HIPERWIND

HIghly advanced Probabilistic design and Enhanced Reliability methods for high-value, cost-efficient offshore WIND

Title: Validation of the newly developed FLS and ULS distribution predictions and quantification of the resulting uncertainty reduction

Deliverable no: D4.5

Delivery date: 30-09-2024
Lead beneficiary: DTU
Dissemination level: Public

This project has received funding from the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 101006689

Author information (alphabetical):		
Name	Organization	Email
Moritz Gräfe	DTU	mograf@dtu.dk
Nikolay Dimitrov	DTU	nkdi@dtu.dk
Mohamed Reda El Amri	IFPEN	mohamed-reda.el-amri@ifpen.fr
Martin Guiton	IFPEN	martin.guiton@ifpen.fr

Acknowledgements/Contributions::		
Bruno R. Faria	Suguang Dou	

Document information:					
Version	Date	Description	Prepared by	Reviewed by	Approved by
1.0	2024	Official	Authors listed above	A. Kolios	A. Kolios

Definition:	

Contents

List of Figures

List of Tables

List of Abbreviations

CFD	Computational Fluid Dynamics
DEL	Damage Equivalent Load
DOE	Design of Experiment
FLS	Fatigue Limit State
GP	Gaussian Process
LSTM	Long Short-Term Memory
ULS	Ultimate Limit State
LAT	Lowest Astronomical Tide

1 Executive Summary

This report presents the results of Task 4.5 in the Hiperwind project, focusing on validating surrogate models developed for wind turbine load predictions. The primary objective was to improve the accuracy and reduce uncertainties in wind turbine design and operational predictions. Specifically, the study investigated using Long Short-Term Memory (LSTM) and Gaussian Process (GP) models to predict wind turbine loads, comparing models trained on simulated data with those using real-world field measurements.

Key findings of the report include:

- LSTM Models: These models were tested using both simulated and field data to predict fore-aft tower base bending moments. Simulation-based models provided higher prediction accuracy due to the absence of measurement uncertainties inherent in field data. However, models using field data showed improved accuracy with increasing amounts of real-world data, though performance gains plateaued after a certain threshold.
- Gaussian Process Models: GP models effectively predicted the mean bending moment but struggled with accurately predicting maximum moments. The models showed high predictivity for mean bending moment estimation but exhibited limitations in high-stress scenarios.
- Model Updating: The report explored how combining simulated data with field measurements could improve model accuracy. Results showed that transferring models trained on simulations to real-world scenarios is promising. However, the achieved prediction accuracies were not satisfactory, primarily due to differences in data correlation patterns between simulated and field conditions. Retraining models with small amounts of field data improved prediction accuracy in some cases, underlining the potential of combined use simulated and field data for virtual sensing.

Overall, the findings highlight the potential of surrogate modeling techniques, such as LSTM and GP, to reduce uncertainties in wind turbine load predictions. However, challenges remain in integrating simulation and field data. Further refinement of these techniques is required to achieve consistent improvements in predictive accuracy across varied operational conditions.

2 Introduction and problem formulation

Wind turbine loads are a key factor in wind turbine design and operations. Fatigue and extreme load estimations are necessary for design integrity assessment during the wind turbine development stage, and are typically carried out using numerical simulations. And for existing assets, the actual fatigue damage accumulated in the structure over its lifetime is one of the primary criteria that can determine whether life extension is feasible. As the majority of existing assets do not have continuous load measurement history, remaining lifetime estimation will typically require numerical load simulations combined with an assessment of the site-specific environmental conditions and wind turbine operation patterns (Dimitrov and Natarajan, 2019).

Load simulation tools such as Hawc2, Diego or DeeplinesWind$^{\text{TM}}$ are rigorously validated in order to ensure that they can provide adequate design loads assessment. Nevertheless, there are several uncertainties associated with using a model. For a quantity y that we want to predict based on some measured inputs X (e.g., environmental inputs such as mean wind speed), we may build a model $g(X)$ that should map the relationship between X and y. Equation 2.1 outlines the types of uncertainties associated with such a model setup. First, the model inputs may be uncertain - there could be measurement uncertainty, or the quantities may not be fully observable, such as for example the spatial- and time- average of the wind speed over the entire domain which is a typical simulation input but can't be measured in such details. These uncertainties are denoted by ε_X. Another factor, the model uncertainty ε_g, represents the inaccuracies of the mapping function due to e.g. imperfect parameterization or not sufficiently capturing the problem physics. Finally, if we are comparing the model predictions to a measured value $\hat{y}$, this may differ from the true quantity y due to the presence of measurement uncertainty, ε_y

$$y = (\hat{y} + \varepsilon_y) = g(X + \varepsilon_X) + \varepsilon_g \tag{2.1}$$

Despite not being as expensive as for example time-domain CFD simulations, aeroelastic load simulations still require a fairly large amount of computations. It is therefore common to replace the aeroelastic tool by a simpler and more computational efficient map of its behavior, a so-called surrogate model, which is typically a regression model. The surrogate model receives a subset of X as inputs, $H \subset X$, and estimates the output of the full-fidelity model. Denoting the true model output by $\tilde{y} = g(X)$, and the surrogate model as $q(H)$ the surrogate model equation can be defined as

$$\tilde{y} = q(H) + \varepsilon_q \tag{2.2}$$

By approximating the aeroelastic model predictions with a surrogate model, we are introducing an additional uncertainty, ε_q, due to the model error of the surrogate model itself. Therefore, when we predict a real quantity by running a surrogate model, our estimate is subject to both the aeroelastic model uncertainty, and the surrogate model uncertainty.

The model uncertainties ε_g and ε_q are both epistemic in nature and could be reduced with improved state of knowledge, e.g., by improving the model accuracy. One potential way of obtaining a surrogate model with higher performance in predicting real data is to train the model directly on measurements. While this completely eliminates the numerical model uncertainty ε_g, it can potentially lead to higher statistical uncertainty (part of ε_q) due to limitations in the measurements. This can also lead to the model not being able to generalize well, as the finite amount of observations will rarely cover the entire range of variation of the input variables.

The aims of the present report are to:

- Evaluate the performance of surrogate modelling methodologies utilized in the Hiperwind project (as discussed in cite D4.1, D4.2, D4.3) in predicting real-world wind turbine load measurements
- Study model parameter updating as means of reducing model uncertainty
- A secondary aim is to investigate ways for data pre-processing for comparison of time series models considering sensor alignment between field and simulation setups.

3 Data description

The load prediction models in this task are developed based on simulated and measured turbine load data from a field measurement campaign. This section describes the measurement campaign in detail.

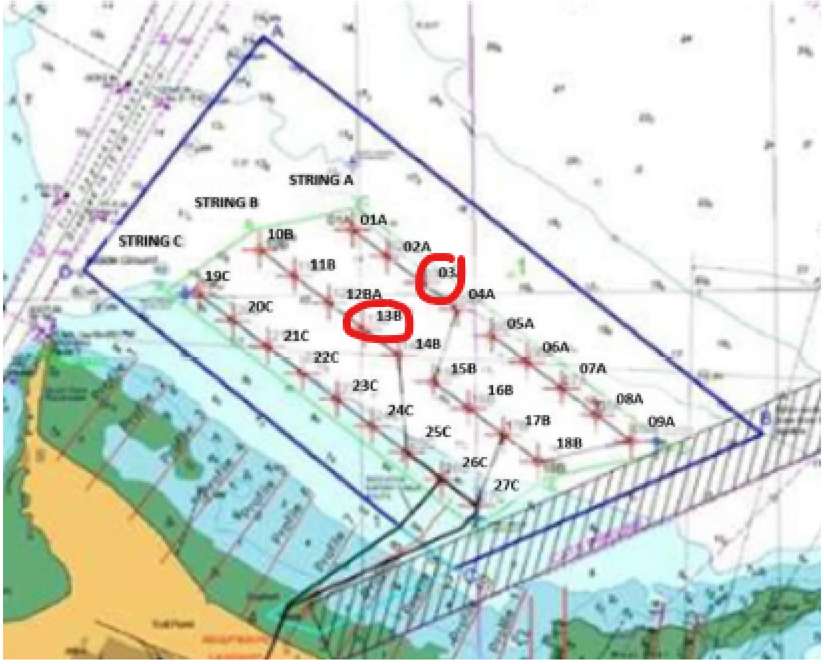

Figure 3.1: Layout of the Teesside wind farm

The Teesside wind farm is an offshore wind farm located in the North Sea, close to the east coast of the UK. It is comprised of 27 Siemens 2.3 MW turbines. Figure 3.1 shows the layout of the Teesside wind farm. The data used in this task originates from Turbine 13 and comprises a total length of one month (November 2020). The available measurements include SCADA and load signals. All the signals used in this task for model training and testing as well as data pre-processing are summarized in table 3.1.

The strain sensors given in table 3.1 are placed around the circumference of the transition piece according to figure 3.2 at height of 4.92 meters above LAT (Lowest Astronomical Tide). The acceleration and inclination sensors are located at the same height in the transition piece while the -X axis points in the north direction and -Y in the east direction.

To provide an overview of the operating conditions present during the measurement campaign, wind speed and wind direction estimations are extracted from the SCADA data using nacelle-based sensors (nacelle wind speed and nacelle orientation respectively). Figure 3.3 shows a wind rose extracted from the SCADA data. It can be seen that the prevailing wind direction is South-West with wind speeds between 0 and 30 m/s occuring during the measurment period. This implies that the A-SG-135 and A-SG-315 sensors measure mostly strain in the side-side direction (assuming that the turbine rotor is always facing the wind), while the T-SG sensor is well aligned with the fore-aft direction.

Figure 3.4 shows the relative distribution of wind speeds and time series plot of the wind speed

Table 3.1: SCADA, load sensors and dynamics measurements T13

Data	Channel	Sampling Freq.	Unit
SCADA	Wind Direction	10 min	$[deg]$
	Wind Speed	10 min	$[m/s]$
Strain Measurements	A-SG-315	20 Hz	microstrain
	A-SG-135	20 Hz	microstrain
	T-SG-30-1	20 Hz	microstrain
	T-SG-210-1	20 Hz	microstrain
	T-SG-60-1	20 Hz	microstrain
	T-SG-240-1	20 Hz	microstrain
Dynamics Measurements	ACC-X-1	20 Hz	$[m/s^2]$
	ACC-X-1	20 Hz	$[m/s^2]$
	INC-X-1	20 Hz	$[deg]$
	INC-Y-1	20 Hz	$[deg]$

throughout the measurement campaign.

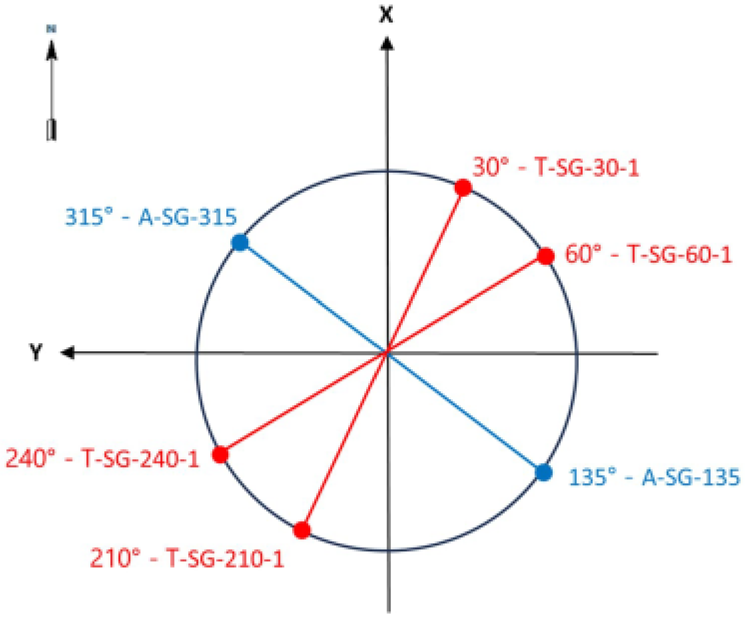

Figure 3.2: Strain measurements sensor placement

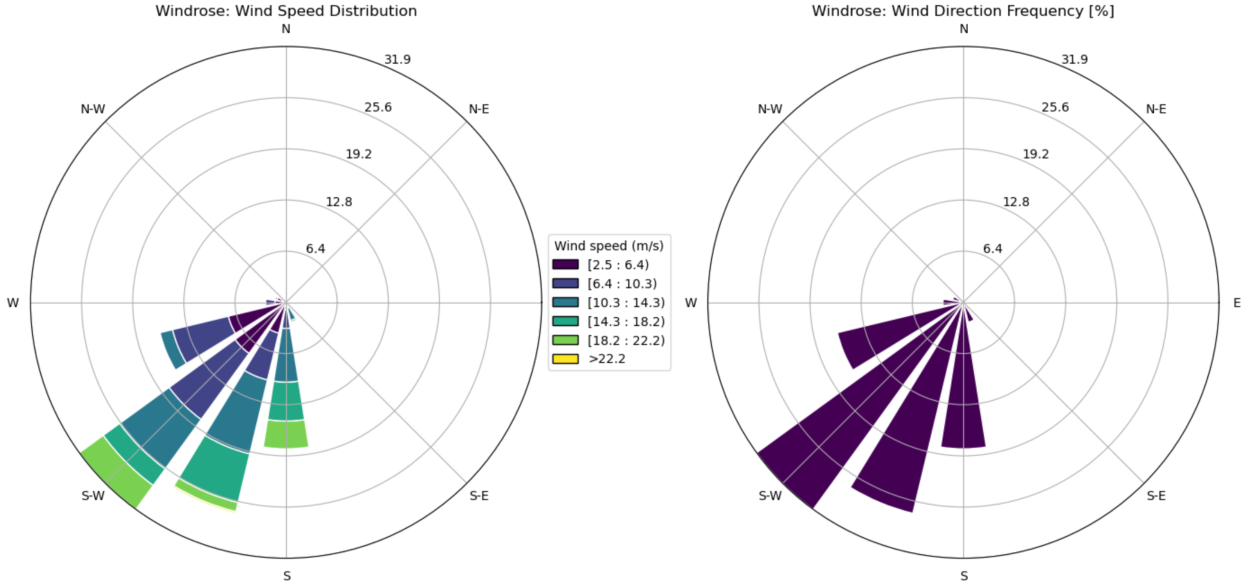

Figure 3.3: Left: Wind rose Teesside, T13. Right: Relative occurrence frequency of wind directions Teesside, T13

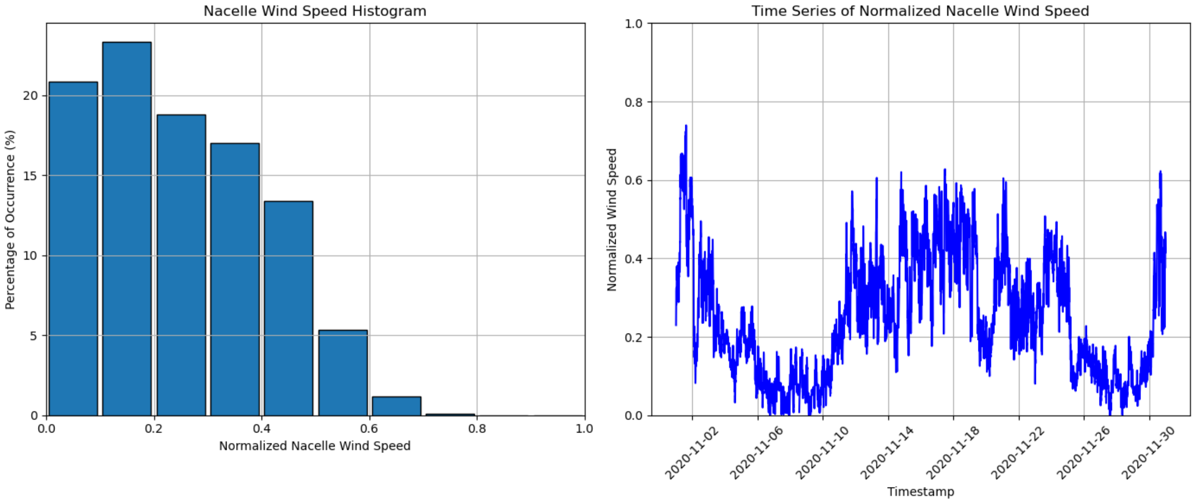

Figure 3.4: Left: Histogram of nacelle wind speeds during measurement campaign. Right: Time series of nacelle wind speed measurement during measurement campaign.

4 Methodology

4.1 Load simulations setup

The surrogate models discussed in this report were originally developed as part of Hiperwind Deliverable 4.1 (Dimitrov et al., 2022). As a result, we are using the same aeroelastic model setup and the same input sampling space for training the simulation-based surrogates. Below, the description of the sampling procedure is repeated from Deliverable 4.1 for completeness.

The IEC 61400-1 design standard provides a reference definition of the scenarios and the environmental conditions to be used for wind turbine design. For onshore sites, the major driving environmental factors are the wind speed and the turbulence, characterized by a joint probability distribution. Wind shear is also considered - defined as an exponential function with a deterministic exponent (either 0.11 or 0.2). When addressing site-specific problems, additional variables or more detailed modelling of a given variable may come into play depending on the specific conditions at the site. As a realistic use case, we choose four environmental variables: wind speed, turbulence, wind shear and air density. The joint distribution of wind speed vs. turbulence is as defined by the IEC61400-1 standard, while the wind shear is taken as conditional on wind speed using the formula defined in (Dimitrov, et al., 2015). Air density is considered independent and following a Gaussian distribution. The exact distribution parameters are outlined below.

- Wind speed u: a truncated Weibull distribution with parameters A=11.28 and k=2, and bounded between the cut-in and cut-out wind speed of the turbine under consideration. The Weibull distribution parameters correspond to IEC wind speed class I (mean wind speed of 10m/s).
- Turbulence σ_u: the standard deviation of wind speed, σ_u, is considered Lognormally distributed, conditional on the wind speed. The mean and standard deviation of the lognormal distribution are defined with the formulas $\mu_{\sigma_u}|u = 0.14(0.75u + 3.8)$ and $\sigma_{\sigma_u} = 1.4 \cdot 0.14$, respectively, which corresponds to IEC turbulence class B.
- Wind shear exponent α: considered Gaussian, with mean and standard deviation conditional on the wind speed: $\mu_\alpha|u = 0.088(ln(u) - 1)$ and $\sigma_\alpha|u = 1/u$, respectively. This is the expression recommended by (Dimitrov, et al., 2015), (Kelly, et al., 2014).
- Air density ρ: independent Gaussian with mean 1.225 and standard deviation of 0.05.

A total of 2,000 random samples are generated from the above distributions. The sample generation is done in a two steps process: 1) a set of uniform, i.i.d. variables ranging from 0 to 1 is drawn from a Halton quasi-random sequence, and 2) the physical values with the correct joint distribution are obtained by a Rosenblatt transformation. The resulting dataset is visualized in Figure 4.1.

Based on the above design of experiment, 2000 load simulations are carried out on a model of the Teesside offshore wind turbine using the Hawc2 aeroelastic simulation tool. All simulations are with 10-minute duration and represent operation under normal (power production) conditions.

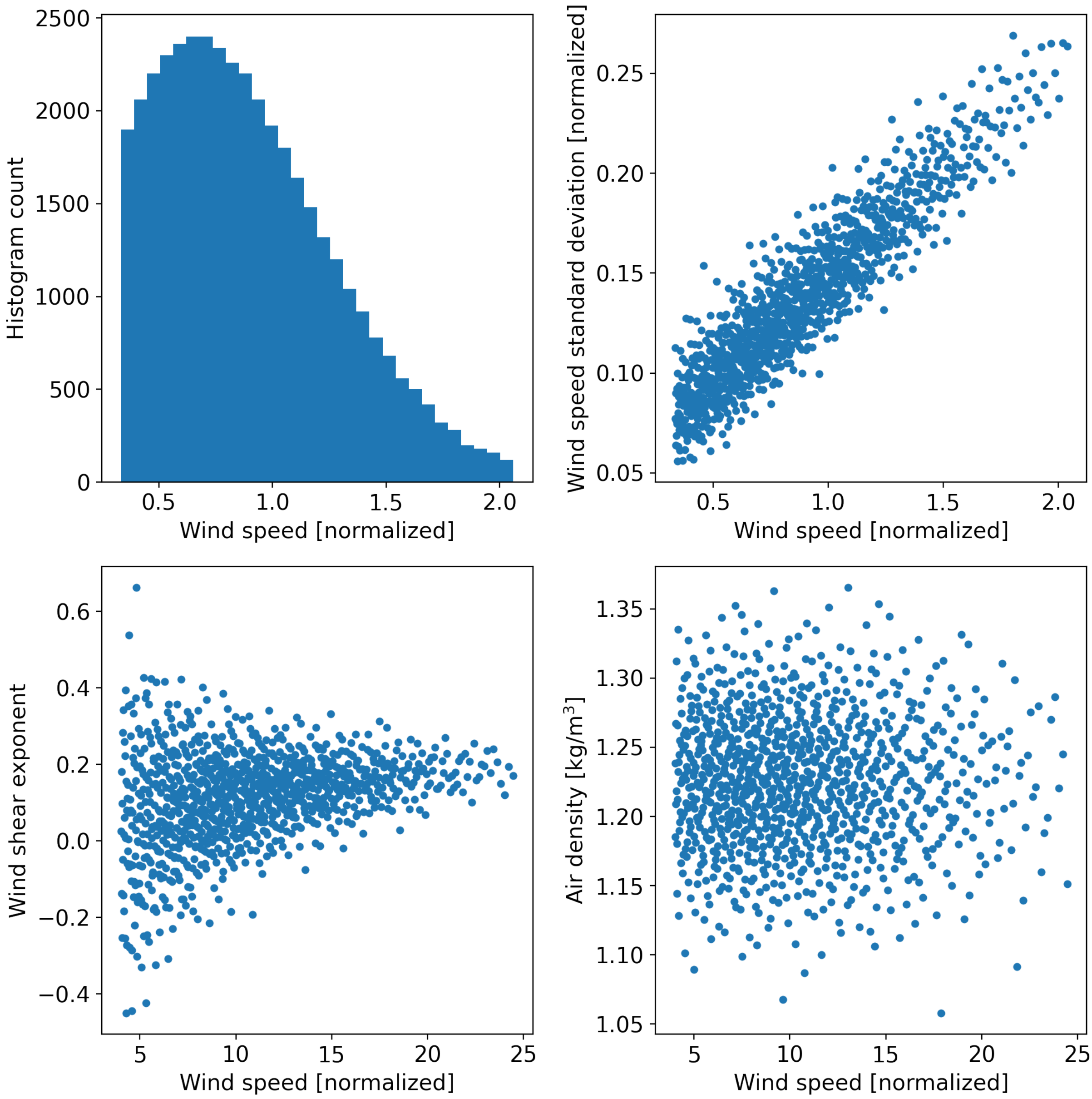

Figure 4.1: Scatter plots of the environmental condition distributions – used as load simulation and surrogate model inputs. Top left: wind speed histogram. Top right: turbulence as function of wind speed. Bottom left: wind shear exponent vs. wind speed. Bottom right: Air density vs. wind speed. Figure taken from Hiperwind D4.1 (Dimitrov et al., 2022)

4.2 Conversion of section strains to loads

In order to complete the primary goals of the study, an additional challenge that needs to be resolved is how to adequately compare the outputs of models trained on simulations and the actual measurements. The typical outputs of aeroelastic models are given in terms of forces and bending moments on cross section level, while the load measurements are obtained in terms of strains. Therefore, the strain measurements from the field measurement campaign are converted into tower bending moments, using information about the tower geometry and material characteristics.

Since the calibration factors of the strain gauge measurements were not known, the offset of each strain pair of strain sensors was found manually. This was performed by filtering the strain measurement for wind aligning with the orientation of the sensors and assuming that the combined bending strain of the pair should be close to zero for wind speeds below 4 m/s. The combined bending strain for for each sensor pair is then shifted by the found offset. Following this correction of strain measurements the strains are converted into bending stresses by:

$$\sigma = \frac{E_{Tower} \cdot \epsilon}{10^6} \tag{4.1}$$

where σ is the resulting stress, E_{tower} is the young's modulus and ϵ is the strain.

The resulting bending stresses are then converted into bending moments by:

$$M = \frac{\sigma \cdot 2 \cdot I_{tower}}{d_{tower}} \tag{4.2}$$

.

I_{Tower} is the tower's moment of inertia and is calculated by:

$$I_{Tower} = \frac{\pi}{64}\left(tower_d^4 - (tower_d - 2 \cdot tower_t)^4\right) \tag{4.3}$$

As the tower bending moments from the aero-elastic simulations are given as fore-aft and side-side bending moments relative to the nacelle's yaw position, the measured bending moments are also converted to fore-aft and side-side bending moments. This is done by projecting the calculated bending moments to a nacelle fixed coordinate system considering the nacelle position information from the available SCADA data. The tower characteristics used for converting strains to loads are summarized in table 4.1. A sketch of the bending moments distributions as function of wind speed is shown in Figure 4.2. As expected, the field moments calculated from the field measurements show higher variability than simulation results. However, the order of magnitude, and main correlation patterns with wind speed are matched.

Table 4.1: Turbine tower characteristics

Parameter	Value
Tower diameter	4.25 [m]
Wall thickness	0.08 [m]
Young's modulus	210e9 [Pa]

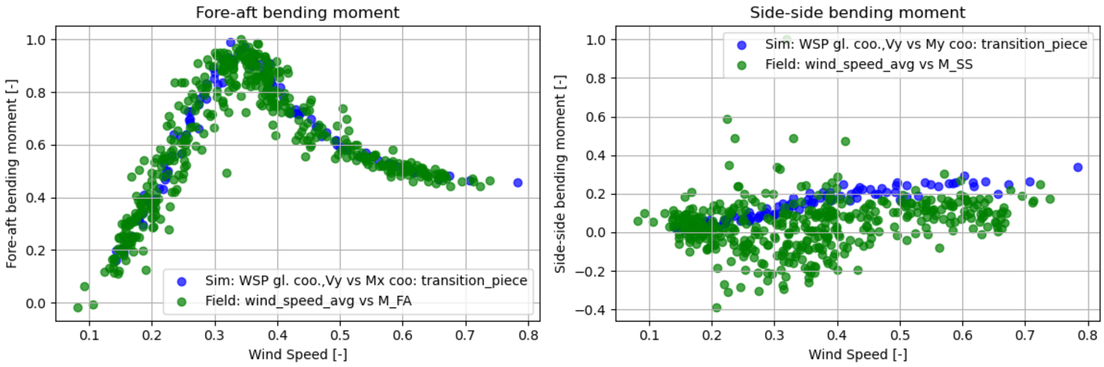

Figure 4.2: Normalized fore-aft and side-side bending moment after sensor offset correction and coordinate transformation.

4.3 LSTM-based time series models

4.3.1 Model definition

LSTM networks represent a specific form of recurrent artificial neural networks, which have remarkably succeeded in capturing temporal dependencies within input data.

What sets LSTMs apart is their use of memory cells, which allow them to retain or forget previous inputs selectively over extended periods. These memory cells store information across multiple time steps, which helps preserve important context over time. Additionally, LSTMs feature a gating mechanism that controls the flow of information between cells, enabling the model to decide which information is relevant, should be kept, and which can be discarded. This selective retention makes LSTMs more effective than traditional neural networks when dealing with time-dependent correlations in data. As this type of correlation can be expected between tower inclinations, accelerations and bending moments, LSTM networks have been selected for this task. The effectiveness of LSTM models for prediction of wind turbine loads and dynamics has been demonstrated e.g. in Gräfe et al. (2024), Gräfe et al. (2024) and Dimitrov and Göçmen (2022).

The LSTM network structure used in this task comprises 6 layers as depicted in figure 4.3. The window size defines the number of timesteps in the input features. Predictions are made for the same number of target time series time steps, while no forecasts into the future are made. The window size used in this task was determined by a hyperparameter tuning process and is set to 50. Considering the data sampling frequency of 10 Hz, this represents a time window of 5 seconds. For visualization, the predicted windows are concatenated to 600 second time windows. Predictions are made in succession while remembering the LSTM cell states for each sub-sequence. Details about the used model hyperparameters are given in table 4.2.

The input and target features used for the simulation and field data based models are summarized in 4.3. The acceleration and inclination signals are used as inputs in all three modeling cases, while predictions are made for the fore-aft bending moment of the tower. The three cases are distinguished by their use of input data sets. The "Field" model uses only data from the measurement campaign for training and testing, while the "Sim" model uses only data from the aeroelastic simulations for training and testing. In contrast, the "update" model uses both the field and the simulation data for model training.

Table 4.2: Model Hyperparameters.

Hyperparameter	Value
Number of LSTM layers	1
Number of LSTM units per layer	100
Window length	50
Number of units in fully connected layer	100
Dropout rate	0.1

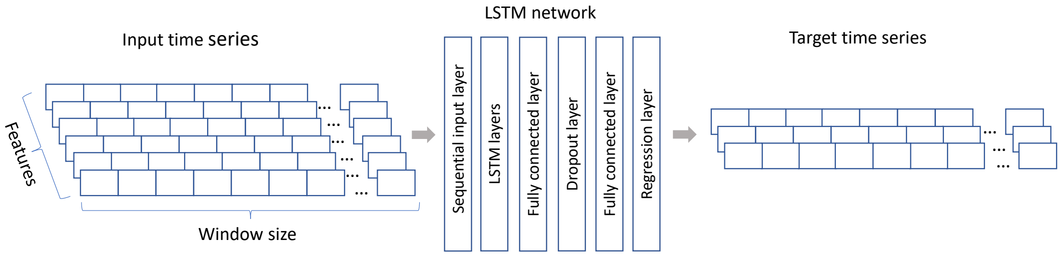

Figure 4.3: LSTM model architecture Gräfe et al. (2024)

Table 4.3: Summary of modeling cases for simulation and field data based models.

Feature	SimA	SimB	SimC
state acc x	✓	x	✓
state acc y	✓	x	✓
eulerang x	✓	✓	x
eulerang y	✓	✓	x
Feature	**FieldA**	**FieldB**	**FieldC**
ACC-X	✓	x	✓
ACC-Y	✓	x	✓
INC-X	✓	✓	x
INC-Y	✓	✓	x

Different cases corresponding to the individual modeling cases are evaluated in the model updating step. First, the transferability of the simulation-based models is tested. Here, the simulation-based models are tested on field input and targets. For the model update, the simulation-based models are first retrained on field data following the testing on field data. The investigated cases are summarized in table 4.4

The time series prediction models in this task are evaluated using the RMSEN (Root Mean Squared Error Normalized) metric. The RMSEN represents the root mean squared error between the predicted time series Y_{pred} and the actual target time series Y, normalized by the standard

Table 4.4: Summary of modeling cases for model updating.

Feature	TransferA	TransferC	TransferC
ACC-X	✓	x	✓
ACC-Y	✓	x	✓
INC-X	✓	✓	x
INC-Y	✓	✓	x
Feature	UpdateA	UpdateB	UpdateC
ACC-X	✓	x	✓
ACC-Y	✓	x	✓
INC-X	✓	✓	x
INC-Y	✓	✓	x

deviation σ_y of the reference time series. The formula for RMSEN is given in equation 4.4.

$$RMSEN = \frac{1}{\sigma_y}\sqrt{\frac{\sum_{i=1}^{N}(Y_{pred} - Y)^2}{N}} \tag{4.4}$$

In this equation, Y_{pred} refers to the predicted values, Y represents the actual reference values, and N indicates the total number of data points in the sequence.

4.3.2 Data preparation and analysis

This task's training testing and validation data originates from field measurements (SCADA and strain sensors) and simulations. The task's scope is to train prediction models on both data sets and eventually use field measurements to update models trained on the simulation data with field measurements. Both data sets have been downsampled to a sampling rate of 10 Hz to enable this combination of data sets. The field measurement data set has been divided into 10 batches of data while the influence of the used amount of training data on the prediction accuracy is investigated and presented in the results.

Figure 4.4 compares simulated and measured acceleration signals. Similar to the bending moments, the acceleration signals from the measurements have been projected to the fore-aft and side-side directions, considering the yaw position of the nacelle. This step is necessary to make signals comparable between simulation and field measurements as the x- and y- accelerations in the simulation data always represent the fore-aft and side-side direction respectively. In the field data the fore-aft and side-side directions changes depending on the orientation of the nacelle. Additionally the units of acceleration signals have been converted from g to m/s^2. While both signals show the same order of magnitude and zero mean characteristics, it can be seen that the simulation data shows a lower standard deviation. The power spectral density plot shows that both data sets show a similar frequency content. The comparison of the y- y-acceleration signal is not presented here but shows similar behavior.

Figure 4.5 compares inclination signals between simulation and field measurements. As for the acceleration signals, the inclination signals of the from the field measurements have been

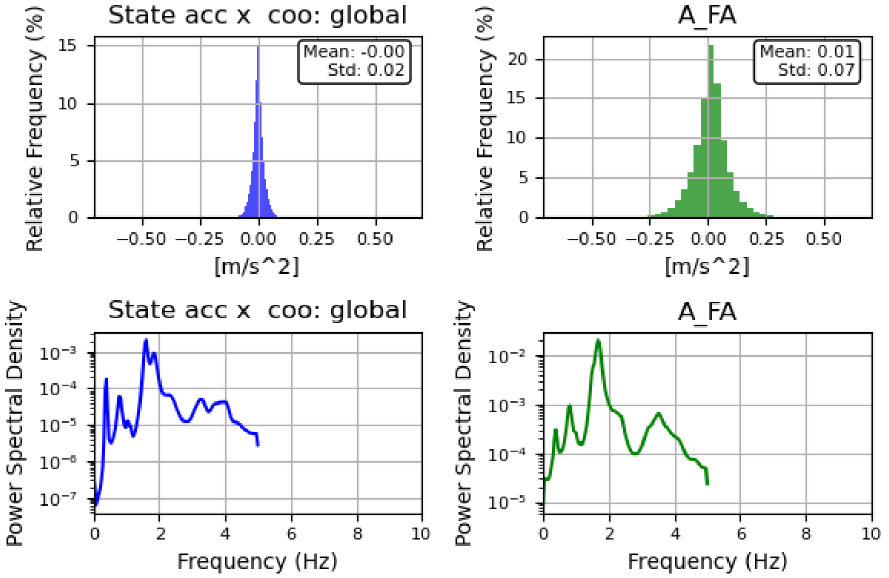

Figure 4.4: Comparison of acceleration signals. Blue: simulation data. Green: Field data

converted to fore-aft and side-side directions. The simulation heavily underestimates the variation of the inclination and does not show similarity in scale to the field data. This is also true for the y-acceleration signals. However simulated and measured signal show similar trends relative to the ambient (nacelle) wind speed (see figure 4.6). While measured and simulated values are on a different scale, both show a similar relationship to the wind speed and the related rotor thrust force, indicating that both signals can serve as input features for time series model after standardization.

Figure 4.7 shows the comparison for the fore-aft bending moment, which is used as the target feature for the time series predictions in this task. It can be seen, that simulation and field data show very similar characteristics in distribution and frequency content. Figure 4.8 shows equivalent findings for the side-side bending moments.

In conclusion, the comparison of simulated and field data shows that the acceleration, inclination and bending moment signals show similar characteristics, which might enable a transfer of models from simulation to field data after standardization.

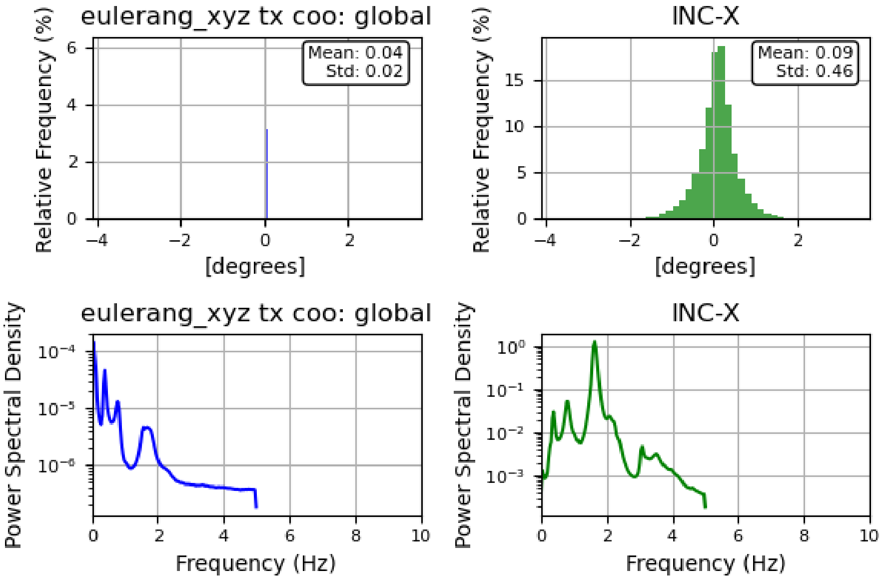

Figure 4.5: Comparison of inclination signals. Blue: simulation data. Green: Field data

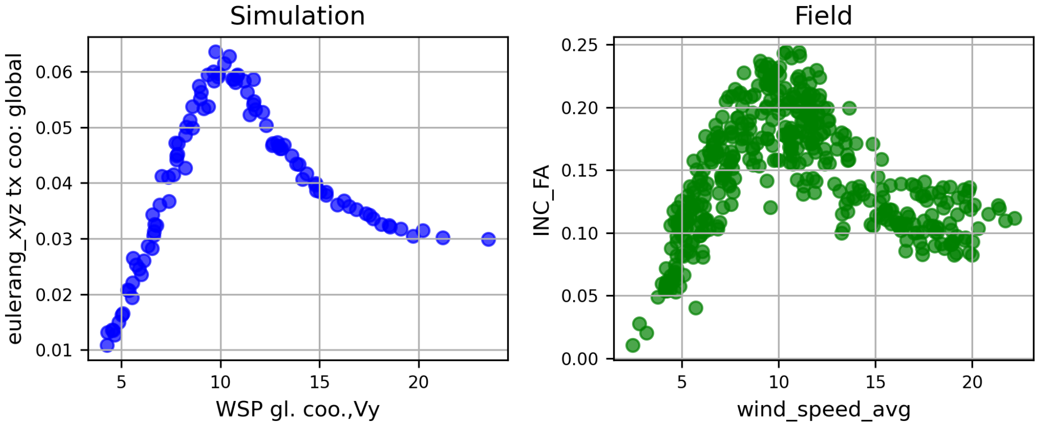

Figure 4.6: Fore-aft inclination of simulation (left) and field (right) relative to wind speed.

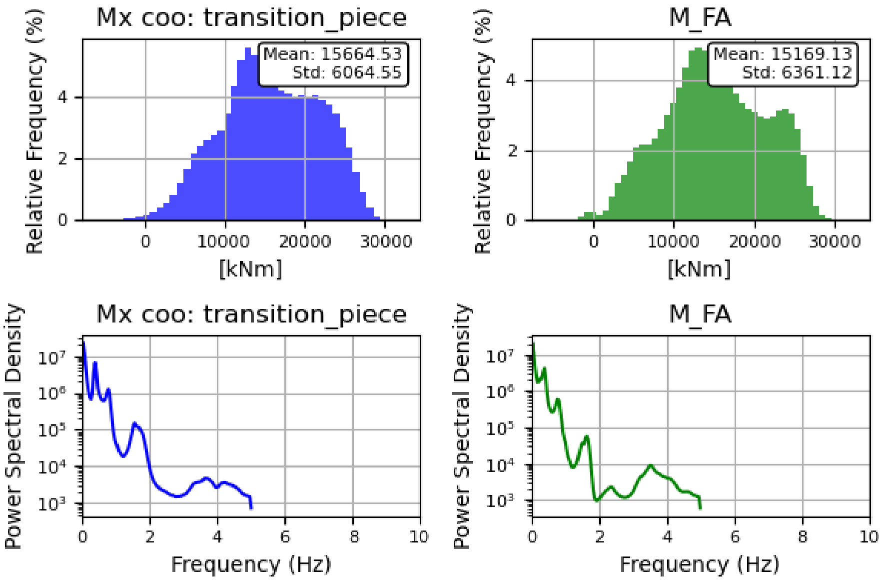

Figure 4.7: Comparison of fore-aft bending moment. Blue: simulation data. Green: Field data

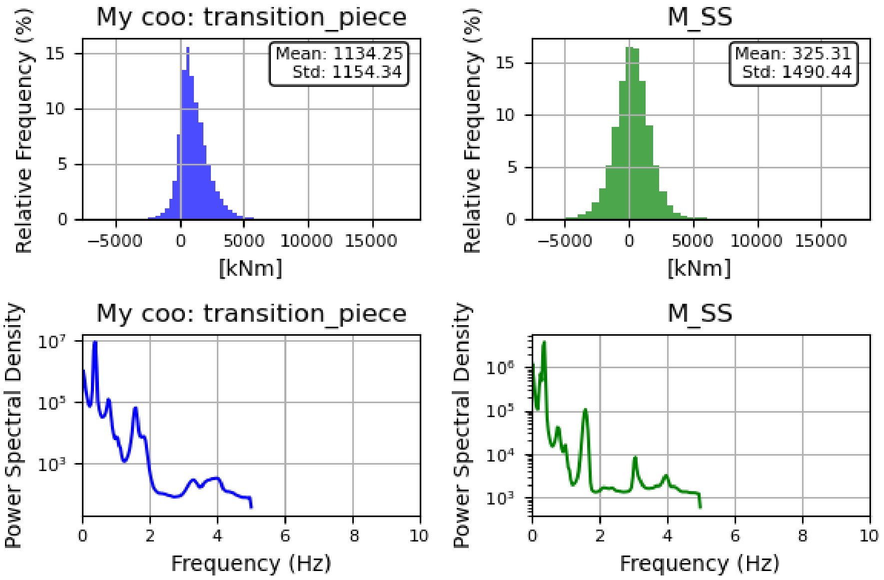

Figure 4.8: Comparison of side-side bending moment. Blue: simulation data. Green: Field data

4.4 Gaussian process modeling with functional inputs

4.4.1 Problem formulation

We model the virtual sensor by a function $f : \mathcal{V} \to \mathbb{R}$ where $\mathcal{V}$ is the functional space in which the random process $\mathbf{V}$ takes its values.

4.4.2 Random process finite dimensional representation

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. We assume that the random process $\mathbf{V}$ belongs to $\mathcal{H} = \mathbb{L}^2(\Omega, \mathcal{F}, \mathbb{P}; \mathcal{V})$ with

$$\mathcal{V} = \left\{ \mathbf{V} : [0, T] \to \mathbb{R}, ||\mathbf{V}|| = (< v, v >)^{1/2} = \left(\int_0^T \mathbf{V}(t)^2 \mathrm{d}t \right)^{1/2} < +\infty \right\}.$$

We assume that $\mathbf{V} \in \mathcal{H}$ has zero mean and continuous covariance function $C(t, s)$. Then

$$\forall t \in [0, T], \ \mathbf{V}(t) = \sum_{i=1}^{\infty} U_i \psi_i(t), \tag{4.5}$$

where $\{\psi_i\}_{i=1}^{\infty}$ is an orthonormal basis of eigenfunctions of the integral operator corresponding to C such that:

$$\lambda_i \psi_i(t) = \int_0^T C(t, s) \psi_i(s) \mathrm{ds}, \tag{4.6}$$

and with $\{U_i\}_{i=1}^{\infty}$ denoting a set of uncorrelated random variables with zero mean and variance λ_i. Decomposition (4.5) is known as the Karhunen-Loève (KL) expansion of $\mathbf{V}$ (Le Maître and Knio (2010)). In the following we denote the truncated version of $\mathbf{V}$ as $\mathbf{V}_m$:

$$\forall t \in [0, T], \ \mathbf{V}_m(t) = \sum_{i=1}^{m} U_i \psi_i(t), \tag{4.7}$$

which represents, in the mean square error sense, the optimal m-term approximation of $\mathbf{V}$ (Le Maître and Knio (2010)). The value of the parameter m should be chosen such that the approximation is accurate enough.

Computational details of functional PCA In practice, the covariance structure of the process $\mathbf{V}$ might be unknown and has to be estimated from the data. More precisely, $C(s, t)$ can be estimated from the sample $\{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}\}$ by:

$$\hat{C}(t, s) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{v}^{(i)}(t) \mathbf{v}^{(i)}(s). \tag{4.8}$$

The eigenvalue problem defined by Eq. (4.6) is then solved by replacing C by $\hat{C}$ (see, Cardot et al. (1999) for convergence results). That approximated eigenvalue problem can be solved by discretizing the trajectories at several discrete time points $t_1, \dots, t_{N_T}$. This yields the matrix of n discretized trajectories $\mathrm{V} \in \mathcal{M}_{n,N_T}(\mathbb{R})$ wih $\mathrm{v}_{i,j} = [v^{(i)}(t_j)]$ for all $i = 1, \dots, n$ and $j = 1, \dots, N_T$. It leads to the empirical covariance matrix defined as $\hat{C}_n = \frac{1}{n} \mathrm{V}^\intercal \mathrm{V} \in \mathcal{M}_{N_T,N_T}(\mathbb{R})$. We then have to solve a classical multivariate PCA with N_T variables given by a sample of size n. Other approaches to implement functional PCA can be found in the literature. In (Ramsay (2006)), e.g., the authors propose to expand the curves as linear combinations of spline basis functions, and to apply PCA to the coefficients of the curves on the spline basis. There also exist different criteria for the choice of the truncation argument m (Jackson (1993)). One can cite the Kaiser-Guttman criterion which consists of choosing the first components with eigenvalues higher than 1. Instead of using the variance provided by λ, the choice of m could be based on the "percentage of variance" given by the ratio $\lambda_m / \sum_{i=1}^{n} \lambda_i$. By this way, we choose q so that the cumulated ratio of variance explained by the first components exceeds a given threshold. Alternatively, to avoid the arbitrary choice of the threshold, the eigenvalues are often displayed in a downward curve and m is chosen according to the stagnation of the slope.

4.4.3 Gaussian Process modeling

In the following, we model f as a realization of a Gaussian Process $F_{\mathbf{u}}$ defined on $\mathbb{R}^m$ (Rasmussen and Williams (2006)), where $\mathbf{u} = (<\mathbf{v}, \psi_1> \cdots <\mathbf{v}, \psi_m>)^\intercal$. Let m_F be the mean function of $F_{\mathbf{u}}$ and k_F its covariance function,

$$\begin{aligned} \mathbb{E}[F_{\mathbf{u}}] &= m_F(\mathbf{u}), \\ \mathrm{Cov}(F_{\mathbf{u}}, F_{\mathbf{u}'}) &= k_F(\mathbf{u}, \mathbf{u}'). \end{aligned} \tag{4.9}$$

Let us denote $F^{(n)}$, the Gaussian process F conditioned on the set of n observations $\mathbf{F}_n = \{f(\mathbf{v}^{(1)}), \ldots, f(\mathbf{v}^{(n)})\}$ of F at $\mathcal{U}_n = \{\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(n)}\}$ where $\mathbf{u}^{(\bullet)} = (<\mathbf{v}^{(\bullet)}, \psi_1> \cdots <\mathbf{v}^{(\bullet)}, \psi_m>)^\intercal$ is the projection of the observation $\mathbf{v}^{(\bullet)}$ on the m-dimensional truncated space given by Karhunen-Loève expansion (Eq. (4.5))

$$F_{\mathbf{u}}^{(n)} = [F_{\mathbf{u}} | F_{\mathcal{U}_n} = \mathbf{F}_n]. \tag{4.10}$$

The mean and covariance of F^n are given by

$$\begin{aligned} \mathbb{E}[F_{\mathbf{u}}^{(n)}] &= m_F(\mathbf{u}) + k_F(\mathbf{u}, \mathcal{U}_n)\Sigma_{F,n}^{-1}(\mathbf{F} - m_F(\mathcal{U}_n)), \\ \mathrm{Cov}(F_{\mathbf{u}}^{(n)}, F_{\mathbf{u}'}^{(n)}) &= k_F(\mathbf{u}, \mathbf{u}') - k_F(\mathbf{u}, \mathcal{U}_n)\Sigma_{F,n}^{-1} k_F(\mathcal{U}_n, \mathbf{u}'). \end{aligned}$$

where $\Sigma_{F,n} = k_F(\mathcal{U}_n, \mathcal{U}_n)$. The next section will cover the preparation steps for sensor-derived time series data.

4.4.4 Data preparation

Time series Preparing time series data involves two essential steps. First, the data is downsampled from 20Hz to 1Hz, reducing the number of data points while preserving critical information. This step is important for making the dataset more manageable and matching the desired temporal resolution. Second, the time series is denoised using a Kalman filter (KF) method, an unsupervised algorithm that tracks a single object in continuous state space. By processing noisy measurements, the Kalman filter can estimate the "true state" of the object being tracked. Its strength lies in effectively handling noisy data, providing a clearer signal. However, the algorithm has some drawbacks. Its computational complexity increases cubically with the size of the state space, reducing efficiency for larger systems. Additionally, parameter optimization using the Expectation-Maximization (EM) algorithm is a non-convex problem, often leading to convergence at local optima rather than the global optimum. To address this issue, we use multistart, a technique employed to improve the chances of finding a global optimum solution. This approach involves running an optimization algorithm multiple times, but starting from different initial points. In this work, we used the `KalmanFilter` class and the EM algorithm `KalmanFilter.em` from the python library `pykalman`. Figure 4.9 illustrates the result of KF-based denoising applied to a noisy signal.

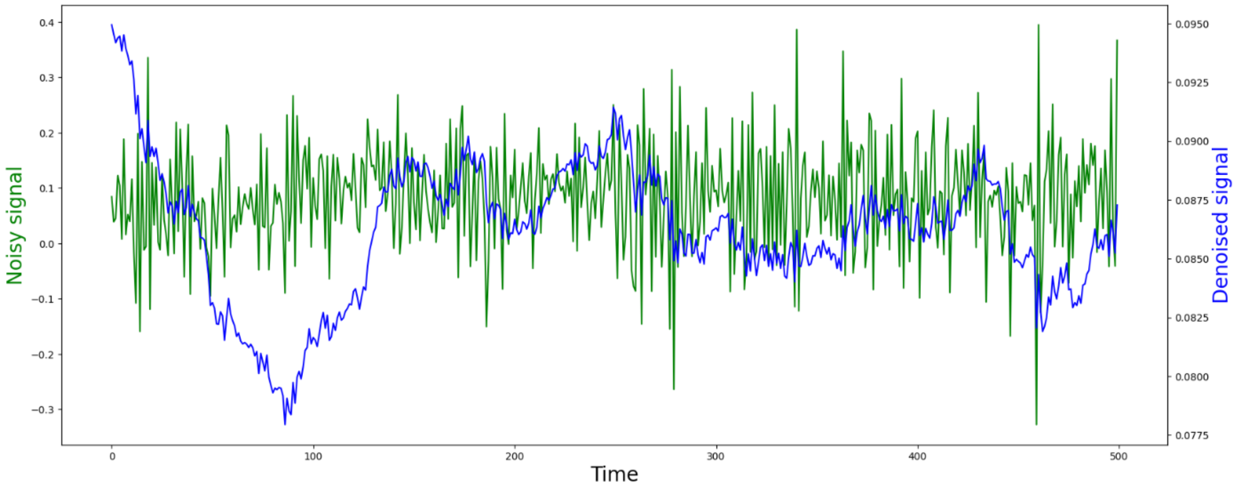

Figure 4.9: An example of Kalman-filter denoising signal. The noisy signal (green curve) and the KF-based denoised signal (in blue).

Scalar outputs Regarding the outputs of interest, we will focus on the maximum and the 1-hour average of the time series of the bending moments.

4.4.5 Gaussian process model trained on data

This study considers four functional inputs: accelerations and inclinations in the X and Y directions. To build the metamodel, we begin by reducing the dimensionality of the functional inputs through projection into a lower-dimensional space, using the decomposition described in Section 4.4.2. The Gaussian process model will then be constructed based on the coefficients obtained from this decomposition (refer to Section 4.4.3).

i) Defining the truncation parameter for the KL expansion We apply a dimensionality reduction technique and determine the optimal truncation parameter m for each of the four inputs. As indicated in Table 4.5, with $m = 4$, more than 99% of the variance in the X and Y inclinations can be captured. For the accelerations, setting $m = 3$ is sufficient to achieve a high explained variance ratio. Consequently, the functional inputs can be represented by a 14-dimensional vector variable.

ii) Gaussian process models We consider a Gaussian Process prior with constant mean function and Matérn covariance kernel with $\nu = 5/2$. The covariance kernel hyper-parameters are estimated by maximizing the likelihood, and all computations are performed using the Python library `smt`. The dataset is split into training and testing sets in which the 70% of data points are allocated as the training set and the 30% as the testing set. All the target variables are

	$m = 1$	$m = 2$	$m = 3$	$m = 4$
Inclinaison-X	93.9%	96.94%	98.2%	98.83%
Inclinaison-Y	95.69%	97.84%	98.65%	99.04%
Acceleration-X	98.36%	99.18%	99.5%	99.64%
Acceleration-Y	96.54%	98.38%	99.10%	99.44%

Table 4.5: The explained variance according to the truncation argument m for the four different functional inputs.

standardized to have 0 mean and unit variance according to the training set, while the same factors scale the testing set.

5 Results from time series predictions

5.1 Time series surrogate models trained on simulation data and measured time series

This section presents the results from the "Field" and "Sim" time series prediction models. Both models input acceleration and inclination signals from the respective data sets and predict the fore-aft tower bending moment, respectively. Figure 5.1 shows one time series example of a predicted 600-second time series for both models for case A. Additionally, the power spectral density of the predicted and reference values is shown. For the simulated data set, the predicted time series follows the reference signal accurately. For the field data, the model predicts low-frequency variation correctly, while higher-frequency fluctuations in the signal is not captured accurately. The comparison of the power spectral density plots also confirms this. While the simulation model accurately captures frequencies up to 2 Hz, the field model can only capture lower frequencies up to approximately 1Hz.

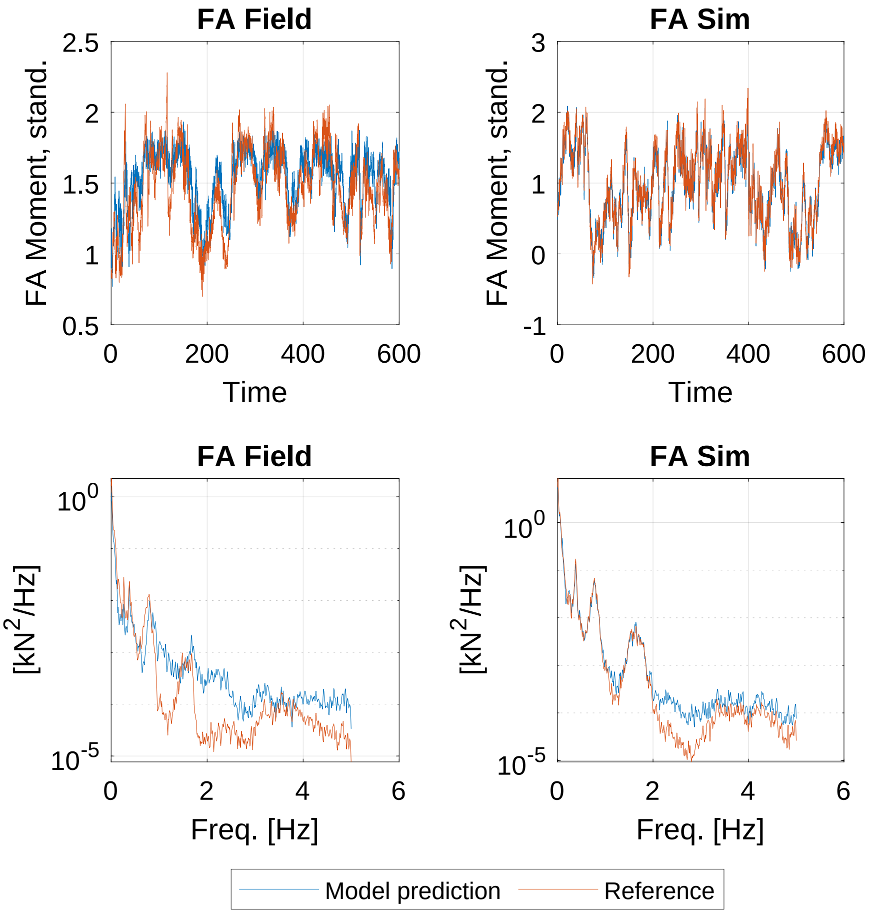

Figure 5.1: Example of time series prediction from field and simulation model, case A

Figure 5.2 summarizes the achieved prediction accuracies for case A for both models. In each boxplot, the box's lower and upper edges correspond to the data's 25th and 75th percentiles, respectively. The median of the samples is shown as a horizontal line inside the box. Data points represented as circles are considered outliers, falling more than 1.5 times the interquartile range beyond the box's boundaries. The whiskers extend from the box to the most extreme data points, still within the defined range, excluding any outliers. The whiskers capture the highest and lowest values that do not meet the criteria for outliers.

Results show that the simulation data-based model achieves better prediction accuracies than the field measurement-based model. This is true in terms of median values as well as the scatter of RMSEN. For the field model, it can be seen that the prediction accuracy is improved for larger training data sets but does not improve further for more than three batches of data used for training the model. Three batches of field data represent circa nine days of operational data.

The higher accuracy of the simulation-based models can be expected, given that we have no measurement uncertainty on the simulations, and that the overall variance in simulated data (see e.g. Figure 4.4) is smaller than in the measured data.

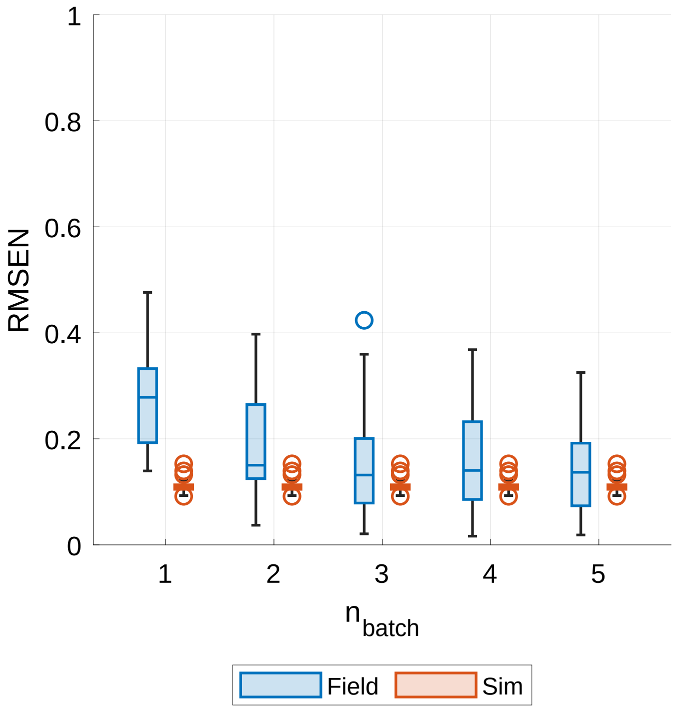

Figure 5.2: RMSEN of time series predictions for sim and field model, case A.

Figure 5.3 presents the RMSEN results for modeling cases B and C. In case B (using inclination as an input), the accuracy is comparable to that of case A, suggesting that the inclination feature offers better descriptive power than the acceleration feature. Conversely, in case C (using acceleration as an input), the prediction error increases notably for both field- and simulation-based models, highlighting the reduced effectiveness of acceleration in capturing relevant system dynamics.

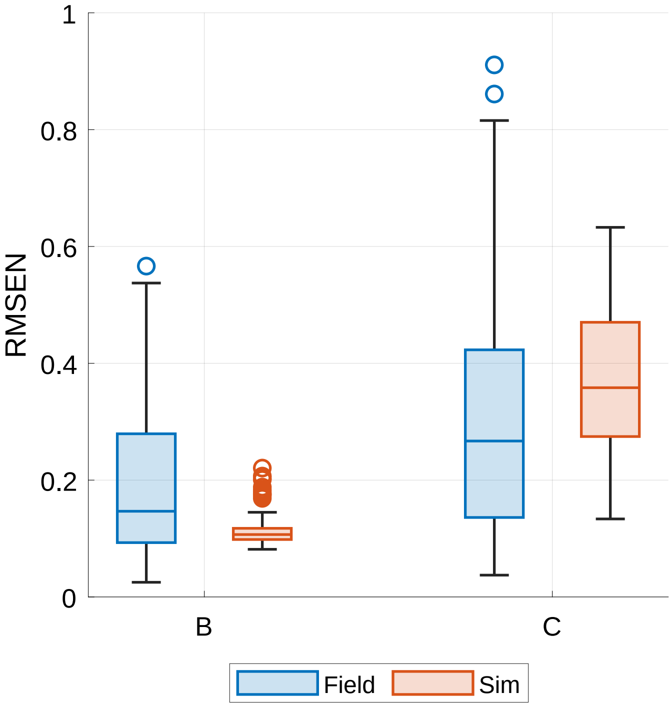

Figure 5.3: RMSEN of time series predictions for cases B and C, $n_{batch} = 3$.

6 Results from predictions of 1-hour statistics

6.1 Evaluation metrics

In the context of evaluating the predictive models, three key metrics are used to assess their performance: the root mean squared error, the relative root mean squared error, and the predictivity coefficient.
Root Mean Square Error (RMSE) quantifies prediction accuracy by averaging the magnitude of errors between predicted and actual values. It squares differences, calculates their mean, then takes the square root, yielding a single value in the original units that represents typical model error

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \tag{6.1}$$

where N is the number of data points, $\hat{y}_i$ is the predicted value and y_i is the true value.
The relative root mean squared error (RRMSE) is a variation of RMSE that expresses the error as a relative value. RRMSE is calculated by dividing RMSE by the mean of the true values and then multiplying by 100% to express it as a percentage

$$RRMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N} y_i^2}}, \tag{6.2}$$

A classical measure for assessing the predictive ability of a model is the predictivity coefficient Q^2. It is usually calculated by the following formula

$$Q^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{6.3}$$

where $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i^2$. Note that the predictivity coefficient Q^2 compares the prediction errors obtained with the model with those obtained when prediction equals the empirical mean of the observations. Thus, the closer Q^2 to one, the more accurate the model is. The coefficient close to zero indicates poor prediction abilities, as there is a little improvement compared to prediction by a naive sample-mean.

6.2 Results based on GP regression

This section presents the results of the proposed strategy based on Gaussian Process regression. As mentioned in the previous section, independent models were built and trained for 2 target variables (mean and maximum bending moments). The evaluation metrics are shown in Table 6.1. We note the very good match between the GP model predictions and true mean bending moment. We would also like to note somewhat lower performance in terms of the maximum bending moment.

Figure 6.1 shows the mean and maximum bending moment probability distributions, while Figures 6.2 and 6.3 show the empirical exceedance probability calculated from the observations and the GP models. Due to the probabilistic nature of GPs, we obtain a confidence interval for the exceedance probability. Regarding the mean bending moment, we obtain very good results. For the maximum, in certain areas, we underestimate. However, the true values remain within the confidence interval.

Evaluation metrics	Mean bending moment		Max bending moment	
	Training	Testing	Training	Testing
RMSE (Unit)	0.006	0.028	0.07	0.11
RRMSE (%)	0.59	6.04	6.31	21.93
Q^2	-	0.9963	-	0.9431

Table 6.1: GP results for the prediction of the mean and the maximum bending moment.

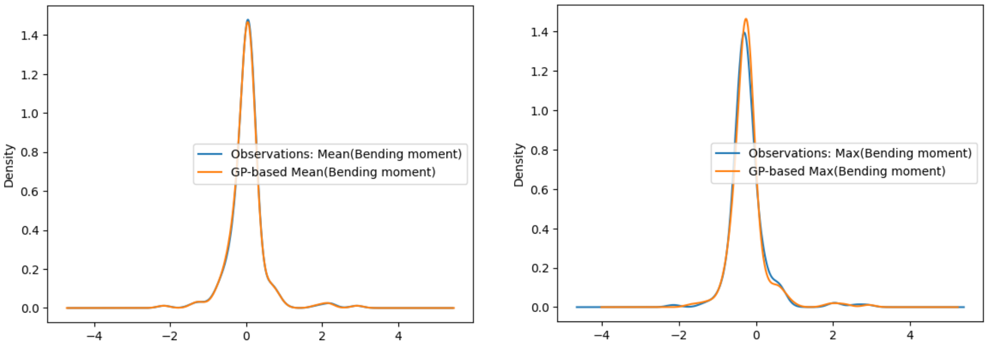

Figure 6.1: The probability densities of the mean and maximum bending moments from data and GP models.

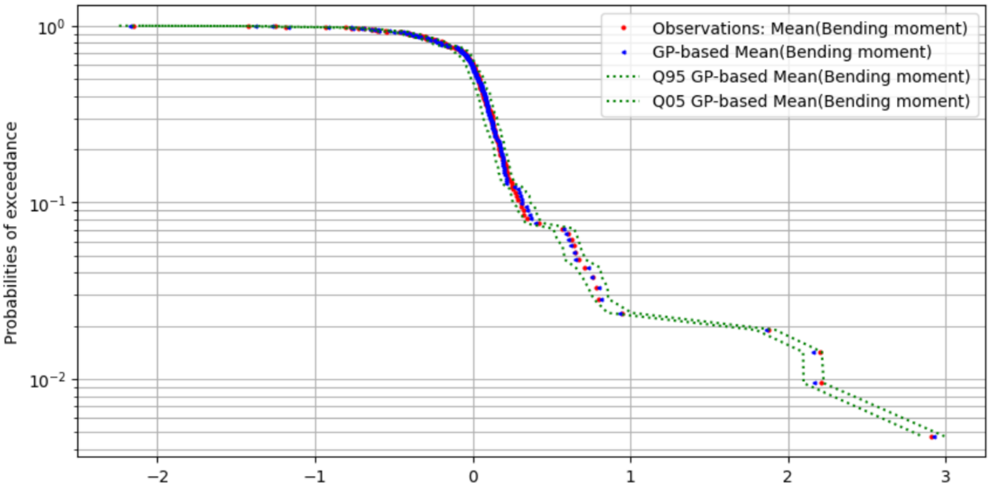

Figure 6.2: Mean of the bending moment: Probabilities of exceedance based on the observations (in red), the Gaussian process model (in blue) and the confidence interval (in green).

6.3 Results based on aggregating LSTM model-based data

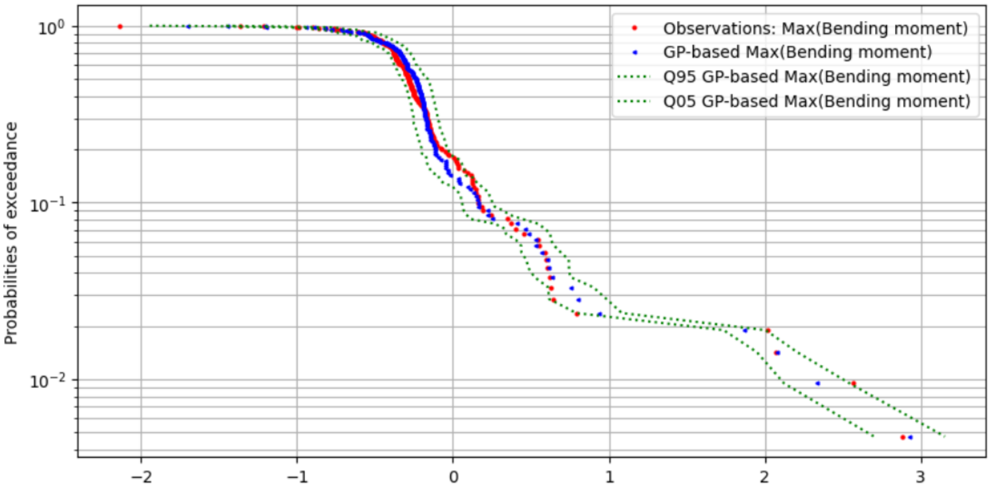

Figure 6.3: Max of the bending moment: Probabilities of exceedance based on the observations (in red), the GP model (in blue) and the confidence interval (in green).

7 Model updating

This section presents the results from the model transfer and updating as cases as summarized in table 4.4.

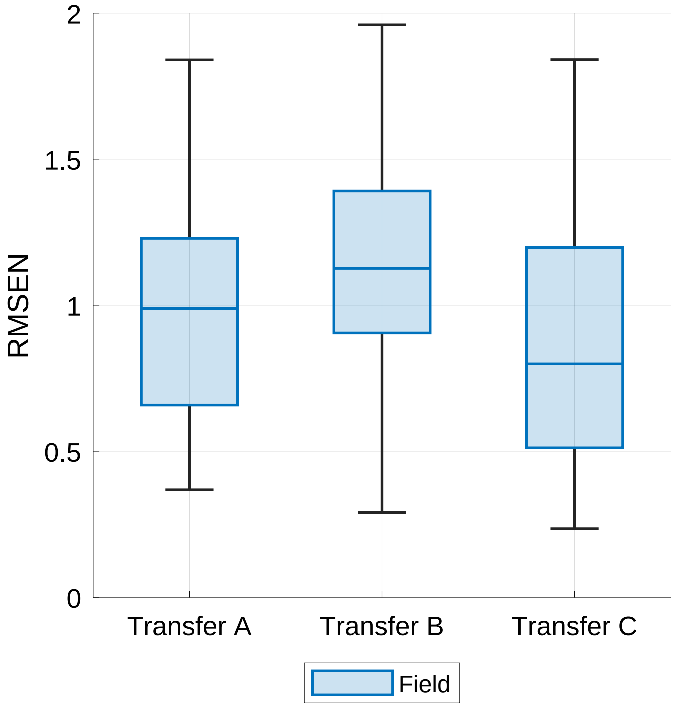

Figure 7.1: Prediction RMSEN of simulation data based model, tested on field measurements.

Figures 7.1 and 7.2 show the results from the model transfer. In this case, the model trained on simulation data is directly used to predict the field measurements of the fore-aft bending moment. 7.1 shows the prediction accuracy of transfer cases A to C. For A and B, the mean RMSEN is close to or above one, which means that the prediction error is equal to or above the standard deviation of the signal itself. However, for case C, which only takes acceleration signals as input, the mean prediction error is around 0.75, indicating that the model can somewhat predict the target signal. These findings are also confirmed by analyzing the time series of predictions and reference signals. While for cases A and B, no correlation between prediction and reference is visible, the predictions of case C follow the low-frequency fluctuations of the the reference signal. This indicates the potential for the transfer of simulation-based models to field conditions.

In the second step, the models trained on simulation data are updated using the field measurements. This is done by retraining the simulation models using different quantities of measurement data. The prediction accuracy is then tested on unseen data from the field measurements.

Figure 7.3 shows the prediction accuracy for the updated models for cases A to C. For case A, the prediction accuracy of the updated model increases with the amount of training data used for the model update. However, the results do not indicate that the updated models have a higher prediction accuracy, compared to the models purely trained on field data.

In contrast, for cases B and C a trend towards better prediction accuracy for the updated models is visible for mean prediction errors and spread of prediction errors. This is particularly

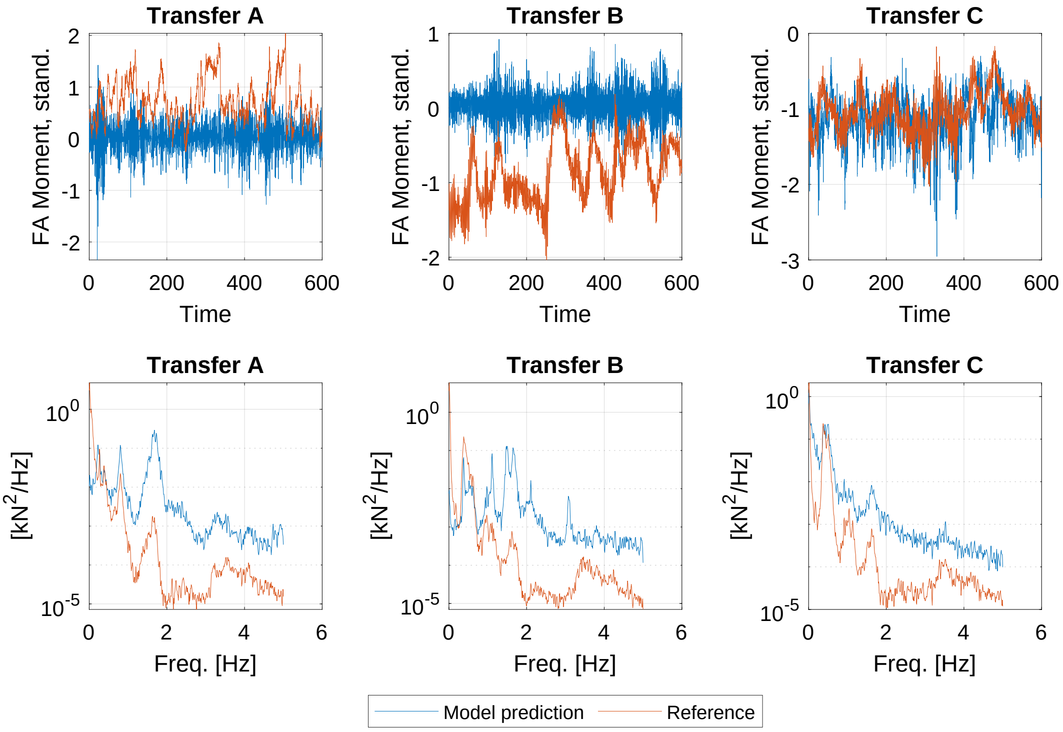

Figure 7.2: Prediction RMSEN of simulation data based model, tested on field measurements.

visible for the update with small amount of field training data (one batch represents circa 3 day of field measurements). This finding indicates potential in situations where little field measurements are available. Here the model accuracy can benefit from a extensive simulated training data set, covering a wide range of operating conditions, updated with limited data from the field.

In conclusion, the results indicate that combining simulation and field measurement data for model training is promising but needs further refinement. The prediction accuracy of transferred models is limited. Several factors may contribute to this outcome. One key issue is the differing correlation patterns between input and output data in the simulation versus the field data. This problem could be tackled by further improving the representation of the turbine and measurement setup in the simulations, leading to a better correlation between simulated and measured signals. Additionally, further data pre-processing techniques, such as filtering and cleaning of field measurements, should be explored. For the updated models, the results of this study show potential for the approach, which should be investigated further.

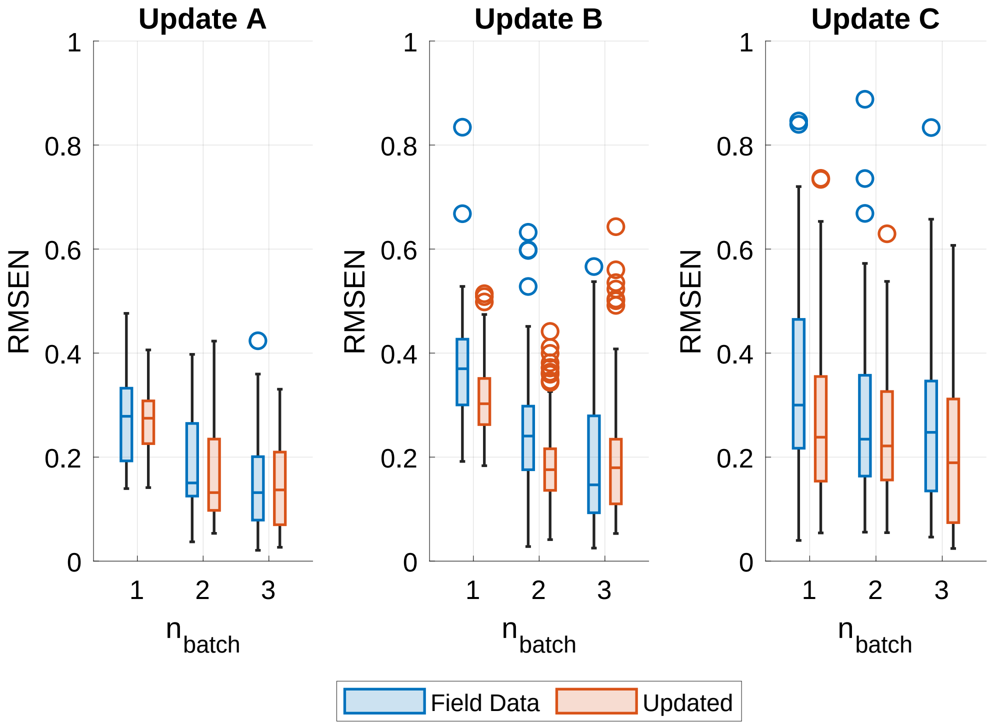

Figure 7.3: Prediction RMSEN of field model vs. updated model for 1 to 5 batches of training data.

8 Conclusions

In conclusion, this project successfully validated and evaluated the surrogate modeling techniques developed for predicting wind turbine loads, focusing on reducing uncertainties in wind turbine design and operational load predictions. The study examined both Long Short-Term Memory and Gaussian Process models, which were trained on simulated and field data. The results showed that simulation-trained models generally outperformed those trained on field data due to the lower variability and absence of measurement uncertainty in simulation environments. However, field-based models showed significant improvements with increased data but reached a point of diminishing returns beyond a certain amount of training data.

The Gaussian Process models proved effective in predicting mean bending moments, but they struggled with maximum moments, pointing to areas where further refinement is needed.

Model updating, where simulation models were retrained using field data, showed promise, particularly when limited field data was available. These findings suggest that while integrating simulation and field data has the potential to improve load prediction models, challenges remain, particularly in addressing discrepancies between simulated and real-world conditions.

The project highlights the potential for surrogate modeling techniques to reduce uncertainties in wind turbine design and operation. Still, it also emphasizes the need for continued research to improve model accuracy, especially when blending simulation data with real-world measurements. Future efforts should focus on enhancing the generalizability of these models and refining data pre-processing techniques to ensure better alignment between simulation and field data.

Acknowledgements

This work is a part of the HIghly advanced Probabilistic design and Enhanced Reliability methods for the high-value, cost-efficient offshore WIND (HIPERWIND) project, which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101006689. The support is greatly appreciated. The authors gratefully acknowledge the computational and data resources provided on the Sophia HPC Cluster at the Technical University of Denmark, DOI: 10.57940/FAFC-6M81 .

References

H. Cardot, F. Ferraty, and P. Sarda. Functional linear model. *Statistics & Probability Letters*, 45 (1):11–22, 1999.

N. Dimitrov and T. Göçmen. Virtual sensors for wind turbines with machine learning-based time series models. *Wind Energy*, 25(9):1626–1645, 2022. doi: https://doi.org/10.1002/we.2762. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/we.2762.

N. Dimitrov and A. Natarajan. From scada to lifetime assessment and performance optimization: how to use models and machine learning to extract useful insights from limited data. *Journal of Physics: Conference Series*, 1222(1):012032, may 2019. doi: 10.1088/1742-6596/1222/1/012032. URL https://dx.doi.org/10.1088/1742-6596/1222/1/012032.

N. Dimitrov, S. Marelli, and S. Schär. Hiperwind deliverable report D4.1 - novel surrogate modelling approaches for wind turbine reliability assessment. Technical report, ETH Zürich and Danmarks Tekniske Universitet, 2022.

M. J. Gräfe, V. Pettas, N. Dimitrov, and P. W. Cheng. Machine learning based virtual load sensors for mooring lines using motion and lidar measurements. *Wind Energy Science Discussions*, 2024:1–29, 2024. doi: 10.5194/wes-2024-25. URL https://wes.copernicus.org/preprints/wes-2024-25/.

M. Gräfe, V. Pettas, and P. W. Cheng. Data-driven forecasting of fowt dynamics and load time series using lidar inflow measurements. *Journal of Physics: Conference Series*, 2767 (3):032025, jun 2024. doi: 10.1088/1742-6596/2767/3/032025. URL https://dx.doi.org/10.1088/1742-6596/2767/3/032025.

D. A. Jackson. Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. *Ecology*, 74(8):2204–2214, 1993.

O. Le Maître and O. M. Knio. *Spectral Methods for Uncertainty Quantification*. Scientific Computation. Springer, Dordrecht, 2010. doi: 10.1007/978-90-481-3520-2.

J. O. Ramsay. *Functional data analysis*. Wiley Online Library, 2006.

C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.