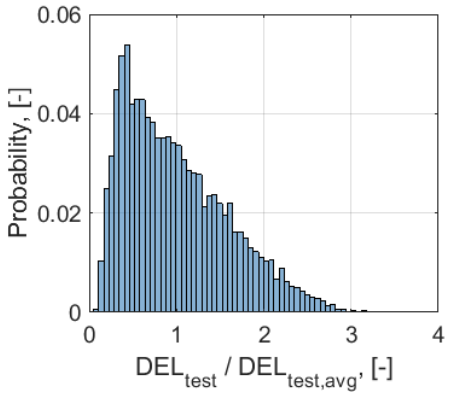
The tower-top bending load about the X-axis is analyzed in Figure 6.8. From Figure 6.8(a) it is seen that the predicted loads closely match the test loads, achieving an R^2 of 0.9859 and a MAPE of 5.10%. Like the load emulators for the previous hotspots, the error is approximately Gaussian (Figure 6.8(d)), with a mean of -0.22% and a standard deviation of 6.16%. These metrics indicate that the accuracy of this load emulator is slightly better than the accuracy of the load emulators for mudline and interface level loads. The test data is again shown to have a skewed distribution, with a mode just below the mean value (Figure 6.8(c)). The load emulator is shown to have a good performance across the range of test DELs considered, with a slight tendency to overestimate lower loads and underestimate the higher loads (Figure 6.8(b)). As was the case in the mudline and interface level load emulators, the load about the Y-axis is predicted to a very similar accuracy as the load about the X-axis (see Figure 6.9). The RMSEs and total DEL errors obtained for this hotstpot were 6.54% and -2.49% for the loads about the X-axis and 8.99% and -4.51% for the loads about the Y-axis, respectively.



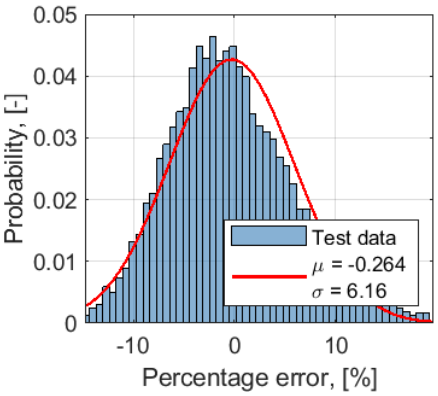
(a) Predicted vs test DELs



(b) Binned percent error distribution of DEL prediction

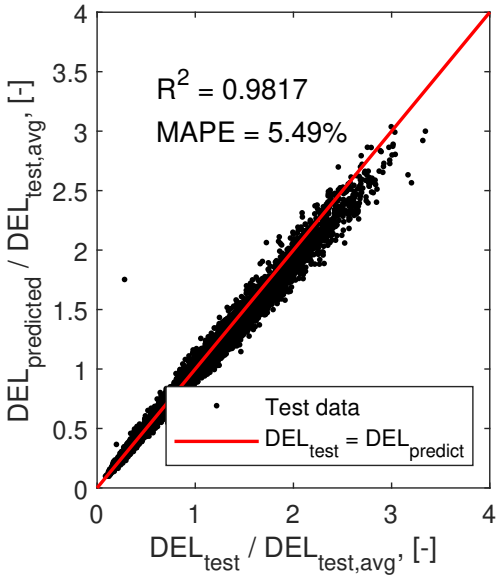


(c) Distribution of test DELs

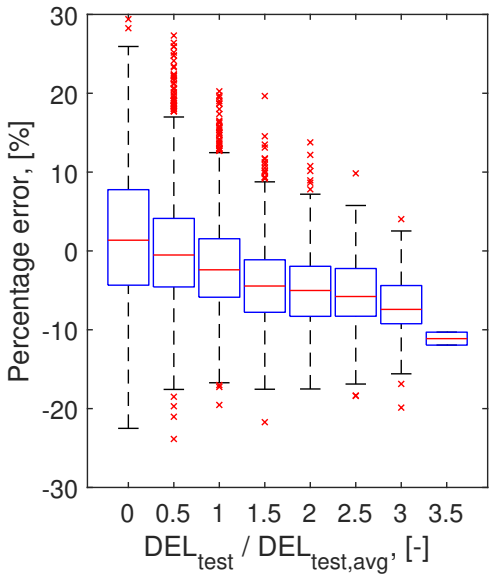


(d) Total percent error distribution of DEL prediction

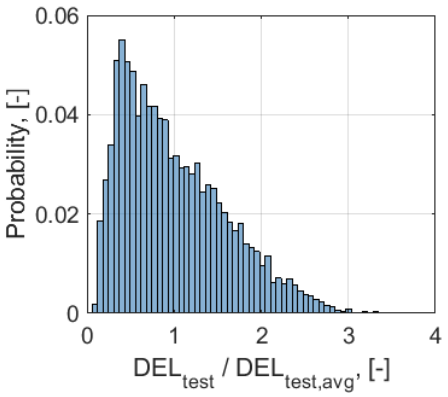
Figure 6.8: Performance of the load emulator for tower top bending loads (X-axis) on the test data.



(a) Predicted vs test DELs



(b) Binned percent error distribution of DEL prediction



(c) Distribution of test DELs

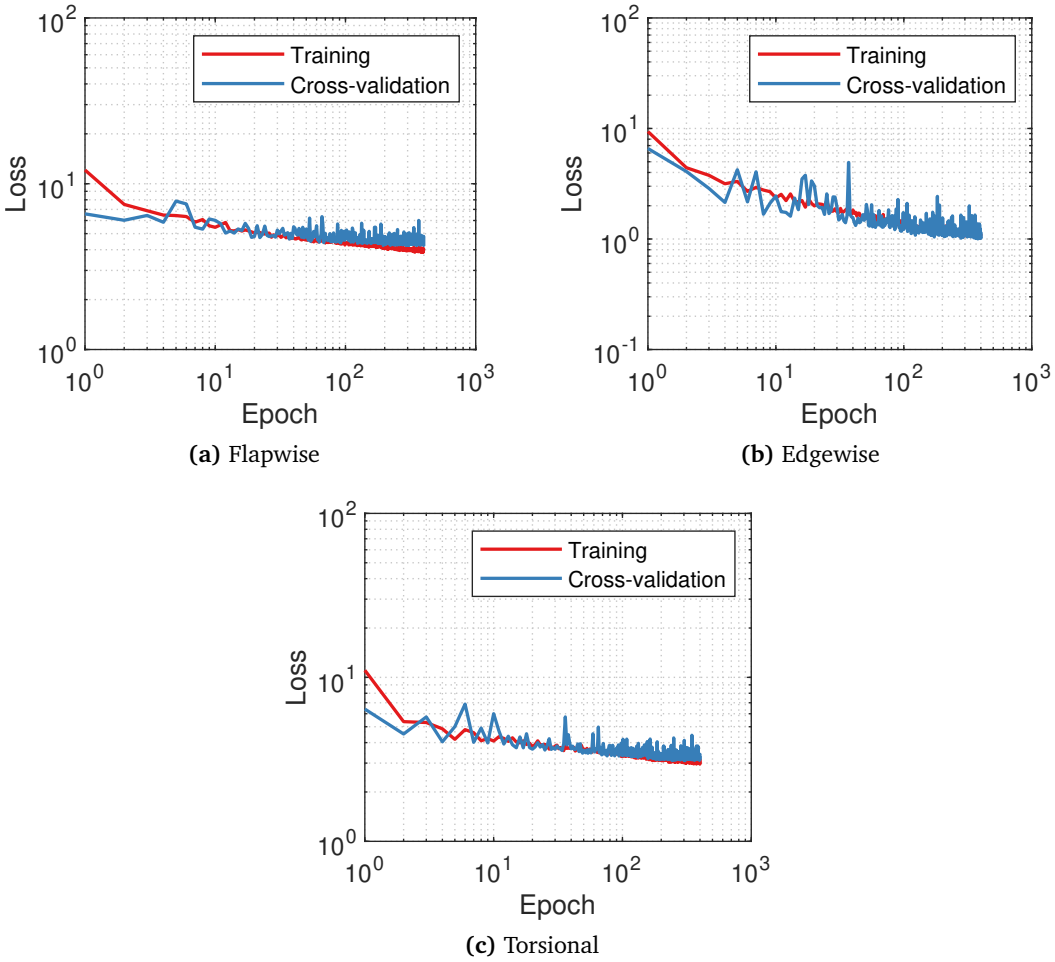
(d) Total percent error distribution of DEL prediction

Figure 6.9: Performance of the load emulator for tower top bending loads (Y-axis) on the test data.

6.3.4 Blade root load emulator

Finally, the results of the blade-root load emulators are shown in this section. The blade root loads were emulated using three separate NNs, therefore three different training loss convergence plots are presented in Figure 6.10. In all cases, the loss function is approximately converged after 400 epochs. The training loss and cross-validation loss are very similar to each other, indicating low variance in the trained

model.



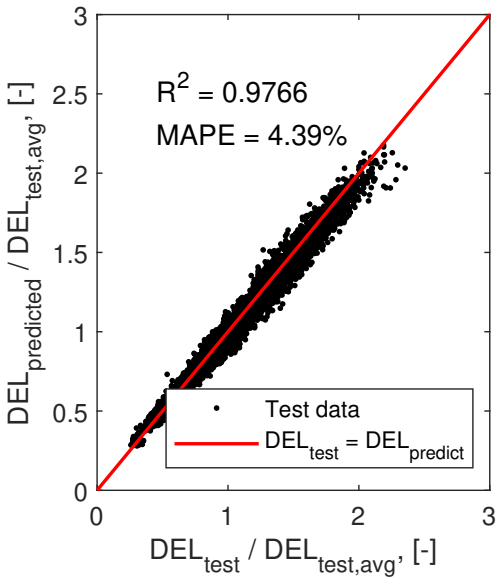
(a) Flapwise

(b) Edgewise

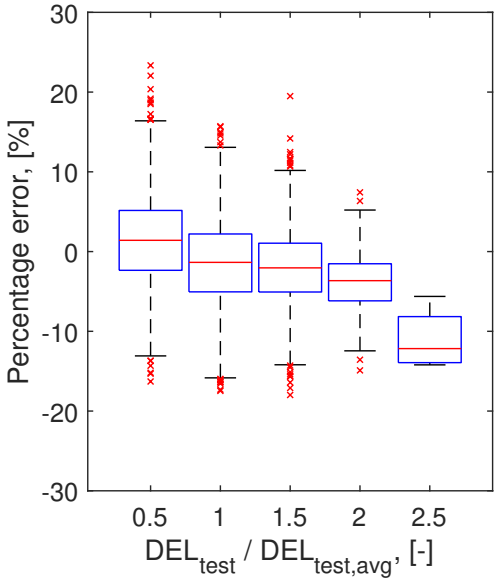
(c) Torsional

Figure 6.10: Convergence of loss function of training and cross-validation sets for the blade-root load emulators.

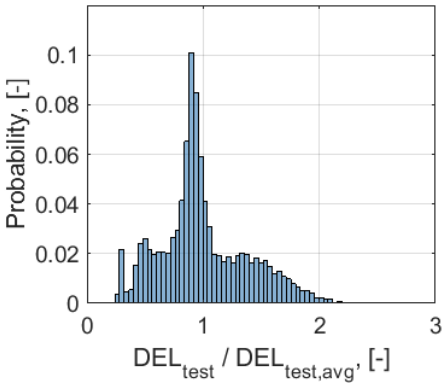
The emulation of the flapwise moment at the blade-root is evaluated in Figure 6.11. It is seen that the loads are predicted with high accuracy, achieving a MAPE of 4.39% and an R^2 of 0.9766 (Figure 6.11(a)). The distribution of this load channel is different to that of the previous loads analyzed. The distribution shows a strong peak around the mean value and is approximately symmetric around this value (Figure 6.11(c)). In Figure 6.11(d), the prediction error is approximately Gaussian, with a mean value of -1.02% and a standard deviation of 5.19%. The box-plots in Figure 6.11(b) show that the error is distributed very similarly independent of the intensity of the load that is to be emulated, with a slight tendency to under-estimate very high loads. The RMSE for this channel is 5.33% while the error in the total DEL load is -2.73%.



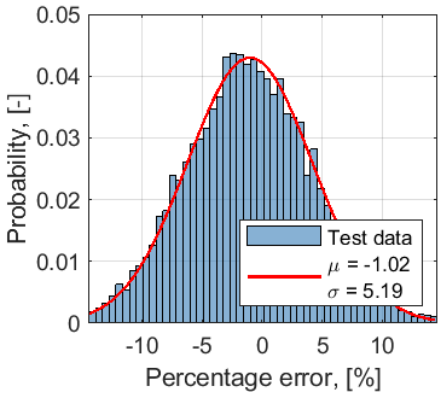
(a) Predicted vs test DELs



(b) Binned percent error distribution of DEL prediction



(c) Distribution of test DELs

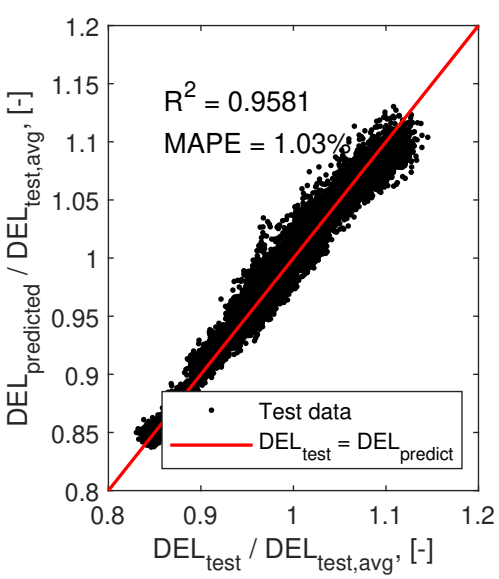


(d) Total percent error distribution of DEL prediction

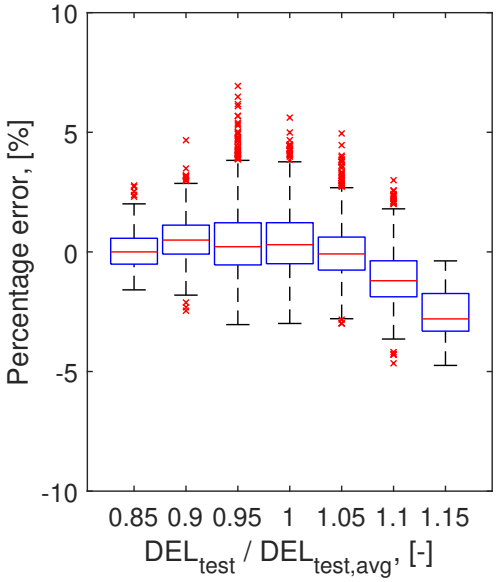
Figure 6.11: Performance of the load emulator for blade root bending loads (flapwise) on the test data.

A similar analysis is shown in Figure 6.12 for the edgewise load. It is highlighted that the distribution of the DELs of this location is much narrower than the previous locations, with the majority of the loads being within 10% of the average test DEL (Figure 6.12(c). As such, the performance of the model as measured by the MAPE is extremely good; the load is estimated on average within a 1.03% error (Figure 6.12(a). However, the performance as measured by the R^2 is 0.9581, which is lower than the other load emulators. The reason for this is that the narrow distribu-

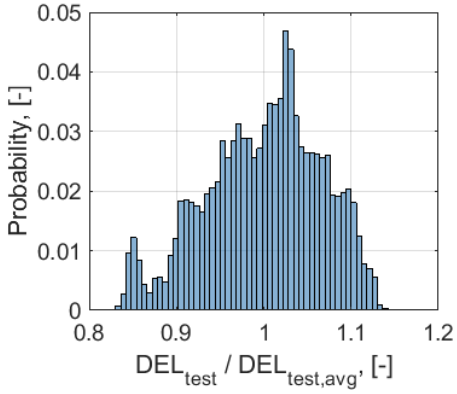
tion of target values causes the load emulator to estimate close to the mean value and this is not punished severely by the training algorithm as the MAPE is kept low. The error for this load is approximately Gaussian with an average of 0.07% and a standard deviation of 1.26% (Figure 6.12(d)). There is a slight tendency to over-estimate lower loads and underestimate the higher loads, as shown in Figure 6.12(b). The RMSE for the edgewise moment at the blade root is 1.33% and the total DEL is estimated with an error of -0.09%.



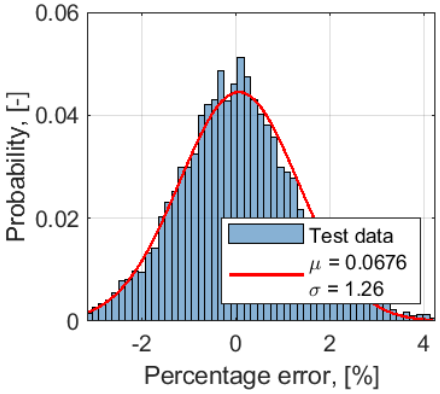
(a) Predicted vs test DELs



(b) Binned percent error distribution of DEL prediction



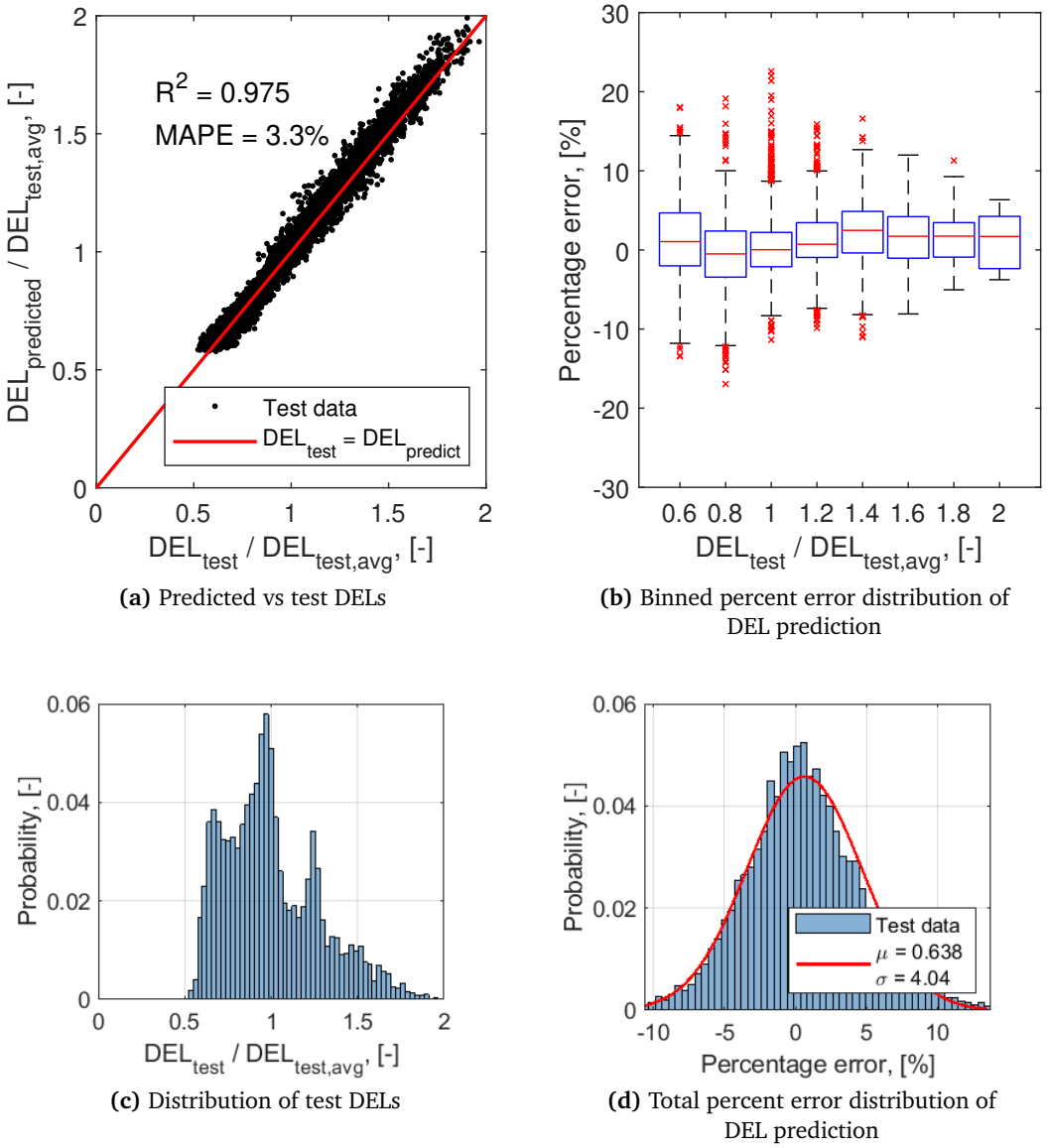
(c) Distribution of test DELs



(d) Total percent error distribution of DEL prediction

Figure 6.12: Performance of the load emulator for blade root bending loads (edgewise) on the test data.

Finally, the torsional moment at the blade root is analyzed in Figure 6.13. As in previous load emulators, the accuracy is shown to be high, with a MAPE of 3.30% and a coefficient of determination of 0.9750. The RMSE was found to be 4.32% while the error in the total DEL is 2.03%.



(a) Predicted vs test DELs

(b) Binned percent error distribution of DEL prediction

(c) Distribution of test DELs

(d) Total percent error distribution of DEL prediction

Figure 6.13: Performance of the load emulator for blade root bending loads (torsional) on the test data.

Table 6.3: Neural network architectures selected for each load emulator.

Load	MAPE [%]	RMSE [%]	R^2	MPE [%]	tDELE [%]
$M_{mud,X}$	7.95	9.74	0.9606	-6.62	-5.91
$M_{mud,Y}$	7.17	9.15	0.9701	-4.40	-3.86
$M_{int,X}$	6.70	8.46	0.9670	-0.86	-0.74
$M_{int,Y}$	7.45	9.46	0.9591	-3.64	-3.46
$M_{towertop,X}$	5.10	6.54	0.9859	-0.22	-2.49
$M_{towertop,Y}$	5.49	8.99	0.9817	-1.78	-4.51
$M_{b,flap}$	4.39	5.53	0.9766	-1.00	-2.73
$M_{b,edge}$	1.03	1.33	0.9581	0.08	-0.09
$M_{b,tors}$	3.30	4.32	0.9750	0.65	2.03

6.3.5 Summary

The results of the six load emulators are presented in Table 6.3. In general terms, the load emulators showed that they can accurately predict the fatigue loads at the locations studied for site-specific cases. This proves the generic concept of the load emulator methodology presented in this work, as the load emulator is successfully applied to predict loads of two detailed models it has not seen during training. The results show that the load emulator could be applied to any site which is within the ranges of geometric, dynamic, and environmental DoFs used to create the SMs.

The value of studying various error metrics is shown in Table 6.3. No single error metric can fully describe the accuracy of each load emulator. Using several error metrics that measure goodness of fit, central tendencies, and dispersion of estimates allows for a holistic assessment of each load emulator. Dispersion metrics for the load emulators at mudline and interface level are the highest of all the load emulators. However, when compared in terms of the coefficient of determination or the total DEL error, these load emulators compared well with the other load channels.

A common trend throughout the load emulators is the tendency of these models to under-predict the DEL loads. This may come as a consequence of having used the MAPE as the loss function. This loss function has some known disadvantages which affect the performance of the trained model [36]. Notably, the MAPE is unbounded for over-prediction error but is bounded at 100% for under-prediction. This means that over-estimates are penalized more severely by this loss function and therefore the trained model is incentivized to under-predict. Alternatives to the MAPE loss function are proposed in literature to overcome this issue, however, they were not considered within the scope of this work.

Chapter 7

Feature Importance and Sensitivity of Training Data

In this chapter, the load emulators created in chapter 6 are studied more closely. The two sampling techniques presented in chapter 5 are compared in section 7.1. There, NNs are compared when trained on data obtained through Monte Carlo sampling and quasi-Monte Carlo sampling. The comparison includes a study on the size of the training data is performed to determine how much training data is required. Afterwards, a study on the importance of each input feature is performed in subsection 7.1.2. This is done through a global variance analysis that can quantify the relative importance of each input feature.

7.1 Sensitivity to training data

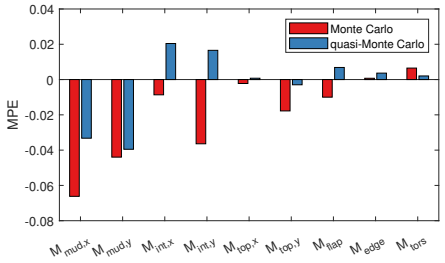
The relationship between the training data provided to the NN and the performance of the resulting load emulator is not well understood at this stage. It is unknown what effect the two sampling strategies considered can have on the overall performance of the trained model. This section aims to gain a deeper understanding by comparing load emulators trained on random data and quasi-random data. First, NNs trained on data from a quasi-Monte Carlo sampling method are compared to NNs trained on data from a Monte Carlo sampling method in terms of performance metrics. Afterwards, an analysis on the dependence of the NN on the size of training data set is performed on both variants.

7.1.1 Sampling method

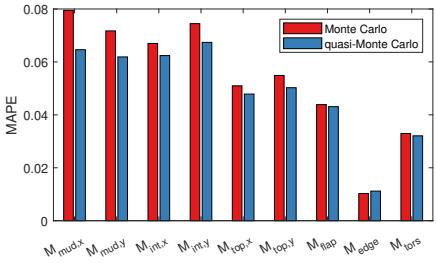
Using the hyperparameters obtained in the previous chapter for each load emulator, 6 new load emulators are trained using data obtained via a quasi-Monte Carlo sampling method. The size of the training data set and the distributions are identical, only the method in which the samples are taken is changed. Performance metrics are calculated on the test data set which is composed of simulations of site-specific, detailed

structural models at 2 sites in the North Sea. Training is performed in both cases for 400 epochs.

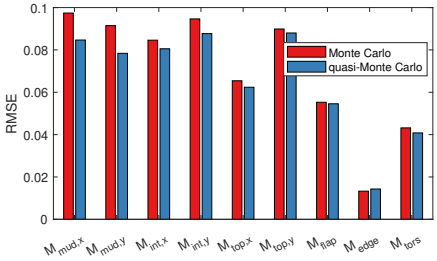
Comparison of each metric to those obtained with MC data is shown in Figure 7.1. The results show that the quasi-MC sampling method performs well on the test set, and even outperforms the NNs trained on MC sampling in most metrics. For most models, quasi-MC sampling performs better than the MC sampling for all metrics. Only the load emulator concerned with the edgewise blade-root load channel is shown to perform better when using quasi-MC sampling, and the differences in this case are in the order of 0.1% across all metrics. Overall, the quasi-MC method outperforms the MC method for more than 80% of the metrics.



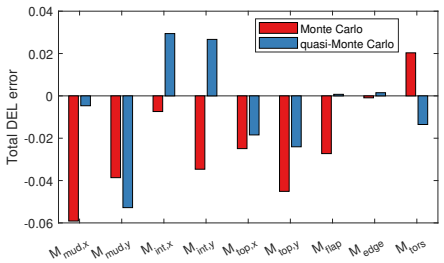
(a) MPE



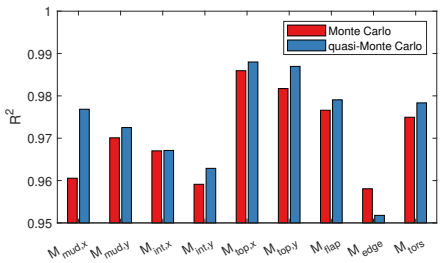
(b) MAPE



(c) RMSE



(d) Total DEL error



(e) Coefficient of determination

Figure 7.1: Comparison of performance metrics for load emulators trained on random and quasi-random data.

Moreover, the improvement of each performance metric can be quantified. On

average, NNs trained on quasi-MC data resulted in MPEs and tDELEs 1% closer to 0 than the NNs trained on MC data. Similarly, the quasi-MC NNs obtained MAPEs and RMSEs 0.5% lower than their counterparts, and an R^2 0.003 greater than for the NNs trained on MC data.

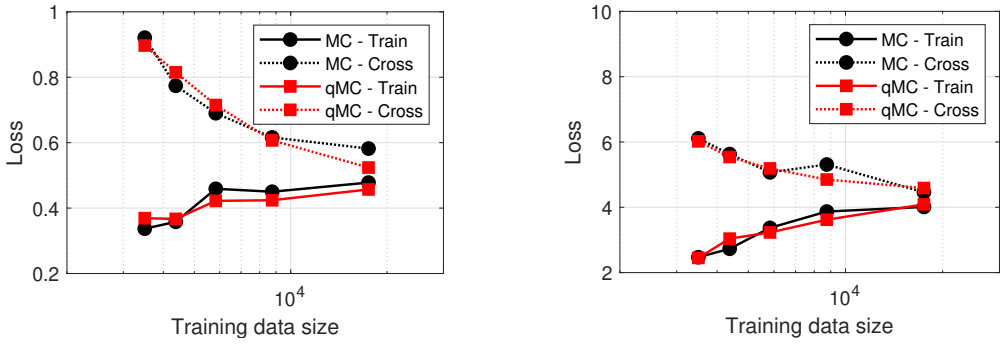
7.1.2 Training data size dependence

It is desirable to understand the effect of the size of training data used on the overall performance of the network. So far, a sample of approximately 17500 samples is used for training, and 7500 samples for cross-validation. However this sample size was not rigorously determined. Therefore, it is not known if less samples are sufficient to train the model (reducing computational time) or if additional samples are required. The use of the cross-validation data during the optimization of parameters does have an effect of limiting high variance (overfitting) of the trained models.

To perform this analysis, several NNs are trained for two different load emulators in which some data is withheld during training. The load emulators selected were the mudline load emulator and the flapwise moment at the blade root. Training is performed on the network for 100%, 50%, 33%, 25% and 20% of the available training data. For each case, cross-validation is performed on the same set of approximately 7500 samples. The size of the training data set can be compared to the training and cross-validation losses to visualize the convergence of the model and identify possible overfitting. For consistency across each case, the number of epochs used to train is defined inversely proportional to the training data size. For example, when 100% of the data is used, the NN is trained for 400 epochs. When 25% of the data is used, the NN is trained for 1600 epochs. This ensures that the same number of gradient descent steps are made in each case.

The results of this analysis is shown in Figure 7.2. For the mudline load emulator shown in Figure 7.2(a), it is seen that using less data with the current network architecture can result in high variance as a result of overfitting. This is visible from the differences between the training loss and cross-validation loss for both MC and quasi-MC (qMC) sampling methods. This observation also holds true for the the flapwise blade load emulator in Figure 7.2(b). The amount of training data used is shown to have adequate results, greatly limiting the differences between training and cross-validation losses. However, for both load emulators there is some room for improvement by increasing the training data size. Following the trends of both figures, it is expected that with training data in the order of 30,000 samples the cross-validation loss and training loss would be nearly identical. This however represents a doubling of the computational cost required to obtain the training data for relatively small improvements in performance.

Comparing the behavior of the quasi-MC NNs to the MC NNs, the quasi-MC NNs have consistently lower training losses and a faster convergence of the cross-validation losses. This means that in general, quasi-MC data is easier for the NN to fit to, and fitting to this data generally aids in reducing overfitting.



(a) Mudline **(b)** Blade (flapwise)

Figure 7.2: Dependence of load emulator performance on training data size.

7.2 Feature importance

As black-box models, it is difficult to quantify the relative importance of each input feature in NNs. This is specially true for deeper NNs, where the relationship between inputs can be confounded. One method to understand the sensitivities of the built model to the inputs given to the model is through global variance analysis. This analysis relates the total variance observed in the output of the model to the variance of the inputs.

A black-box model $Y = f(X)$ can be decomposed in several sub-models in the form [37]:

$$Y = f_0 + \sum_{i=1}^{d} f_i(X_i) + \sum_{i<j}^{d} f_{ij}(X_i, X_j) + ... \tag{7.1}$$

In this representation, f_0 is the expected value of $f(X)$, while $f_i(X_i)$ is the expected value of $f(X|X_i)$. Just as the model can be decomposed in terms of contributions of each input individually, and their combinations, so can the variance. Taking the variance of Equation 7.1 yields the following expression for the variance of the model:

$$Var(Y) = \sum_{i=1}^{d} V_i + \sum_{i<j}^{d} V_{ij} + ... \tag{7.2}$$

where V_i is the 1st order contribution of the parameter X_i to the variance of the model amd V_{ij} is the joint contribution of second order of the parameters X_i and X_j.

Using this definition, the Sobol index is defined. The first order Sobol index [37] is

$$S_{1,i} = \frac{V_i}{Var(Y)} \tag{7.3}$$

The total Sobol index, which captures the higher order interactions with other variables can be similarly defined by including all the higher order interactions [37]:

$$S_{T,i} = \frac{V_i + V_{ij} + V_{ijk} + ...}{Var(Y)} \tag{7.4}$$

In order to compute the variance terms to determine the Sobol index of each input, integration is carried out over the d-dimensional space. To optimize this integration, the Sobol sequence is used and the SALib library in Python is used to perform the calculations.

In this way, only the ranges of each input parameter need to be defined. A very large sample of inputs can be collected using the Sobol sequence, which can be given to the trained NN for quick evaluation. The results can be integrated and the Sobol index of each input calculated.

Conveniently, the inputs used when training the model were first normalized as their z-score. Therefore, each parameter is varied from -3 to 3 as this will roughly represent 99% of the observed variation in each parameter. The exception to this rule is the dynamic amplification factors defined as engineered features which were not normalized. In these cases, the range is defined from 0 to 3.5, which also roughly represents 99% of these inputs.

This analysis is performed on all six load emulators to determine the contribution of each input variable on the 9 different load channels. The results are summarized in Figure 7.3. Here, the total Sobol index of each input is shown in a colormap. In this figure, it is apparent that the accelerometer data at the nacelle is critical for emulating the tower and foundation loads. In these load channels, the features involving the standard deviation of the nacelle accelerations are among the most important inputs as measured by the Sobol index.

The sine and cosine of the wind direction, along with the projections of the nacelle acceleration according to the bearing angle (features EF1-EF4) are critical features. This is an expected result as the differences between the local and global frame of reference need to be accounted for by the SM. It is also seen that the relative importance of the cosine and sine of wind or wave angles are dependent on the direction of the load. For loads about the X-axis, the cosine is more important. For loads about the Y-axis, the sine is more important.

It is also visible in Figure 7.3 that the wave properties have a high degree of importance for loads below the water depth but their relevance recedes as the location of the load is closer to the rotor. This is visible in the wave direction, H_s, T_p, α, and the engineered features EF7-EF12 which all involve the projection of wave statistics. This is consistent with expectations, as wave loading will be more critical for foundation loads. The loads at tower top and blade root depend more on the wind loading and therefore have low Sobol indices for most of the wave-related features.

For the engineered features, EF13 and EF15, a moderate importance is found across the majority of load channels. These features are the $f_1^{0.5}$ and f_1^3 respectively. It is therefore seen that these powers of the first mode frequency are critical in predicting the DELs. Especially when comparing with the Sobol indices found for f_1,

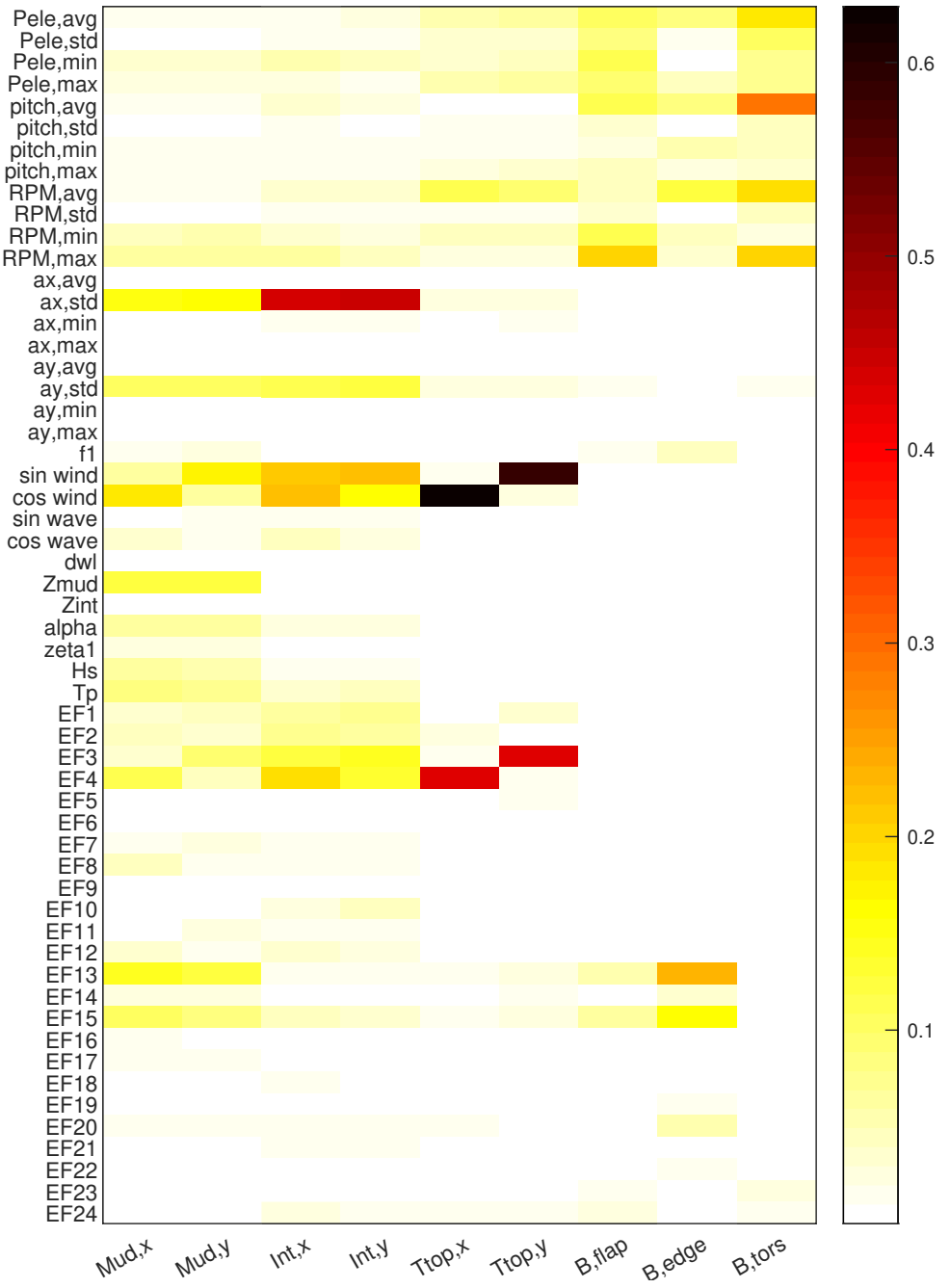


Figure 7.3: Convergence of loss function of training and cross-validation sets for the interface load emulator.

it is seen that the engineered features are more helpful to the model than the basic feature.

The engineered features which attempt to resemble the absolute value of a transfer function (EF18-EF24) have a relatively low importance across the different load channels, however their contribution is not negligible.

SCADA-like metrics encoded in the first 12 basic features are shown to be fundamental for blade load emulation, as the variance in the load emulator is almost entirely described by these features. Their influence on tower and foundation loads is less dominant but still present.

Lastly, some features can be identified to have negligible influence on the load emulators. For example, the average, minimum and maximum nacelle accelerations in both directions, as well as the ranges of the accelerations (EF5 and EF6). This analysis provides a quantitative justification for trimming these features from the model in future iterations. Similar pruning could also be performed for each emulator individually, such as eliminating the wave related inputs for the blade load emulator.

Chapter 8

Conclusions

8.1 Conclusions

The work presented here-in has successfully demonstrated the potential use of load emulators in a type-generic process for prediction of fatigue loads of any potential site. Load emulators were developed using simplified models representative of multiple generic sites. These were then tested using data from simulations of two site-specific designs in the North Sea to test the surrogate model's ability to generalize the load emulation to different sites.

In chapter 3 it was shown that the simplified models are sensitive to changes in parameters such as the diameter at water level and the pile penetration depth. These parameters have a strong effect on the dynamic behavior of the model as well as the hydrodynamic loading that would correspond to this model. It was also shown that the act of reducing the number of degrees of freedom inherently introduces differences in the dynamics of the model, as the act of simplifying changes the overall model. Nevertheless, a strategy for performing a calibration of the simplified model was developed in this chapter, such that a simplified model could be built with the same dynamics of a detailed model by affecting the stiffness, damping, and pile-penetration depth. Finally, this chapter showed a methodology for selecting appropriate hydrodynamic hyperparameters which control the generation of wave load files to ensure an accurate representation wave loading on the simplified models.

The methodology for creating simplified models was tested in chapter 4, where the fatigue load calculation was performed on a simplified model and a corresponding detailed model. The simplified model, when appropriately tuned using the techniques described in chapter 3, proved to accurately match the fatigue loading of the detailed model within 2% RMSE for DLC 1.2 and 5% RMSE for DLC 7.2. This shows that the methodology successfully creates simplified models capable of accurately representing potential site-specific designs.

In chapter 5, the strategies used to select the DoFs of each simulation that would be included in the training database were presented. Here two strategies were analyzed, one in which the training data is obtained from a purely random process

(Monte Carlo), and one in which low-discrepancy sequences are used to deterministically select samples which are evenly spaced across the domain (Quasi-Monte Carlo). The effects on the resulting load emulator are compared afterwards in chapter 7.

To train the load emulator chapter 6 showed a method for selecting suitable hyperparameters which describe the optimizer and network architecture using random sampling. A total of 6 load emulators for the 9 load channels were trained, and subsequently tested on data from simulations of two detailed models of separate site-specific designs. It was shown that surrogate model trained on generic data performed well when testing on site-specific cases as measured by several error metrics such as MAPE, MPE, RMSE, R^2, and the total DEL error. Across all channels the coefficient of determination obtained was above 0.95, the MAPE below 7%, and bias (MPE and total DEL error) was generally found in the order of 1-5%.

In chapter 7, the final research questions were answered successfully via a series of analyses on the trained NNs. It was shown that the sampling strategy of using low-discrepancy sequences such as the Sobol sequence results in improved performance overall as measured by several error metrics when compared to using a simple Monte Carlo approach. This strategy was shown to have the additional advantage in limiting the variance of the trained model. These results show that a quasi-Monte Carlo sampling method should be the preferred strategy for constructing a training database for fatigue load emulation. Finally, it was also shown in this chapter that the features related to nacelle accelerations, wind direction, rotor speed, and tower frequency were the most important overall. For mudline and interface loads, features related to wave statistics such as H_s, T_p, and the wave direction proved to be critical for load emulation at these locations.

8.2 Future Work

Although this work successfully answered the research questions initially proposed, there are still many aspects of the load emulation methodology that are yet to be explored. In this section, the research questions which emerge from this thesis and can be used to guide future research are listed.

- How do the load emulators presented perform on measurement data? This thesis focused on using simulation data for training, cross-validation, and testing, however the load emulator should later be tested on real measurement data to fully evaluate it's capabilities. This brings new challenges of data management, as measurement data is more noisy than simulation data and it's possible that not all of the input channels used in this thesis are available when using measurements.

- Is it possible to relate the degrees of freedom that define the structural model? Throughout the thesis, the degrees of freedom that define the dynamics and geometry of the structural model were considered independent. However, it is recognized that the two are intrinsically coupled. It may be beneficial for the

training of the load emulator to derive a relationship between the geometric degrees of freedom and the dynamics which may result in more realistic structural models. This could potentially be done using physics-informed criteria or through machine learning techniques using data of existing turbines.

- How do other machine learning algorithms compare to neural networks for load emulation? The thesis presented only the use of neural networks for load emulation, however there is a great variety of machine learning algorithms that can be applied for regression modeling such as decision trees and gradient boosting. It is not known if these algorithms could perform better than the neural networks.

- Can a similar methodology be derived to estimate design loads as an initial loads assessment? The methodology for creating simplified structural models and creating a training database could potentially be applied for estimating design loads of new projects. In this case, instead of using SCADA-like measurements, wind and wave properties would be used as inputs to estimate fatigue loads.

Bibliography

[1] IEA. *Net Zero by 2050: A Roadmap for the Global Energy Sector*. 2021.

[2] GWEC. *Global Offshore Wind Report 2022*. 2022.

[3] Kristof Maes et al. “Continuous strain prediction for fatigue assessment of an offshore wind turbine using Kalman filtering techniques”. In: *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings*. IEEE. 2015, pp. 44–49.

[4] Emmanuel Branlard, Dylan Giardina, and Cameron S. D. Brown. “Augmented Kalman filter with a reduced mechanical model to estimate tower loads on a land-based wind turbine: A step towards digital-Twin simulations”. In: *Wind Energy Science* 5 (3 2020). ISSN: 23667451. DOI: 10.5194/wes-5-1155-2020.

[5] Javad Baqersad, Christopher Niezrecki, and Peter Avitabile. “Extracting full-field dynamic strain on a wind turbine rotor subjected to arbitrary excitations using 3D point tracking and a modal expansion technique”. In: *Journal of Sound and Vibration* 352 (2015). ISSN: 10958568. DOI: 10.1016/j.jsv.2015.04.026.

[6] Alexandros Iliopoulos et al. “Fatigue assessment of offshore wind turbines on monopile foundations using multi-band modal expansion”. In: *Wind Energy* 20 (8 2017). ISSN: 10991824. DOI: 10.1002/we.2104.

[7] N. Noppe et al. “Full load estimation of an offshore wind turbine based on SCADA and accelerometer data”. In: vol. 753. 2016. DOI: 10.1088/1742-6596/753/7/072025.

[8] Alexandros Iliopoulos et al. “A modal decomposition and expansion approach for prediction of dynamic responses on a monopile offshore wind turbine using a limited number of vibration sensors”. In: *Mechanical Systems and Signal Processing* 68 (2016), pp. 84–104.

[9] Nicolai Cosack. “Fatigue load monitoring with standard wind turbine signals”. In: (2010).

[10] TS Obdam, LWMM Rademakers, and H Braam. “Flight leader concept for wind farm load counting: offshore evaluation”. In: *Wind Engineering* 34.1 (2010), pp. 109–121.

[11] U Smolka, D Kaufer, and PW Cheng. “Are sea state measurements required for fatigue load monitoring of offshore wind turbines?” In: *Journal of Physics: Conference Series*. Vol. 555. 1. IOP Publishing. 2014, p. 012095.

[12] Marinos Souliotis. “Machine learning techniques for load monitoring of offshore wind turbines”. Delft University of Technology, Nov. 2017.

[13] Jaehyeun Lee. “Fatigue Assessment of Offshore Wind Turbines Using Measurements of Individual Turbines and Machine Learning Techniques”. Delft University of Technology, Norwegian University of Science, and Technology (NTNU), July 2019.

[14] Anish Venu, Mike Lüdde, and John Omole. “Prediction and Validation of Fatigue loads using Artificial Intelligence on Real World Measurement Data”. In: vol. 1618. 2020. DOI: 10.1088/1742-6596/1618/2/022006.

[15] Francisco de Nolasco Santos et al. “Two-tier model for wind turbine fatigue assessment based on SCADA-dependent neural networks”. In: *10th International Conference on Structural Health Monitoring of Intelligent Infrastructure*. 2021.

[16] Laura Schröder, Nikolay Krasimirov Dimitrov, and John Aasted Sorensen. “Uncertainty propagation and sensitivity analysis of an artificial neural network used as wind turbine load surrogate model”. In: vol. 1618. 2020. DOI: 10.1088/1742-6596/1618/4/042040.

[17] Nikolay Dimitrov. “Surrogate models for parameterized representation of wake-induced loads in wind farms”. In: *Wind Energy* 22 (10 2019). ISSN: 10991824. DOI: 10.1002/we.2362.

[18] Juan Pablo Murcia et al. “Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates”. In: *Renewable Energy* 119 (2018). ISSN: 18790682. DOI: 10.1016/j.renene.2017.07.070.

[19] Nikolay Dimitrov et al. “From wind to loads: wind turbine site-specific load estimation using databases with high-fidelity load simulations”. In: *Wind Energy Science Discussions* (2018). ISSN: 2366-7443. DOI: 10.5194/wes-2018-18.

[20] Laura Schröder et al. “Wind turbine site-specific load estimation using artificial neural networks calibrated by means of high-fidelity load simulations”. In: vol. 1037. 2018. DOI: 10.1088/1742-6596/1037/6/062027.

[21] Pamela J Haley and DONALD Soloway. “Extrapolation limitations of multilayer feedforward neural networks”. In: *[Proceedings 1992] IJCNN international joint conference on neural networks*. Vol. 4. IEEE. 1992, pp. 25–30.

[22] Turan Dirlik. “Application of computers in fatigue analysis”. PhD thesis. University of Warwick, 1985.

[23] Charu C. Aggarwal. *Neural Networks and Deep Learning*. 2018. DOI: 10.1007/978-3-319-94463-0.

[24] Lisa Ziegler et al. “Sensitivity of wave fatigue loads on offshore wind turbines under varying site conditions”. In: *Energy Procedia* 80 (2015), pp. 193–200.

[25] Ana Glisic, Gonçalo T Ferraz, and Peter Schaumann. “Sensitivity analysis of monopiles’ fatigue stresses to site conditions using Monte Carlo simulation”. In: *The 27th International Ocean and Polar Engineering Conference*. OnePetro. 2017.

[26] Ramtin Rezaei, Paul Fromme, and Philippe Duffour. “Fatigue life sensitivity of monopile-supported offshore wind turbines to damping”. In: *Renewable Energy* 123 (2018), pp. 450–459.

[27] MC Holtslag, WAAM Bierbooms, and GJW Van Bussel. “Wind turbine fatigue loads as a function of atmospheric conditions offshore”. In: *Wind Energy* 19.10 (2016), pp. 1917–1932.

[28] Marc Seidel. “Wave induced fatigue loads: Insights from frequency domain calculations”. In: *Stahlbau* 83 (8 2014). ISSN: 14371049. DOI: 10.1002/stab.201410184.

[29] Ingrid B Løken and Amir M Kaynia. “Effect of foundation type and modelling on dynamic response and fatigue of offshore wind turbines”. In: *Wind Energy* 22.12 (2019), pp. 1667–1683.

[30] Clemens Hübler, Cristian Guillermo Gebhardt, and Raimund Rolfes. “Hierarchical four-step global sensitivity analysis of offshore wind turbines based on aeroelastic time domain simulations”. In: *Renewable Energy* 111 (2017), pp. 878–891.

[31] JR Morison, JW Johnson, and SA Schaaf. “The force exerted by surface waves on piles”. In: *Journal of Petroleum Technology* 2.05 (1950), pp. 149–154.

[32] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).

[33] IEC. *Wind energy generation systems – Part 3–1: Design requirements for fixed offshore wind turbines*. en. Standard IEC 61400-3:2019. Geneva, Switzerland: International Electrotechnical Commission, 2019. URL: https://webstore.iec.ch/publication/29360.

[34] Amrit Shankar Verma et al. “Effects of Wind-Wave Misalignment on a Wind Turbine Blade Mating Process: Impact Velocities, Blade Root Damages and Structural Safety Assessment”. In: *Journal of Marine Science and Application* 19.2 (2020), pp. 218–233.

[35] Russel E Caflisch. “Monte carlo and quasi-monte carlo methods”. In: *Acta numerica* 7 (1998), pp. 1–49.

[36] Chris Tofallis. “A better measure of relative prediction accuracy for model selection and model estimation”. In: *Journal of the Operational Research Society* 66.8 (2015), pp. 1352–1362.

[37] I. M. Sobol. “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”. In: *Mathematics and Computers in Simulation* 55 (1-3 2001). ISSN: 03784754. DOI: 10.1016/S0378-4754(00)00270-6.

TU Delft Delft University of Technology

SIEMENS Gamesa RENEWABLE ENERGY