Integrating Physics-Based Models and Data-Driven Methods for Input Load Estimation and Digital Twining of Offshore Wind Turbines

A dissertation submitted by

Azin Mehrjoo

In partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Civil and Environmental Engineering

Tufts University

August 2024

Advisor:

Prof. Babak Moaveni

Abstract

Offshore wind turbine support structures are subjected to a wide range of loading events, from routine operational conditions to extreme weather, making them vulnerable to fatigue and potential failure. These failures can lead to significant economic losses due to downtime in wind energy production. As turbines increase in size and are located further offshore, the challenges and costs associated with their preventive maintenance escalate. This research addresses these challenges by exploring the potential of condition-based monitoring and predictive maintenance, moving away from the traditional reactive approach.

Central to this strategy is the concept of the digital twin, a virtual replica of a physical turbine that simulates structural behavior under various conditions. For digital twins to be effective in real-time monitoring, they must integrate accurate methods for estimating input loads, which are essential for predicting structural responses and performing virtual sensing at unmeasured locations. This dissertation proposes a comprehensive framework that extends digital twin capabilities through advanced data collection and input load estimation techniques.

The first component of the framework focuses on collecting high-quality data with minimal sensors, employing theoretical observability and optimal sensor placement strategies to ensure accurate estimation of key structural parameters. The second component addresses the challenge of input load estimation, proposing a range of methods from computationally intensive physics-based approaches to more efficient hybrid models that combine machine learning with physics-based methods. These include transfer learning, multitasking, and Monte Carlo dropout for uncertainty quantification. While these efficient models significantly reduce computational costs,

they still require real data for training, and real input load data from offshore wind turbines are often unavailable.

To overcome the limitations of supervised learning in scenarios where measured input loads are unavailable, the framework further advances and proposes a physics-informed neural network strategy. These networks incorporate physical laws directly into the model, allowing for accurate input load estimation without relying on extensive training data. The research demonstrates that by integrating these innovative approaches, it is possible to enhance the reliability, efficiency, and cost-effectiveness of SHM for offshore wind turbines, paving the way for more sustainable wind energy production.

Dedication

To my wonderful parents, Azita and Masoud,

This thesis is dedicated to you with all my love and gratitude. Your unwavering support, endless encouragement, and boundless love have been the foundation of my journey. Your belief in me has given me the courage to face every challenge and continue moving forward. I am deeply thankful for your sacrifices, your guidance, and the unconditional love you have shown me every step of the way.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Dr. Babak Moaveni, for his guidance and support throughout my Ph.D. studies. His expertise and advice have been crucial in the development of this research. I deeply appreciate his mentorship and the opportunities he provided me during this journey. I also wish to thank the members of my committee, Dr. Costas Papadimitriou, Dr. Manolis Chatzis, Dr. Eric Hines, and Dr. Anna Haensch for their valuable feedback and contributions. Their insights and suggestions have significantly enhanced the quality of this work.

I would like to specifically thank Dr. Eleonora Tronci, who not only contributed as a committee member but also offered her support as a friend. Her willingness to sit with me through numerous meetings and help solve issues was immensely helpful, and I am truly grateful for her generosity and kindness. I would also like to express my gratitude to my friend and peer, Bridget, whose kindness, friendship, and support in the office made a significant difference in my Ph.D. experience. I am truly thankful for the bond we formed during this time.

I am deeply grateful to my wonderful partner and best friend, Amin, whose support over the past years made this journey possible. His dedication and care were essential in helping me achieve this milestone. I cannot thank him enough for being by my side, not just as a source of support but as a peer who helped me work through the professional challenges I encountered along the way.

I would like to thank my brother, Amid, for always being there, for his humor that helped me navigate challenging moments, and the strength of our bond, which has been a constant source of comfort throughout these years.

Last but not least, I extend my heartfelt thanks to my dearest friends Mina, Nazanin, and Golshan, for decades of friendship, mental support, and countless other forms of encouragement. Your enduring presence in my life has been invaluable.

Table of Contents

List of Figures

List of Tables

Integrating Physics-Based Models and Data-Driven Methods for Input Load Estimation and Digital Twining of Offshore Wind Turbines

1 Chapter 1: Introduction

1.1 Background

Offshore wind turbine (OWT) support structures endure various loading events throughout their operational lifespan, ranging from routine conditions to extreme weather events. These structures are constantly exposed to cyclic wind and wave loads, increasing their susceptibility to fatigue, which is a leading cause of support structure failure. Structural failures and downtime in offshore wind energy production can have significant economic implications. Therefore, it is crucial to take proactive measures to address structural issues before they escalate and to ensure the reliability of support structures.

Operating in challenging conditions, preventive maintenance of OWTs is difficult and costly, especially as turbines grow larger and more distant from shore. Therefore, instead of relying solely on frequent and costly preventive maintenance, this work explores the potential of condition-based monitoring and predictive maintenance. By leveraging advanced monitoring techniques, we aim to move away from the traditional 'fix-it-when-it-breaks' approach. This transition allows us to monitor the condition of the turbines in real-time and predict potential issues before they escalate.

Condition-based monitoring and predictive maintenance techniques are based on the vibrational response of the monitored structure. However, obtaining comprehensive vibrational response data directly from the structure is challenging due to the limited number of sensors that can be deployed. This limitation necessitates the use of virtual sensing, where the vibrational response at unmeasured locations is estimated using available data and computational models. This is where the concept of a digital twin becomes crucial.

A digital twin is a virtual replica of a physical wind turbine, capable of simulating system behavior under various environmental and operational conditions. It serves as a powerful tool for structural response predictions and virtual sensing, which are integral to condition-based monitoring techniques. Previous and current digital twinning practices for monitoring and performance assessment of OWTs focus on modeling accurate digital replicas by updating their parameters with new monitoring data. However, these models still require real-time input from the structure to estimate quantities of interest, such as strain time histories for fatigue life estimation.

To make digital twins truly effective for real-time monitoring and virtual sensing, it is essential to integrate methods for estimating input loads applied to the structure. Without the ability to provide real-time responses for the structure, a digital twin remains merely a sophisticated finite element (FE) model. Therefore, this research proposes a comprehensive framework that extends digital twin concepts by incorporating input load estimation methods, making it possible to predict structural responses accurately and perform virtual sensing.

The proposed framework consists of two main components: Data collection techniques and input load estimation techniques. The first component addresses the need to collect high-quality data with minimal sensors. The sensors must guarantee the estimation of quantities of interest (QoI) such as strains and parameters, ensuring we have the minimum necessary information for accurate estimation. This is achieved through theoretical observability, which determines whether the defined system is observable under the assumed measurements. Additionally, to obtain the best possible data, sensors should be optimally placed, which involves determining the optimal number of sensors in their optimal locations with optimal cost, a methodology known as optimal sensor placement.

The second component of the framework focuses on the estimation of input loads, which is essential for virtual sensing. Different methods were explored for various scenarios, starting with a purely physics-based approach using a recursive Bayesian estimation method. While accurate, this method was computationally expensive and unsuitable for real-time applications. Consequently, hybrid methods combining the computational power of machine learning (ML) with the accuracy of physics-based methods were proposed.

The first hybrid method involves transfer learning, multitasking, and Monte Carlo dropout for uncertainty quantification. The method works by training based on simulations and fine tuning using limited real data. This method significantly reduces computational costs but relies on supervised learning, requiring measured input loads for training. Since load measurements are often unavailable in the offshore wind industry, an alternative approach was needed.

To address this, the framework includes physics-informed neural networks (PINNs). In the first PINN method, physics laws are enforced in the loss function of the neural network, allowing the input load to be learned alongside network parameters as a trainable parameter. In the second PINN method, an integrated neural network with intertwined sub-networks estimates the input load without assuming it is measured, using a loss function that enforces physics laws without relying on input load data.

1.2 Objectives and Original Contributions

The key objectives and contributions of this dissertation include:

- Developed a novel OSP framework integrating Bayesian inference, information theory, and a cost-benefit objective function to optimize sensor placement for parameter estimation and strain prediction in OWTs, balancing accuracy with installation costs.
- Applied a theoretical observability framework to determine minimal sensor networks for capturing essential structural information in wind turbines. Extended observability analysis to nonlinear systems, providing insights into optimal sensor placement for full observability.
- Developed a recursive Bayesian estimation method for accurate input load prediction using output-only measurements.
- Demonstrated virtual sensing at unmeasured locations, enhancing digital twinning for comprehensive SHM of OWTs.
- Created a hybrid model combining data-driven and physics-based methods that integrates transfer learning and multitask learning with Monte Carlo dropout for robust real-time input load estimation.
- Introduced PINN frameworks that integrate physical laws into neural networks for accurate input load estimation in OWTs.
- Developed an integrated network with interconnected sub-networks for simultaneous prediction of input loads and dynamic responses.

1.3 Thesis Organization

Chapter 2 of this dissertation focuses on the optimal sensor placement (OSP) framework for parameter estimation, virtual sensing, and condition monitoring using information theory. The framework used a Bayesian OSP method combined with modal expansion to minimize the

information entropy about quantities of interest (QoI), such as strain time histories at critical locations of the structure, without the knowledge of input excitation. The proposed optimization framework also accounted for variations in sensor installation cost at different locations on the monitored structure. The framework was evaluated numerically using a realistic model of an offshore wind turbine on a jacket support structure under installation cost assumptions and considering information entropy of the QoI. The QoI in this numerical study were the strain time history at one or more locations on the support structure in one problem and the parameters of the structure in the other. A correlation length was considered to account for the spatial correlation of data between adjacent sensors. Effects of the correlation length and input loads on the OSP results for parameter estimation were studied. The considered structural parameters for estimation in this study include (1) modulus of elasticity of tower elements (tower stiffness), (2) modulus of elasticity of jacket elements (jacket stiffness), and (3) vertical foundation spring (soil stiffness). The effect of a subjective weight between the information entropy and sensor configuration cost on the OSP results was also investigated. Different optimal designs were achieved for different weight factors, and the Pareto solutions for OSP were presented.

In Chapter 3, the theoretical observability framework was employed to formulate recommendations for designing sensor networks that ensure sensors can capture the essential information needed to infer critical quantities of interest. Given the modular nature of wind turbines, these recommendations are intended to have broad applicability. The primary quantities of interest include the states of the structure, which are vital for effective structural health monitoring. Additionally, in the future, we plan to address the inversion problem by estimating the loads acting on the structure from the measurements. Of particular interest is the separate estimation of wave and wind loads, due to their distinct characteristics. By leveraging this

theoretical observability framework, we aim to recommend a minimal sensor placement strategy that guarantees the precise estimation of these critical quantities of interest.

Chapter 4 discusses a recursive window-based Bayesian estimator for input load estimation and virtual sensing of acceleration and strain time history at unmeasured critical locations using output-only measurements. The estimator was numerically and experimentally validated in the context of input load estimation for an offshore wind turbine, and the results were compared with a traditional augmented Kalman filter and a linear regression approach for the quasi-static component of loading. The method was first demonstrated through a numerical study using a finite element model of a 6 MW offshore wind turbine, where input wind load time histories were accurately estimated. Then, the framework was applied to the real data measured from an offshore wind turbine in the North Sea. The study demonstrated the window-based Bayesian input estimator as a promising tool for the digital twinning of offshore wind turbines, demonstrating superior accuracy in input load estimations and full response predictions compared to the augmented Kalman filter and linear regression methods without the constraint of co-located measurements at input load locations.

Chapter 5 presents a novel hybrid surrogate model for real-time input load estimation that integrates data-driven and physics-based methodologies to address these limitations. The proposed framework focused on training a neural network-based regression model that can predict the input load time history using the dynamic structural response (accelerometers and strain measurements) as input. The model was optimally designed to overcome the lack of experimental data by following a transfer learning strategy: the model was initially trained on synthetic data generated from a physics-based model and then fine-tuned using experimental data. Additionally, a

multitasking approach was implemented to extend the applicability of the model to different regions of operation. Finally, a dropout Montecarlo strategy was proposed to quantify the uncertainty of the regression model. This hybrid surrogate model demonstrated superior computational efficiency and accuracy in input load estimation compared to conventional physics-based models and it offers several practical benefits. It is scalable, does not necessitate continual recalibration, and exhibits robustness across diverse datasets and operational conditions. The model's potential for real-time applications underscores its contribution to advancing maintenance strategies and design improvements in offshore wind turbine engineering.

Chapter 6 introduces two novel frameworks based on physics-informed neural networks to tackle the complex inverse problem of estimating input loads in offshore wind turbines. These frameworks are uniquely crafted to function in scenarios where input loads are not directly measurable. Both frameworks utilize the concept of physics-informed learning by incorporating the equations of motion into their loss functions, which guarantees that the predictions comply with the basic physical laws that regulate the system. The first framework uses time as the input and measured displacements and accelerations as the outputs. In this setup, the input load is considered an additional parameter that can be learned. The network dynamically estimates the input load using a loss function that minimizes data loss, which accounts for discrepancies between predicted and actual values, with a physics-based loss that enforces the equations of motion. The second framework introduces an integrated network that uses two interlinked networks to predict input loads and dynamic responses. This framework includes a comprehensive loss function composed of three elements: equation of motion loss, differentiation law loss, and data loss. The equation of motion loss ensures that the predicted input load complies with physical laws, while the differentiation law loss preserves the consistency of relationships between displacement,

velocity, and acceleration. The data loss minimizes the differences between predicted and actual measurements, ensuring the accuracy of observed data. Currently both frameworks are validated for MDOF systems with a plan to extend the application of the frameworks on OWTs.

Chapter 7 provides a summary of the key findings and contributions of the research presented in this thesis. It synthesizes the insights gained from the various studies, highlights the overall significance of the work, and discusses its implications for the field. The chapter also outlines potential areas for future research, suggesting directions for further development and exploration based on the outcomes of this study.

2 Chapter 2: Optimal Sensor Placement for Parameter Estimation and Virtual Sensing of strains on an Offshore Wind Turbine Considering Sensor Installation Cost

The findings provided in this chapter are from the material Published in:

- *Mehrjoo A, Song M, Moaveni B, Papadimitriou C, Hines E. Optimal sensor placement for parameter estimation and virtual sensing of strains on an offshore wind turbine considering sensor installation cost. Mechanical Systems and Signal Processing. 2022 Apr 15;169:108787.*

2.1 Introduction

The global offshore wind energy (OWE) capacity is projected to expand 15-fold over the next two decades [1]. In light of our increasing dependence on this form of renewable energy, offshore wind developers, operators, and investors will benefit from stronger assurances of offshore wind turbines' (OWT) reliability through structural health monitoring (SHM). In addition to lowering investment risk, increased reliability can reduce the cost of maintenance and extend the design life of future wind farms. High-performance SHM consists of systematic monitoring and efficient condition-based maintenance strategies that are based on the right sensors in the right places for the right price, known as optimal sensor placement. Moreover, corrosion and fatigue are the most common failure mechanisms of an OWT support structure[2, 3]. To evaluate the remaining life of such structures, it is necessary to know the level of stresses and strains over the life of the structure.

SHM implements damage identification strategies for engineering structures and estimating their remaining useful life. Damage refers to any variation of the mechanical properties of the structure over time. The techniques to estimate a structural system's mechanical properties from measurements are referred to as system identification methods. System identification methods can be classified into output-only methods—where only the system's response is observed, and input-output methods—where the inputs and the response of the system are measured [4]. The objectives of SHM include investigation of uncertainties in structural behavior, condition-based maintenance, remaining lifetime prognosis, and optimization of future designs. In a rapidly growing global industry like OWE, these methods can help minimize Operation and Management (O&M) costs, protect against catastrophic failures, and provide feedback to improve the design of future turbines [5]. SHM methods can employ probabilistic Bayesian frameworks to account for uncertainties and

errors in the measured data, estimation process, and structural model [6]. The application of Bayesian inference for SHM of civil structures was first introduced in 1998 [7]. Several studies include Bayesian inference applications for model updating and SHM of large-scale civil structures, which can be found in [8-13].

The reliability of an SHM procedure depends on the method's accuracy and the information available in the measured data. The location and type of sensors on the structure determine the level of information about the QoI in the data. SHM includes the experimental procedures to estimate a QoI using partially measured data [14]. Moreover, SHM includes virtual sensing, which is the process of estimating a QoI using a validated model and partially measured data. The experimental setup in SHM, corresponding to the number and location of the sensors, can be optimized to provide cost-effective and informative measurements about the QoIs [15]. This process is referred to as Optimal Sensor Placement (OSP). This chapter introduces an approach to OSP for estimating the parameters and strain history, and hence the fatigue life, of an OWT support structure. OSP methods hold promise for efficient condition-based monitoring at scale and could enable precise decision-making for entire OWT fleets regarding support structure maintenance and life extension.

Information theory-based approaches have been used by a number of researchers for OSP problems. These approaches can account for measurement uncertainties as well as modeling errors. Different measures of the information contained in the data were used to find the OSP solution. Some researchers have maximized the determinant or trace of the Fisher Information Matrix (FIM) [16-23], while others have employed the expected Bayesian loss function [24, 25] to minimize the trace of the inverse of the FIM. A Bayesian framework for OSP in SHM applications was proposed

in [26], where the OSP method optimized a metric related to the probability of damage detection of all regions of the structure. Shannon entropy [27], a measure of the uncertainty in model parameter estimation, was introduced in [28] for the OSP problem using a Bayesian framework. Shannon entropy is used in OSP for (i) parameter estimation when the input excitation is not measured [29], (ii) load identification of linear and nonlinear models [30], and (iii) model class selection for damage detection [31]. Shannon entropy depends on the determinant of the FIM for an asymptotically large number of data [15]. In [32], the expected information gain [33], which was derived from Kullback-Leibler divergence (KLD) [34], was maximized for finding OSP for response reconstruction. The effect of spatial correlation of the prediction error due to modeling error on the OSP problem was investigated in [35]. Estimating strains using virtual sensing is useful for fatigue estimation [36, 37]. There are several studies in the literature using the modal expansion method for strain estimation of OWT [38-42]. Fatigue hotspots in an OWT with a jacket support structure are commonly underwater and difficult to access [43].

Solving the optimization problem of an OSP solution using discrete-valued design variables requires an exhaustive search over all the possible combinations of sensor types and locations. This is computationally expensive, even for a structure with a small number of degrees of freedom (DOFs). Heuristic algorithms such as Forward Sequential Sensor Placement (FSSP) and Backward Sequential Sensor Placement (BSSP) were proposed in [15, 31], which overcome this computational problem but often provide a suboptimal sensor configuration. Genetic Algorithms [44-47] are alternatives for exhaustive search and can be used to complement FSSP and BSSP algorithms for improved results [35]; however, they are more computationally demanding than the heuristic algorithms.

Past OSP studies have not adequately explored the effect of cost on the OSP result. This chapter investigates both single-objective optimization and multi-objective (Pareto efficiency) optimization to find OSP for the purpose of condition monitoring of an OWT for input-output parameter estimation and output-only strain estimation cases. The single-objective formulation minimizes the information entropy of the sensor configuration and does not consider the sensor configuration cost. The multi-objective OSP optimizes the cost-benefit objective function defined with different weight factors to balance the ratio of information and cost in the objective function. The present study suggests a balanced multi-objective OSP that provides insight into the effect of cost on the optimal sensor configuration.

The defined multi-objective function, Pareto study, and the application of the framework on OWT comprise the novelty of this work, where a combination of FSSP and BSSP search methods is used. The combined method is called sequential sensor placement (SSP). The SSP gives near-optimal results comparable with the exhaustive search. Implementation of the algorithm is evaluated on optimal sensor configuration design of a realistic offshore wind turbine model for (1) parameter estimation and (2) strain estimation. The Pareto study presented in this chapter provides insight to help decision-makers with the OSP design given a limited budget. A sensitivity analysis is performed to find the most influential parameters on the dynamical behavior of the structure and the selection of updating structural parameters. The study also considers the effect of spatially correlated prediction errors, which controls the minimum distance between sensors, as well as the effect of input excitation assumption on OSP results for parameter estimation in an input-output formulation. The effectiveness of the OSP algorithm is checked by comparing the best and worst configurations.

2.2 Methodology

This section presents the formulation of OSP for (1) parameter estimation and (2) strain estimation. Structural parameters of interest can include mass and/or stiffness of different components or substructures of a structure. In this study, the parameters reflecting the uncertainties in the model are the modulus of elasticity of the tower, E_{tower}, modulus elasticity of the jacket, E_{jacket}, and the vertical stiffness, k_v, of soil springs at the bottom of the jacket. These parameters are chosen based on a sensitivity analysis done on the structure. In the first two subsections, the Bayesian inference frameworks for each of these two purposes are briefly reviewed, and OSP formulation using Shannon Entropy and Expected Utility is outlined afterward.

2.2.1 Bayesian inference framework for parameter estimation

The Bayesian inference framework for estimating unknown parameters of a structural model is briefly reviewed here. Assume $\boldsymbol{\theta} \in R^{N_\theta}$ be the vector of parameters of interest. Let $\mathcal{D} = \left\{ \mathbf{y}_k \in R^{N_o} \right\}$ be the vector of measured acceleration response time history sampled over N points (k = 1, .., N) by Δt intervals where $\mathbf{y}_k \in R^{N_o}$ refer to output data and N_o is the number of measured DOFs, i.e., number of sensors. Let $\mathbf{x}_k(\boldsymbol{\theta}) \in R^{Nd}$ be the vector of predicted response time history from the Finite Element (FE) model at all DOFs (N_d) of the structure at each time step. The difference between the measurements and the model predictions is defined as the prediction error $\mathbf{e}_k(\boldsymbol{\theta})$, which includes the combined effects of modeling error and measurement noise. The relationship between the measured response $\mathbf{y}_k$ and $\mathbf{x}_k(\boldsymbol{\theta})$ follows Eq. (2.1):

$$\mathbf{y}_k = \mathbf{L}\mathbf{x}_k(\boldsymbol{\theta}) + \mathbf{e}_k(\boldsymbol{\theta}) \tag{2.1}$$

where $\mathbf{L} \in R^{N_o \times N_d}$ is a Boolean matrix consisting of ones at the measured DOFs and zeros elsewhere. It defines the number and location of sensors on the structure; therefore, it can be interpreted as the sensor configuration matrix. The prediction error is postulated to follow a Gaussian distribution with zero mean and covariance matrix $\mathbf{\Sigma}_t \in R^{N_o \times N_o}$. This assumption is justified by the principle of maximum information entropy, which states that the Gaussian distribution provides maximum uncertainty among all distributions with a specified mean and covariance. This imposes the least amount of constraint on the choice of distribution. However, other distributions can be assumed for the prediction error if there is prior knowledge about its statistical properties. Assuming prediction errors at different time steps are independent, the posterior probability density function (PDF) of the structural model parameters given the measured data has the following form [7]:

$$p(\boldsymbol{\theta}|\mathbf{\Sigma}_t, \boldsymbol{\mathcal{D}}) \propto \frac{1}{(\sqrt{2\pi})^N \sqrt{\det \mathbf{\Sigma}_t}} \exp[-\frac{NN_o}{2} J(\boldsymbol{\theta}|\mathbf{\Sigma}_t, \boldsymbol{\mathcal{D}})] p(\boldsymbol{\theta}) \tag{2.2}$$

where $p(\boldsymbol{\theta})$ is the prior PDF of updating parameters; and

$$J(\boldsymbol{\theta}|\mathbf{\Sigma}_t, \boldsymbol{\mathcal{D}}) = \frac{1}{NN_o} \sum_{k=1}^{N} [\mathbf{y}_k - \mathbf{L}\mathbf{x}_k(\boldsymbol{\theta})]^{\mathrm{T}} \mathbf{\Sigma}_t^{-1} [\mathbf{y}_k - \mathbf{L}\mathbf{x}_k(\boldsymbol{\theta})] \tag{2.3}$$

represents the measure of fit between the measured data and corresponding model predictions. This method requires both the input and output of the system to evaluate the posterior PDF.

2.2.2 Bayesian inference framework for strain estimation

This section provides a review of the Bayesian inference framework and modal expansion technique for estimating strains at desired locations using measured time history structural

response without the knowledge of input. Let $\mathcal{D} = \left\{ \mathbf{y}_k \in R^{N_o} \right\}$ be the vector of displacement time history measurements where $\mathbf{y}_k$ is the output data at N_o DOFs and $\mathbf{e}_k$ be the Gaussian measurement error with zero mean and covariance matrix. The zero mean assumption is valid when the model's predictions are, on average, accurate, indicating that the model is unbiased. This means that prediction errors are equally likely to be positive or negative, effectively canceling each other out over multiple predictions. In the context of OSP, which occurs during the design phase of the sensor network, any bias present in the model can be corrected. This ensures that the zero mean assumption holds true before applying the methodology. The measured time history responses can be expanded using modal expansion formulation as:

$$\mathbf{y}_k = \mathbf{\Phi}_m(\mathbf{L})\mathbf{q}_{m_k} + \mathbf{e}_k \tag{2.4}$$

where $\mathbf{\Phi}_m(\mathbf{L}) \in R^{N_o \times m}$ is the mode shape matrix of *m* contributing modes at N_o DOFs and $\mathbf{q}_{m_k}$ is the modal coordinate vector of *m* contributing modes at time step *k*.

Subsequently, Bayesian inference is employed to estimate $\mathbf{q}_{m_k}$ and its uncertainty, i.e., posterior PDF of $\mathbf{q}_{m_k}$. Based on Bayes theorem, the posterior PDF of $\mathbf{q}_{m_k}$ given the measured time history $\mathbf{y}(t)$ is:

$$p(\mathbf{q}_{m_k} | \mathbf{y}_k) \propto p(\mathbf{y}_k | \mathbf{q}_{m_k})\, p(\mathbf{q}_{m_k}) \tag{2.5}$$

where $p(\mathbf{y}_k | \mathbf{q}_{m_k})$ is the likelihood function. Assuming the prior PDF $p(\mathbf{q}_{m_k})$ to be Gaussian with zero mean and covariance matrix $\mathbf{\Sigma}_{pr} \in R^{m \times m}$, the posterior PDF of $\mathbf{q}_{m_k}$ is Gaussian with $\bar{\mathbf{q}}_k$ mean and $\mathbf{\Sigma}_q$ covariance [32, 48, 49] given by:

$$\bar{\mathbf{q}}_k = \left[\boldsymbol{\Phi}_m^{\mathrm{T}}(\mathbf{L})\boldsymbol{\Sigma}_t^{-1}(\mathbf{L})\boldsymbol{\Phi}_m(\mathbf{L}) + \boldsymbol{\Sigma}_{pr}{}^{-1}\right]^{-1}\boldsymbol{\Phi}_m^{\mathrm{T}}(\mathbf{L})\boldsymbol{\Sigma}_t^{-1}(\mathbf{L})\mathbf{y}_k \tag{2.6}$$

$$\boldsymbol{\Sigma}_q(\mathbf{L}) = \left[\boldsymbol{\Phi}_m^{\mathrm{T}}(\mathbf{L})\boldsymbol{\Sigma}_t^{-1}(\mathbf{L})\boldsymbol{\Phi}_m(\mathbf{L}) + \boldsymbol{\Sigma}_{pr}{}^{-1}\right]^{-1} \tag{2.7}$$

The modal expansion technique is then employed for virtual sensing, i.e., to predict strains at the desired DOFs. This can be done using the estimated modal coordinate and assuming a Gaussian prediction error with zero mean and covariance $\boldsymbol{\Sigma}_\varepsilon \in R^{N_z \times N_z}$:

$$\mathbf{z}_k = \boldsymbol{\Psi}\mathbf{q}_{m_k} + \boldsymbol{\varepsilon}_k \tag{2.8}$$

where $\mathbf{z}_k \in R^{N_z \times 1}$ is the vector of predicted strains at desired locations (N_z) and $\boldsymbol{\Psi} \in R^{N_z \times m}$ is the strain mode shape matrix of *m* contributing modes.

By propagating the uncertainty of estimated $\mathbf{q}_{m_k}$ to $\mathbf{z}_k$ in Eq. (2.8), $\mathbf{z}_k$ follows a Gaussian distribution with $\bar{\mathbf{z}}_k$ mean and $\boldsymbol{\Sigma}_z \in R^{N_z \times N_z}$ covariance given by:

$$\bar{\mathbf{z}}_k = \boldsymbol{\Psi}\left[\boldsymbol{\Phi}_m^{\mathrm{T}}(\mathbf{L})\boldsymbol{\Sigma}_t^{-1}(\mathbf{L})\boldsymbol{\Phi}_m(\mathbf{L}) + \boldsymbol{\Sigma}_{pr}{}^{-1}\right]^{-1}\boldsymbol{\Phi}_m^{\mathrm{T}}(\mathbf{L})\boldsymbol{\Sigma}_t^{-1}(\mathbf{L})\mathbf{y}_k \tag{2.9}$$

$$\boldsymbol{\Sigma}_z(\mathbf{L}) = \boldsymbol{\Psi}\left[\boldsymbol{\Phi}_m^{\mathrm{T}}(\mathbf{L})\boldsymbol{\Sigma}_t^{-1}(\mathbf{L})\boldsymbol{\Phi}_m(\mathbf{L}) + \boldsymbol{\Sigma}_{pr}{}^{-1}\right]^{-1}\boldsymbol{\Psi}^{\mathrm{T}} + \boldsymbol{\Sigma}_\varepsilon \tag{2.10}$$

Eq. (2.10) indicates that the uncertainty in estimated strains does not depend on the measurements $\mathbf{y}_k$. Consequently, optimal sensor placement for strain estimation, which relies on minimizing a scalar measure of $\boldsymbol{\Sigma}_z$, does not directly require the input or output measurements.

The strain mode shape matrix is a property of the structure and can be calculated utilizing the displacement mode shape matrix and shape functions used in the FE model (for example, Hermitian shape function for beam-column element).

2.2.3 Spatial correlation of prediction error

In this study, the effect of spatial correlation of prediction errors between neighboring DOFs is considered. The covariance of total prediction error should accord with the errors and correlations inhere in the measurements and the model. However, no measurements are available during the sensor design stage. In [35], the authors used a measure of spatial correlation length and assumed an exponential correlation function to consider the effect of spatially correlated prediction error. This procedure is briefly introduced here. Considering independency between measurement noise and modeling error, the total prediction error matrix $\mathbf{\Sigma}_t$ takes the form:

$$\mathbf{\Sigma}_t = \overline{\mathbf{\Sigma}} + \mathbf{\Sigma} \tag{2.11}$$

where $\overline{\mathbf{\Sigma}}$ and $\mathbf{\Sigma}$ are the covariance matrices of the measurement noise and modeling error, respectively. As measurement noise is independent of the location of the sensors, $\overline{\mathbf{\Sigma}} = s^2\mathbf{I}$ is considered as a diagonal matrix where $\mathbf{I}$ is the identity matrix and s^2 denotes the variance of sensor noise. However, due to the correlation between neighboring locations because of modeling error, the covariance matrix $\mathbf{\Sigma}$ can have non-zero off-diagonal terms. The correlation between the prediction errors at DOFs *i* and *j* is assumed to be:

$$\Sigma_{ij} = \sqrt{\Sigma_{ii}\Sigma_{jj}}R(\delta_{ij}) \tag{2.12}$$

where R is a function of δ_{ij}, which is the spatial distance between the DOFs. The same correlation function as [35] is used in this study

$$R(\delta_{ij}) = \exp[-\delta_{ij} / \lambda] \tag{2.13}$$

where λ is a measure of the spatial correlation length, and a larger value λ assigns a stronger correlation to distant DOFs. The correlation length λ should be chosen such that the covariance matrix in Eq. (2.12) be consistent with the actual errors and correlations as observed from measurements. However, no such measurements are available in the design phase. The correlation length is recommended to be significantly smaller than the "wavelength" of the highest contributing mode. In [35], it is shown that for up to the characteristic length of the highest contributing mode of the structure, the spatial correlation between prediction errors forces the minimum distance between sensors to be of the order of the correlation length.

2.2.4 Optimal sensor placement formulation

The information entropy method has been used in this study for OSP. For parameter estimation, the formulation is derived using the definition of information entropy and asymptotic approximation. For strain estimation, KLD information gain and information entropy are used. In general, information entropy is a scalar measure of uncertainty associated with a QoI, and information gain is the reduction in uncertainty about the QoI after measuring data. Optimizing either measure would lead to the same optimal experimental setup.

2.2.4.1 OSP framework for parameter estimation

For the purpose of parameter estimation, the information entropy *H* is defined as:

$$H(\mathbf{L}|\boldsymbol{\Sigma}_t,\boldsymbol{\mathcal{D}}) = E_{\boldsymbol{\theta}}\left(-\ln p\left(\boldsymbol{\theta}|\boldsymbol{\Sigma}_t,\boldsymbol{\mathcal{D}}\right)\right) = -\int \ln p\left(\boldsymbol{\theta}|\boldsymbol{\Sigma}_t,\boldsymbol{\mathcal{D}}\right) p\left(\boldsymbol{\theta}|\boldsymbol{\Sigma}_t,\boldsymbol{\mathcal{D}}\right) d\boldsymbol{\theta} \tag{2.14}$$

which depends on measured data $\boldsymbol{\mathcal{D}}$ and sensor configuration $\mathbf{L}$. $E_{\boldsymbol{\theta}}(...)$ denotes mathematical expectation with respect to vector $\boldsymbol{\theta}$. Minimizing the entropy would result in OSP. For

uninformative (i.e., uniform) prior PDF of the model parameters, the information entropy can be asymptotically approximated when a large amount of data is available [15]:

$$H(\mathbf{L}|\boldsymbol{\Sigma}_t, \mathcal{D}) = \frac{1}{2} N_{\boldsymbol{\theta}} \ln(2\pi) - \frac{1}{2} \ln\left[\det \mathbf{Q}(\mathbf{L} \mid \boldsymbol{\Sigma}_t, \hat{\boldsymbol{\theta}})\right] \tag{2.15}$$

where $\hat{\boldsymbol{\theta}}$ is the vector of optimal values for the parameters and $\mathbf{Q}(\mathbf{L} \mid \boldsymbol{\Sigma}_t, \hat{\boldsymbol{\theta}}) \in R^{N_\theta \times N_\theta}$ is the FIM which contains information about the uncertainty of estimated parameters, given by:

$$\mathbf{Q}(\mathbf{L} \mid \boldsymbol{\Sigma}_t, \hat{\boldsymbol{\theta}}) = \sum_{k=1}^{N} \left(\mathbf{L} \nabla_{\boldsymbol{\theta}} \mathbf{x}_k\left(\hat{\boldsymbol{\theta}}\right)\right)^{\mathrm{T}} \left(\mathbf{L} \boldsymbol{\Sigma}_t \mathbf{L}^{\mathrm{T}}\right)^{-1} \left(\mathbf{L} \nabla_{\boldsymbol{\theta}} \mathbf{x}_k\left(\hat{\boldsymbol{\theta}}\right)\right) \tag{2.16}$$

in which $\nabla_{\boldsymbol{\theta}} = \left[\partial/\partial\theta_1 \quad ... \quad \partial/\partial\theta_{N_\theta}\right]$ is the gradient vector with respect to the parameter vector $\boldsymbol{\theta}$. The FIM in Eq. (2.16) is created using the sensitivities of response in sensor locations with respect to the model parameters to be inferred. The logarithm of the determinant of FIM which is present in Eq. (2.15) is an overall scalar measure of this sensitivity. In the absence of experimental data, the optimal value $\hat{\boldsymbol{\theta}}$ of the model parameter set is not available. To proceed with the optimal design, the value of $\hat{\boldsymbol{\theta}}$ is assigned a nominal value. The nominal value of the parameters is considered as the true value of model parameters to represent the actual structural system. The uncertainty in the nominal value $\hat{\boldsymbol{\theta}}$ can be considered by using the expected information entropy over all possible values $\hat{\boldsymbol{\theta}}$ quantified by the prior probability distribution $p(\hat{\boldsymbol{\theta}})$.

2.2.4.2 OSP framework for strain estimation

Optimal sensor placement for strain estimation is formulated using the expected information gain $U(\mathbf{L})$ [50]:

$$U(\mathbf{L}) = \int_{\Upsilon} u(\mathbf{L}, \mathbf{y})\, p(\mathbf{y}|\mathbf{L})\, d\mathbf{y} \tag{2.17}$$

where $u(\mathbf{L}, \mathbf{y})$ is the utility function using the KLD between the prior and posterior probability distribution of the response QoI $\mathbf{z}_k$ introduced in Eq. (2.8). KLD measures the amount of information added to the prior after measuring the data. This scalar value shows the usefulness of the experimental design. Maximizing this measure would result in the OSP. In [50], it is shown that Eq. (2.17) can be rewritten in terms of the change in information entropy after measuring the data by

$$U(\mathbf{L}) = H_{z,prior} - \int_{\Upsilon} H_z(\mathbf{L}, \mathbf{y}|\boldsymbol{\mathcal{D}})\, p(\mathbf{y}|\mathbf{L})\, d\mathbf{y} \tag{2.18}$$

where $H_{z,prior}$ is the information entropy before measuring data and $H_z(\mathbf{L}, \mathbf{y}|\boldsymbol{\mathcal{D}})$ is information entropy after measuring data. Following the fact that the QoI $\mathbf{z}_k$ is Gaussian distributed with covariance matrix $\boldsymbol{\Sigma}_z(\mathbf{L})$ in Eq. (2.10) that does not depend on the data $\mathbf{y}$, the information entropy for the response QoI $\mathbf{z}_k$ takes the form [32]:

$$H_z(\mathbf{L}, \mathbf{y}|\boldsymbol{\mathcal{D}}) = H_z(\mathbf{L}|\boldsymbol{\mathcal{D}}) = \frac{1}{2} N_z \left[\ln(2\pi) + 1\right] + \frac{1}{2} \ln\left(\det\left(\boldsymbol{\Sigma}_z(\mathbf{L})\right)\right) \tag{2.19}$$

which is independent of the data $\mathbf{y}$. Using this independence of the posterior information entropy $H_z(\mathbf{L}, \mathbf{y}|\boldsymbol{\mathcal{D}}) = H_z(\mathbf{L}|\boldsymbol{\mathcal{D}})$ on the data $\mathbf{y}$, the integral in Eq. (2.18) simplifies to $U(\mathbf{L}) = H_{z,prior} - H_z(\mathbf{L}|\boldsymbol{\mathcal{D}})$. Considering that the prior information entropy $H_{z,prior}$ is constant and independent of the sensor configuration, minimizing posterior information entropy $H_z(\mathbf{L}|\boldsymbol{\mathcal{D}})$ is the mathematical counterpart of maximizing information gain. This study's objective is to find

a sensor configuration $\mathbf{L}$ that minimizes the posterior information entropy $H_z\left(\mathbf{L}|\boldsymbol{\mathcal{D}}\right)$ for predicting strains at desired hotspot locations.

The optimal sensor configuration assures that no other setup could result in a more informative observation. It is worth noting that the information entropy presented in Eq. (2.15) for parameter estimation and Eq. (2.19) for strain estimation do not directly depend on the sensor measurements ($\mathbf{y}_k$) but sensor locations ($\mathbf{L}$). This is ideal for the instrumentation design phase, where no measurements are yet available.

The following metric is also defined to measure the effectiveness of the OSP

$$u = \frac{H_{\max} - H_{\min}}{H_{\min}} \tag{2.20}$$

where $H_{\max}$ and $H_{\min}$ are maximum and minimum information entropies corresponding to the worst and best sensors configurations. A small value of *u* (close to zero) indicates that the best and worst sensor configurations provide a similar level of information; thus, an OSP process is not beneficial for the problem, and the higher this measure, the more beneficial the OSP algorithm.

2.2.5 Consideration of sensor configuration cost

The cost of sensor configuration is considered using an objective function with a weight factor to balance the benefit of minimizing the information entropy and the cost of sensor configuration. The cost and information entropy values are not in the same order of magnitude, and therefore their values are normalized in the objective function. Two reference terms H_{ref} and Cost_{ref} are defined as the minimum information entropy and maximum cost obtained by assuming a certain number of sensors (e.g., eight sensors in this study). The benefit term for each sensor configuration

$H(\mathbf{L})$ is normalized by H_{ref} and the cost term $\text{Cost}(\mathbf{L})$ is normalized by Cost_{ref}. The information entropy is a decreasing function of the number of the sensors, and the normalized benefit could be controlled by H_{ref} which refers to the minimum entropy for a reference number of sensors. This normalization will not affect the results and only makes the values of the weight factors more reasonable. $H(\mathbf{L})$ is used for notation brevity for posterior information entropy instead of $H(\mathbf{L}|\boldsymbol{\Sigma}_t, \boldsymbol{\mathcal{D}})$ for parameter estimation and $H_z(\mathbf{L}|\boldsymbol{\mathcal{D}})$ for strain estimation. It should be noted that cost value is always between zero and one, and benefit values are always positive.

$$\begin{cases} b(\mathbf{L}) = \left(H(\mathbf{L}) - H_{ref}\right) / H_{ref} \\ c(\mathbf{L}) = \text{Cost}(\mathbf{L}) / \text{Cost}_{ref} \end{cases} \tag{2.21}$$

where sensor configuration cost is defined by a cost function, including sensor cost, which is constant, and installation costs which is variable for different locations, a non-uniform installation cost is considered for different locations, as is the case for OWTs. The cost-benefit objective function is defined in Eq. (2.22). The weight factor *w* in this equation defines the relative importance of information entropy versus cost.

$$\begin{cases} \text{f}(w, \mathbf{L}) = w \times b(\mathbf{L}) + (1-w) \times \text{c}(\mathbf{L}) \\ \mathbf{L}^* = \arg\min_{\mathbf{L}} \left[\text{f}(w, \mathbf{L})\right] \end{cases} \tag{2.22}$$

A Pareto front study is performed considering varying weights between cost and information entropy. An advantage of the proposed formulation is that it offers the ability to use approximate numerical methods over an expensive exhaustive search. A simplified diagram for the OSP framework is presented in Figure 2.1.

2.2.6 Computational algorithm for OSP

Consider a structure with N_d DOFs that is going to be instrumented with N_o sensors. The total number of sensor configurations on this structure is

$$\binom{N_d}{N_o} = \frac{N_d!}{N_o!(N_d - N_o)!} \tag{2.23}$$

which for practical cases is a considerably large number. Finding the optimal sensor configuration among this set requires an exhaustive search, which is computationally expensive . There are two computationally efficient approximate alternative approaches to solve this problem, namely the FSSP and the BSSP algorithms, which provide near-optimal solutions. In the FSSP approach, the sensors' position is chosen sequentially by placing one sensor in the location that results in the highest decrease in information entropy. Specifically, given that *i*-1 sensors have been placed on the structure, the position for the i^{th} sensor is determined among $N_d - i + 1$ possible locations. Starting with i=1, this process continues until the total number of desired sensors is reached. This approach can also be used in a backward manner (BSSP). In the application of BSSP, the algorithm starts with N_d sensors placed at all of the structure's DOFs, and then sensors are removed one by one from the location resulting in the smallest increase of the information entropy until the desired number of sensors is achieved. It is worth noting that the FSSP and BSSP only provide the optimal solution when the optimal sensor configuration for an *i*-sensor configuration is a subset of an *i*+1-sensor configuration. However, the resultant information entropy using these algorithms is demonstrated to be close to the absolute optimal sensor configuration [35]. The FSSP

approach needs $\sum_{i=1}^{N_o}(N_d - i + 1)$ optimization iterations, and BSSP needs $\sum_{i=1}^{N_d - N_o + 1}(N_d - i + 1)$ optimization iterations, which are much smaller than the number required by the exhaustive search.

In this chapter, a combined SSP approach is used for OSP. In this approach, for a given number of sensors, a simultaneous FSSP and BSSP search is done. Later, the information entropies for the resultant configurations are compared, and the configuration with lower information entropy is chosen as the optimal configuration. The combined SSP approach is found to perform better than FSSP or BSSP and provides very accurate results which are close to those obtained by exhaustive search.

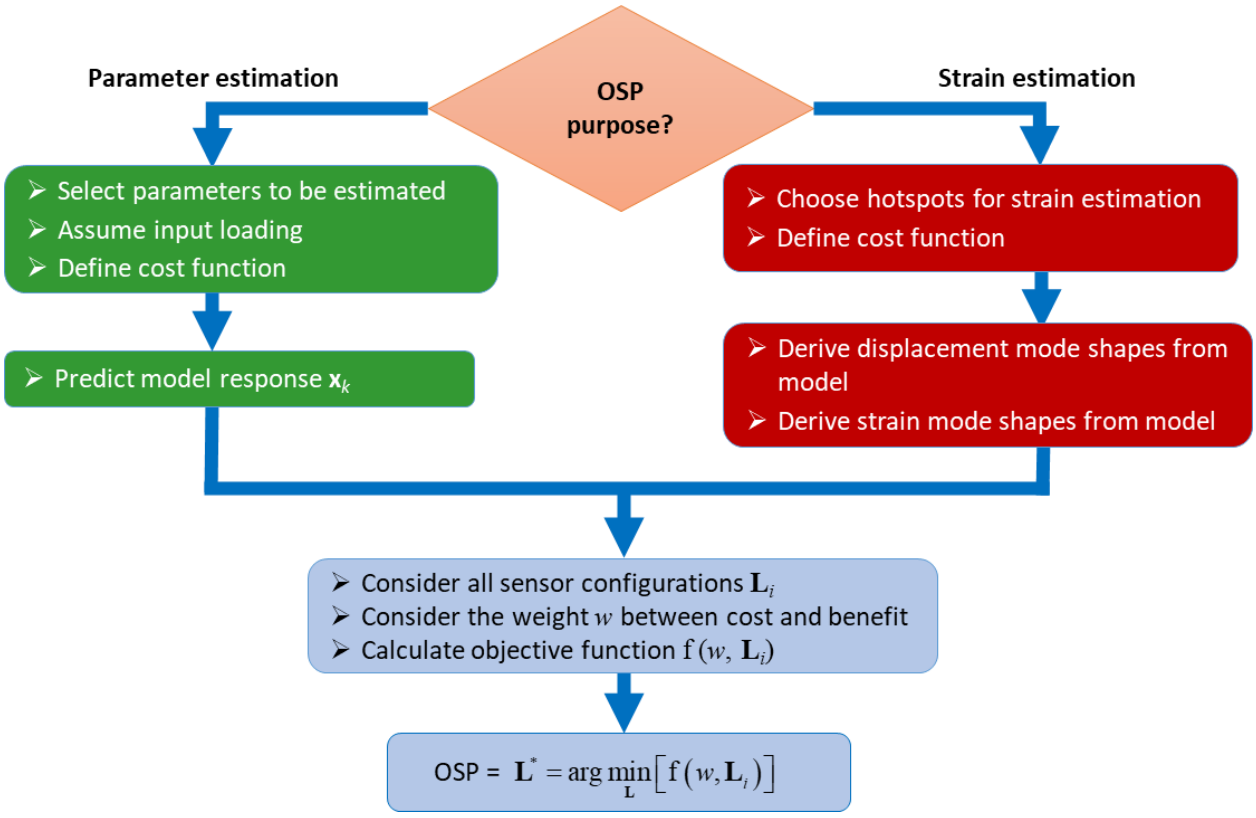


Figure 2.1 The OSP procedure framework

2.3 Description of the OWT Model

The case study structure in this chapter is a numerical model of an offshore wind turbine on a jacket support structure based on a real offshore wind turbine. A linear elastic FE model of this wind turbine was created using the structural analysis software OpenSees [51]. The tower was modeled using 32 beam elements with varying cross-sections, and an assumed concentrated mass of 500 tons at the top representing the rotor-nacelle assembly (RNA). The masses of tower and jacket elements were lumped at their end nodes. The modulus of elasticity was assumed to be 200 GPa for the steel material, which was used for all elements. A rotational mass of 1.8×10^7 kg-m^2 was assigned at the top of the tower for RNA mass moment of inertia about X and Y axes (see Figure 2.2 for axis orientations). The soil-structure interaction was modeled using three linear springs at the bottom of the four jacket legs, i.e., k_L, k_V, and k_R. The lateral stiffness of soil constraint (k_L) was assumed to be the same in X and Y directions, and the vertical stiffness (k_V) and rotational constraint (k_R) are assumed to be the same for four legs.

The presented OSP framework is applied on a realistic but simple numerical model to demonstrate the OSP process on an OWT. In the simplification of the model, the following OWT complexities are not considered: asymmetry of RNA, the effect of the rotor on dynamics of the structure, distributed wind load along the height of the structure, wave loads, and wave/wind misalignment. However, the framework can be applied to a more detailed model and loading conditions.

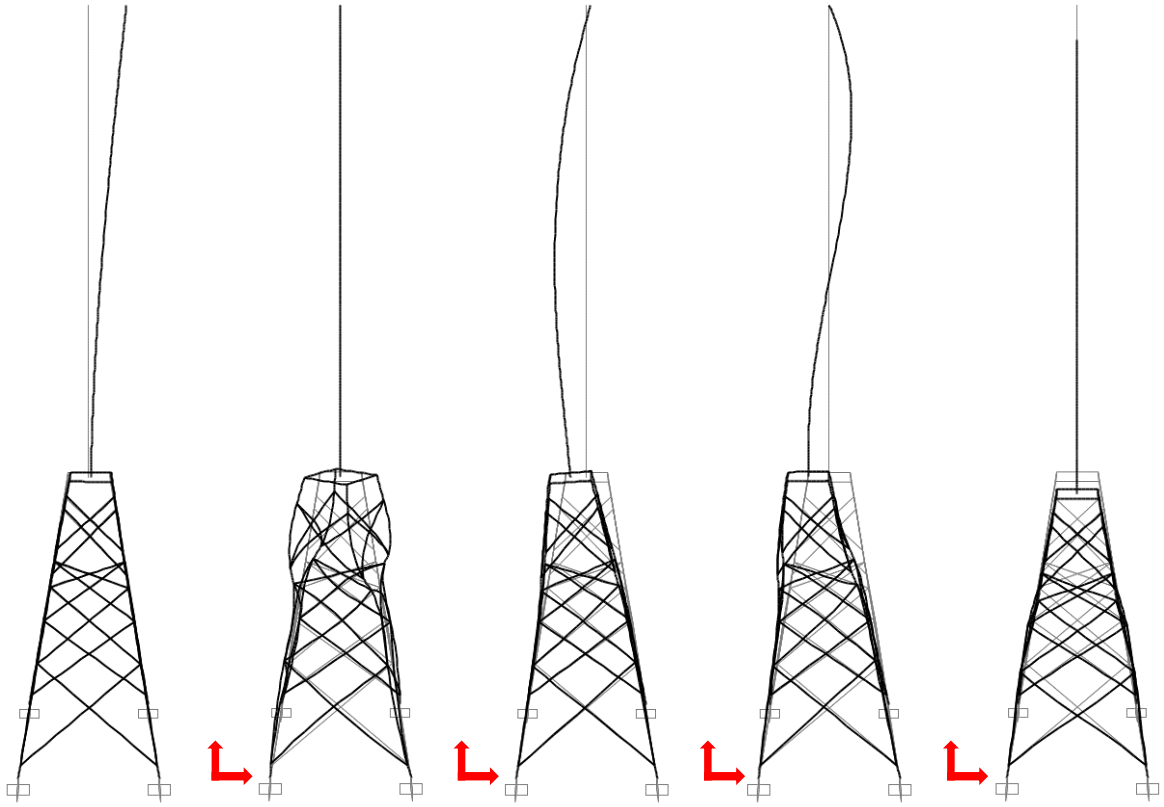
Figure 2.2 shows the natural frequencies and mode shapes of the OWT. Due to the symmetry of the structure, modes 1 & 2 and 4 & 5 have the same natural frequencies and similar mode shapes

in X and Y directions, respectively, while the third mode is torsional. All mode shapes except for mode 3 have relatively large deformations at mid-height of the tower.

In the application of OSP for parameter estimation, the following considerations have been taken into account for the selection of updating model parameters: (1) the number of updating parameters should not get too large, so the inverse updating problem is observable, (2) model parameters with considerable prior uncertainty such as foundation stiffness should be considered, (3) parameters should have sensitivity to measurements, and (4) selected updating parameters must be able to compensate for unmodeled components and modeling errors.

A sensitivity analysis is performed on the three foundation springs (k_L, k_V, and k_R) to study their influence on dynamic properties (natural frequencies) of the OWT. Figure 2.3 shows the sensitivity of natural frequencies of modes 1, 3, 5, and 6 with respect to the soil springs. The values of springs have been normalized to their initial values, which are 42.66 GN/m for k_L & k_V and 136.04 GN-m/rad for k_R. These initial values are based on soil constraints used for an offshore wind turbine in the North Sea [36]. It can be observed that only the higher modes (fifth and sixth) natural frequencies are affected by the vertical spring k_V, while k_L and k_R have little effect on any of the vibration modes considered. Therefore, only vertical spring stiffness for the foundation is considered as an updating parameter.

Mode 1 & 2 Mode 3 Mode 4 & 5 Mode 6 & 7 Mode 8



0.28 Hz 1.99 Hz 2.30 Hz 4.46 Hz 4.90 Hz

Figure 2.2 The contributing modes of the structure

Furthermore, the OWT structure is divided into two substructures with two updating parameters representing their moduli of elasticity: E_{jacket} for all elements in the jacket and E_{tower} for the tower. The moduli of elasticity are influential on the dynamic response and can potentially compensate for modeling errors at different parts of the structure. Therefore, the final updating parameters are $\boldsymbol{\theta} = [E_{jacket}, E_{tower}, k_v]^{\mathrm{T}}$.

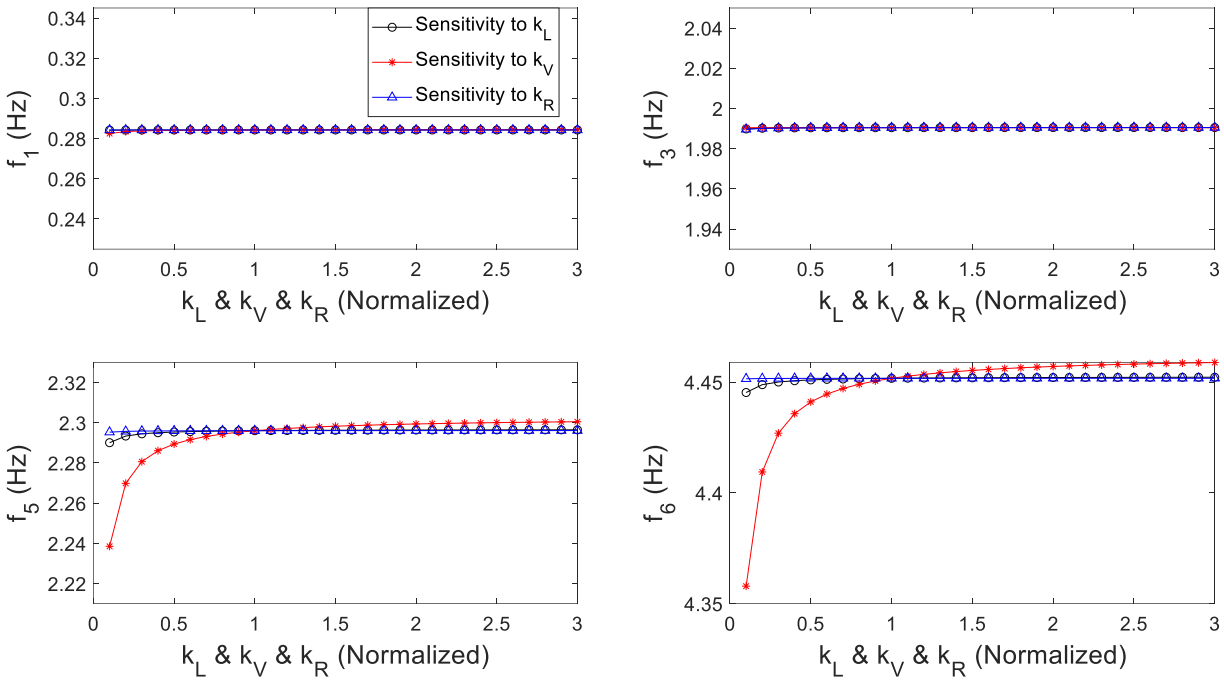


Figure 2.3 Sensitivity analysis results with respect to the stiffness of soil constraints

2.4 OSP Results for Parameter Estimation

This section describes the results of OSP for parameter estimation for the considered OWT and studies the effects of data correlation, input loads, and numerical optimization algorithms on OSP results. Due to the symmetry of the structure, only X-direction DOFs are considered as possible sensor measurements. This assumption reduces the computational cost, especially for the exhaustive search.

2.4.1 Effects of Data Correlation

In this section, the importance of considering the spatial correlation between measurements caused by prediction errors on the OSP result is studied. Results of OSP are presented and

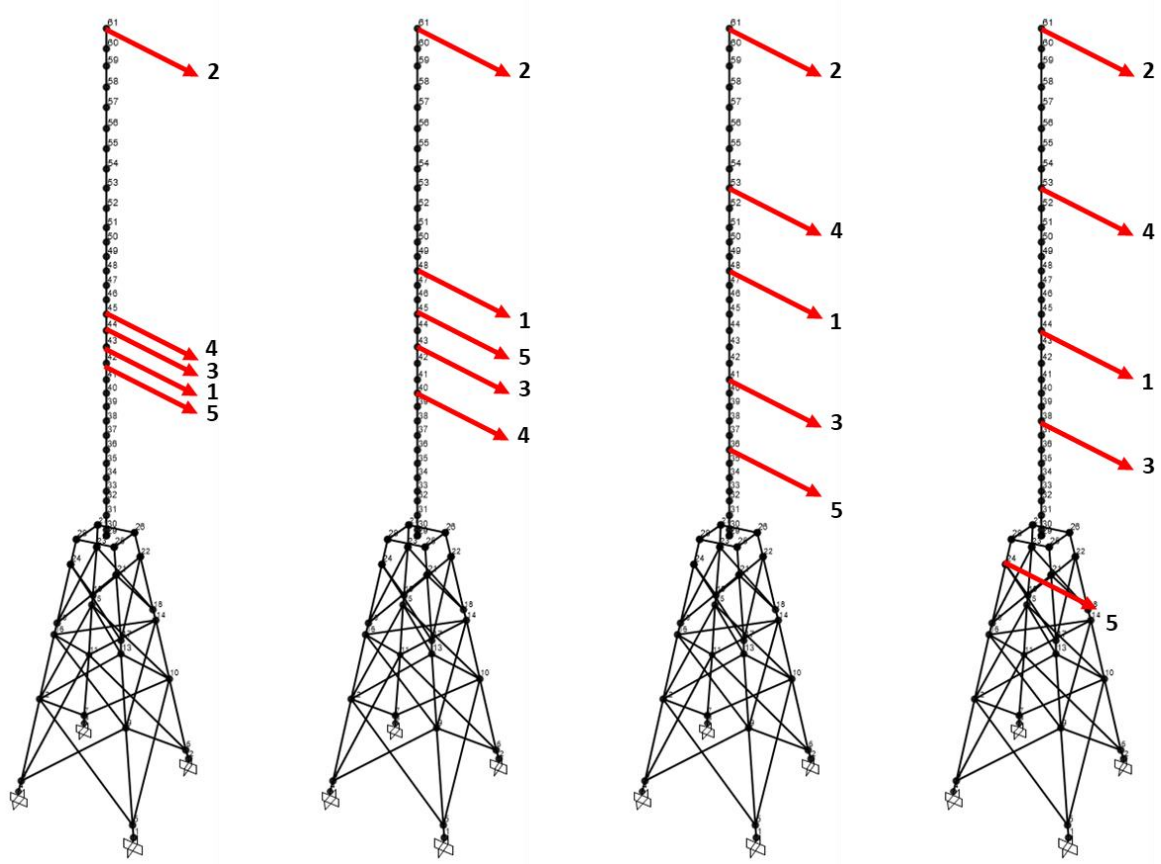
compared for two cases where data correlation is considered or not. The prior distribution of updating parameters is assumed to be uniform.

The OSP is performed for estimating $\boldsymbol{\theta} = [E_{jacket}, E_{tower}, k_v]^{\mathrm{T}}$ considering up to ten sensors. Due to the computational efficiency of SSP and its accuracy, this numerical approach is used to solve the optimization problem iteratively. The sensors' order is an indicator of their relative importance in gaining information and reducing uncertainty about the updating parameters. Figure 2.4a shows the OSP result for the case when no data correlation is considered. The results show that 4 out of 5 sensors are clustered in the middle of the tower, which is justified considering the structure's dominant mode shapes have larger amplitudes in the middle of the tower, as shown in Figure 2.2. However, sensors placed at adjacent nodes of the finite element mesh may provide redundant information. This approach clearly ignores the correlation between the data acquired in neighboring locations, which is not a realistic assumption.

In order to address this issue, a spatial correlation length of prediction errors may be considered. The spatial correlation length can be accounted for in the covariance matrix of the total prediction error, as shown in Eq. (2.13). To investigate the effect of correlation length, given the number of $N_o = 5$ sensors, OSP analysis is performed for $\lambda = 2,5,10$. The correlation length controls the correlation among neighboring DOFs, and a larger correlation length indicates a correlation between farther DOFs, and therefore, the optimal sensors' locations would be more spread out. The results for $\lambda = 2,5,10$ are shown in Figure 2.4 b-d, respectively. The red arrows in all the corresponding figures in this chapter are an indicator of the location and direction of the measurements. It can be seen that the optimal sensor locations get farther away from each other by increasing the correlation length measure. While the correlation between sensors exists, a large

correlation length may also be unrealistic. In the lack of measurement correlation consideration, OSP results depend on the finite element discretization as the sensors would be placed in adjacent nodes.

Figure 2.5 shows the information entropy versus the number of sensors for different spatial correlation lengths. It can be seen that more than 80 percent of the information entropy reduction can be obtained by optimally placing six sensors, and the reduction becomes less significant as the number of sensors increases. Moreover, increasing the spatial correlation length results in higher information entropy. This increase in entropy is reasonable since the correlation length adds a constraint in sensor placement and reduces the total amount of information by implying that some measurements are correlated.



(a) (b) (c) (d)

Figure 2.4 OSP results: (a) no correlation is considered; (b) $\lambda = 2$; (c) $\lambda = 5$; and (d) $\lambda = 10$

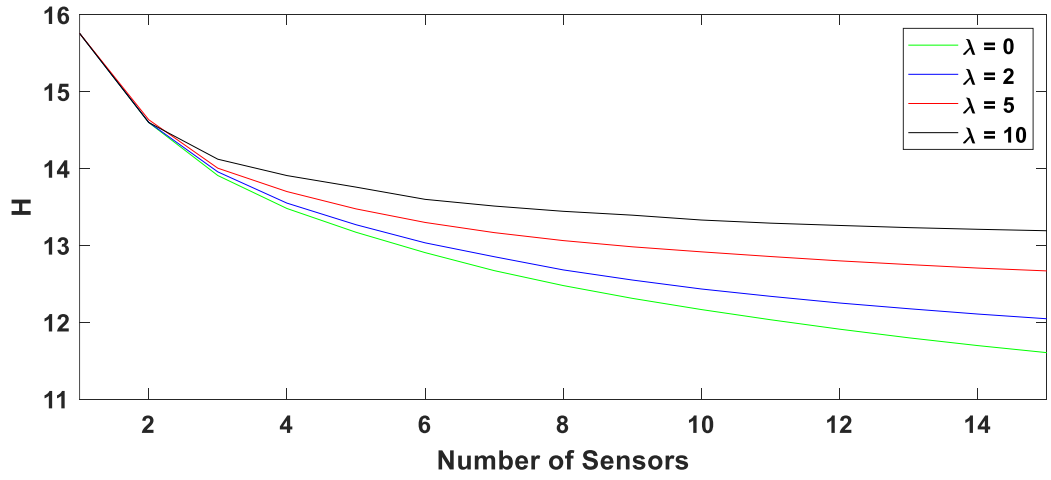


Figure 2.5 Relative information entropy comparison

2.4.2 Effects of Input Loads

OSP for parameter estimation in this chapter is formulated assuming known input excitation. The input load is required to simulate the model-predicted response ($\mathbf{x}_k$) in Eq. (2.16). In this section, the sensitivity of OSP results to different input load assumptions is demonstrated. Two different loading conditions on the OWT are considered: (1) only lateral load and (2) a combination of lateral and torsional loads. Both loads are assumed as resultant point loads and applied at the top of the structure (nacelle level). The Kaimal spectrum [52] is used as an approximation of the wind power spectrum. Figure 2.6 shows the Kaimal spectrum with a mean wind speed of 10 m/s and a standard deviation of 4. The simulated wind load time history is depicted in Figure 2.7. The torsional load is also assumed to follow the same Kaimal spectrum, and its time history is simulated using the lateral load time history multiplied by a factor of 4.

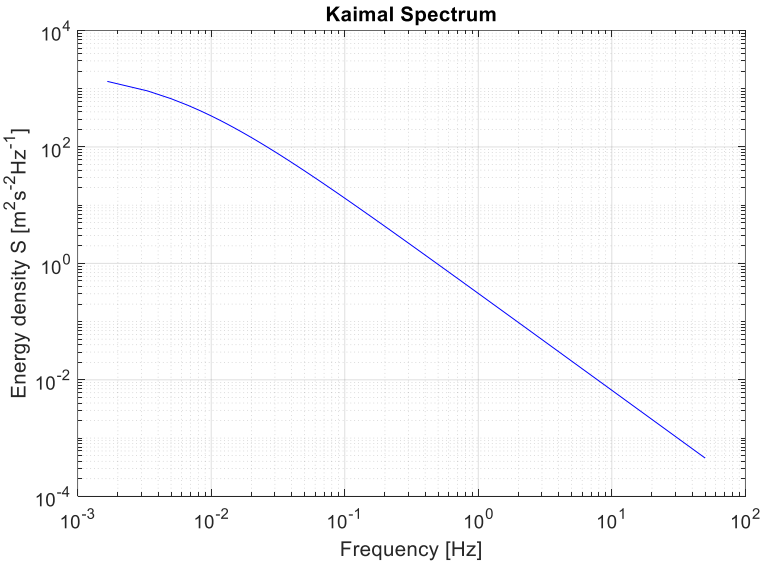


Figure 2.6 The Kaimal spectrum

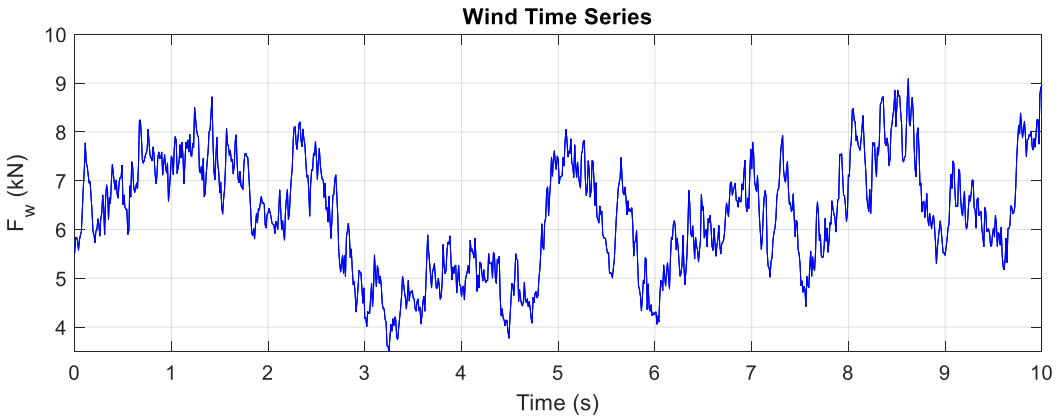


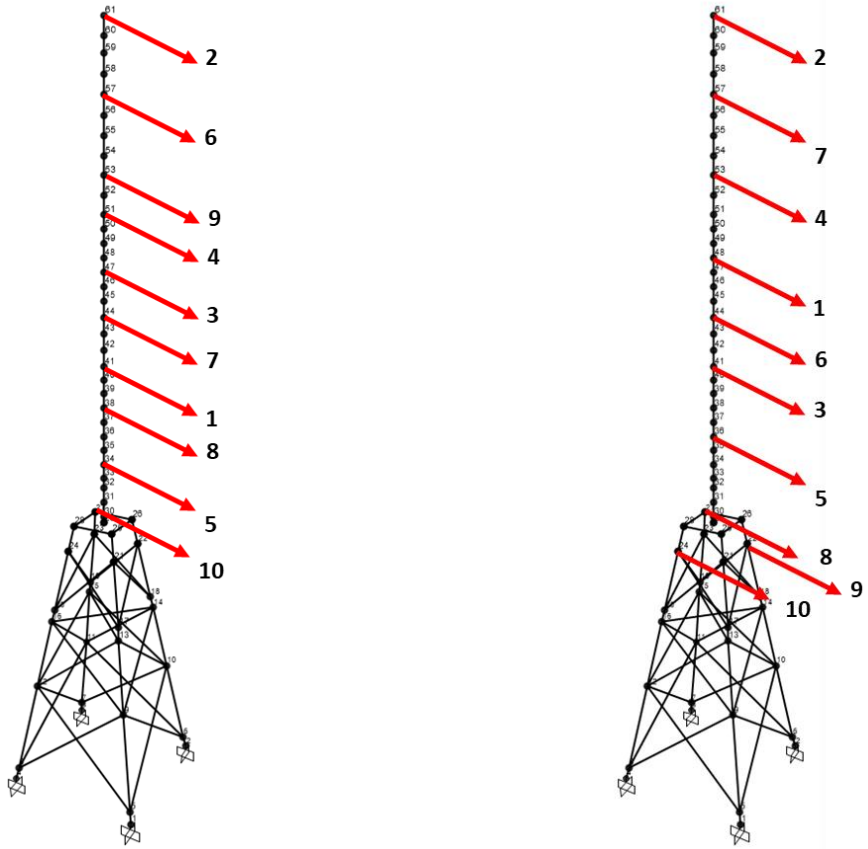
Figure 2.7 Time history of the lateral wind load

The analysis is performed assuming $N_o = 10$ (number of sensors) and $\lambda = 5$ (correlation length). The results are shown in Figure 2.8. It can be observed that the input loads can significantly affect the optimal sensor configurations. It is important to clarify that while the exact values of the input loads are not necessary, the presence and type of load can still impact sensor placement. The key point is that the load must sufficiently excite the structure to influence the optimal sensor configuration, regardless of its specific magnitude. More sensors are placed on the jacket when the torsional load is present, which is reasonable as the torsional response of the structure is mainly represented by the torsional movement of the jacket (see mode shape 3 in Figure 2.2). Moreover, in this case study, the tower is modeled as a cantilever beam, and thus the sensor on the tower cannot capture torsional motion. The location of the first couple of sensors which are the most informative ones, varies for different input loads. This means that input loads directly affect the OSP results.

2.4.3 Comparing SSP and Exhaustive Search

There are several search algorithms available to find the OSP configuration, including FSSP, BSSP, and the exhaustive search. Only the exhaustive search method guarantees the optimal

solution as all combinations of sensor configurations are considered and compared. However, this method is computationally expensive. The alternative search method which is used in this study is the SSP method which combines the results of FSSP and BSSP methods at each iteration. In this section, results of OSP using SSP and exhaustive search are compared under the combined torsional and lateral input loads for a 1-sensor to 5-sensor configurations. A spatial correlation length of $\lambda = 5$ is assumed. Figure 2.9 depicts the sorted information entropy of all configurations using the exhaustive search for a total of 4.613029×10^6 configurations along the X-axis. The red dots in Figure 2.9 show the information entropy of the optimal configuration for all the configurations of 1-5 sensors. In this plot, it can be observed that entropy can be drastically reduced by changing sensor locations and thus highlighting the value of OSP. Increasing the number of sensors results in lower optimal information entropy, which is expected. However, the information entropy for optimal configuration of 3, 4, and 5 sensors are very close. Figure 2.10 presents the OSP results for parameter estimation for 1-5 sensor configurations. The results show that the information provided by only lateral measurements is adequate for estimating the three parameters of interest. This would not be the case if only torsional motion provided unique information about a parameter of interest. For example, if the estimation of the soil spring under only one leg or the stiffness of a single diagonal truss member of the jacket was sought after. Figure 2.11 compares the accuracy of the SSP method to the exhaustive search. In Figure 2.11, the information entropy for the best and worst configurations and for the 1-5 sensor configurations are compared using two search methods. The results are almost identical, confirming the accuracy of the SSP method.



(a) (b)

Figure 2.8 OSP results considering (a) lateral load; (b) a combination of lateral and torsional load

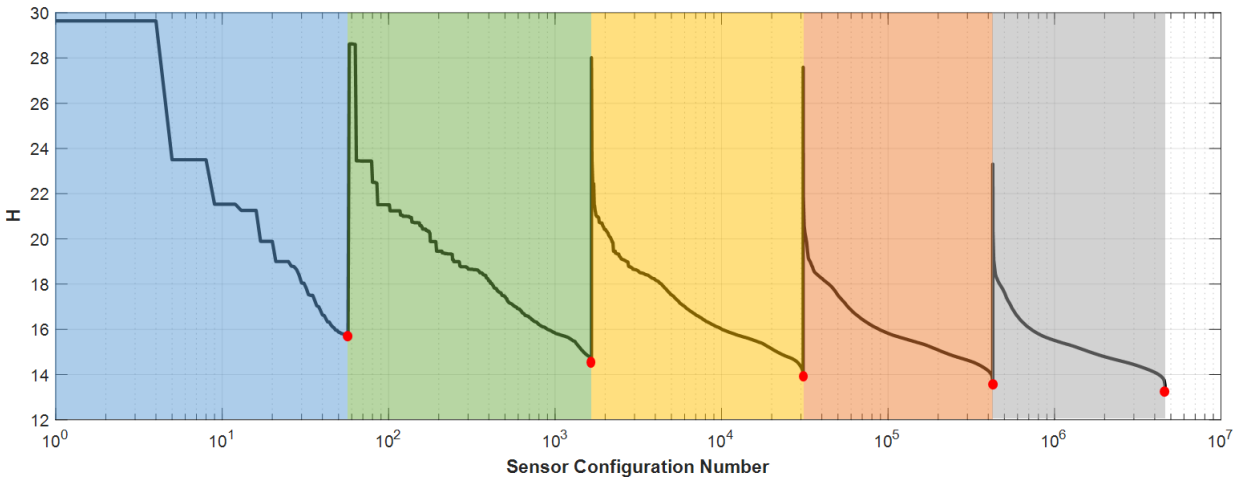
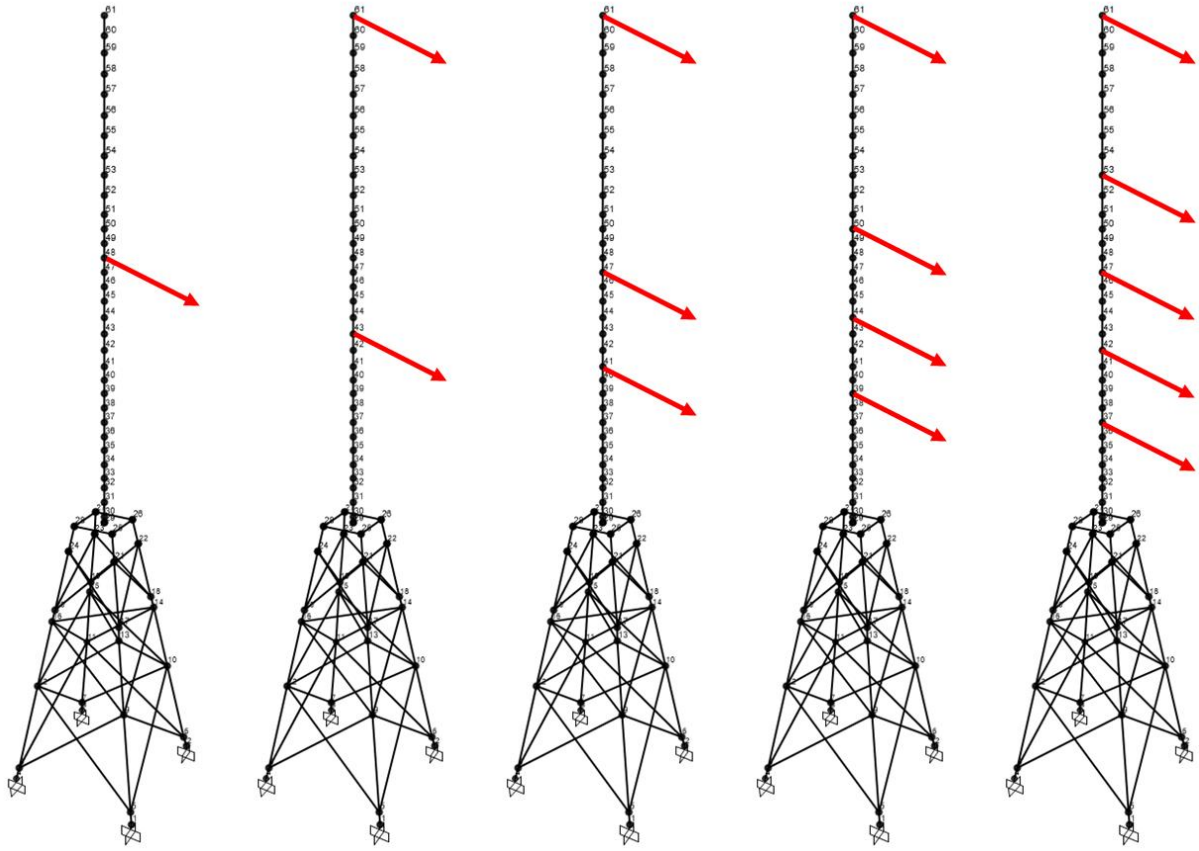
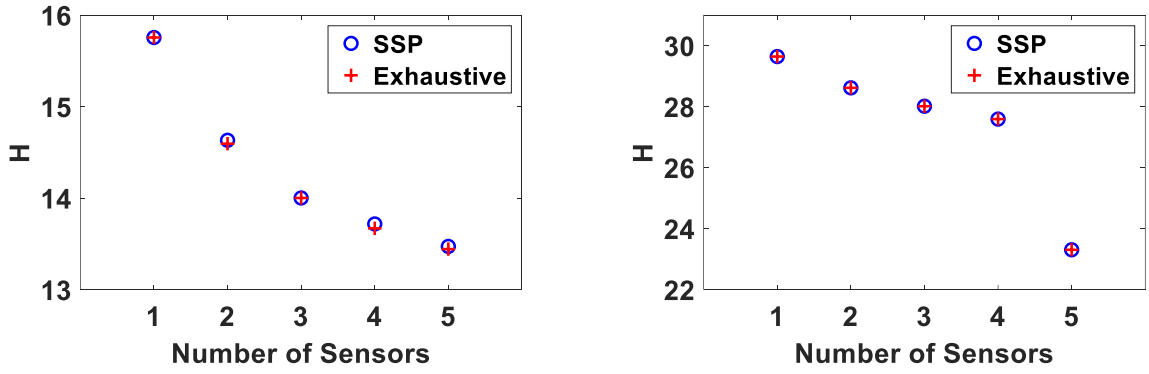


Figure 2.9 Sorted information entropy for 1-5 sensor configurations



1-sensor configuration | 2-sensor configuration | 3-sensor configuration | 4-sensor configuration | 5-sensor configuration

Figure 2.10 Optimal sensor configuration results corresponding to each specific sensor number



a) The best configuration

b) The worst configuration

Figure 2.11 Comparing information entropy of optimal configurations using SSP and exhaustive search methods

2.4.4 Cost Consideration

The previous section provided OSP results for parameter estimation without consideration of cost, where sensor locations were provided sequentially for a known number of sensors. This section provides the OSP results where sensor installation cost can be drastically different at different locations such as offshore wind turbine structures. Using Eq. (2.21), the cost and benefit of this case are calculated. The reference minimum information entropy ($H_{\min}$) is calculated by assuming 8 sensors. The reason behind this choice is the tradeoff between information entropy and cost. Assuming a higher number of sensors would force the weight factor for balancing cost and benefit to be infinitesimal. In this section, the cost-benefit objective function (Eq. 2.22) is minimized with a weight factor of $w = 0.55$. The weight factor value is chosen such that it gives a balanced result. The SSP method is implemented, and its results are compared with those from the exhaustive search. The OSP analysis is performed assuming a spatial correlation length of $\lambda = 5$ under both lateral and torsional loads.

The installation cost for one sensor is assumed to be \$10,000 at underwater locations, \$5,000 within the splash zone, and linearly increasing from \$1,000 to \$2,000 along the tower height. The sensor installation cost is shown in Figure 2.12. Furthermore, the price of each sensor is assumed to be \$500, which is constant and is added to the installation cost. Figure 2.13 shows the cost-benefit objective function of all 4.613029×10^6 candidate sensor configurations using exhaustive search, where the results are grouped based on the number of the sensors in each candidate configuration and sorted in ascending order based on the value of the objective function. It can be seen that the objective function (f) of Eq. (2.22) within each group of sensors can be reduced drastically by sensor placement, which highlights the importance and value of OSP. The red dots

represent the best configuration within each group of sensors. The best configuration for the considered weight of w = 0.55 is when 2 sensors are used, which provides the lowest value of the objective function. It is worth noting that the results are sensitive to the weight between cost and benefit, w, in the objective function. Figure 2.14 shows the optimum sensor configurations for each number of sensors. It can be observed that cost consideration slightly moves the optimal locations closer to the bottom of the tower where the installation cost is low. A smaller weight factor would reduce the weight on information entropy and increase the weight on cost, which results in sensors being pulled down until all of them are placed in the minimum cost locations for the extreme case of $w = 0$. The accuracy of the OSP results from SSP and the exhaustive search is compared in Figure 2.15. The results confirm the previous conclusion about the accuracy and efficiency of SSP for the case of using an objective function with cost consideration.

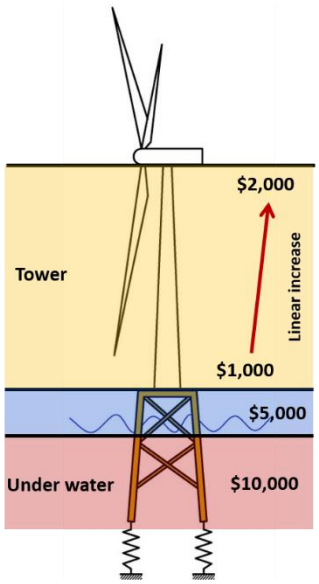


Figure 2.12 Sensor installation cost

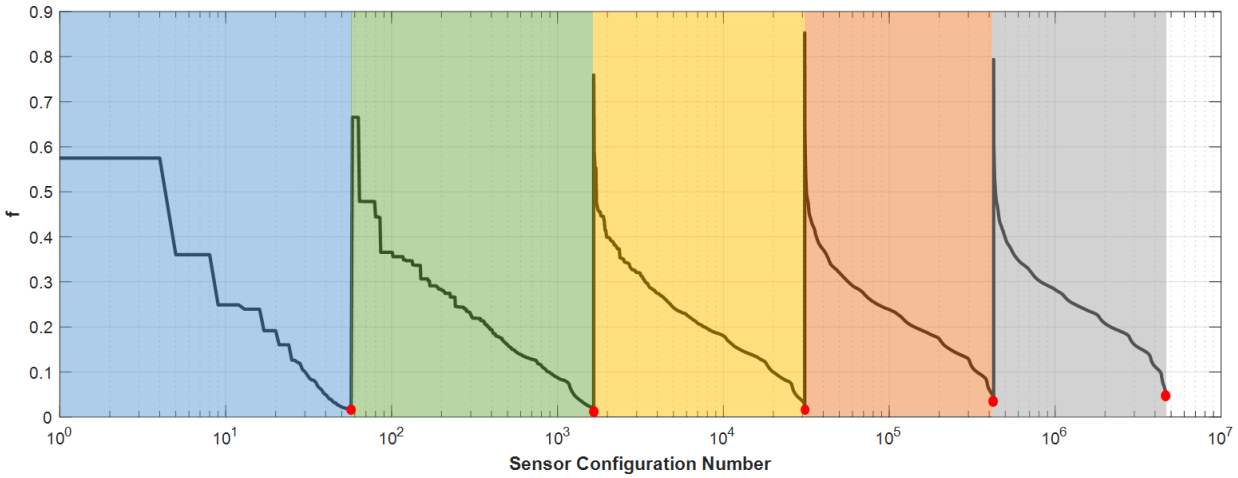
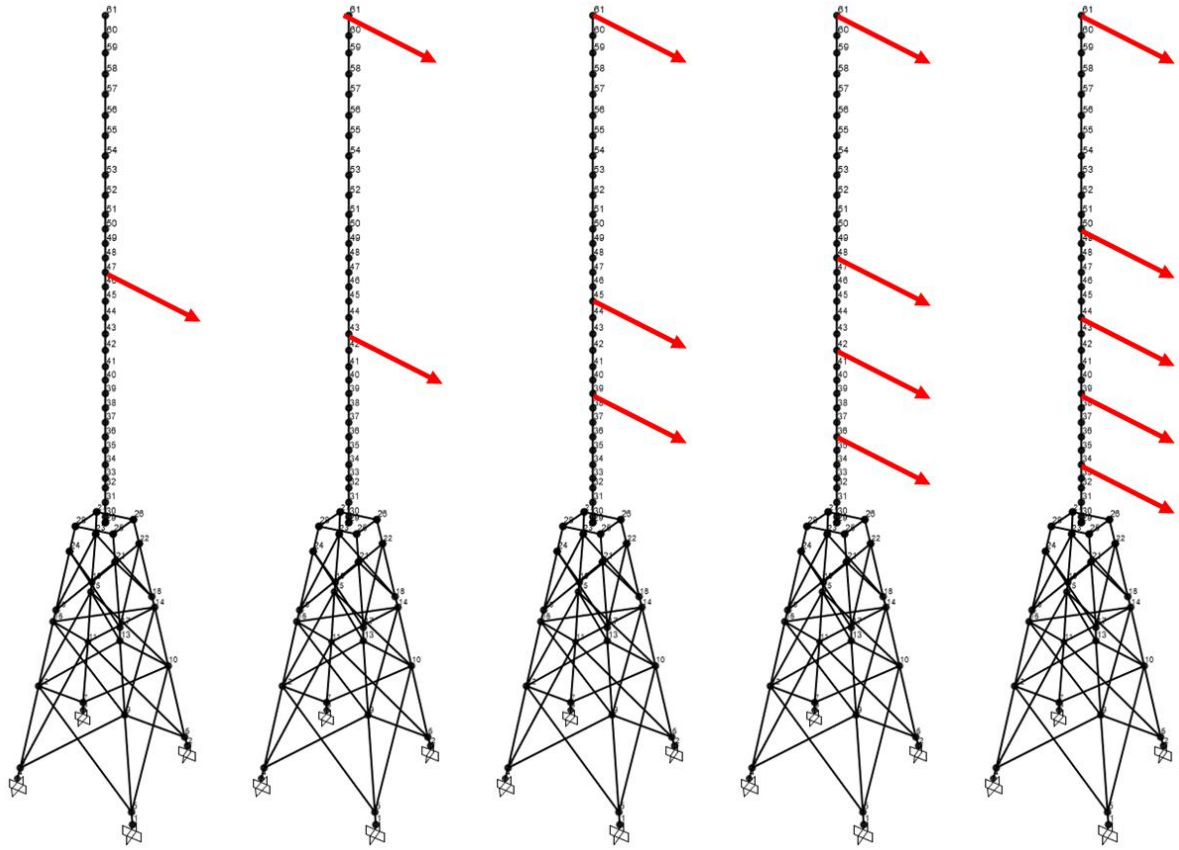
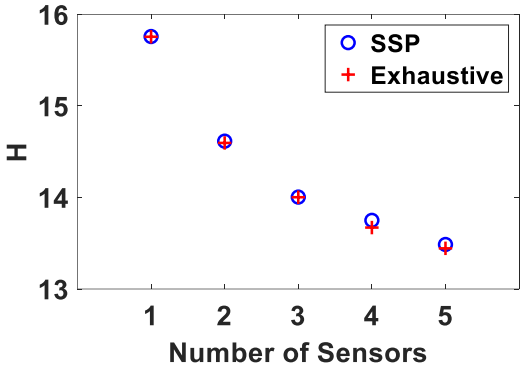


Figure 2.13 Sorted objective function f for 1-5 sensor configurations

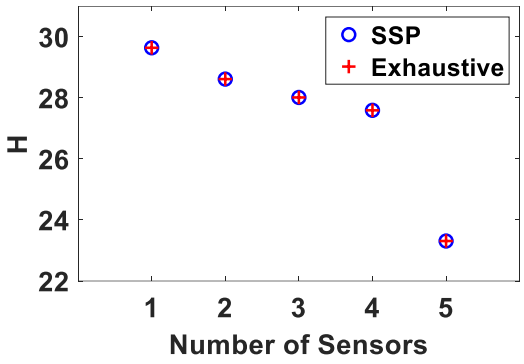


1-sensor configuration | 2-sensor configuration | 3-sensor configuration | 4-sensor configuration | 5-sensor configuration

Figure 2.14 Optimal sensor configuration results for different number of sensors with the consideration of installation cost



a) The best configuration



b) The worst configuration

Figure 2.15 Information entropy of the best and the worst configurations compared for SSP and exhaustive search

Table 2.1 compares the effectiveness of the OSP when estimating parameters using 1-sensor to 5-sensor configurations. SSP search method results are used for comparison. A u value that is more than zero shows that using this framework is beneficial.

Table 2.1 Comparing the measure of effectiveness for parameter estimation for different number of sensors

	1-sensor	2-sensor	3-sensor	4-sensor	5-sensor
u	0.88	0.96	1.00	1.01	0.73

Results of the OSP for parameter estimation are also compared with three cases of naïve/common-sense sensor placements. The three commonsense cases include: (c1) uniform along the whole structure, (c2) uniform along the height of the tower, and (c3) locations obtained by Kammer's method [19] which maximizes the linear independence of mode shapes. The first 8 modes of the structure are considered in the application of Kammer's method. The entropy values for these three cases are compared with the OSP results for parameter estimation in Figure 2.16. It can be observed that the entropy from case c2 (uniform along the tower) is close to OSP results but cases c2 and c3 provide inferior results.

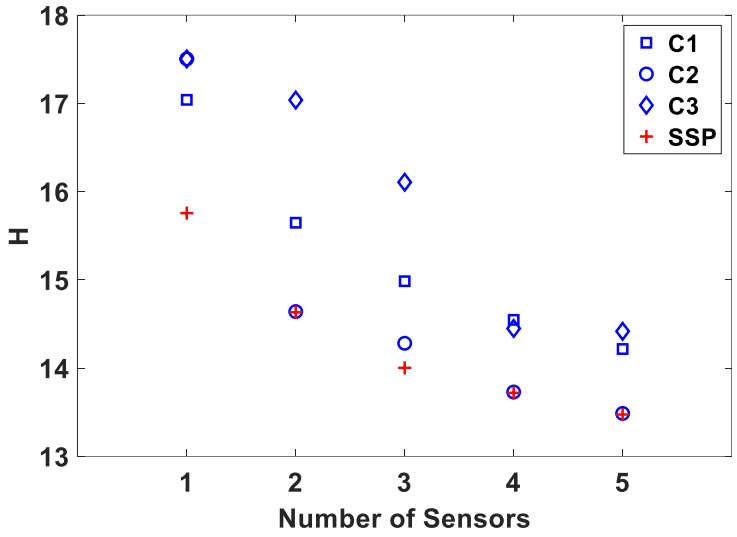
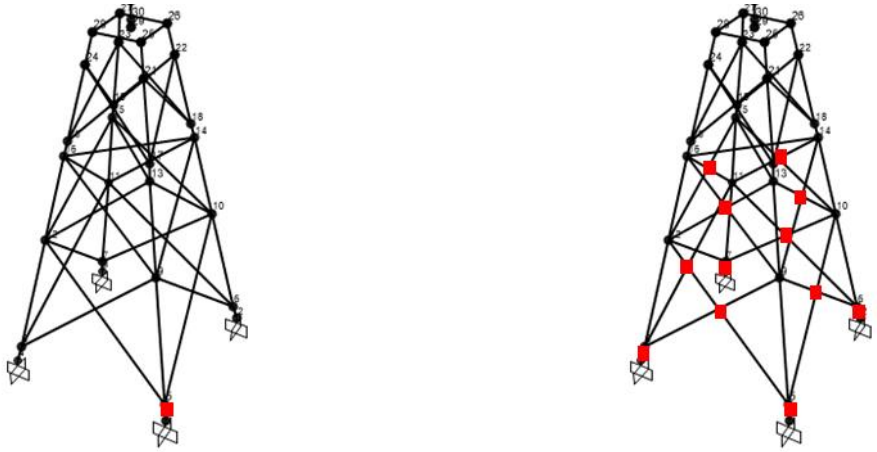


Figure 2.16 Comparing three common-sense configurations with SSP results for parameter estimation

2.5 OSP Results for Strain Estimation

Two OSP problems are studied in this section, as shown in Figure 2.17, for estimating the axial strain at different fatigue hotspots, namely (1) one location at the bottom of the jacket just above the support, since lower joints are more prone to fatigue and (2) at twelve locations, middle of eight diagonal members and four elements at the bottom of the jacket. Both problems are solved with and without cost consideration. The first five vibration modes of the structure, as shown in Figure 2.2, are considered in this study. The prior knowledge about the modal coordinate in Eq. (2.10) is assumed to be Gaussian with a large $\boldsymbol{\Sigma}_{pr} = 10^{13}$, which implies unknown prior knowledge similar to the assumption of uniform distribution. The covariance of strain prediction error is postulated as $\boldsymbol{\Sigma}_{\varepsilon} = 10^{-1}$ at all locations.



a) Problem 1 b) Problem 2

Figure 2.17 Strain estimation locations

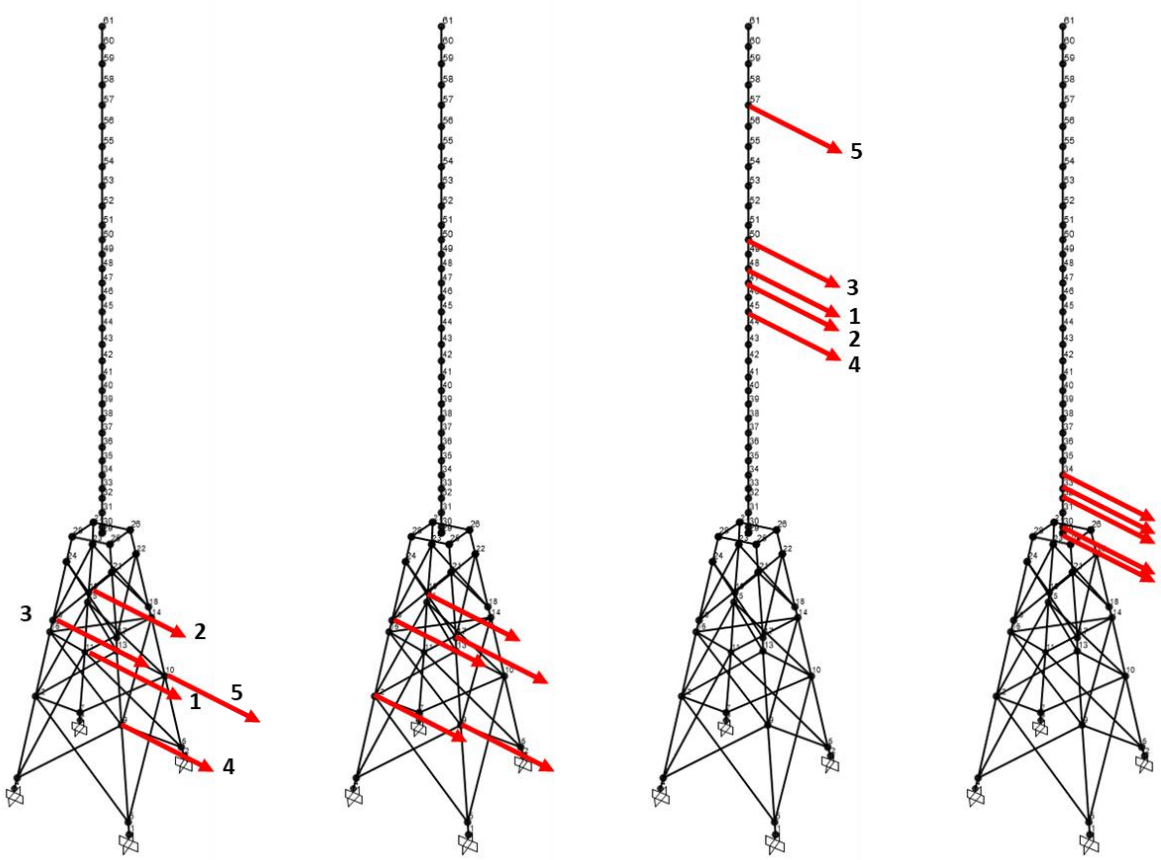
2.5.1 OSP for estimating strains at one location

2.5.1.1 Case 1: No cost consideration

Using the formulation presented in Sections 2.2.2, 2.2.3, and 2.2.4.2 of this chapter, OSP is found using the SSP algorithm and the exhaustive search for a 5-sensor configuration. Prediction error correlation is considered using a correlation length of $\lambda = 5$. Figure 2.18 depicts OSP results together with the worst sensor placements for 5-sensor configurations. The information entropy of OSP results for strain estimation using SSP and exhaustive search are compared in Figure 2.19 for the best and worst sensor configurations.

This investigation confirms the previous observation about the accuracy of the SSP method, which states that SSP provides near-optimal solutions for the OSP problem. It is worth noting that the order of placing sensors is available only when using the SSP and can be interpreted as the relative importance of each sensor. The exhaustive search does not automatically provide such ordering. In Figure 2.19, it is seen that adding the fifth sensor provides little information compared to the first four sensors. Figure 2.19 shows the accuracy of the SSP method, as well as the

usefulness of the OSP algorithm by comparing the information entropy of the best and worst configurations. Although Figure 2.18b shows that SSP and exhaustive search provide different worst sensor configurations, Figure 2.19b shows that the resultant information entropies are generally similar.

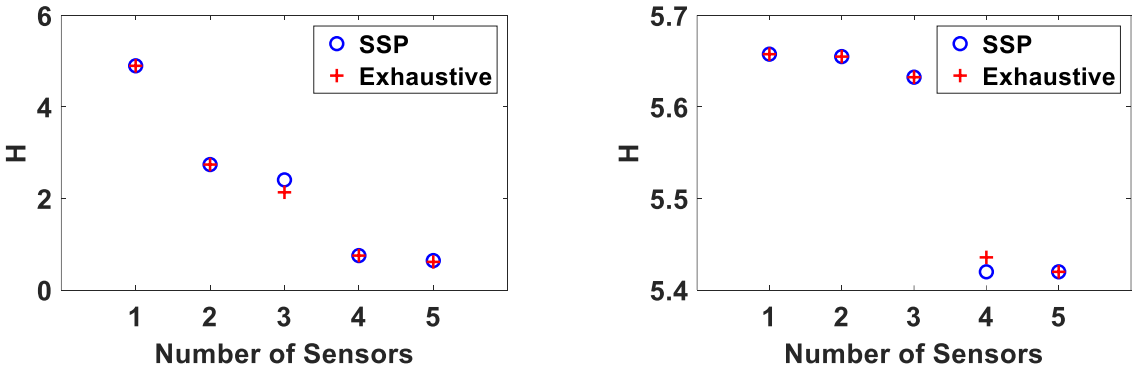


SSP **Exhaustive** **SSP** **Exhaustive**

a) The best configuration

b) The worst configuration

Figure 2.18 Comparing OSP results for strain estimation at a single hotspot using SSP and exhaustive methods



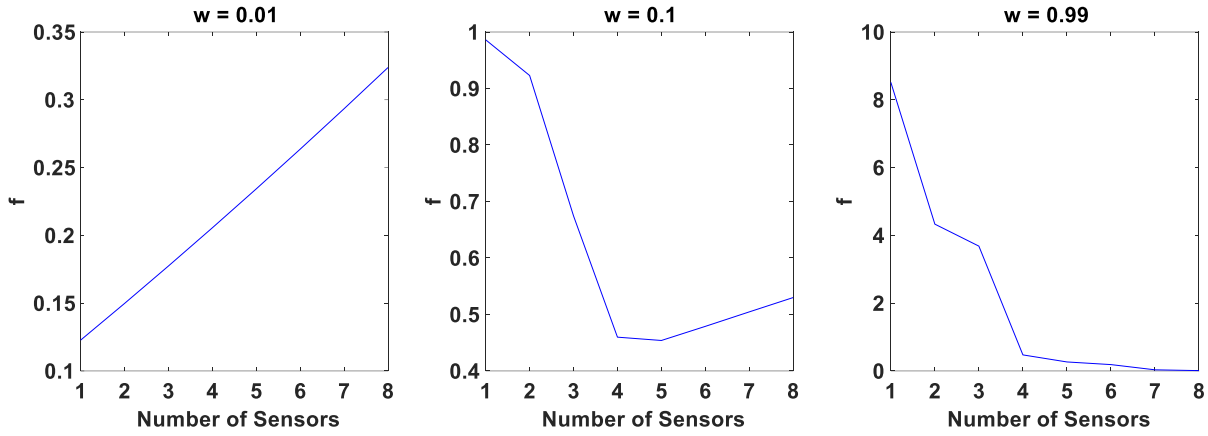
a) b)

Figure 2.19 The information entropy of OSP results for strain estimation at a single hotspot using SSP and exhaustive methods for a) the best configurations; b) the worst configurations

2.5.1.2 Case 2: OSP with cost consideration

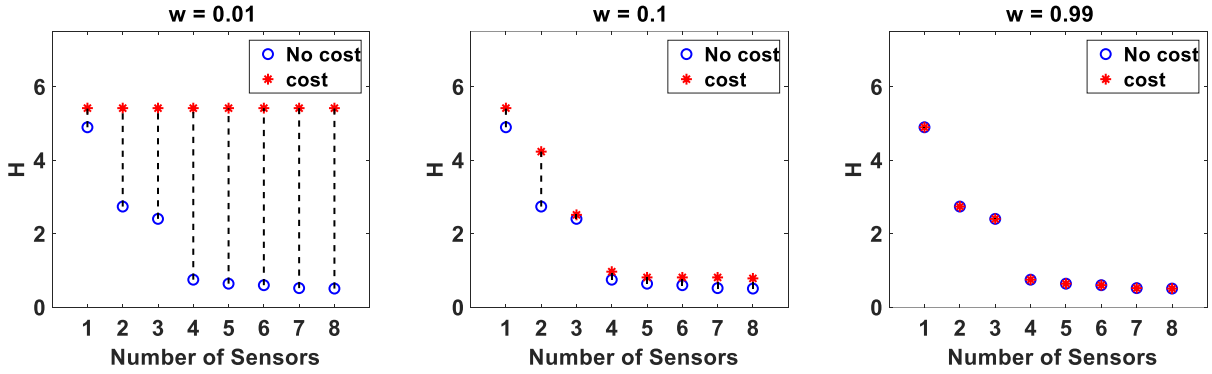
To account for the cost in the OSP, a multi-objective optimization is employed by varying the weight factor between the information entropy and cost in Eq. (2.22). Similar to section 2.4.4, the reference minimum information entropy (H_{ref}) is calculated by assuming 8 sensors. Weight $w=0$ indicates the case where the OSP result is dominated by the cost and $w=1$ is the other extreme where information entropy dominates the solution. Three weight factors of 0.01, 0.1, and 0.99 are used. These weight values are chosen to show two extreme and one balanced scenarios. Note that $w=0.55$ was a balanced case for parameter estimation. However, for strain estimation, based on the value of information entropy and cost, $w=0.1$ is found to be more balanced. In this case, 1-sensor to 8-sensor configurations are investigated. The SSP algorithm is used, and prediction error correlation is considered using a correlation length of $\lambda=5$. The values of objective functions are displayed in Figure 2.20 for different sensor numbers, and the three considered weights. As shown in Figure 2.20a, the best setup that minimizes the objective function for $w=0.01$ is a 1-sensor configuration, and the objective function is monotonically increasing by adding more sensors. This is due to the high weight of the cost component compared to the normalized benefit. For the case

of $w = 0.1$, the objective function is not monotonically increasing or decreasing and has a minimum at 5 sensors due to the balance between the values of cost and information, as shown in Figure 2.20b. Therefore, the OSP, which minimizes the objective function, has 5 sensors. The last scenario with $w = 0.99$ shows control of the information entropy over the OSP results. The objective function, in this case, is monotonically decreasing like the case where cost is not considered. The best sensor setup is an 8-sensor configuration where the objective function is minimum. Figure 2.21 shows the corresponding information entropy of the OSP designs for different sensor setups with the three considered weights of 0.01, 0.1, and 0.99 together with the case when the cost is not considered. In this figure, the dashed line provides a visual measure to show the difference between information entropy when the cost is considered and when it is not. It can be seen that this difference is the largest for $w = 0.01$ while it becomes negligible for $w = 0.99$. Furthermore, in Figure 2.21(a), where the weight of information entropy is very low, it remains almost constant by adding sensors. This is because the sensors are added in locations with the minimum cost, which in this particular structure coincide with the least informative locations (See Figure 2.18b using exhaustive search). However, adding more sensors would result in a decrease in information entropy eventually. By increasing the weight of information entropy in the objective function, the discrepancy between cost and no-cost cases diminishes. The weight factor should be chosen to maintain a balance between cost and information entropy and such that not much information is lost when the cost is considered. It is worth mentioning that for the case $w = 0.1$, the results in Figure 2.21(b) suggest that the information gained by adding more than 5 sensors at their optimal locations is not significant to justify the cost from the additional sensors.



a) b) c)

Figure 2.20 The objective function of OSP results for strain estimation using the SSP method for the best configurations



a) b) c)

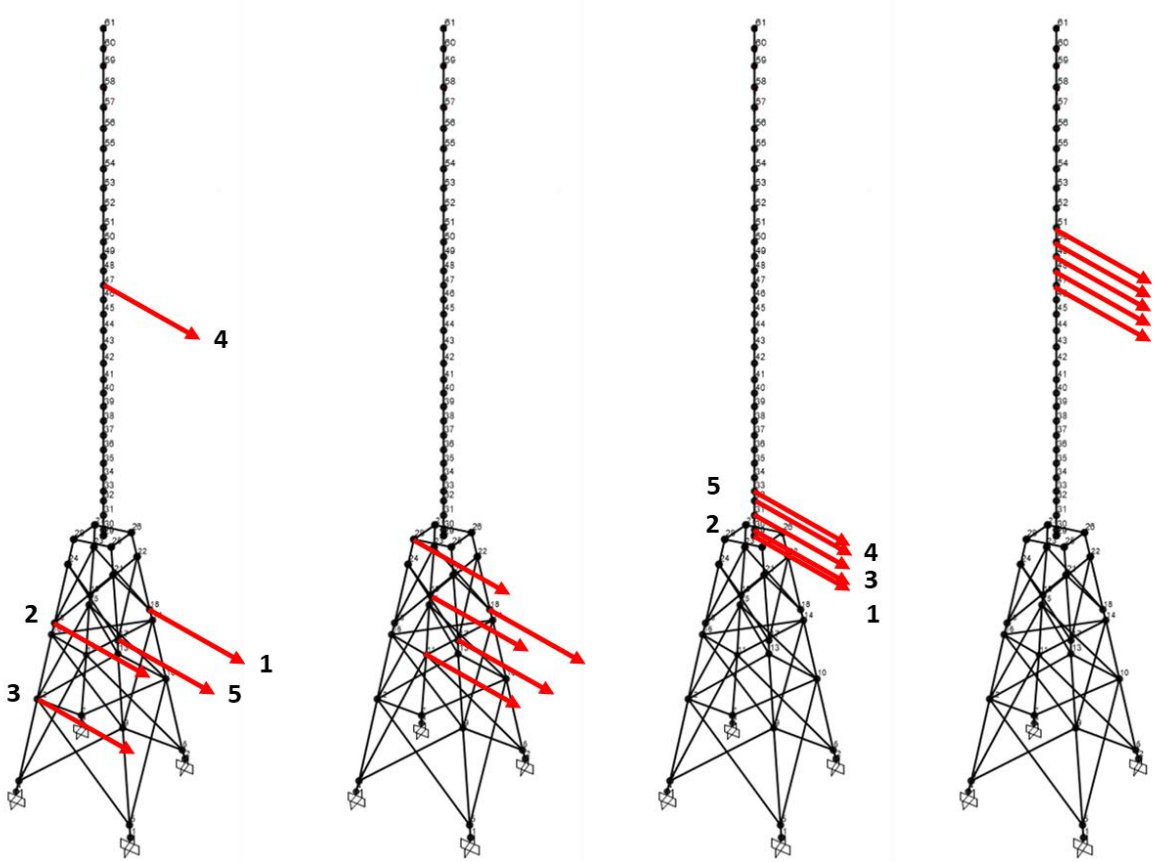
Figure 2.21 The difference in information entropy of OSP results with and without cost consideration

2.5.2 OSP for estimating strains in twelve hotspot locations

Previous studies have shown that X joints are more susceptible to fatigue compared to K joints [43]. Thus, the X joints located at the middle of eight lower members, as well as the bottom of the four jacket legs, are chosen as fatigue hotspots for strain estimation, as shown in Figure 2.17b. OSP is performed with and without cost constraint, assuming a $\lambda = 5$ correlation length and utilizing both SSP and exhaustive search algorithms.

2.5.2.1 Case 1: OSP without cost consideration

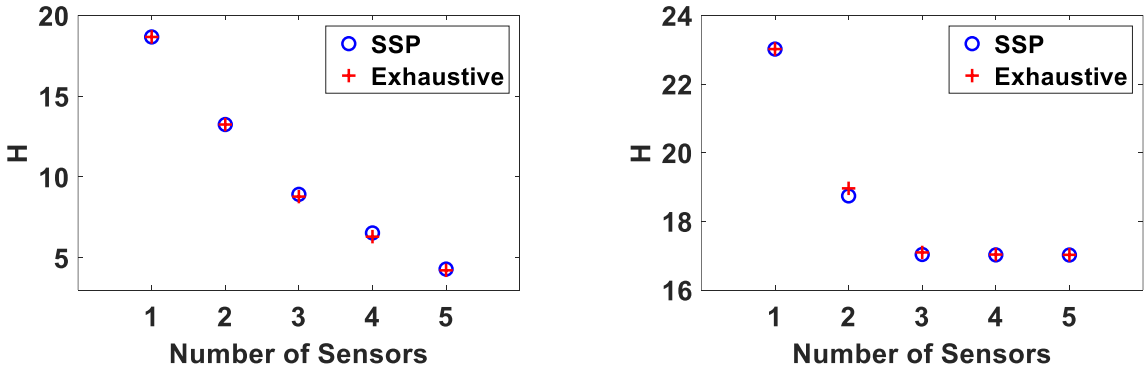
The same framework as that of Section 2.5.1 is used here. Figure 2.22 illustrates the best and worst configurations for a 5-sensor configuration. There are some discrepancies in the location of sensors for both the best and the worst cases using the two search methods. However, the information entropy values shown in Figure 2.23 are very similar. It can be seen that the differences between the two methods are minor in all cases, and the SSP is preferred due to its lower computational cost.



SSP **Exhaustive** **SSP** **Exhaustive**

a) The best configuration b) The worst configuration

Figure 2.22 Comparing results of OSP for strain estimation at 12 hotspots using SSP and exhaustive methods



a) The best configurations

b) The worst configurations

Figure 2.23 The information entropy of OSP results for strain estimation at 12 hotspots using SSP and exhaustive methods for a) the best configurations; b) the worst configurations

Table 2.2 compares the effectiveness of the OSP when estimating strains at one location and 12 locations using 1-sensor to 5-sensor configurations. SSP search method results are used for comparison. Table 2.2 shows that by increasing the number of strain estimation locations, the effectiveness of OSP decreases, especially for the cases with a higher number of sensors. This shows the importance of using OSP when the number of strain estimation locations is small. An intuitive explanation for this observation is that when more QoIs are estimated, every DOFs of the structure potentially contains more information on some of the QoIs, and the relative importance between DOFs decreases, i.e., the effectiveness of OSP decreases.

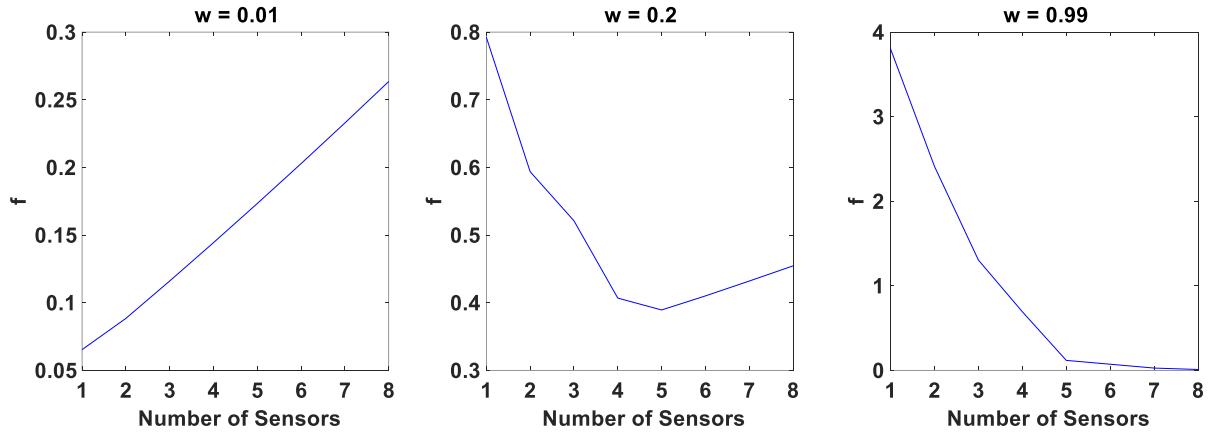
Table 2.2 Comparing the measure of effectiveness for estimating strains at one and twelve locations

	1-sensor	2-sensor	3-sensor	4-sensor	5-sensor
1 location	0.15	1.06	1.34	6.20	7.42
12 locations	0.23	0.41	0.91	1.61	2.96

2.5.2.2 OSP with cost consideration

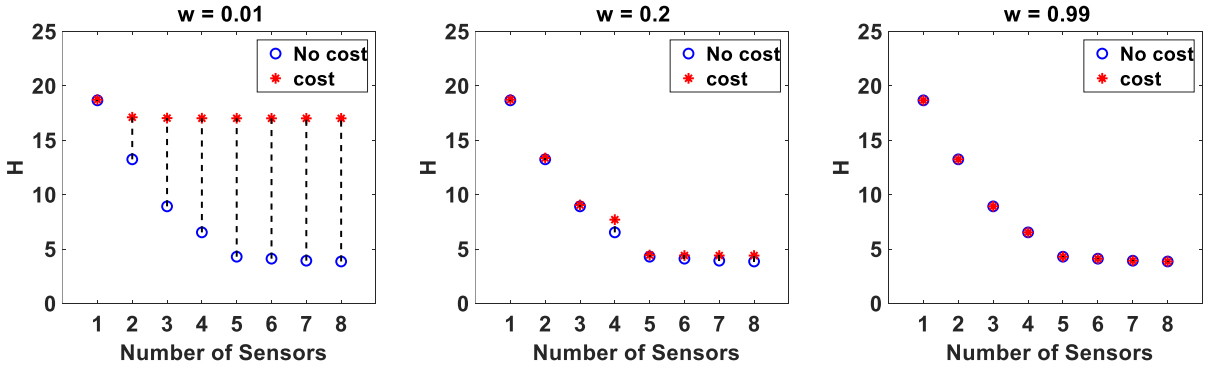
In this case, OSP for strain estimation is performed considering sensor configuration cost. Similar to sections 2.4.4 and 2.5.1, the reference minimum information entropy (H_{ref}) is calculated by assuming 8 sensors. Three weight factors of 0.01, 0.2, and 0.99 are considered in the cost-benefit objective function of Eq. (2.22). The balanced weight factor ($w = 0.2$) is chosen to give a balanced result, and it is different compared to section 5.1 ($w = 0.1$) as the information entropy, and accordingly, the normalized benefit of configurations are different. Due to the high computational cost of exhaustive search and the confirmed accuracy of SSP, only the SSP method is implemented in this section. Figure 2.24 shows the objective function (for OSP results) as a function of different number of sensors under three considered weight factors. Similar to Section

2.5.1, the best configuration for $w = 0.01$ is a 1-sensor configuration, for $w = 0.2$ is a 5-sensor configuration, and for $w = 0.99$ is an 8-sensor configuration. Figure 2.25 illustrates the difference between information entropy when the cost is considered and when the cost is not considered. As expected, the differences between the entropy of with/out cost are the most when *w* is small and become negligible for *w* = 0.99.



a) b) c)

Figure 2.24 The objective function of OSP results for strain estimation using the SSP method for the best configurations



a) b) c)

Figure 2.25 The difference in information entropy of OSP results with and without cost consideration

To further investigate the effects of weight factors, a Pareto front study is performed, and the OSP is repeated considering 50 different weight factors ranging from 0 to 1 with intervals of 0.0204 and 1-sensor to 8-sensor configurations, i.e., a total of 50×8 = 400 OSP optimizations. Figure 2.27 shows the Pareto front plots of OSP results for each number of sensors. The Pareto front plot includes solutions that are optimal, and no objective can be improved without sacrificing at least one other objective. This study is a multi-objective optimization with information entropy and cost as its objectives. The black circles correspond to OSP results using the considered 51 different weight factors. The blue line provides the values of cost and information entropy for different OSP and shows the tradeoff between them as their weight in the objective function varies. There are exactly 50 black circles in each plot; however, the information entropy and cost of some of the configurations with different weight factors are the same resulting in the black circles being overlapped. Figure 2.27 shows that by increasing the number of the sensors, as more configurations are available, there are more Pareto front solutions at different levels of entropy. For example, for 1-sensor configuration 50 different weight factors resulted in only three levels of information entropy, one high, one balanced, and one low. For 8 sensors, 8 levels are achieved. By means of these plots, the decision-maker can consider the required cost to achieve a certain level of information entropy or determine the amount of accuracy that can be achieved for a specific budget.

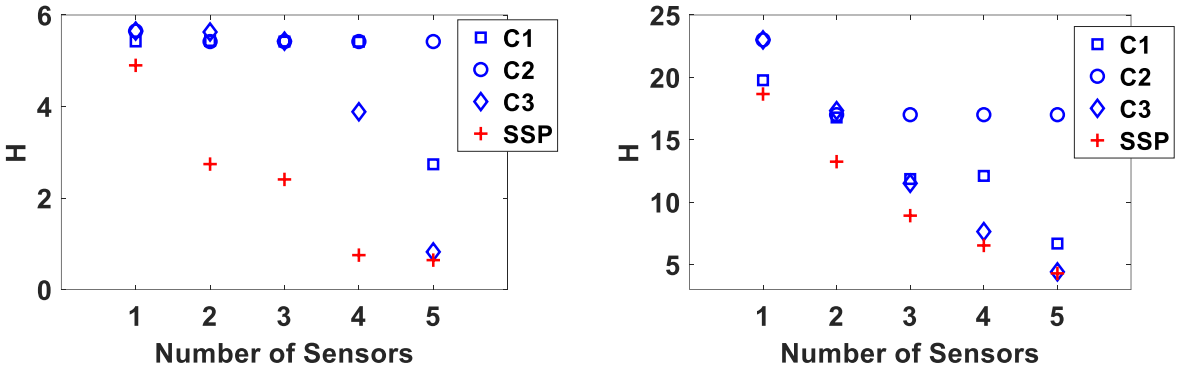
The information of the eight subplots in Figure 2.27 is presented and condensed in a Pareto plot as Figure 2.28. This plot can help to optimize the number of sensors as well as optimize the weight factor. The black circles are clustered in different levels of information entropy. In each cluster, there are configurations with different numbers of sensors and different weight factors overlapping but, there is only one configuration with minimum cost. This configuration is

interpreted as the Pareto optimal configuration. For example, at the entropy level of ~12, there are 12 different configurations fitted (with 6 distinct circles visible). The configurations include 3 to 8 sensors, and with weights varying between 0.0612 to 0.1837, and the Pareto solution (least cost at a constant entropy level) is the one with 3 sensors and $w = 0.1837$. The Pareto solutions are connected with the red line. Every point on this line denotes the best configuration under the corresponding cost constraint, i.e., the information entropy cannot be decreased further without increasing the budget. The Pareto solutions are listed in Table 3. Figure 2.28 and Table 3 could help the decision-makers with the OSP design, including the number of sensors given a budget, without a need to concern with the weight factor in the objective function.

Table 2.3 Pareto solutions at different levels of information entropy

Information entropy	H>18	16< H< 18	12< H< 14	10< H< 12	6< H< 8	4< H< 6
# of Sensors	1 sensor	2 sensors	2 sensors	3 sensors	4 sensors	5 sensors
Cost	$1,500	$3,031	$7,000	$8,562	$14,218	$24,843

In the case of OSP for strain estimation, the entropy from OSP is compared with the same three cases of naïve/common-sense sensor placements, i.e., (c1) uniform along the whole structure, (c2) uniform along the height of the tower, and (c3) locations obtained by Kammer's method considering the first 8 modes. The entropy values for these three cases are compared with the OSP results for strain estimation (1 location and 12 locations) in Figure 2.26. It can be observed that in general, the commonsense cases are non-optimal while there are instances (e.g., 5 sensor setup) that c3 sensor configuration gets close to the optimal solution.



a) For one strain location

b) For 12 strain locations

Figure 2.26 Comparing three common-sense configurations with SSP results for strain estimation

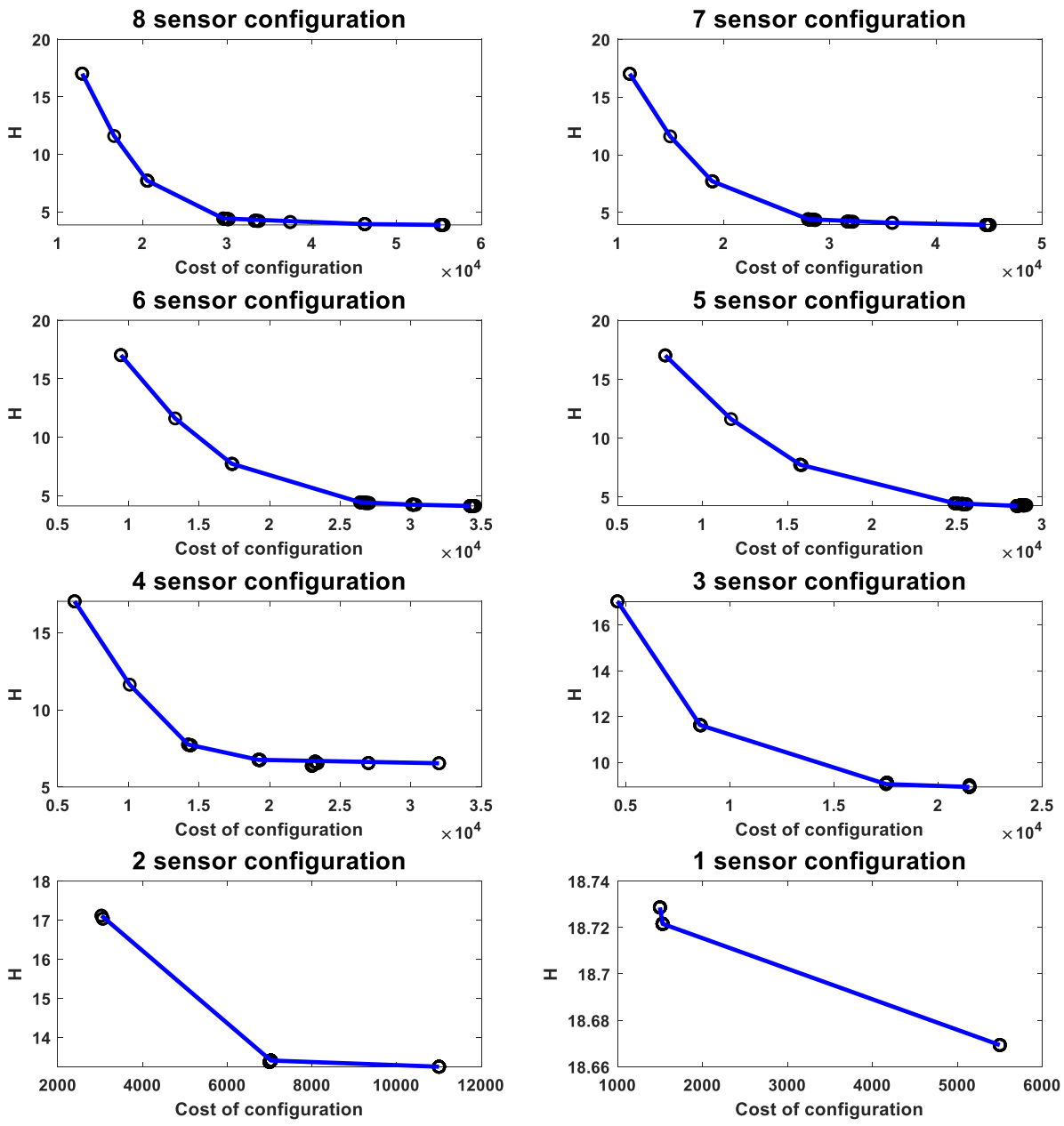


Figure 2.27 Pareto plot (cost vs information entropy) for 1-8 sensor configurations

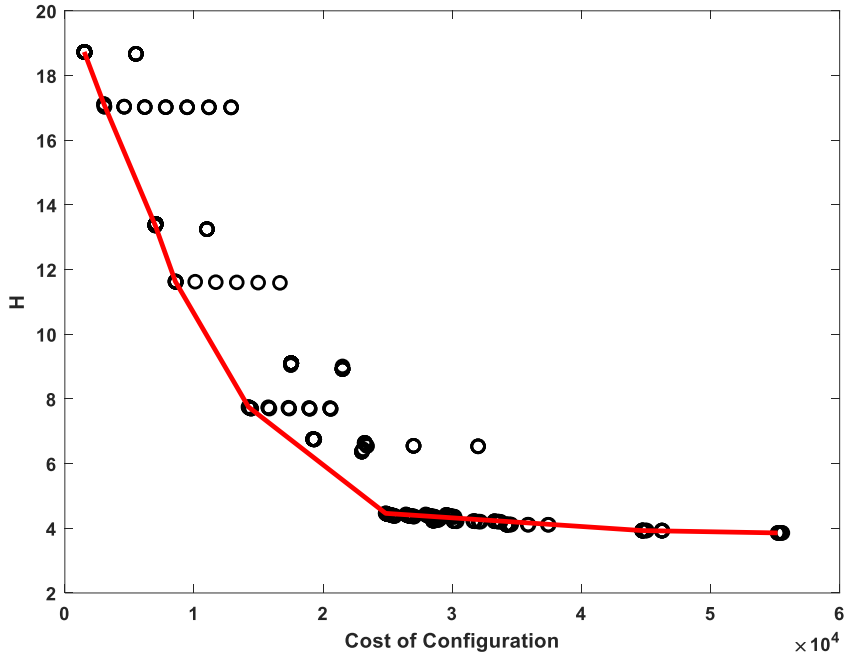


Figure 2.28 Pareto plot (cost vs. information entropy) for all sensor configurations

2.6 Conclusion

This chapter has formulated and evaluated an information theory-based OSP framework for parameter estimation and strain estimation considering sensor configuration cost. The OSP framework for parameter estimation requires the input information, while the OSP framework for strain estimation is an output-only method. The approach is evaluated numerically when applied to a realistic model of an offshore wind turbine. A computationally efficient approach, the SSP method, which is a combination of FSSP and BSSP methods, is employed to solve the OSP problem and is compared with the exhaustive search approach. The results are analogous to optimal configurations from the exhaustive search. The proposed optimization framework accounts for the sensor configuration cost on the structure. The sensor configuration cost is the summation of sensor installation cost, which is defined unevenly based on the location and sensor

cost, which is constant. A cost-benefit objective function is defined and has been used in this study to account for both information entropy and cost constraint. In the objective function, a weight factor is used to balance the importance of information entropy versus cost. The OSP results are found to be sensitive to the weight factor, and thus, a Pareto front study is performed for different weight factors and different numbers of sensors. The Pareto plots can help the decision-makers to quantify the required cost to achieve a certain level of information entropy or alternatively determine the amount of accuracy/entropy that can be achieved for a specific budget.

Acknowledgements

The authors acknowledge partial support of this study by the National Science Foundation grant 1903972, Bureau of Safety and Environmental Enforcement (BSEE), U.S. Department of the Interior, Washington, D.C., under Contract 140E0119C0003, and Massachusetts Clean Energy Center under AmplifyMass program. The opinions, findings, and conclusions expressed in this chapter are those of the authors and do not necessarily represent the views of the sponsors and organizations involved in this project.

3 Chapter 3: Minimal Sensor Placement

3.1 Introduction

Offshore wind turbines operate in harsh and remote environments, necessitating continuous monitoring to ensure their reliability and longevity. Relying solely on preventive maintenance is impractical due to the significant costs and logistical challenges involved in accessing and servicing these structures. Instead, condition-based monitoring offers a more efficient and cost-effective solution. This approach relies on real-time data collection and analysis to predict and prevent potential failures, ensuring the reliability and longevity of the turbines.

To implement effective condition-based monitoring, it is crucial to collect suitable data that accurately reflects the structural health of the wind turbines. This is where the concept of theoretical observability becomes essential. Observability ensures that the collected data enables the complete state of the system to be inferred from its outputs, providing a framework for designing sensor setups that gather informative and sufficient data. The observability of a dynamic system was first introduced by Kalman in 1960 [53]. For a system to be observable, it must be possible to uniquely determine its state from its outputs over time. If a state, an input, or a parameter is unobservable, it cannot be expected that any system identification method would successfully estimate their value correctly. Kalman initially detailed observability for linear dynamical systems, with the condition often checked using the rank of the observability matrix [54]. where the condition is often checked using the rank of the observability matrix. However, for nonlinear systems, the concept required further exploration.

Depending on the nature of the nonlinear system—whether analytic or rational—two general methods for assessing observability have been established: geometric and algebraic approaches. Analytical systems, characterized by analytic functions, are smooth and infinitely differentiable, making them useful for detailed and precise modeling in control theory. Rational systems, defined by functions that are ratios of polynomials, are analyzed using algebraic geometry. These differences influence the mathematical tools and techniques used for analysis, with analytical systems often requiring differential equations and power series, while rational systems use algebraic methods and polynomial equations. Consequently, analytical systems are preferred for applications requiring smooth, continuous modeling, whereas rational systems are suited for contexts where algebraic simplicity and piecewise behavior are adequate.

Building on the foundational work of Kalman, Hermann and Krener [55] extended controllability and observability concepts to nonlinear systems based on the works of multiple researchers in the early 1970s [56-60]. They utilized Lie algebra and differential geometry to define and analyze controllability and observability for analytic systems, presenting techniques to reduce the state space complexity while retaining the system's input-output behavior. They proposed observability rank condition (ORC) as a method to infer the observability of a nonlinear system.

Later, Diop [61] introduced new criteria for observability and identifiability using differential geometry and algebraic methods, known as algebraic observability. Subsequent research bridged the gap between geometric and algebraic theories of observability, demonstrating that for a broad class of rational systems, algebraic observability is equivalent to local generic observability [62]. The paper demonstrated that algebraic observability is equivalent to local generic observability for

a broad class of rational systems. In [63], the researchers combined a review of differential geometric and algebraic methods with practical application. They applied observability theories to a kinetic model of a biological model.

Building on the foundational works of Ljung and Glad [64, 65], where the groundwork for the standard bases computation essential for observability of nonlinear rational systems was explored, Sedoglavic [66] proposed a probabilistic polynomial-time algorithm designed to test the local algebraic observability of nonlinear rational systems. This research contributes to developing an efficient computational approach that can determine the set of observable variables and identify the number of non-observable variables that need to be known to achieve an observable system. Both observability approaches lead to the rank test, which determines the observability of a control system by calculating the dimension of the space spanned by gradients of the Lie derivatives of its output functions.

A comparison of geometric and algebraic observability methods was conducted in [67]. This paper provided an in-depth comparison of these methods and algebraic identifiability algorithms, demonstrating their theoretical foundations and practical applications through engineering examples. The paper highlighted the use of these methods to address real-world problems, discussed their computational efficiency, and outlined their limitations, particularly for non-analytic systems.

Shi et al. proposed a novel algorithm for evaluating the observability of linear systems with unknown parameters [68]. The algorithm addressed the limitations of the traditional ORC by reducing computational cost and memory requirements, making it applicable to large engineering systems. It introduced a recursive algorithm to compute the observability matrix of linear systems

with unknown parameters using numerical realizations and modular operations to improve efficiency. This method significantly reduced the computational complexity associated with symbolic differentiation in the traditional ORC method, enabling the analysis of systems with thousands of states and parameters on standard desktop computers. However, using modular operations introduces a small probability of incorrect rank determination, which, although very low, cannot be entirely eliminated.

Observability Rank Condition with Direct Feedthrough (ORC-DF) [69] was introduced as a geometric algorithm to investigate the theoretical observability of nonlinear systems with partially measured inputs and outputs, tailored for systems with direct feedthrough, which enhanced its applicability to real-world engineering systems. ORC-DF also explored the ability to track unmeasured inputs, referred to as system invertibility.

Later, Shi and Chatzis introduced an efficient algorithm to assess the observability and identifiability of rational nonlinear systems with unmeasured and unknown inputs [70]. The algorithm was based on ORC-DF, which handles rational nonlinear systems with unmeasured inputs and direct feedthrough. They utilized a power series-based framework to efficiently compute the observability matrix, making it applicable to large and complex engineering systems. The paper demonstrated the algorithm's performance and capability through applications to various large and complex engineering models. While the algorithm improved computational efficiency, it still relied on random numerical realizations, which could introduce variability in the results. Additionally, the method was primarily applicable to rational nonlinear systems and might require further adaptation for other types of systems. In another work by Maes et al [71], the

observability framework was introduced for modally reduced-order models, which can also be applied to large structural systems in a computationally efficient way.

In this study the theoretical observability framework is employed to formulate recommendations for designing sensor networks that ensure sensors can capture the essential information needed to infer critical quantities of interest. Given the modular nature of wind turbines, these recommendations are intended to have broad applicability. The primary quantities of interest include the states of the structure, which are vital for effective structural health monitoring. Additionally, in the future, we plan to address the inversion problem by estimating the loads acting on the structure from the measurements. Of particular interest is the separate estimation of wave and wind loads, due to their distinct characteristics. By leveraging this theoretical observability framework, we aim to recommend a minimal sensor placement strategy that guarantees the precise estimation of these critical quantities of interest.

This chapter will cover the methodology and results of our study on the observability of offshore wind turbines. The methodology section is divided into three subsections, detailing the approaches for linear and nonlinear observability and a robust computational algorithm designed to handle large systems efficiently. The third section presents the results and discussion for two categories of problems: one linear and the other nonlinear. Finally, the chapter concludes with a discussion of future ideas and potential improvements.

3.2 Methodology

This section outlines the methodology for analyzing observability in dynamic systems, divided into three subsections: Linear Observability, Nonlinear Observability and computational

algorithm. Each subsection details the mathematical framework and methods used to determine the observability of the system. Observability is a property of a dynamical system that determines whether the internal states of the system can be inferred from its external outputs. A system is considered observable if, for any possible sequence of state and control vectors, the current state can be determined in finite time using only the outputs. Essentially, it reflects the ability to reconstruct the complete internal state of the system from the measurements available.

3.2.1 Linear Observability

Linear observability pertains to linear dynamical systems, where the relationship between the state variables and the outputs can be described using linear equations. Consider a linear time-invariant system described in state space form as:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \tag{3.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{ns}$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^{np}$ is the input vector, $\mathbf{y}(t) \in \mathbb{R}^{ny}$ is the output vector or measured data, $\mathbf{A} \in \mathbb{R}^{ns \times ns}$ is the state matrix, $\mathbf{B} \in \mathbb{R}^{ns \times np}$ is the input matrix, $\mathbf{C} \in \mathbb{R}^{ny \times ns}$ is the output matrix, and $\mathbf{D} \in \mathbb{R}^{ny \times np}$ is the feedforward matrix. To determine whether the states of the system can be inferred from the measurements, one can follow these steps. At time $t = t_0$, which is the initial time, the measured response is:

$$\mathbf{y}(t_0) = \mathbf{C}\mathbf{x}(t_0) + \mathbf{D}\mathbf{u}(t_0) \tag{3.2}$$

By taking the derivative of this equation and substituting $\dot{\mathbf{x}}$ from Eq. (3.1):

$$\dot{\mathbf{y}}(t_0) = \mathbf{C}\dot{\mathbf{x}}(t_0) + \mathbf{D}\dot{\mathbf{u}}(t_0) = \mathbf{C}\big(\mathbf{A}\mathbf{x}(t_0) + \mathbf{B}\mathbf{u}(t_0)\big) + \mathbf{D}\dot{\mathbf{u}}(t_0) \tag{3.3}$$

By repeating this process successively, we obtain:

$$\frac{d^{ns-1}\mathbf{y}(t_0)}{dt^{ns-1}} = \mathbf{C}\mathbf{A}^{ns-1}\mathbf{x}(t_0) + ff\left(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{u}(t_0), \dot{\mathbf{u}}(t_0), \dots, \frac{d^{ns-2}\mathbf{u}(t_0)}{dt^{ns-2}}\right) \tag{3.4}$$

where the function *ff* encapsulates the input and its derivatives up to order (*ns*−2). It is important to note that this structure is also a linear function of the inputs and their derivatives. These terms are eliminated by the ORC where the inputs and their derivatives are all known. Writing these equations for different derivatives in matrix format, we have:

$$\mathbf{Y}(t_0) = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^{ns-1} \end{bmatrix} \times \mathbf{x}(t_0) + ff\left(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{u}(t_0), \dot{\mathbf{u}}(t_0), \dots, \frac{d^{ns-2}\mathbf{u}(t_0)}{dt^{ns-2}}\right) \tag{3.5}$$

where **Y** is the concatenated matrix, which includes all the derivatives of the output and the output itself. This system of equations must have a unique solution to infer the initial conditions from the measured responses. This implies that the number of independent equations must equal the number of states to be estimated. Therefore, the condition for the system to be observable is:

$$rank\left(\begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^{ns-1} \end{bmatrix}\right) = ns \tag{3.6}$$

This matrix is called the observability matrix, $\boldsymbol{\mathcal{O}^L} \in \mathbb{R}^{(ns \times ny) \times ns}$, and a rank test can determine if the system is observable. It should be noted that the observability of linear systems is independent of the input load. By knowing the system and output matrices, we can determine whether the system is observable.

3.2.2 Nonlinear Observability

The concept of observability in control theory is referred to the ability to infer the complete state of a system from its output measurements. For nonlinear systems, this concept is more complex than for linear systems. The ORC is a generalization of the rank condition used in linear systems. Consider an analytic (infinitely differentiable) nonlinear system described in state space form as:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \\ \quad \mathbf{y}(t) = \mathbf{h}\big(\mathbf{x}(t)\big) \end{cases} \tag{3.7}$$

where $\mathbf{f}$: $\mathbb{R}^{ns} \times \mathbb{R}^{np} \rightarrow \mathbb{R}^{ns}$ represents the system dynamics, and $\mathbf{h}$: $\mathbb{R}^{ns} \rightarrow \mathbb{R}^{ny}$ represents the output map. Following similar steps as in linear systems here and taking the derivative of the output equation:

$$\begin{aligned} \dot{\mathbf{y}}(t) = \frac{\partial \mathbf{y}(t)}{\partial t} &= \frac{\partial \mathbf{h}\big(\mathbf{x}(t)\big)}{\partial t} = \frac{\partial \mathbf{h}\big(\mathbf{x}(t)\big)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}(t)}{\partial t} \\ &= \frac{\partial \mathbf{h}\big(\mathbf{x}(t)\big)}{\partial \mathbf{x}} \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) = L_f \mathbf{h}\big(\mathbf{x}(t)\big) \end{aligned} \tag{3.8}$$

where L_f **h(x)** is the Lie derivative of **h** along **f**. The Lie derivative provides a metric to understand how the output function evolves along the system's trajectory. Similar to the process in linear observability, by taking successive derivatives of the output function, we iteratively see how the output function evolves with higher-order effects induced by the dynamics. The observability matrix for the nonlinear system is built by the gradients of the Lie derivatives of the output function with respect to the state vector.

$$\boldsymbol{\mathcal{O}^{NL}} = \begin{bmatrix} \dfrac{\partial L_f^0 \mathbf{h}(\mathbf{x}(t))}{\partial \mathbf{x}} \\ \dfrac{\partial L_f^1 \mathbf{h}(\mathbf{x}(t))}{\partial \mathbf{x}} \\ \dfrac{\partial L_f^2 \mathbf{h}(\mathbf{x}(t))}{\partial \mathbf{x}} \\ \vdots \\ \dfrac{\partial L_f^{n-1} \mathbf{h}(\mathbf{x}(t))}{\partial \mathbf{x}} \end{bmatrix} \tag{3.9}$$

where, L_f^k **h(x)** is the kth order Lie derivative. Then the rank test can determine the observability of the nonlinear system, which is observable if $rank(\boldsymbol{\mathcal{O}^{NL}}) = ns$.

3.2.3 Algorithms to calculate the observability matrix

In calculating the observability matrix for both linear and nonlinear systems, two major issues arise: the computational complexity and practical limitations of using the ORC. For large linear systems, the primary challenge lies in the sheer size of the matrices involved. As the number of states and parameters increases, the observability matrix becomes increasingly large and computationally demanding to calculate. This issue is compounded in nonlinear systems, where the ORC requires symbolic computations of Lie derivatives and their gradients. These symbolic computations are not only complex but also highly memory-intensive and time-consuming, especially for systems with a large number of states and parameters. Therefore, algorithms are needed to test the observability of large systems, as well as efficient methods for handling the symbolic calculations required for nonlinear systems.

In an effort to propose a practical method to estimate the observability of a large linear system with unknown parameters, Shi et al. [68] proposed an algorithm based on the inherent linearity of the system. While the standard Kalman observability matrix would generally be sufficient when

no parameters need to be estimated, as is the case here, it can become computationally intensive and memory-consuming when dealing with large linear systems, especially if calculated symbolically. The algorithm developed in [68] particularly advantageous in this context because it is designed to handle large-scale systems more efficiently, reducing computational overhead and preventing issues like excessive RAM usage. This makes it a practical choice even when the system remains linear and parameters are not being estimated. The system is defined as :

$$\begin{cases} \dot{\mathbf{x}}_{\boldsymbol{t}}(t) = \mathbf{f}\big(\mathbf{x}_{\boldsymbol{t}}(t)\big) + \sum_{i=1}^{m} \mathbf{g}_{\boldsymbol{i}}\boldsymbol{u}_{\boldsymbol{i}} \\ \mathbf{y}(t) = \mathbf{h}\big(\mathbf{x}_{\boldsymbol{t}}(t)\big) \end{cases} \tag{3.10}$$

where $\mathbf{x}_{\boldsymbol{t}}(t) = [\mathbf{x}(t)^T\ \boldsymbol{\theta}^T]^T$ is the augmented state including the states $\mathbf{x}(t) \in \mathbb{R}^{ns}$ and the unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^{n\theta}$, $\mathbf{f} = [\mathbf{Ax}\ \ \mathbf{0}_{n\theta\times 1}]^T$ and $\mathbf{g}_{\boldsymbol{i}} = [\mathbf{B}_{\boldsymbol{i}}\ \ \mathbf{0}_{n\theta\times 1}]^T$ in which Bi is the input matrix. Given that the system functions are linear matrices, the calculations of the Lie derivatives and their gradients with respect to the states can be simplified. This results in a simplified observability matrix for the linear system with unknown parameters, represented by the following equation:

$$\boldsymbol{\mathcal{O}^{NL}} = \begin{bmatrix} \mathbf{C} & \mathbf{CA} & \mathbf{0}_{ny\times ns} & \cdots & \mathbf{0}_{ny\times ns} & \cdots & \mathbf{CA}^{n-1} & \mathbf{0}_{ny\times ns} & \cdots & \mathbf{0}_{ny\times ns} \\ \frac{\partial \mathbf{Cx}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{CAx}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{CB_1}}{\partial \boldsymbol{\theta}} & \cdots & \frac{\partial \mathbf{CB}_{\boldsymbol{m}}}{\partial \boldsymbol{\theta}} & \cdots & \frac{\partial \mathbf{CA}^{n-1}\mathbf{x}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{CA}^{n-2}\mathbf{B_1}}{\partial \boldsymbol{\theta}} & \cdots & \frac{\partial \mathbf{CA}^{n-2}\mathbf{B}_{\boldsymbol{m}}}{\partial \boldsymbol{\theta}} \end{bmatrix}^T \tag{3.11}$$

In [68] it was proven that the symbolic derivatives in constructing the observability matrix can be calculated recursively with less computational complexity compared to direct symbolic differentiation. Moreover, numerical values are assigned to the states and parameters instead of using symbolic expressions which makes the calculation more efficient. These values are chosen randomly but represent the system's operating range. This initial numerical assignment allows the

algorithm to compute the required matrices numerically. Modular arithmetic can further improve numerical stability and computational efficiency. Instead of working with floating-point number realizations of the matrices, the modular remainder is used with respect to a very large prime number. This approach ensures that the computations remain within a manageable numeric range, enhancing both stability and efficiency.

3.3 Results and Discussion

3.3.1 Observability of Large Linear Systems

Direct measurement of strains at all critical points on a wind turbine is impractical due to the complexity and cost involved. Virtual sensing offers a promising solution by estimating strains at unmeasured locations using a limited number of physical sensors and an accurate system model. This approach is particularly beneficial for offshore wind turbines because it reduces the need for numerous expensive and hard-to-maintain physical sensors. By employing virtual sensing, we can infer the strain at unmeasured locations using a limited number of strategically placed sensors. This method relies on the system's observability, ensuring that the state of the structure can be accurately inferred from the available measurements. By applying theoretical observability principles, we can design a minimal sensor network that provides sufficient data to estimate the strains throughout the structure accurately.

3.3.1.1 Numerical Model

This study examines a numerical model of an offshore wind turbine on a jacket support structure, inspired by a real turbine. Using the structural analysis software OpenSees [72], a linear elastic finite element (FE) model is created. The tower is modeled with 32 beam elements of

varying cross-sections and a concentrated mass of 500 tons at the top to represent the Rotor-Nacelle Assembly (RNA). The masses of the tower and jacket elements are lumped at their end nodes. The steel material used for all elements has a modulus of elasticity of 200 GPa. A rotational mass of 1.8×1071.8 \times 10^71.8×107 kg-m² is assigned at the top of the tower for the RNA mass moment of inertia about the X and Y axes. Soil-structure interaction is modeled using three linear springs at the bottom of the four jacket legs: lateral stiffness, vertical stiffness, and rotational constraint. The lateral stiffness is identical in the X and Y directions, and the vertical stiffness and rotational constraint are the same for all four legs. Figure 3.1 shows the model of the turbine. It should be noted here that the system under study is not autonomous; input(s) are applied, and all of these inputs are measured.

3.3.1.2 Observability Matrix calculation

From the model, the mass, stiffness, and damping matrices are extracted to calculate the state and output matrices for the observability matrix. Using the algorithm developed in [68] for large linear systems, and assuming no parameters need to be estimated, the system remains linear, simplifying the calculation of the observability matrix. Different sensor configurations including accelerometers in x and y directions $(\ddot{x}, \ddot{y})$ and displacement sensors measuring in x, y and torsional directions (x, y, θ) are assumed for this problem. Table 3.1 presents the results of the observability analysis. It shows that any sensors placed above the platform do not contribute to making the system observable. However, the system becomes observable when there is only one sensor in any direction on the jacket. In this context, observability means that all the states of the system, including all degrees of freedom (DOFs), displacements, and velocities, can be inferred from the measurements.

Table 3.1. Observability Results for the Linear Case

	x	y	x,y	x,θ	$\ddot{x}$	$\ddot{y}$	$\ddot{x},\ddot{y}$	$\ddot{x},\theta$
Above platform	×	×	×	×	×	×	×	×
Below Platform	✓	✓	✓	✓	✓	✓	✓	✓

Although this algorithm is designed for large systems, the use of double-precision values in calculations makes it slow. For each one of these configurations, it took about an hour to infer the observability.

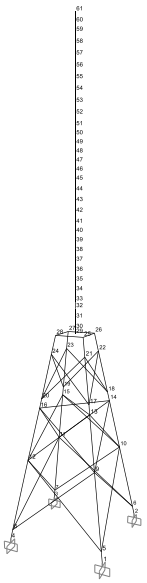


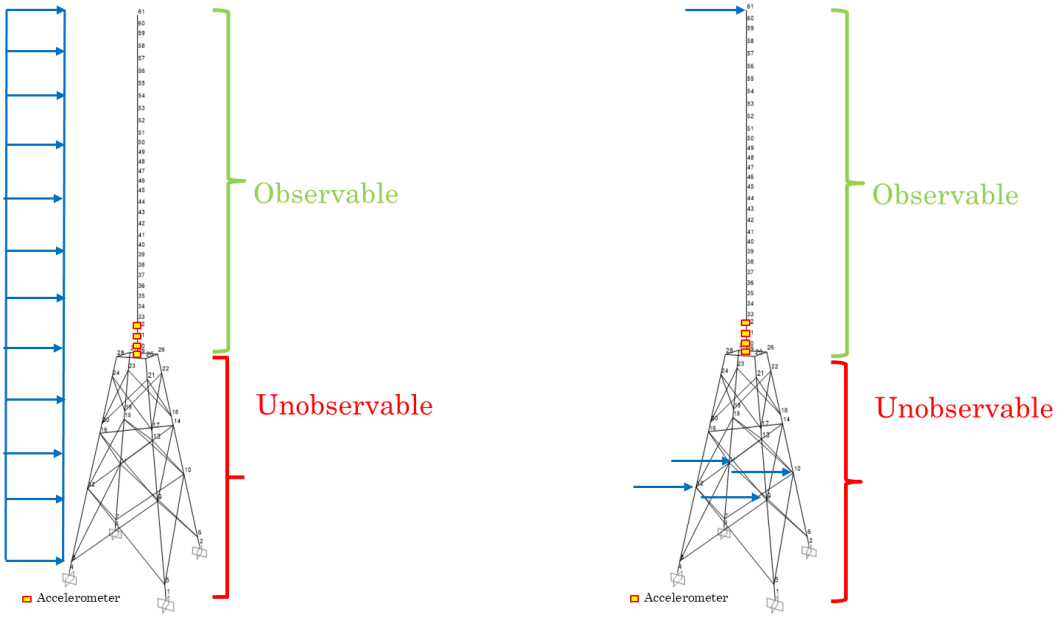
Figure 3.1 Numerical model of the offshore wind turbine on a jacket substructure

3.3.2 Observability of Large Linear Systems with unknown parameters

After addressing the observability of large linear systems, we were also interested in determining if the system remains observable when unknown parameters are added. Adding these unknown parameters makes the inherently linear system nonlinear due to augmentation. Two unknown parameters were added to the system: the modulus of elasticity of the tower and the jacket. The same algorithm used in the previous problem was employed, utilizing modular

arithmetic and numerical realization. The observability of the entire system, including the identifiability of the parameters and the observability of the states, was checked. Through this systematic analysis, the impact of unknown parameters on the system's observability was investigated, ensuring that the system's states and parameters could still be accurately inferred from the available measurements.

First, the effect of different input loading scenarios was explored, as input could affect nonlinear observability. A similar case was defined where four accelerometers were placed at the lowest points of the tower closest to the jacket, shown by yellow rectangles in Figure 3.2. In one scenario, independent loads were applied on all structure nodes. In the other scenario, a point load representative of wind was applied at the top of the tower, and four loads representative of wave forces were applied to the four legs of the jacket, shown by blue arrows. The results are shown in Figure 3.2.

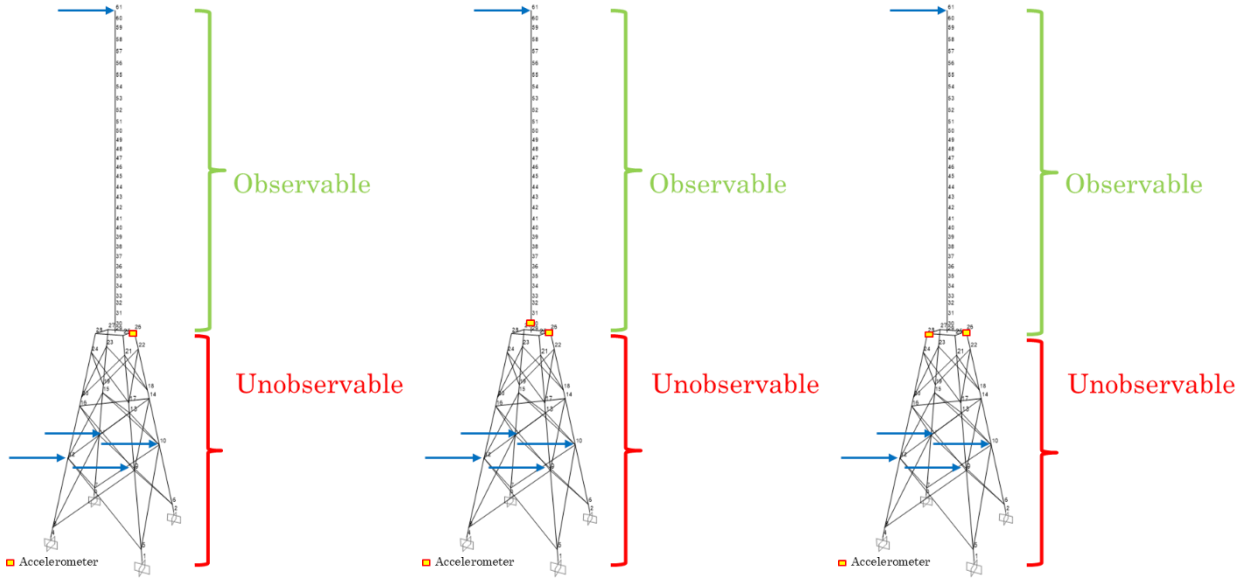


Ej and Et are identifiable Ej and Et are identifiable

Figure 3.2 Observability results for different input loading scenarios

The results show that in both loading scenarios, the whole system is unobservable. This outcome aligns with the results from the linear observability analysis, where it was found that no sensors on the tower alone could make the system observable. As expected, adding more unknowns, such as parameters, maintained the system's unobservable state. However, to determine which specific states are unobservable, the corresponding column of the observability matrix can be removed, and the rank test performed. If the rank remains the same, the state is unobservable. After conducting this analysis, it was found that the states on the tower were all observable, while those on the jacket were unobservable. Interestingly, both parameters (the modulus of elasticity of the tower and jacket) were identifiable in both loading scenarios. In the previous chapter, where parameter estimation was the primary focus, sensor placement on the jacket was not critical because any location on the structure provided sufficient information for accurate estimation. Consequently, the method favored placing sensors on the tower, which was also more cost-effective. However, when the objective shifted to estimating strains specifically on the jacket, the sensor placement strategy adapted accordingly, directing sensors to the jacket.

The next problem involved determining if adding accelerometers on the platform could help make the system observable under the point loading scenario. Three scenarios were checked: one accelerometer on the platform, one accelerometer on the tower and one on the platform, and two accelerometers on opposite sides of the platform to potentially capture torsion effects. The results are shown in Figure 3.3.



Ej and Et are identifiable Ej and Et are identifiable Ej and Et are identifiable

Figure 3.3 Observability results for configurations including sensors on the platform

The results remained the same across all three scenarios: none of the configurations made the system observable. In each case, the states above the platform were observable, but anything below the platform remained unobservable. Additionally, both unknown parameters were identifiable in all scenarios. This indicates that adding accelerometers on the platform does not enhance the system's overall observability.

The final problem investigated whether the findings for the linear system also apply to the augmented nonlinear system. Four different scenarios were checked: (1) two accelerometers on the tower and two on the jacket, (2) two on the tower and one on the jacket, (3) one on the tower and one on the jacket, and (4) only one accelerometer on the jacket, they are shown in Figure 3.4.

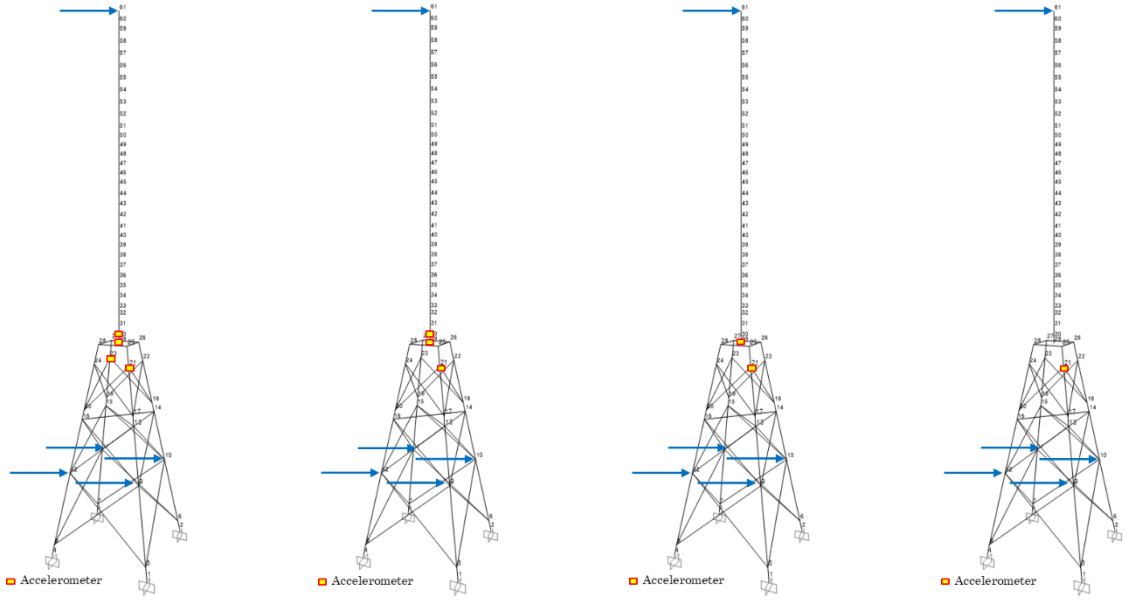


Figure 3.4 Different configurations of accelerometers

All four scenarios resulted in a fully observable system. This outcome mirrors the results seen in the linear system, where the entire structure becomes observable with just one accelerometer on the jacket. Moreover, even a single sensor on the jacket is sufficient for the identifiability of the two unknown parameters. These findings reinforce the conclusion that strategically placing a minimal number of sensors on the jacket can ensure full observability of the system and accurate identification of key parameters.

3.4 Conclusions

In this chapter, the observability of offshore wind turbines was analyzed to formulate recommendations for designing sensor networks that ensure critical quantities of interest can be accurately inferred. The study focused on both linear and nonlinear systems, utilizing robust computational algorithms to handle the complexities of large systems.

The results from the observability analysis demonstrated that for large linear systems, a minimal number of strategically placed sensors could make the system observable, ensuring that the states of the structure, including displacements and velocities, can be inferred from the measurements. When unknown parameters were introduced, turning the system nonlinear, the observability was similarly assessed. Various sensor configurations were tested, and it was found that placing sensors on the jacket of the wind turbine is crucial for achieving full observability and identifiability of parameters just like the linear systems.

Moving forward, the study aims to address the inversion problem which is estimating the input load on the wind turbine, which could potentially be used for virtual sensing at various unmeasured locations. Input loads, such as wind and wave forces, are vital for understanding the structural response. However, these loads are often unmeasured and must be inferred from other available data. Accurately understanding the input loads allows for better prediction of the structural response and potential failure modes. Later, the plan is to specifically focus on the separate estimation of wind and wave loads acting on the structure. One of the ongoing challenges in wind turbine monitoring is the separation of these loads, as they act simultaneously on the structure and their combined effects complicate the analysis. This distinction is important because the two types of loads have different characteristics and effects on the structure. By developing recommendations for minimal sensor placement to separate these loads, more precise load estimation can be achieved, leading to a better understanding of the individual contributions of wind and wave forces to the overall structural response.

The findings of this study will contribute to the design of efficient and effective sensor networks for offshore wind turbines, enhancing their reliability and longevity. By leveraging the theoretical

observability framework, the recommendations will ensure minimal sensor placements can provide comprehensive monitoring capabilities, facilitating timely maintenance and reducing operational costs.

4 Chapter 4: Recursive Bayesian Estimation of Wind Load on a Monopile-Supported Offshore Wind Turbine Using Output-Only Measurements

The findings provided in this chapter are from the materials under review in:

- *Mehrjoo A, Tronci EM, Moynihan B, Moaveni B, Rüdinger F, McAdam R, Hines E. Recursive Bayesian Estimation of Wind Load on a Monopile-Supported Offshore Wind Turbine Using Output-Only Measurements. Mechanical Systems and Signal Processing. Under review*

4.1 Introduction

Offshore wind turbine (OWT) support structures endure various loading events throughout their operational lifespan, ranging from routine conditions to extreme weather events. These structures are constantly exposed to cyclic wave and wind loads, increasing their susceptibility to fatigue, a leading cause of support structure failure [73]. Assessing fatigue life is crucial for ensuring the reliability and longevity of OWT support structures, as well as creating opportunities for design improvements [74]. Due to limited measurements available for support structures, the concept of a digital twin has emerged as a tool for structural response predictions and virtual sensing. Acting as a virtual replica of a physical wind turbine, a digital twin integrates measured data with a virtual model for structural health monitoring and optimizing maintenance strategies. By simulating system behavior in various environmental and operational conditions, digital twins can reduce the need for on-site monitoring, leading to decreased downtime and maintenance costs [75, 76]. In the context of digital twins, input load estimation is a valuable tool that can be used for virtual sensing and response prediction. This study represents an advancement in the field of OWT digital twinning by introducing a tailored estimation tool for input load characterization that could be valuable in the context of design and monitoring.

Previous and current digital twinning practices for monitoring and performance assessment of OWTs [77] focus on modeling accurate digital replicas of actual operational wind turbines by updating their parameters as new monitoring data becomes available. However, these models still require real-time input from the structure to estimate quantities of interest characterizing its dynamic response, such as current strain time histories for fatigue life estimation. This study aims to overcome the lack of direct measurements for input forces by providing an input load estimation

tool for a comprehensive digital twinning framework that enables virtual sensing and structural response simulations. In this context, the introduction of load estimation marks a new addition complementing current digital twinning strategies for OWTs.

The proposed input load estimation strategy also has the potential to play an important role in the design of support structures. Current design methodologies rely on load assumptions that have the potential to be updated through the use of input load estimation, thereby enhancing the accuracy of service-life assessment. Input loads on wind turbine structures have different characteristics, including stationary, periodic, stochastic, and transient variations induced at higher modes that are not explicitly identified in design specifications but that contribute to variability in OWT behavioral characteristics. Estimating the actual input load on the offshore wind turbine enables a more accurate representation of structural dynamic response and loading conditions.

Wind turbine load estimation techniques in the literature can be classified into three categories: 1) Surrogate Model Load Estimation, which utilizes simplified load mapping functions like polynomial chaos expansion [78], deconvolution [79], and lookup tables [80] to estimate site-specific loads based on a database of high-fidelity load simulations, offering interpolation between linear models for different wind conditions; 2) AI-enabled load estimation employs machine learning models [81], including neural networks [82], to learn the relationship between measurements and structural loads, which can easily adapt to the changing conditions of turbines and optimize load control strategies; 3) load estimation based on data combined with virtual sensing, which uses methods such as Kalman filtering [83] to estimate the internal loads acting on the wind turbines [76, 84].

This study focuses on Kalman filter (KF) and Bayesian inference methods for input load estimation. The Kalman filter combines information from a system's model with its measurements to predict additional variables, such as internal states of the system (displacements, velocities), unknown parameters of the structures, and loads where direct measurements are unavailable. Kalman filter methods have been widely used for estimating the dynamic system response of civil, electrical, mechanical, and aerospace systems. This section of the introduction will focus on the literature and background concerning the development and use of Kalman filter strategies for input load estimation of wind turbines.

The different applications of Kalman filters exist for estimating the internal states of a system depending on the target output and prior assumptions. One common approach consists of assuming a known input load to the system or modeling the input load as white noise excitation. In the context of practical applications, this KF strategy was employed to predict strain responses by utilizing measured accelerations and strains on a 3 MW offshore wind turbine in the Belwind wind farm [85]. A comparative analysis between modal expansion and Kalman filter was conducted under the same assumption to estimate stress time histories at fatigue hotspots of a 3 MW offshore monopile wind turbine in different operating conditions [86]. The effectiveness of KF assessing fatigue in metallic structures was further demonstrated by predicting power spectral density from output-only vibration measurements at limited locations [37]. However, the unmeasurable input forces in offshore wind turbines and the inaccurate assumption of white noise for wave and wind loads can impact estimation results.

An alternative KF-based approach involves the joint estimation of internal states and input loads, eliminating the need for assumptions about the input loads. In one study, the actual value of

the input was replaced by an optimal estimate, and KF was then employed for internal state estimation [87]. Another method, the Augmented Kalman Filter (AKF), augments the state vector with unknown inputs and utilizes a standard KF to jointly estimate the unknown states and inputs. AKF demonstrated higher accuracy in cases where inputs were assumed to be collocated with sensors, accounting for modeling errors [88]. The state-input estimation method was applied to an OWT with a lattice foundation for simulation and estimation [89]. A separate study compared KF with joint input-state estimation and modal expansion method for estimating dynamic strain time history at the base of an OWT model [90]. AKF with a reduced model was implemented to assess the loads on a wind turbine [91]. Additionally, a substructure-based input-state estimation framework using AKF was proposed in [92], and a dual KF for joint state-input estimation was proposed and compared with the AKF and Gillijn and De Moor's method [87] on a laboratory frame structure [93, 94].

Another approach considers the uncertainty associated with specific parameters in the OWT model. These structural parameters, such as springs, which represent soil-structure interaction, are uncertain as their values are anticipated to change under different operating conditions [77]. Researchers have employed this method for joint input-state-parameter estimation with two different techniques: Extended Kalman Filter (EKF) methods used by [36, 95-98] and Unscented Kalman filter methods (UKF) [99-102].

This study implements a recursive window-based Bayesian input estimator (WBIE) to predict input loads. The method is validated through numerical simulations and practical application on an operational 6 MW OWT, highlighting its real-world applicability. In applying the WBIE, an initial finite element model of the turbine, which has been updated using measured data, is utilized;

see [77] for more details. The updated model is used in the recursive estimation process to estimate the input load time history in overlapped time windows. By focusing solely on input estimation, WBIE enhances converging stability. Specifically, the study deviates from previous works by performing the method without the need for co-location and applying it directly to measured data without preprocessing, as, for example, is done in [36, 98]. This reduces reliance on user intervention and expertise. Following input estimation, the full structural response is computed by applying the estimated loads to the updated FE model, adding virtual sensing capabilities to the OWT digital twin. One of the key contributions of this study, reflected in the presented results, is its ability to accurately reproduce and predict the structural response along the monopile and tower using only two equivalent lateral point loads at the hub height level. These results prove the method's efficiency in capturing complex and time-varying loading dynamics. The method's performance was compared with the well-established AKF method, which underscores that WBIE excels in providing more accurate input load estimations and exhibits superior full-response predictions without the limitation of having collocated measurements at input load locations. This positions WBIE as a highly promising and effective tool for the digital twinning and virtual sensing of OWTs.

4.2 Methodology

This section reviews the input load time history estimation techniques, offering detailed insights into methodologies such as the WBIE and the Augmented Kalman Filter chosen for comparison. It discusses the common practice of separating dynamic and quasi-static components in the input load and highlights a comparison between the AKF, WBIE, and full response predictions. This section serves as a comprehensive review, offering a nuanced understanding of the intricacies

involved in input load estimation and highlighting the distinctive contributions of the methodology. Categorized as a recursive Maximum A Posteriori (MAP) estimation technique, this method leads to an iterative, window-based Bayesian inference approach for estimating the time history of input force. An overview of this method is provided below.

The discrete equation of motion for a linear dynamic system can be written as

$$\mathbf{M}\ddot{\mathbf{u}}_t + \mathbf{C}\dot{\mathbf{u}}_t + \mathbf{K}\mathbf{u}_t = \boldsymbol{S_p}\mathbf{f}_t \tag{4.1}$$

where $\ddot{\mathbf{u}}_t \in \mathbb{R}^{n_{DOF}\times 1}$, $\dot{\mathbf{u}}_t \in \mathbb{R}^{n_{DOF}\times 1}$and $\mathbf{u}_t \in \mathbb{R}^{n_{DOF}\times 1}$ are the nodal acceleration, velocity and displacements at discrete time step *t*; $\mathbf{M}, \mathbf{C}, \mathbf{K} \in \mathbb{R}^{n_{DOF}\times n_{DOF}}$ are the mass, damping, stiffness matrices of the structural system respectively, with n_{DOF} representing the number of degrees of freedom characterizing the system. $\boldsymbol{S_p} \in \mathbb{R}^{n_{DOF}\times n_f}$ is the Boolean selection matrix, n_f is the number of input loads and $\mathbf{f}_t \in \mathbb{R}^{n_f\times 1}$ is the input load vector at time step *t*. The equation of motion can be reformulated into state space form, considering the fact that the nodal responses of the structure can be expressed as a function of the input load, as shown in Equation 2. This function is effectively characterized, in the formulation implemented in this work, by the FE model of the structure. The measured response of the system is a combination of the model response plus a total noise term, including modeling error and measurement noise. Considering this, the measurement vector $\mathbf{y}_t \in \mathbb{R}^{n_y\times 1}$, with n_y representing the number of measured response locations, can be expressed in following form:

$$\mathbf{y}_t = \mathbf{h}(\mathbf{f}_{1:t}) + \boldsymbol{\nu}_t \quad \text{where} \quad \boldsymbol{\nu}_t \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_t) \tag{4.2}$$

where $\mathbf{h}$ is the function or the FE model and $\mathbf{f}_{1:t}$ is the time history of the input load vector from time step 1 to t. The modeling and measurement noise are aggregated in $\boldsymbol{\nu}_t$. This represents the

error between what the FE model estimates and the measurements, following a zero-mean Gaussian white noise process with $\mathbf{R}_t \in \mathbb{R}^{n_y \times n_y}$ as the covariance matrix. In this scenario, the input load is unknown and is yet to be estimated. The estimation process adopts a windowing approach, dividing measured responses into *N* overlapping windows, each with a length of *L*. Each window k is from the time stamp t_1^k to t_2^k. Let's define $\boldsymbol{\psi}_k = (\mathbf{f}_{t_1^k:t_2^k})$ as the quantity of interest (QoI) for estimation in each window and write measurement response in Equation (4.2) for each window as $\mathbf{y}_k = \mathbf{h}\left(\mathbf{f}_{1:t_2^k}\right) + \boldsymbol{\nu}_k$ where $\mathbf{y}_k \in \mathbb{R}^{(L \times n_y) \times 1}$, $\boldsymbol{\nu}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_k)$ and $\mathbf{R}_k \in \mathbb{R}^{(L \times n_y) \times (L \times n_y)}$ which is a block diagonal matrix. Bayes' rule can be used to find the posterior probability distribution of $\boldsymbol{\psi}_k$:

$$p(\boldsymbol{\psi}_k|\mathbf{y}_k) = c_0 \times p(\mathbf{y}_k|\boldsymbol{\psi}_k) \times p(\boldsymbol{\psi}_k) \tag{4.3}$$

where $p(\mathbf{y}_k|\boldsymbol{\psi}_k)$ is the likelihood function, $p(\boldsymbol{\psi}_k)$ is the prior distribution and c_0 is the normalizing constant. The MAP estimate of $\widehat{\boldsymbol{\psi}}_k$ can be found by maximizing the posterior probability distribution.

$$\widehat{\boldsymbol{\psi}}_k = argmax\big(p(\boldsymbol{\psi}_k|\mathbf{y}_k)\big) \tag{4.4}$$

Solving this optimization problem can be done iteratively [103]. By assuming a random walk model for the QoI, its evolution can be written as

$$\boldsymbol{\psi}_{k,i+1} = \boldsymbol{\psi}_{k,i} + \mathbf{w}_{k,i} \quad \text{where} \quad \mathbf{w}_{k,i} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}) \tag{4.5}$$

Here, $\boldsymbol{w}_i$ is the process noise vector at i^{th} iteration with covariance matrix $\boldsymbol{Q} \in \mathbb{R}^{(n_f \times L) \times (n_f \times L)}$. The state space representation of the input load estimation problem for each window can be expressed as:

$$\begin{cases} \boldsymbol{\psi}_{k,i+1} = \boldsymbol{\psi}_{k,i} + \mathbf{w}_{k,i} \\ \mathbf{y}_k = \mathbf{h}\left(\boldsymbol{\psi}_{k,i+1}\right) + \boldsymbol{\nu}_{k,i+1} \end{cases} \tag{4.6}$$

By assuming the prior and likelihood density functions as Gaussian distributions [97] in Equation (4.3) and after algebraic operations the logarithm of posterior distribution function for each iteration takes form

$$\begin{aligned} \log\left(p(\boldsymbol{\psi}_k|\mathbf{y}_k)\right) = c_1 - 1/2\left(\mathbf{y}_k - \hat{\mathbf{y}}_{k,i+1}\right)^T \mathbf{R}_k^{-1}\left(\mathbf{y}_k - \hat{\mathbf{y}}_{k,i+1}\right) \\ - 1/2\left(\boldsymbol{\psi}_{k,i+1} - \widehat{\boldsymbol{\psi}}_{k,i+1}^-\right)^T \left(\widehat{\mathbf{P}}_{k,i+1}^-\right)^{-1}\left(\boldsymbol{\psi}_{k,i+1} - \widehat{\boldsymbol{\psi}}_{k,i+1}^-\right) \end{aligned} \tag{4.7}$$

where $\hat{\mathbf{y}}_{k,i+1}$ is the model response or $\mathbf{h}\left(\boldsymbol{\psi}_{k,i+1}\right)$, $\widehat{\boldsymbol{\psi}}_{k,i+1}^-$ is the prior mean of QoI, $\widehat{\mathbf{P}}_{k,i+1}^-$ is the prior covariance of QoI at k^{th} window and c_1 is a constant. The MAP can be found by minimizing Equation (4.7), i.e., solving for when its derivative with respect to the state vector (QoI) is zero.

$$\left(\frac{\partial \hat{\mathbf{y}}_{k,i+1}}{\partial \boldsymbol{\psi}_{k,i+1}}\right)^{\mathrm{T}} \mathbf{R}_k^{-1}\left(\mathbf{y}_k - \hat{\mathbf{y}}_{k,i+1}\right) - \left(\widehat{\mathbf{P}}_{k,i+1}^-\right)^{-1}\left(\boldsymbol{\psi}_{k,i+1} - \widehat{\boldsymbol{\psi}}_{k,i+1}^-\right) = 0 \tag{4.8}$$

The derivation in Equation (4.8) is performed numerically through the finite difference method. Within each window, an iteration process starts to update the prior mean and covariance $\widehat{\boldsymbol{\psi}}_{k,i+1}^-$ and $\widehat{\mathbf{P}}_{k,i+1}^-$ to posterior mean $\widehat{\boldsymbol{\psi}}_{k,i+1}^+$ and covariance $\widehat{\mathbf{P}}_{k,i+1}^+$ using the Kalman gain matrix $\mathbf{K}_k$. The relative change in the estimated $\widehat{\boldsymbol{\psi}}_{k,i+1}^+$ is then compared to the previous iteration and the iterative process continues until a desired convergence criteria $\varepsilon_{\boldsymbol{\psi}}$ is met, or the maximum allowed iterations i_{max} is reached. The framework then advances one window ahead in time, and the process repeats. For a detailed explanation of the derivation the reader is referred to [97].

$$\mathbf{K}_{k,i+1} = \widehat{\mathbf{P}}_{k,i+1}^- \mathbf{C}^T\left(\mathbf{C}\widehat{\mathbf{P}}_{k,i+1}^- \mathbf{C}^T + \mathbf{R}_k\right)^{-1} \tag{4.9}$$

$$\widehat{\boldsymbol{\psi}}_{k,i+1}^+ = \widehat{\boldsymbol{\psi}}_{k,i+1}^- + \mathbf{K}_{k,i+1}\left(\mathbf{y}_k - \hat{\mathbf{y}}_{k,i+1}\right) \tag{4.10}$$

$$\widehat{\mathbf{P}}_{k,i+1}^{+} = \left(\mathbf{I} - \mathbf{K}_{k,i+1}\mathbf{C}\right)\widehat{\mathbf{P}}_{k,i+1}^{-}\left(\mathbf{I} - \mathbf{K}_{k,i+1}\mathbf{C}\right)^{T} - \mathbf{K}_{k,i+1}\mathbf{R}_{k}\mathbf{K}_{k,i+1}^{\boldsymbol{T}} \tag{4.11}$$

In above equations, $\mathbf{C} = \frac{\partial \hat{\boldsymbol{y}}_k}{\partial \boldsymbol{\psi}_k} \in \mathbb{R}^{(L \times n_y) \times (n_f)}$ is the model response's sensitivity matrix. The utilization of windows for grouping measured data presents a notable advantage compared to the Augmented Kalman filter, enhancing the sensitivity of responses to input loads and consequently refining the precision of the estimated input loads. At each time step *t*, the input not only affects the responses at that specific time step but also extends its influence across the entire window, spanning from step *t* to *L*+*t*. Selecting the appropriate window length requires an important balance. It should be long enough to capture the influence of the input at the start of the window on the subsequent responses. The optimal window size, when combined with overlapping, provides an effective strategy to navigate this challenge. A summary of the framework is presented in Figure 4.1. It is important to highlight that the application of the proposed WBIE method requires prior considerations of tuning parameters. These parameters include the initial state vector and its covariance ($\boldsymbol{\psi}_0$ and $\mathbf{P}_0$), the covariance matrices for process and measurement noise ($\mathbf{Q}$ and $\mathbf{R}$), as well as specifications for window length, overlapping, convergence criteria, and the maximum number of iterations. The initial state vector can be set to zero.

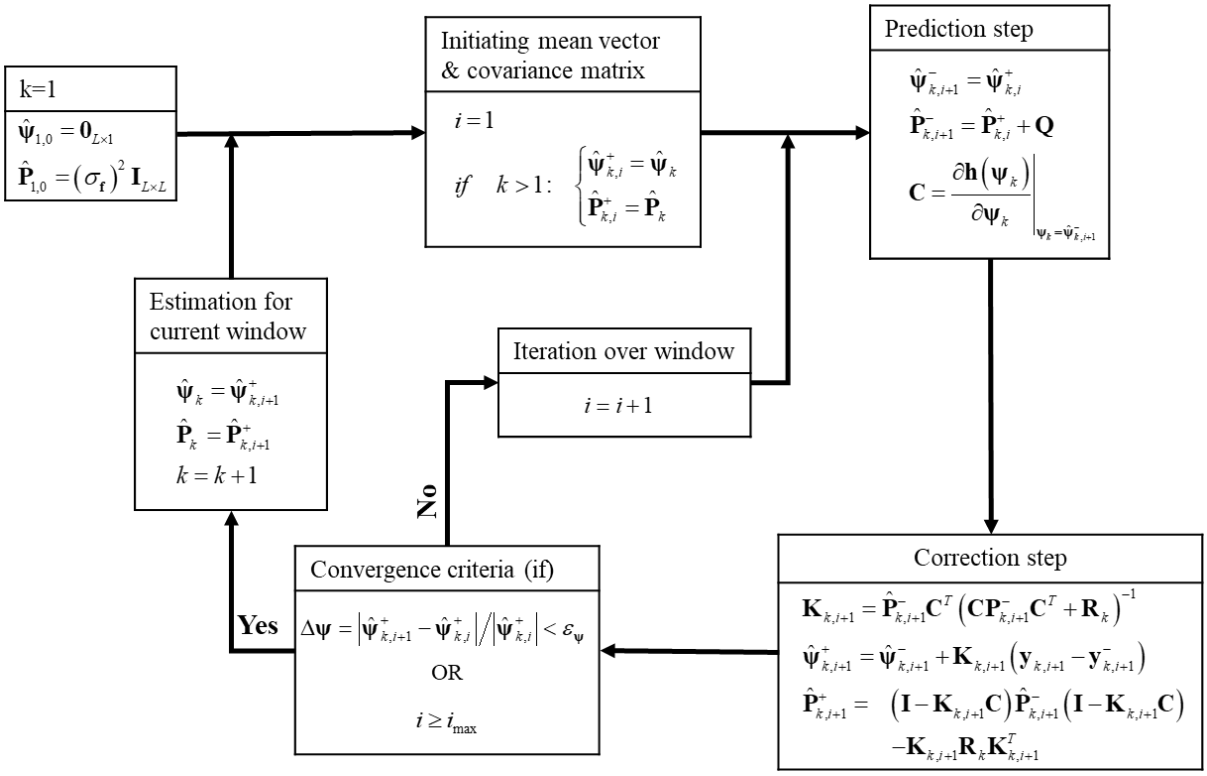


Figure 4.1 WBIE framework

In this study, the results of the proposed WBIE method are assessed through a comparative analysis with the well-established Augmented Kalman filter technique referenced in this paper as AKF [88]. The AKF implemented in this study uses modal space as a reduced-order model of the structural system. It is essential to note that the AKF, unlike the WBIE, does not adopt a window-based approach, emphasizing a key distinction in their respective methodologies. This comparison provides valuable insights into the performance and advantages of the proposed method over the AKF for input load time history estimation. Kalman filter-based strategies, e.g., AKF, are often employed for one-step ahead prediction, which corresponds to a forecasting method that uses the current value of a vector time series to predict its next value. The model is trained on the available data up to a certain point in time and then used to predict the next value in the series, and this process continues. However, WBIE, which is not a traditional Kalman filter, estimates the input

time history given the data up to that point and can reproduce responses by running the FE model with the estimated input up to that point. The difference between the two methods in response estimation is shown in Figure 4.2. The key distinction lies in that WBIE does not directly estimate the vibrational responses; instead, it estimates the input load, which is subsequently used in a FE model to reproduce the responses. In contrast, the AKF method estimates both the input and vibrational responses simultaneously. Due to its nature of windowing data and iterating over each window, WBIE can estimate the input load with better accuracy. This, in turn, leads to more precise response estimations.

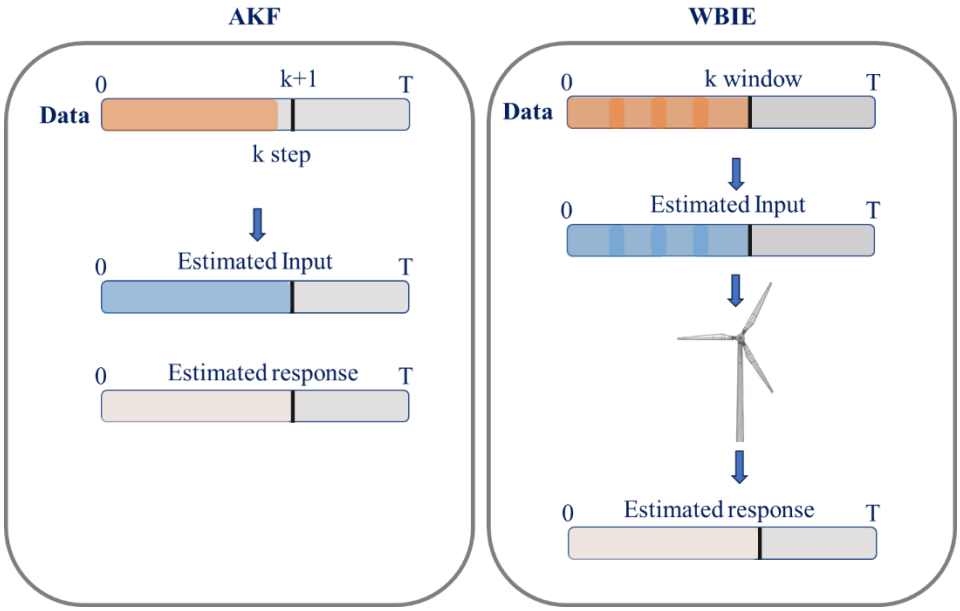
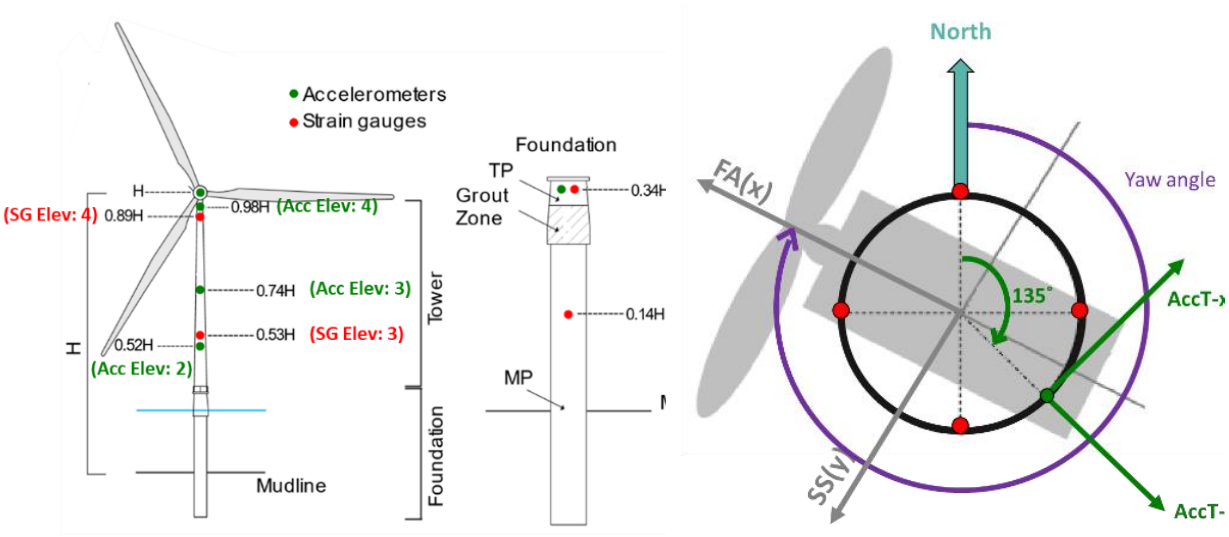


Figure 4.2 AKF and WBIE comparisons

4.3 OWT Model and Available Measured Data

The WBIE framework presented in the previous sections is tested and validated on the measured data from a 6 MW monopile-supported wind turbine located in the North Sea. The target turbine is instrumented with 8 perpendicular accelerometers across four levels, 16 strain gauges across

four levels, and Supervisory Control And Data Acquisition (SCADA) at the nacelle. Two weeks of continuous data are available, recorded at a sampling frequency of 25Hz for vibration channels and 10Hz for SCADA. The strain gauges data are calibrated. Subsequently, they are used to calculate the resultant bending moments in two perpendicular directions (North and East). The reported measured strains are presented in this format, representing the calculated bending moments rather than the raw strain measurements. The sensor layout is illustrated in Figure 4.3.



(a) (b)

Figure 4.3 Instrumentation layout a) Elevation View b) Top View

A linear elastic FE model of this turbine is constructed using the structural analysis software OpenSees [72]. The model is built using the available structural design details of the tower and monopile, which consist of the material properties, tower geometry, foundation, and masses. Linear elastic-beam column elements with varying cross sections are employed to model the tower and foundation. Tower and monopile masses are lumped at their end nodes. The Rotor Nacelle Assembly (RNA) is modeled using a lumped mass and three mass moments of inertia at the center of gravity of the RNA at the top of the tower. Soil-structure interaction is modeled using a

decomposed macro element stiffness matrix, separating lateral and rotational stiffnesses. Further details about the modeling of this OWT and its soil springs can be found in [77].

Due to the symmetry of the structure, modes 1 & 2 and 3 & 4 have closely spaced natural frequencies and similar mode shapes but in perpendicular directions. Figure 4.4 shows the first four modes of the FE model in the x and y directions, along with natural frequencies.

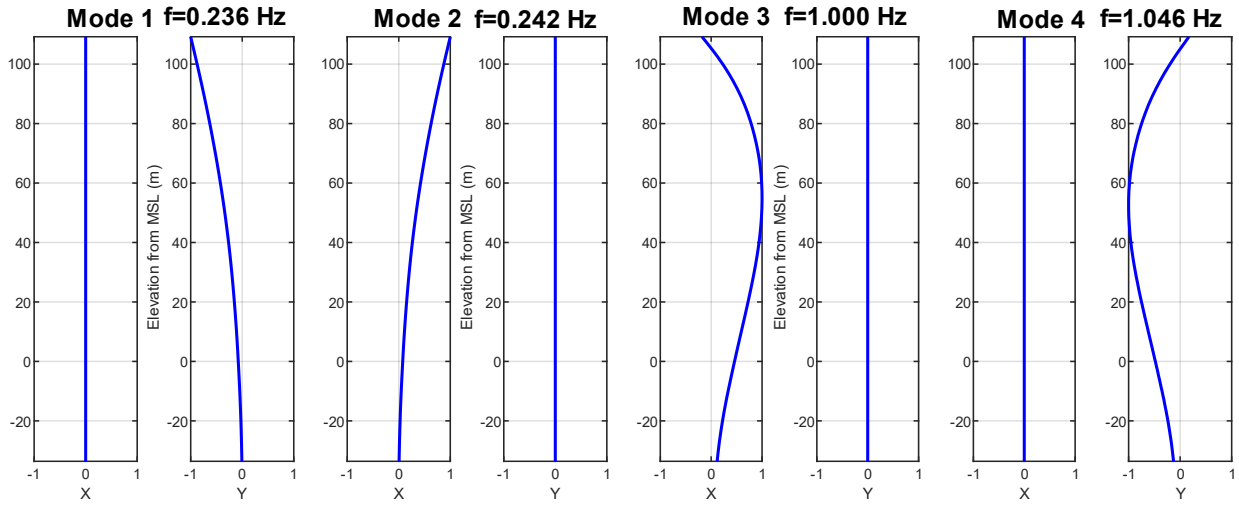


Figure 4.4 First 4 modes of the structure

4.4 Input Load Estimation Results

This section employs WBIE and AKF to estimate two independent perpendicular effective load time histories at the top of the turbine tower in fore-aft (FA) and side-to-side (SS) directions. These lateral input loads serve as representatives of wind forces, wave loads, and nacelle dynamics. It is worth noting that for the operating conditions considered in this work (i.e., region 3 of power production), the wave loads are expected to be negligible compared to wind loads. While the full FE model is utilized for WBIE, a reduced-order modal model is employed for AKF. In structural dynamics, the discretization of the equation of motion in large-scale dynamical systems leads to

high computational costs due to the large number of DOFs involved. To address this challenge, model order reduction techniques are utilized to develop simplified models. These models are designed to keep the dynamic behavior of the system while reducing computational expenses. One such technique is modal reduction, which focuses on capturing the system's vibrational modes. By retaining a subset of these modes, a reduced model is constructed, thereby facilitating more efficient analysis and computation.

Results of the two methods are compared using two quantitative metrics: 1) Relative Root Mean Square Error (RRMSE), which is defined as $\text{RRMSE} = \sqrt{\frac{(y-\hat{y})^T(y-\hat{y})}{y^T y}} \times 100$, and 2) Time Response Assurance Criterion (TRAC) $\text{TRAC} = \frac{(y^T \hat{y})^2}{(y^T y)(\hat{y}^T \hat{y})}$. Through these comparisons, the authors aim to highlight the strengths and limitations of each method, providing insights into their applicability to OWTs.

The WBIE and AKF methods are evaluated using two different applications. First, the techniques are applied to the numerically simulated data generated from the FE model of the 6MW turbine, comparing the performance of the WBIE strategy with AKF. Second, the evaluation is carried out using real-world data, covering a broader evaluation that also includes full response prediction.

4.4.1 Application on Numerically Simulated Data

To numerically simulate the input load at the top of the tower, a multi-physics model is constructed using the FAST tool developed by the National Renewable Energy Laboratory, incorporating a turbulence intensity of 20%. The local axes of the structure are consistently defined by the yaw measurements, where the yaw direction, also referred to as FA, is defined as the local

x-axis, and the SS is defined as the local y-axis shown in Figure 4.4. The wind load is assumed to have a 52-degree angle with the local x-axis and subsequently decomposed into its x and y components, which are then applied to the structure. The simulated measured responses, including accelerations and moments in the x and y directions, are obtained from the model and then polluted with a 1% noise-to-signal ratio to mimic real-world conditions.

For the WBIE framework, the initial value of the load is assumed to be zero, and the process noise **Q** for the input load is modeled as a diagonal matrix with equal entries (10^{-4}) for both input loads in FA and SS. The choice of measurement covariance is based on the root mean square (RMS) of the simulated responses, set to $(0.01\times\text{RMS}(y))^2$. The initial covariance of the input loads is set to be equal to (10^{-2}). Both covariance matrices are diagonal. Window parameters are defined with a length of 16 seconds and 50% overlap. The convergence criterion is set to 0.01, allowing a maximum of 5 iterations. For the AKF, a reduced-order modal model incorporates the first four modes. Process noise **Q** for the state is assumed to be (10^{-8}), **S** input covariance is (10^{12}), and measurement covariance mirrors the values used in WBIE. Note that the input covariance in the AKF method compensates for the order difference between the two estimated values, the states, and the input load and acts as a scaling factor. This information is summarized in Table 4.1.

Table 4.1 Input load RRMSE comparison for the numerical study

WBIE hyperparameters	AKF hyperparameters
$\mathbf{Q} = 10^{-4}$	$\mathbf{Q} = 10^{-8}$
$\mathbf{R} = (0.01\times\text{RMS}(y))^2$	$\mathbf{S} = 10^{12}$
$\mathbf{P_0} = 10^{-2}$	Number of modes = 5
$i_{max}=5,\ \varepsilon_\Psi=0.01$	

The input load estimation results are shown in Figure 4.5. WBIE seems to converge slower to the true input compared to AKF. The predictions for accelerations and moments using both methods are compared in Figure 4.6 and Figure 4.7, respectively. Both methods provide very accurate predictions, but AKF shows slightly better performance, which is an interesting observation. However, as will be observed in the real-world application, the full response results show completely different trends. The full response prediction is where the whole estimated input is fed into the FE model of the structure, and the responses are predicted. Table 4.2 shows the RRMSE comparison between the two methods for input load in both x and y directions, which shows that it takes longer for WBIE to converge to the true input compared to AKF. Table 4.3 compares the one-step-ahead predictions using the two methods. The four elevations defined in the table refer to TRAC values for both methods, and the values for all the simulated responses are 0.99.

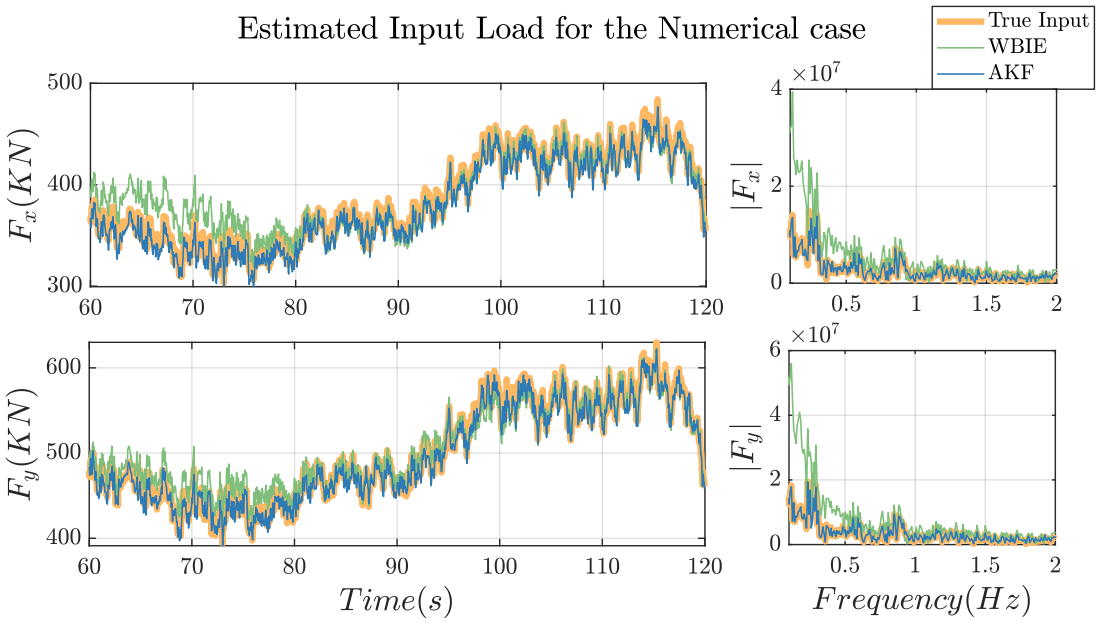


Figure 4.5 Comparison of estimated input load for the numerical case

Table 4.2 Input load RRMSE comparison for the numerical study

RRMSE	including the first 20 seconds		excluding the first 20 seconds	
	WBIE	AKF	WBIE	AKF
x direction	13%	2%	9%	2%
y direction	15%	1%	6%	1%

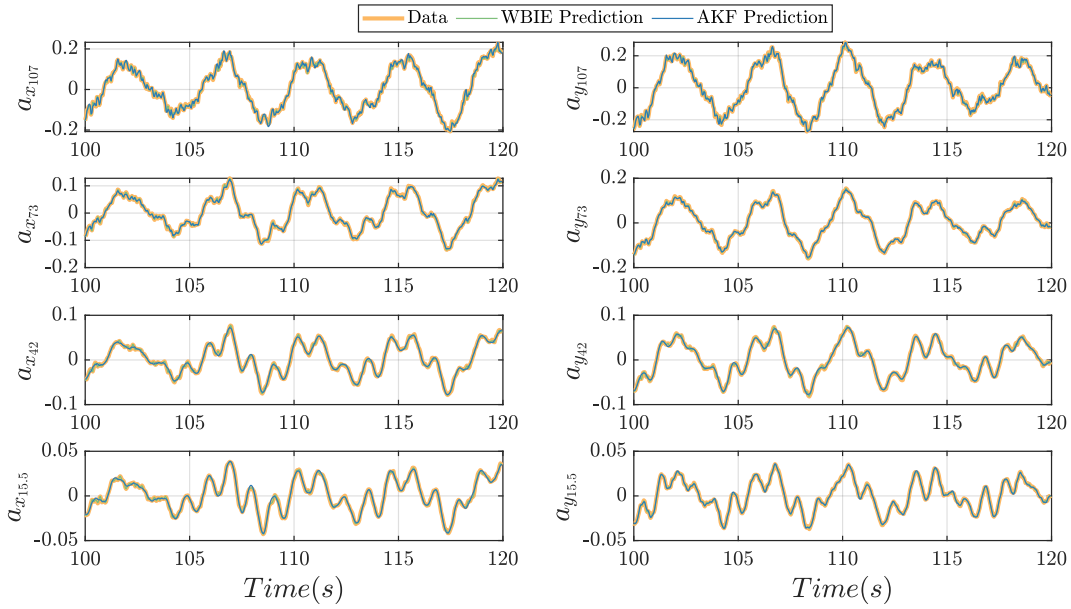


Figure 4.6 Acceleration predictions (in m/s2)

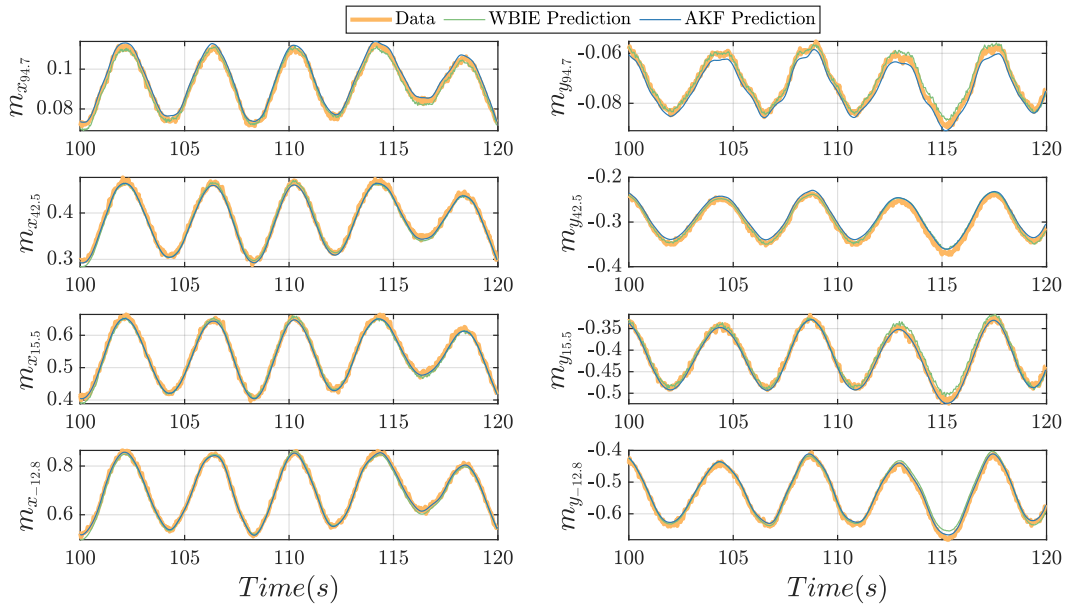


Figure 4.7 Moment predictions (in MN.m)

Table 4.3 Predictions' RRMSE comparison for the numerical study

Elev	a_x		a_y		m_x		m_y	
	WBIE	AKF	WBIE	AKF	WBIE	AKF	WBIE	AKF
4	16%	2%	15%	3%	16%	5%	14%	4%
3	16%	1%	15%	3%	16%	4%	13%	3%
2	16%	1%	15%	2%	16%	5%	14%	3%
1	16%	1%	16%	2%	16%	5%	14%	3%

4.4.2 Application on the Real-World Data

The recorded data is saved in 10-minute intervals. In this application, a dataset characterized by high wind speed and full-power generation operational conditions has been chosen. The parameters of the two methods are the same as the numerical case, except for the measurement covariance. The measurement covariance is set to 4% of the Root Mean Square (RMS) of the measured responses. Input load estimation is conducted using WBIE and AKF methods. An additional comparison is carried out by implementing a quasi-static estimation of loads by dividing

the measured moments by the distance to the load. Since there are four distinct moments at different structural levels, a linear regression approach is used to back-calculate the input. This Linear Regression (LR) method aims to identify the equivalent input that can reproduce these moments with the least average error (L2 norm). It should be noted that this approach assumes the load to be static and can be executed for each time step of the moment's time history. However, it does not capture the high-frequency dynamic content in the estimated input load. The LR method is only applied to estimate input load in the FA or local x-direction.

As discussed in the methodology section, acceleration measurements convey information about the dynamic part of the input, while the moment time histories provide insights into the quasi-static part. The relationship between moments and input loads is described as follows: the moments in the local x-direction (m_x) result from the lateral load in the negative local y direction (SS), denoted as F_y, and the bending moment arising from the torque in the nacelle—a consequence of the generator's reaction. Simultaneously, the moments in the local y direction (m_y) are a result of the lateral load in the negative local x direction (FA), denoted as F_x, and the overhanging moment coming from the nacelle weight at its center of gravity. Consequently, the following equations are employed to solve for F_x and F_y:

$$\begin{aligned} m_x &= F_y v + m_{Torque} \\ m_y &= -F_x v + w_{RNA} h_{cg} \end{aligned} \quad (4.12)$$

where v is the distance between the measured elevation and hub height, m_{Torque} is the bending moment coming from generator torque in the nacelle, w_{RNA} is the RNA weight and h_{cg} is the horizontal offset between RNA center of gravity and center of tower cross-section. m_{Torque} can be calculated from measured power and rpm values available in SCADA. Figure 4.8 shows the

coordinate system and parameters of Equation (4.11). In the numerical study, the contribution of torque and RNA mass offset were not incorporated into the simulated responses.

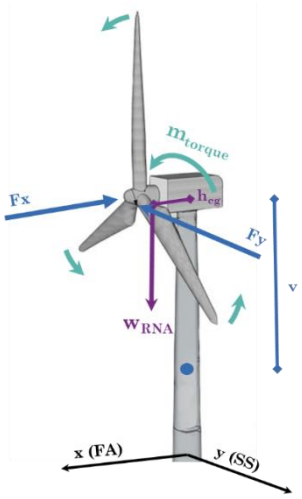


Figure 4.8 Equilibrium equation for the turbine

In past studies of input load estimation [36, 98, 104], researchers have separated the measured responses into dynamic and quasi-static components and subsequently estimated the input load in two distinct steps, shown in Figure 4.9. While this approach is effective for input load estimation, the preprocessing phase can be subjective and requires a more expensive computational effort.

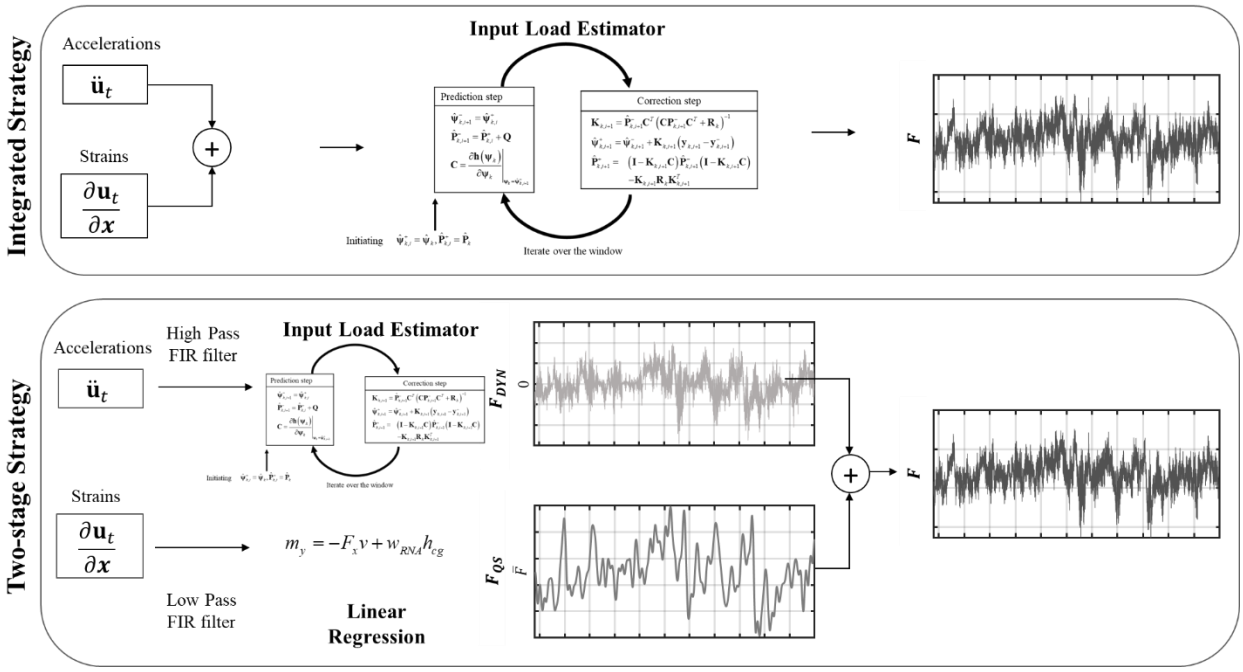


Figure 4.9 Integrated approach vs two-stage separation approach for estimation of the total load

The measured responses are transformed from their global axes to the local x and y axes using the measured yaw angle. The estimated input load in x and y directions using WBIE, AKF and LR for a high wind speed dataset (w_s=12.8m/s) during operational conditions are shown in Figure 4.10. Figure 4.11 provides a zoomed-in view of the input loads, enhancing clarity for comparison. Both figures show the input estimates from the three methods to be in good agreement with each other at low frequencies. As anticipated, linear regression exhibits limitations in capturing higher dynamics in the input load. Consequently, it exhibits high energy around the first natural frequency of the structure, mirroring the frequency content observed in moments. In contrast, WBIE and AKF estimates include additional high-energy peaks at 0.5Hz, corresponding to 3P harmonics, and 1-1.2 Hz region, corresponding to the second bending modes of the structure (in FA and SS) and 6P. It should be noted that the WBIE-estimated input load exhibits higher energy at the 1-1.2Hz region, which proves to be more realistic and advantageous for subsequent full response

reproduction. The input load in the y direction lacks a quasi-static component as expected since, during this operational condition, the turbine faces the wind in x-direction. Figure 4.11 displays the close estimation of the mean thrust load by all three methods. Although the least accurate, LR estimates still provide a reasonable prediction of quasi-static wind thrust load with minimal effort. AKF, while demanding user expertise and filter parameter tuning, boasts quick execution. On the other hand, WBIE demands expertise similar to AKF but incurs higher computational costs. It takes ~6000s for the WBIE to run on the mentioned setting, which can be further increased if iterations are increased to improve accuracy.

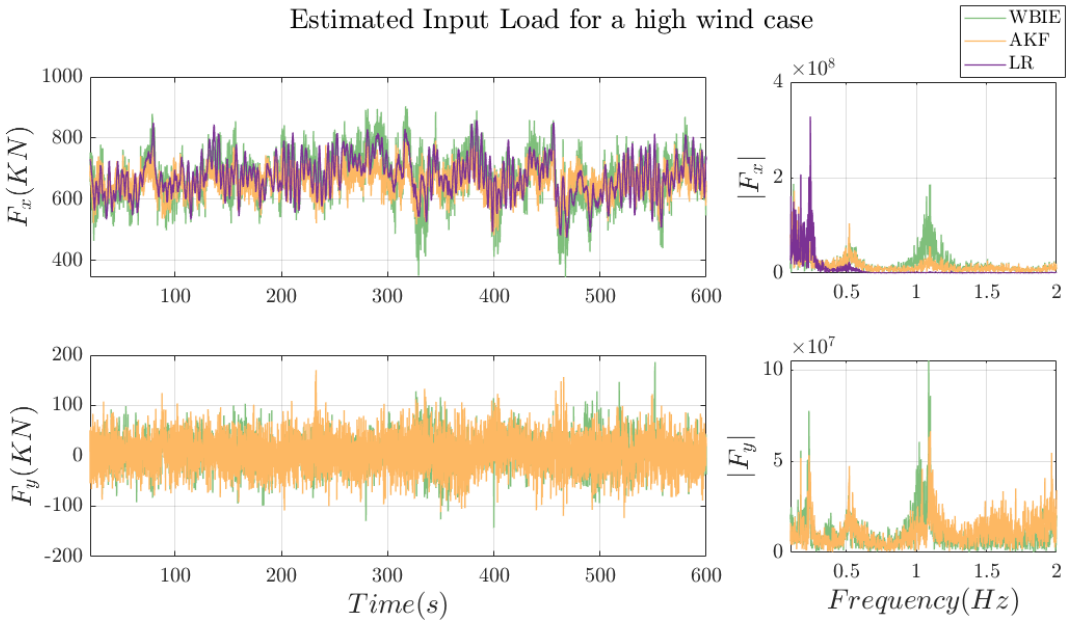


Figure 4.10 Comparison of the estimated input load using LR, WBIE & AKF

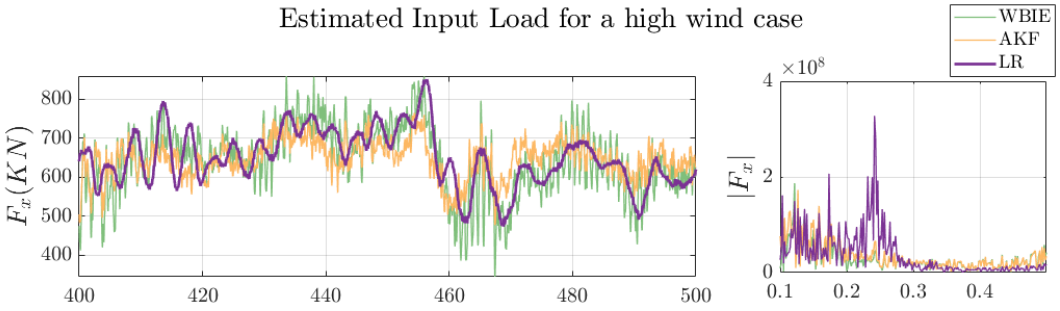


Figure 4.11 Comparison of the estimated input load using the three methods zoomed in 100s

Figure 4.12 shows the prediction of the accelerations at the location of sensors using WBIE and AKF, revealing a generally satisfactory match with measurements, with the exception of the highest level of acceleration, which is close to hub height. Predicting these accelerations is challenging due to the influence of dynamics originating from nacelle dynamics. The results show that AKF outperforms WBIE in predicting the measured responses. Nevertheless, across the lower levels, predictions from both methods exhibit similarity, as confirmed by their reported RRMSE and TRAC values in Table 4.4 and Table 4.5, respectively.

Table 4.4. Predictions' RRMSE comparison for experimental study

Elev	a_x		a_y		m_x		m_y	
	WBIE	AKF	WBIE	AKF	WBIE	AKF	WBIE	AKF
4	61%	23%	62%	20%	54%	31%	16%	15%
3	49%	60%	57%	41%	52%	34%	14%	13%
2	44%	53%	51%	39%	43%	23%	12%	8%
1	40%	49%	50%	34%	49%	38%	11%	7%

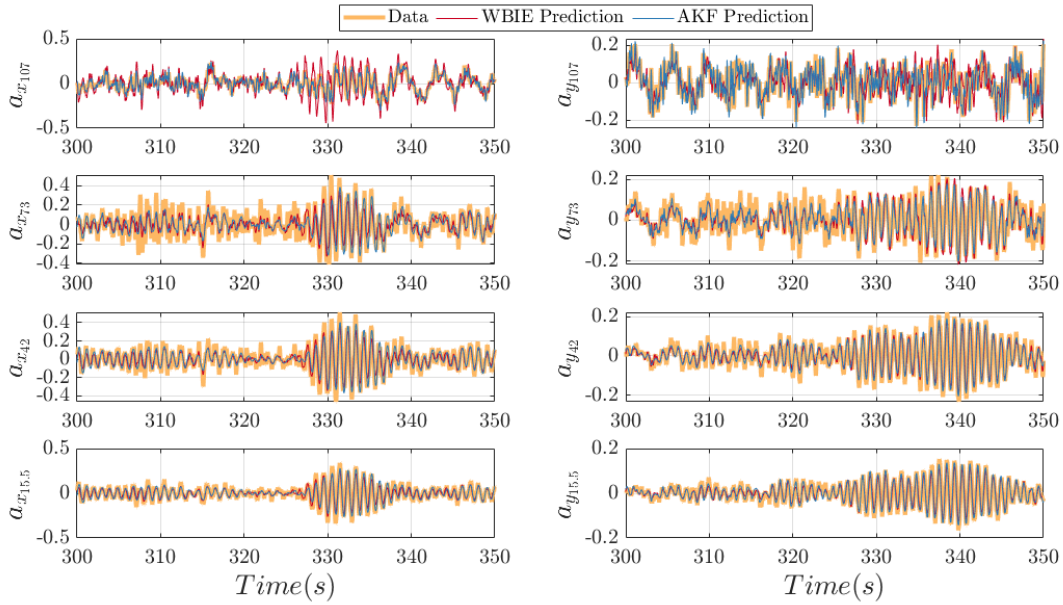


Figure 4.12 Comparison of acceleration predictions at different levels using WBIE and AKF

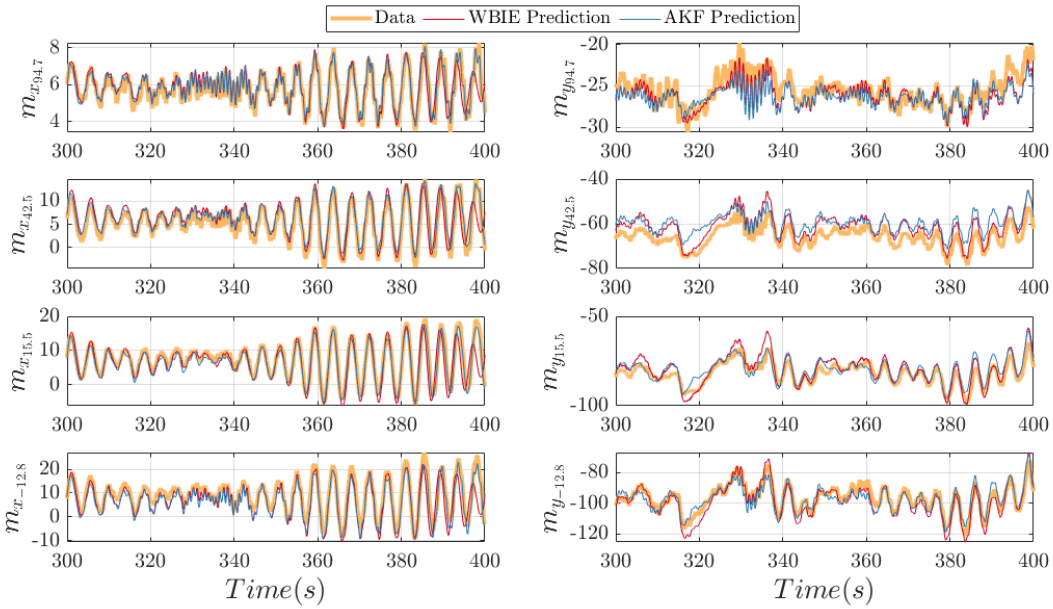


Figure 4.13 Comparison of moments predictions at different levels using WBIE and AKF

Figure 4.13 shows the moment prediction time histories at the measurement's location, indicating that overall, m_x moments (in SS) are better predicted compared to m_y moments (in FA). While AKF outperforms WBIE in the prediction of My moments at certain elevations, the reverse holds true for other elevations where WBIE surpasses AKF. Notably, both methods succeed in capturing the mean of the moments, a crucial aspect directly influencing the mean thrust load. Table 4.4 and Table 4.5 present the RRMSE and TRAC values of moments.

Table 4.5 predictions' TRAC comparison for experimental study

Elev	a_x		a_y		m_x		m_y	
	WBIE	AKF	WBIE	AKF	WBIE	AKF	WBIE	AKF
4	0.755	0.948	0.620	0.963	0.726	0.911	0.980	0.987
3	0.789	0.644	0.675	0.836	0.737	0.890	0.987	0.994
2	0.856	0.724	0.769	0.857	0.826	0.952	0.987	0.994
1	0.889	0.764	0.791	0.884	0.761	0.994	0.988	0.995

The full response reproduction employing the estimated inputs serves as a valuable means to assess the accuracy of the input estimations, as explained in the methodology section, including a validation step in the creation of the model. Consequently, the estimated input loads are used and input in the OpenSees model of the structure, leading to the prediction of the measured accelerations and moments. The full response acceleration predictions are illustrated in Figure 4.14. Notably, WBIE outperformed AKF in acceleration reproduction, aligning with expectations due to the higher frequency dynamic content in the input load estimated by WBIE. The presence of high-frequency content in the input load stimulates the higher vibration modes of the structure. This, in turn, leads to more pronounced accelerations in the structure. As WBIE is adept at capturing these higher frequency components, it performs better than AKF in accurately reproducing acceleration responses. This distinction is further evident in the RRMSE and TRAC values, as detailed in Table 4.6 and Table 4.7.

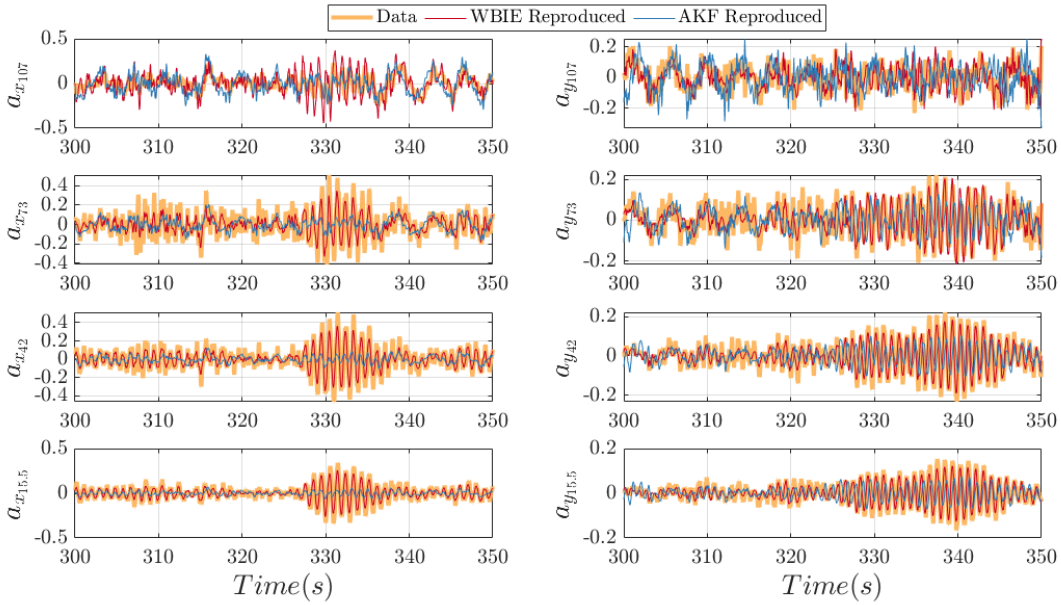


Figure 4.14 Comparison of full response acceleration reproduction at different levels using WBIE and AKF

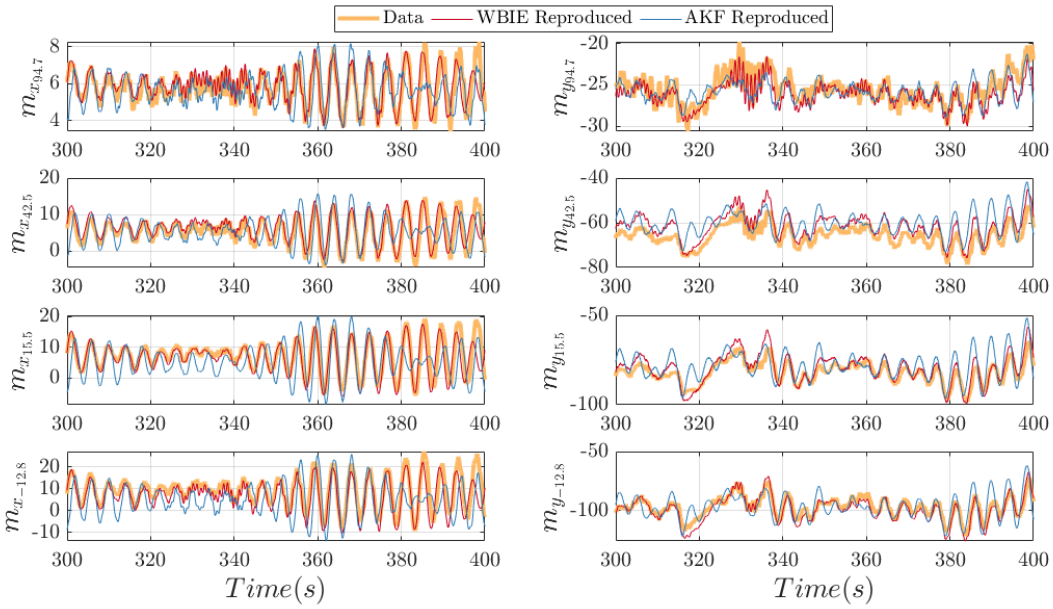


Figure 4.15 Comparison of full response moment reproduction at different levels using WBIE and AKF

The full response prediction of moments is presented in Figure 4.15, highlighting WBIE's superior performance over AKF, especially in x-direction. AKF demonstrates comparable performance in the y-direction, making it advantageous when moments in the FA direction are required for further analyses, such as fatigue assessments, given AKF's notably faster processing speed. Conversely, WBIE performs much better, with metrics of full response reproduction closely aligning with normal predictions and establishing itself as a more reliable tool. Table 4.6 and Table 4.7 provide the performance metrics of the two methods. It is observed that AKF performs well in one-step-ahead predictions even when the estimated input is not reliable. In contrast, WBIE metrics remain consistent between the full response reproduction and normal predictions, indicative of accurate input estimations. This divergence underscores the limitations of AKF as an input estimator, and consequently, its applicability as a virtual sensing tool may be limited.

Table 4.6 Full response predictions' RRMSE comparison for experimental study

Elev	a_x		a_y		m_x		m_y	
	WBIE	AKF	WBIE	AKF	WBIE	AKF	WBIE	AKF
4	61%	110%	63%	153%	56%	135%	16%	16%
3	61%	97%	63%	127%	54%	135%	14%	15%
2	52%	94%	57%	107%	44%	145%	12%	13%
1	48%	92%	55%	104%	50%	141%	11%	12%

Table 4.7 Full response predictions' TRAC comparison for experimental study

Elev	a_x		a_y		m_x		m_y	
	WBIE	AKF	WBIE	AKF	WBIE	AKF	WBIE	AKF
4	0.756	0.520	0.603	0.021	0.708	0.155	0.980	0.978
3	0.634	0.197	0.607	0.036	0.722	0.176	0.987	0.986
2	0.758	0.139	0.695	0.070	0.811	0.142	0.987	0.985
1	0.800	0.157	0.718	0.068	0.752	0.101	0.988	0.985

The performance of the three methods, WBIE, AKF, and LR, are also compared for the estimation of average wind thrust loads at different wind speeds. These estimated average thrust loads are obtained from measured 10-minute data windows at different wind speeds. Eight datasets featuring varying wind speeds were randomly selected in this scenario, and the input loads were estimated. The comparison involves evaluating the mean of the estimated thrust load (equivalent to the input load in the FA direction) derived from each method against the values provided by the turbine OEM. Good agreement is observed between the estimated thrust loads from all three methods and the OEM values supplied by the turbine designer. The results are shown in Figure 4.16, where the black diamonds are the OEM-provided values.

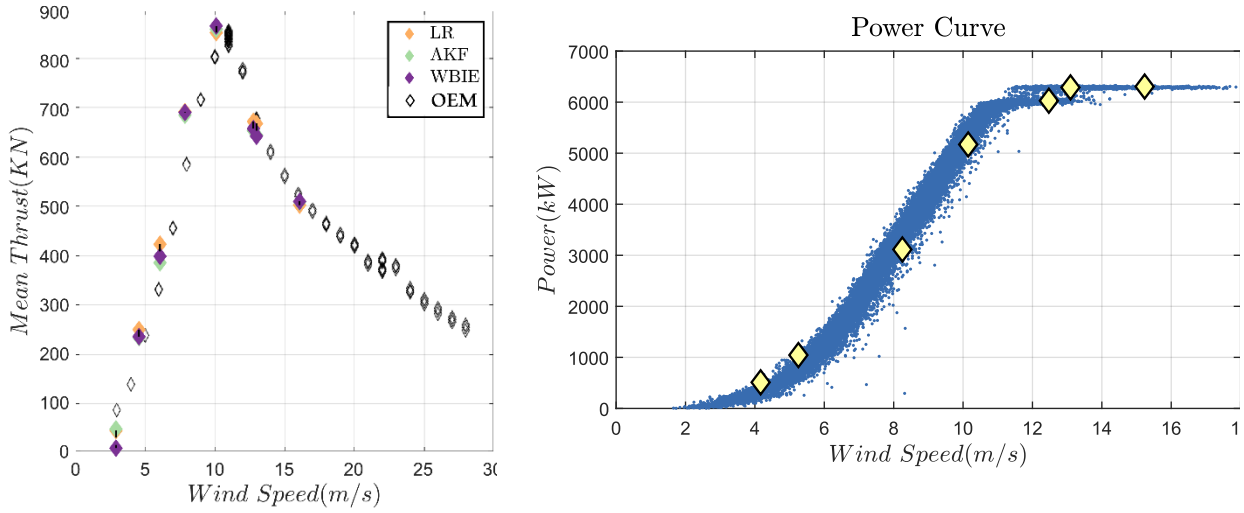


Figure 4.16 (a) Mean thrust load comparison; (b) placement of investigated observations on the power curve

4.5 Virtual Sensing

Finally, the capability of this method for virtual sensing at unmeasured locations is evaluated. This entails the removal of some measured channels, with an attempt to reproduce them using other measurements. The objective is to evaluate how accurately the framework can predict response at unmeasured locations and to assess prediction errors. Virtual sensing is used to estimate the state of a system at locations where no physical sensors are present. The accelerations and moments at their lowest measured level (Acc Elev 1 and SG Elev 1 shown in Figure 4.3) were removed, and the inputs were estimated using the remaining sensor data. These estimated inputs were then used to predict the moments and accelerations at the locations where they were removed, effectively creating a 'virtual sensor' at these locations. This approach allows the spatial coverage of the sensing capabilities to be extended without the need for additional physical sensors, which can be particularly beneficial in scenarios where sensor installation is challenging or costly. The accuracy of the virtual sensing technique using the WBIE method is quantified by comparing the

estimated moments and accelerations with the actual measurements, demonstrating the effectiveness of this approach. Figure 4.17 and Figure 4.18 show the virtual sensing capability of WBIE for virtual sensing of accelerations and moments, respectively. Table 4.8 Virtual sensing metrics reports the virtual sensing accuracy metrics. The predicted moments in the y-direction (FA moments) are found to be more accurate.

Table 4.8 Virtual sensing metrics

a_x		a_y		m_x		m_y	
RRMSE	TRAC	RRMSE	TRAC	RRMSE	TRAC	RRMSE	TRAC
54%	0.763	63%	0.660	65%	0.612	12%	0.988

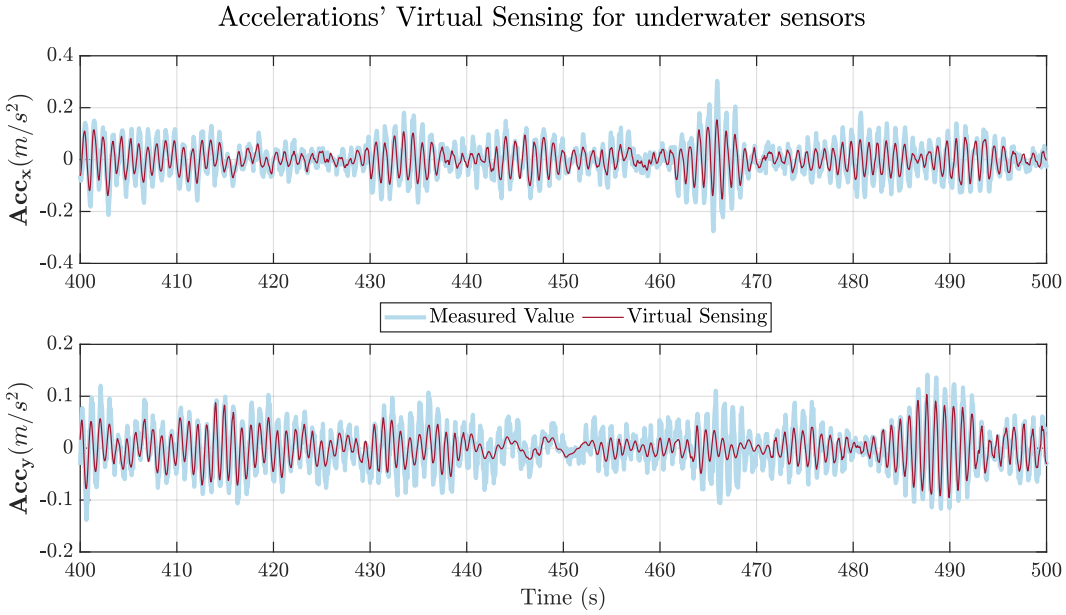


Figure 4.17 Accelerations virtual sensing at the transition piece

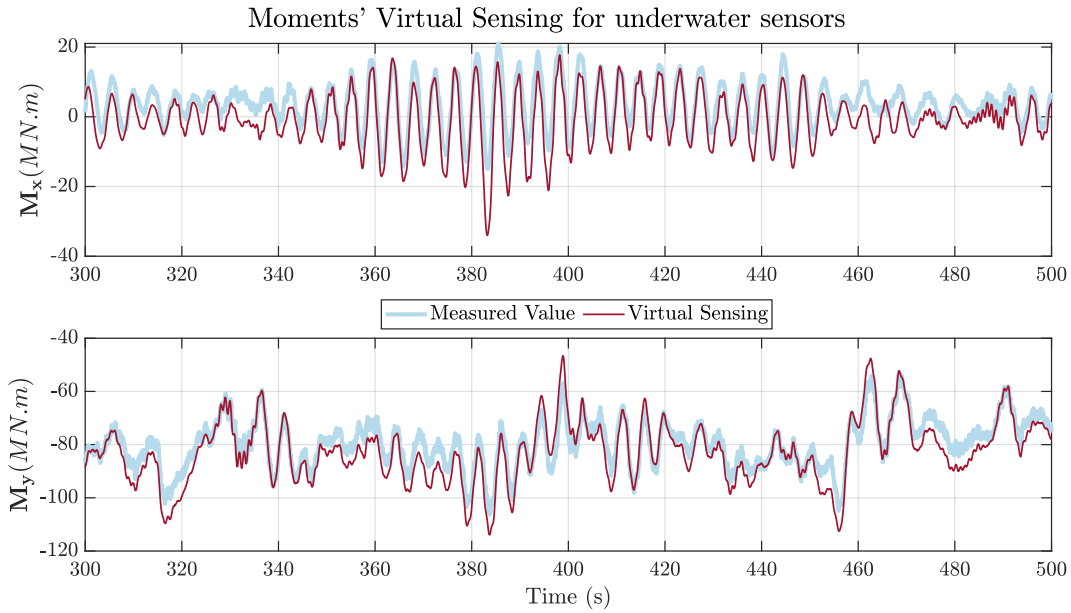


Figure 4.18 Moments virtual sensing for the underwater sensors

4.6 Conclusion

This chapter presents a method for estimating the input loads on a monopile-supported wind turbine using output-only measurements. The method is based on a windowed Bayesian recursive estimator that uses a finite element model of the turbine and measured responses such as accelerations and moments to infer the unknown input loads. The method is validated using both numerically simulated and real-world data from a 6 MW offshore wind turbine. The results show that the proposed method can accurately estimate the input loads and the full structural responses, as well as perform virtual sensing at unmeasured locations. The WBIE method is compared with the simpler AKF and LR methods and has shown to perform better. Comparative analysis of WBIE, AKF, and LR methods underscores the trade-offs between accuracy, computational complexity, and applicability. While AKF exhibits good performance in one-step-ahead predictions, WBIE stands out for its ability to provide realistic input load estimations crucial for

comprehensive full response reproduction. The proposed method allows for a better estimation of the input load, which can inform future designs of offshore wind turbines. The chapter demonstrates the potential of the proposed input load estimation method as a step in the digital twinning of offshore wind turbines, which can enhance their reliability, durability, and maintenance strategies.

Acknowledgements

Partial support for this study is provided by National Science Foundation grant 2230630. A special acknowledgment goes to Ørsted for providing the data needed for this study. The opinions, findings, and conclusions expressed in this document are those of the authors and do not necessarily represent the views of the sponsors and organizations involved in this project.

5 Chapter 5: Hybrid Surrogate Input Load Estimation Model in Offshore Wind Turbines using Transfer Learning and Multitask Learning

The findings provided in this chapter are from the material under review in:

- *Mehrjoo A, Tronci EM, Moaveni B, Hines E. Hybrid Surrogate Input Load Estimation Model in Offshore Wind Turbines using Transfer Learning and Multitask Learning. Renewable Energy. Under review*

5.1 Introduction

Input load estimation can play an important role in Offshore Wind Turbine (OWT) operations and maintenance (O&M). Input load estimation enables the structural responses of OWTs to be assessed at all locations, even where there are no direct measurements. These responses are key indicators of the turbine's state under varying load conditions. This measured information facilitates direct comparisons between an OWT's expected design state and its actual performance. This process enables proactive maintenance interventions, which can increase long-term OWT reliability and performance.

Input load estimation also presents opportunities for improving OWT design. Current design methodologies depend on simplified load assumptions that were developed for turbines with smaller rotors, and do not fully capture the dynamic loading conditions experienced by OWT structures. By integrating new input load estimation results with existing approaches to design, these assumptions can be re-evaluated and defined for the next generation of OWTs.

The literature features three different approaches to estimating OWT input loads: physics-based, data-driven, and hybrid. Physics-based models rely on fundamental physical principles governing OWT behavior, including aerodynamics, structural mechanics, and control theory. Many of these models rely on Kalman filter-based approaches [83]. For instance, recent studies have used Kalman filters in load monitoring for wind turbine blades to assess the remaining useful lifetime (RUL) for structural health monitoring by estimating static loads on the blade using data from a Supervisory Control And Data Acquisition (SCADA) system and strain measurements at the blade root to estimate dynamic responses [105]. The estimated dynamic responses were used to calculate blade moments and the proposed method was validated through numerical simulations.

In another study, a digital twin solution was introduced specifically for the floating OWT TetraSpar prototype [106]. The digital twin was created using a physics-based model and a Kalman filter to estimate the aerodynamic loads, wind speed, and section loads along the tower. Using available OWT measurements including power, pitch, rotor speed, tower acceleration, inclinometers, and GPS sensors, the digital twin employed Kalman filtering to estimate the OWT's structural states. Advancements such as the Augmented Kalman Filter (AKF) have extended the capabilities of traditional Kalman filtering techniques, enabling joint estimation of unknown states and inputs for enhanced load assessment [88]. An AKF input-state estimation method was applied to an OWT with a lattice foundation for simulation and estimation [89]. A separate study compared KF with joint input-state estimation and modal expansion method for estimating dynamic strain time history at the base of an OWT model [90]. An AKF with a reduced model was implemented to assess the loads on a wind turbine [91]. A substructure-based input-state estimation framework using AKF was proposed in [92]. Furthermore, efforts to address parameter uncertainty within turbine models have led to the development of sophisticated techniques like the Extended Kalman Filter (EKF) [36, 95-98] and Unscented Kalman Filter (UKF) [99-102], enabling joint input-state-parameter estimation to capture dynamic variations in operating conditions.

In contrast to physics-based models, data-driven approaches offer an alternative methodology for input load estimation in offshore wind turbines. These methods rely on abundant, high-quality data, perform fast, provide accurate real-time results, and are scalable. Using these methods, a simple surrogate model can be created for the complex inverse problem of input load estimation, approximating the behavior of the actual system. One such method is linear regression, used in [81] for real-time load estimation of the Goldwind 6 MW wind turbine. This regression model was trained using the training data obtained from simulations conducted in Bladed. These simulations

parameterize the relationship between control input signals (such as generator speed, generator torque, nacelle fore-aft and side-side accelerations, and pitch angle) and structural loads. Meanwhile, other researchers used artificial neural networks (ANNs) for wind turbine load assessments [82]. Their study, based on a database of high-fidelity wind load simulations with six probabilistic wind field parameters as inputs, demonstrated the superiority of ANNs over polynomial chaos expansion models in terms of accuracy and speed. In a similar study [107], researchers showed that fatigue load estimation can be performed for individual turbines within a wind farm using SCADA measurements and an aeroelastic turbine model. The paper suggests that if a certain amount of SCADA measurements are available, load mapping can be performed by training a data-driven regression model using ANNs. Moreover, a detailed comparison of commonly used surrogate modeling techniques in terms of accuracy and computational requirements was performed in [78]. The study defined and demonstrated a procedure for quick assessment of site-specific lifetime fatigue loads using simplified load mapping functions trained on a database with high-fidelity load simulations. Furthermore, Genetic Algorithm Back Propagation [108], Support Vector Machines [109], and Gaussian Processes [110] were explored for predicting mooring effective tension and pitch motion, and the results were compared with the real sea test data in [111]. Lastly, building surrogates to probabilistically predict site-specific loads on an offshore wind turbine was done in [112]. They introduced an approximation of the conditional probability distribution function of the response, enabling the inference of hyperparameters of the probability density function from noisy data. In another work, researchers introduced a deep learning-based evolutionary model to improve wind speed prediction accuracy. This approach combines bidirectional long short-term memory (LSTM) networks with an evolutionary decomposition technique and generalized normal distribution optimization. The

proposed hybrid approach was trained and tested on data gathered from an OWT installed at a Swedish wind farm located in the Baltic Sea with two forecasting horizons: ten minutes ahead and one hour ahead. The results showed promising accuracy compared to other methods [113]. A regularized Hidden Markov Model framework for wind speed prediction was applied to data from a 6 MW offshore wind turbine located in the North Sea [114]. This model incorporated a "smart" initialization strategy and regularization to improve the stability and accuracy of wind speed predictions using structural response measurements. The proposed HMM-based method outperformed traditional models such as k-Means and Gaussian Mixture Models, demonstrating its effectiveness in accurately estimating wind speed from structural data. In another study, a multi-task learning and transfer learning was applied to improve condition monitoring of offshore wind turbines with limited data. This study's deep neural network model, with hard parameter transfer, enhanced gearbox vibration anomaly prediction and demonstrated better performance over conventional models [115]. Another study developed a machine learning framework to predict structural fatigue loads and power output in offshore wind farms. By using control settings, turbine characteristics, and wind conditions as inputs, the framework focused on predicting the equivalent thrust, a key indicator of structural fatigue. The study showed that machine learning models can effectively enhance the accuracy of structural fatigue predictions, which are crucial for improving the reliability and safety of offshore wind operations [116].

Data-driven approaches have proven to be useful, fast, and accurate for input load estimation of offshore wind turbines. Nevertheless, they are also subject to several limitations: they require a large volume of high-quality data to represent the system accurately; the SCADA data necessary for training these models is often confidential; where data is available, it may lack the necessary features for training a data-driven model; and data can be noisy and complex. Consequently, there

is a premium on data-driven models that can be built with minimal data [117], and while purely data-driven models may align well with observations, they can lead to physically inconsistent or implausible predictions [118]. Data-driven models may have limitations in extrapolating beyond the datasets with which they were trained, and they may not always provide detailed insights into the specific physical parameters that influence a system's behavior.

Physics-based approaches can be readily adjusted to operate at varying levels of complexity, they allow for rigorous parametric evaluation, and they can be used to make predictions for new situations. Physics-based approaches are, however, by definition limited to the parameters included within their framework. Scenarios where the physics is not fully understood or described may lead to failures in identifying and addressing new problems and behaviors. Another important challenge is the computing time required for solving complex mathematical models. Physics-based models often necessitate low-fidelity models to achieve computational times that are low enough for real-time input load estimation, which introduces errors in the results. Furthermore, solving inverse problems with hidden and complicated physics, like input load estimation of OWTs, can be prohibitively expensive [119].

Considering these advantages and limitations, hybrid approaches that combine the conceptual clarity of physics-based methods with the numerical efficiency of data-driven methods have the potential to offer the best of both worlds. Hybrid models will differ depending on the type of problem, data quality and quantity, and the accuracy of a given physical model, and the desired computational efficiency. A recent numerical study proposed a physics-informed neural network as a surrogate model to bypass the traditional methods for aerodynamic load estimation, integrating physical constraints into the neural network training to achieve enhanced computational efficiency

and accuracy [120]. However, their model was primarily validated numerically and lacked experimental validation, which may limit its practical applicability. The study described this present chapter achieves a similar level of computational efficiency through the application of hybrid transfer learning, and is distinguished for its robust behavior and validation through high-quality measurements of OWT structural strains and accelerations.

The primary objective of this research is to develop a hybrid surrogate model for input load estimation on OWTs using measured data from a 6MW OWT. This model aims to combine the strengths of data-driven and physics-based approaches to address the limitations of our previous study, which was both computationally expensive and lacked physical complexity in modeling the blades and controller [121]. The model proposed herein leverages the simplicity and physical characterization of the structural model while integrating the power of neural networks to learn the relationship between structural responses and input load estimation. This approach addresses an inverse problem, aiming to provide an accurate and more efficient solution for real-time input load estimation of OWTs using the measured structural response of the system. The framework leverages ideas from transfer learning, multitask learning, and Monte Carlo dropout, which will be extensively discussed further in the methodology section. Essentially, the model is pretrained on simulated data, then knowledge is transferred to real-world data, and this is done across different regions of power production. As a last step, the uncertainty in estimation is assessed using Monte Carlo dropout.

The structure of this chapter is organized as follows: Methodology, Data, Results and Discussion, and Conclusion. The Methodology section will outline the development of the hybrid surrogate model. This will be followed by a description of the dataset used for training and

validation in the Data section. Subsequently, the results and findings of the research will be presented, demonstrating the effectiveness and accuracy of the hybrid surrogate model. Finally, the chapter will conclude by summarizing the findings and suggesting directions for future research.

5.2 Methodology

This section will explain the proposed hybrid surrogate model framework. The framework consists of three major steps. The first part is based on a transfer learning approach, involving two pre-training phases and fine-tuning. The second part addresses input load estimation across different operational regions of the wind turbine using multitask learning. The third part includes uncertainty quantification using the Monte Carlo dropout method. As a supervised learning approach, the framework requires ground truth data for model training. All these parts will be further elaborated in the following subsections.

5.2.1 Transfer Learning

In the context of offshore wind turbines, obtaining a large amount of labeled real-world data for training is often impractical due to the high costs and logistical challenges of data collection. However, simulated data can be generated relatively easily and used to train models. The proposed framework for input load estimation leverages transfer learning to address the scarcity of experimental data.

Transfer learning is a machine learning technique where knowledge learned in one domain, referred to as the source domain $\mathfrak{D}_S = (\mathbf{X}_S, \mathbf{Y}_S)$, is transferred to another, referred to as the target domain $\mathfrak{D}_T = (\mathbf{X}_T, \mathbf{Y}_T)$ where $\mathbf{X}$ are the input features and $\mathbf{Y}$ are the labels. This process allows

the model to leverage the patterns and insights gained from the source domain to improve its performance for the defined task in the target domain. It is particularly useful when the target domain has limited data, as it allows the model to generalize better by building upon what is learned from the source domain [122, 123]. Transfer learning can be categorized as either homogenous or heterogeneous [124]. Homogenous transfer learning occurs when the source and target domains share the same feature space $\mathbf{X}_T = \mathbf{X}_S$ and the label space $\mathbf{Y}_T = \mathbf{Y}_S$, whereas heterogeneous transfer learning is for when source and target domains have different feature spaces $\mathbf{X}_T \neq \mathbf{X}_S$ and/or label space $\mathbf{Y}_T \neq \mathbf{Y}_S$.

Inspired by the principles of homogeneous transfer learning, we identified a practical approach to tackle our problem. Homogeneous transfer learning typically involves transferring knowledge between datasets where the features and labels have the same meaning and representation, e.g., two identical structures. The concept could be applied to our scenario, where simulated data from a model of the structure and real measured data from the actual structure are available. By training the models on simulated data and fine-tuning them with measured data, we leverage the strengths of both data sources. This innovative transfer learning strategy is specifically helpful in addressing the challenges posed by limited real-world data availability and the inherent complexities of input load estimation in offshore wind turbines.

In our problem context, the source domain is the simulated data, where the physical relationship between structural responses and input loads is comprehensively modeled. This domain provides valuable insights into the underlying dynamics of the system but is constrained by certain limitations inherent to numerical simulations. On the other hand, the target domain is the measured

data, and the proposed framework seeks to transfer knowledge from the source domain to the target domain.

The proposed methodology uses transfer learning principles to adapt a pre-trained Long Short-Term Memory (LSTM) model [125], trained on the numerical simulations, to the task of input load estimation on real offshore wind turbines. The input data for this study is defined as $\mathbf{X}_S \in \mathbb{R}^{(n_s \times T) \times n_f}$, representing the simulated input data, where n_f denotes the number of features, n_s represents the number of simulated datasets and T is the length of each dataset. Correspondingly, $\mathbf{Y}_S \in \mathbb{R}^{(n_s \times T) \times n_p}$ represents the simulated output data, where n_p denotes the number of output variables. Additionally, $\mathbf{X}_r \in \mathbb{R}^{(n_r \times T) \times nf}$ represents the real input data derived from n_r real datasets, while $\mathbf{Y}_r \in \mathbb{R}^{(n_r \times T) \times np}$ represents the corresponding real output data.

Given the temporal dynamics inherent in our problem domain, this study uses LSTM, a specialized cell type of Recurrent Neural Network (RNN) [126, 127]. The proposed framework includes two phases. The initial phase involves pre-training the network with simulated data and establishing a baseline understanding of the correlation between input loads and structural responses. This pre-trained network acts as a bridge for knowledge transfer, facilitating the adaptation to the specificities of the real-world problem domain during the fine-tuning process, which is the second phase.

Phase I: Pre-training on the Simulated data

In the initial phase, an LSTM neural network with L total layers is pre-trained on the simulated numerical data defined above. Formally, the LSTM predicts the output $\widehat{\mathbf{Y}}_s$ given the input $\mathbf{X}_s$ using the following formulation:

$$\widehat{\mathbf{Y}}_s = \text{LSTM}\left(\mathbf{X}_s; \left[\mathbf{W}^{(1:lz)}, \mathbf{W}^{(lz+1:L)}\right]\right) \quad (5.1)$$

Where $\mathbf{W}^{(1:lz)}$ and $\mathbf{W}^{(lz+1:L)}$ are the weights of the first lz layers and the remaining layers of the LSTM network, respectively. The goal of this phase is to minimize the loss function by adjusting all the weights of the network. Formally, it can be written as:

$$[\mathbf{W}^{*1:lz}, \mathbf{W}^{*lz+1:L}] = \text{argmin}\, loss\left(\widehat{\mathbf{Y}}_s, \mathbf{Y}_s\right) \quad (5.2)$$

Phase II: Transfer learning with Real data

After completion of Phase I, the trained LSTM model with the simulated data is transferred and reused with the first lz layers frozen. Then, the LSTM model is fine-tuned using the real data while keeping the frozen layers from Phase I frozen. The fine-tuning process involves updating the weights of the remaining layers to adapt to the information in the real data. The LSTM predicts the output $\widehat{\mathbf{Y}}_r$ given the real input $\mathbf{X}_r$, using the following formulation:

$$\widehat{\mathbf{Y}}_r = \text{LSTM}(\mathbf{X}_r; [\mathbf{W}^{*1:lz}, \mathbf{W}^{lz+1:L}]) \quad (5.3)$$

Similar to Phase I, the loss function is minimized, but this time, only on the last L-lz layers' weights.

$$[\mathbf{W}^{**z+1:L}] = \text{argmin}\, loss\left(\widehat{\mathbf{Y}}_r, \mathbf{Y}_r\right) \quad (5.4)$$

With this, the trained model will have $\mathbf{W}^* = [\mathbf{W}^{*1:lz}, \mathbf{W}^{**lz+1:L}]$ assigned weights. The framework is show in Figure 5.1.

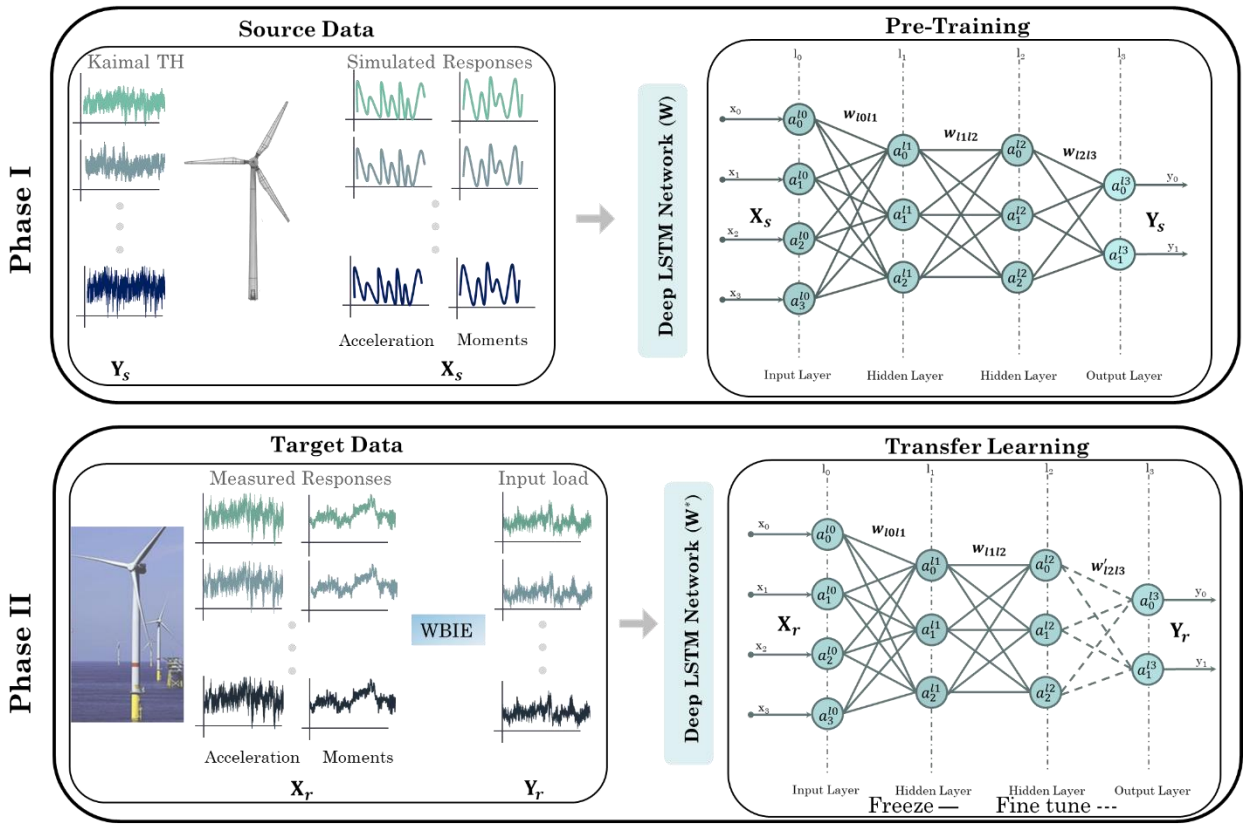


Figure 5.1 Surrogate model for input load estimation – Transfer Learning: X & Y are the vibration responses and input loads respectively.

5.2.2 Multitask Learning

Offshore wind turbines operate as self-regulated structures, dynamically adjusting their loading through controller mechanisms. This regulation depends on wind speed and power production levels. Based on these parameters, the behavior of the turbine can be divided into three distinct regions. The first region occurs below the cut-in wind speed threshold, marking the onset of power generation. The second region spans between the cut-in and rated wind speeds, during which the turbine is actively generating power. Finally, the third region extends from the rated wind speed to the cutoff point, where the turbine operates at maximum capacity. In developing our model, it is important that it can accommodate the diverse behaviors exhibited across different regions. As

there is no wind in region one, the focus of this study is on regions two and three. Thus, a multitask learning approach [128, 129] is taken here to train a comprehensive network, including two of these three regions. Multitask learning based on different regions of performance is not addressed in the literature. This approach allows us to leverage shared information across the regions while still maintaining the unique characteristics of each region. Three tasks for three different regions are defined $T = \{T_2, T_3\}$ and each task has its own training data, including simulated data D_s for pre-training in Phase I, and real data D_r for Phase II:

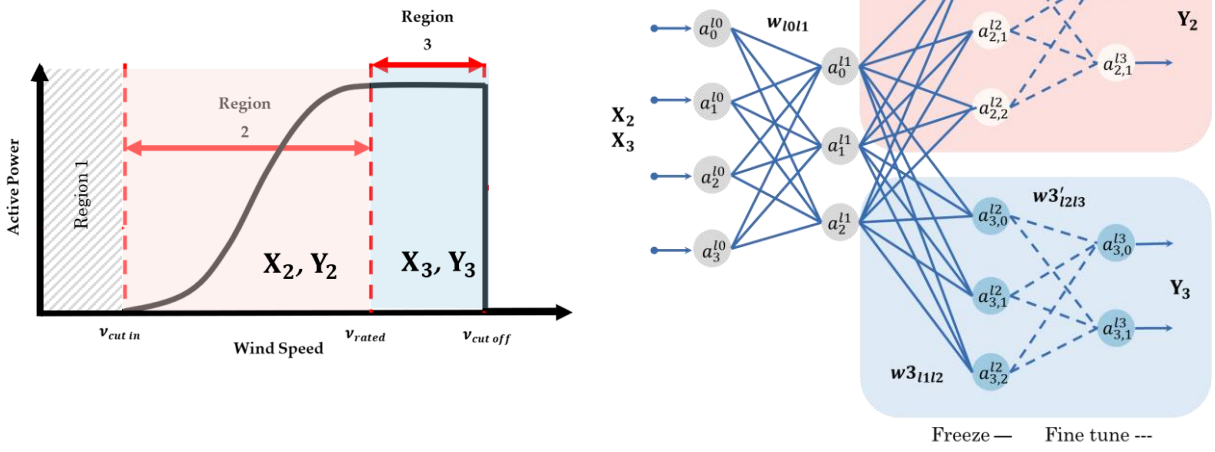
$$\mathbf{D}_s = \{(\mathbf{X}_{s2}, \mathbf{Y}_{s2}), (\mathbf{X}_{s3}, \mathbf{Y}_{s3})\} \tag{5.5}$$

$$\mathbf{D}_r = \{(\mathbf{X}_{r2}, \mathbf{Y}_{s2}), (\mathbf{X}_{r3}, \mathbf{Y}_{r3})\} \tag{5.6}$$

with each pair corresponding to a specific task. The objective of multitask learning is minimizing the sum of the loss functions across the tasks, which is written as:

$$[\mathbf{W}_2^*, \mathbf{W}_3^*] = \arg\min \sum_{i=2}^{3} loss\,(\mathbf{D}_i, \mathbf{W}_i) \tag{5.7}$$

The multitask learning part of the framework is shown in Figure 5.2.



a) Power curve and regions *b) Multi-tasking scheme based on regions*

Figure 5.2 Surrogate model for input load estimation – Multitask Learning: X & Y are the vibration responses and input loads respectively.

5.2.3 Monte Carlo dropout

Although neural networks are effective prediction tools, they often do not account for uncertainty of the trained model, producing deterministic predictions where a set of model parameters is optimized to match the data and remains unchanged after training. Monte Carlo dropout [130] is a technique utilized in neural networks to estimate prediction uncertainties with respect to the model. To quantify the uncertainty of neural network model, let's assume the predictive distribution to be $p(\boldsymbol{y}|\mathbf{x}, \mathfrak{D})$ where **y** is the target, **x** is the input feature for prediction, and $\mathfrak{D}$ is the training domain. The variance of this predictive distribution can be interpreted as the prediction uncertainty. The predictive distribution can be calculated as:

$$p(\boldsymbol{y}|\mathbf{x}, \mathfrak{D}) = \int_{\Omega} p(\boldsymbol{y}|\mathbf{x}, \mathbf{W}) p(\mathbf{W}|\mathfrak{D}) d\mathbf{W} \tag{5.8}$$

where $p(\boldsymbol{y}|\mathbf{x}, \mathbf{W})$ is the likelihood, $p(\mathbf{W}|\mathfrak{D})$ is the posterior of weights of the NN given the training data, and **W** is the weight matrix. A way to calculate this predictive distribution approximately is as follows [130]: During training, a dropout mask randomly turns off a certain percentage of the neurons, helping to prevent overfitting by discouraging the network from relying too heavily on specific neurons. This mask is a binary vector with the same dimension as the number of neurons in a particular layer. The values in the mask are sampled from a Bernoulli distribution with a probability of (1-p), where p indicates the dropout rate. Masking the neurons creates different configurations of the NN, and each configuration corresponds to a different sample from the posterior distribution $p(\mathbf{W}|\mathfrak{D})$. This way, the posterior is approximated to $q(\mathbf{W})$ and the predictive distribution can be approximated using variational inference to:

$$p(\boldsymbol{y}|\mathbf{x},\mathfrak{D}) \approx q(\boldsymbol{y}|\mathbf{x}) = \int p(\boldsymbol{y}|\mathbf{x},\mathbf{W})q(\mathbf{W})d\mathbf{W} \tag{5.9}$$

Assuming that the likelihood $p(\boldsymbol{y}|\mathbf{x},\mathbf{W})$ follows a Gaussian distribution with $\widehat{\boldsymbol{y}}(\mathbf{x},\mathbf{W})$ mean and $\tau^{-1}\mathbf{I}_D$ covariance, the mean and variance of the approximate predictive distribution can be approximated as:

$$\mathbb{E}_{q(\boldsymbol{y}|\mathbf{x})}(\boldsymbol{y}) \approx \frac{1}{T}\sum_{t=1}^{T}\widehat{\boldsymbol{y}}(\mathbf{x},\mathbf{W}_t) \tag{5.19}$$

$$\mathrm{Var}_{q(\boldsymbol{y}|\mathbf{x})}(\boldsymbol{y}) \approx \tau^{-1}\mathbf{I}_D + \frac{1}{T}\sum_{t=1}^{T}\widehat{\boldsymbol{y}}(\mathbf{x},\mathbf{W}_t)^T\widehat{\boldsymbol{y}}(\mathbf{x},\mathbf{W}_t) - \mathbb{E}_{q(\boldsymbol{y}|\mathbf{x})}(\boldsymbol{y})^T\mathbb{E}_{q(\boldsymbol{y}|\mathbf{x})}(\boldsymbol{y}) \tag{5.11}$$

The mean and variance can be calculated by T forward passes through the network, calculating the average and sample variance. During testing, we perform multiple forward passes with different dropout masks, each time creating a different configuration of the trained network by randomly dropping neurons. For each different configuration, we obtain a different output, effectively generating a distribution of outputs. This means that by applying dropout during inference, we are sampling from an approximate posterior distribution of the network's weights, which allows us to estimate the uncertainty in the model's predictions. This enables us to calculate the mean and standard deviation of predictions, providing a measure of uncertainty in the model's predictions. Figure 5.3 illustrates the difference between the predictions from a general neural network and one with Monte Carlo dropout. It is important to note that this approach is not a Bayesian framework but rather an approximation of neural network model uncertainty, which means it only reflects how the model's distribution affects the results, without involving a prior or a real posterior.

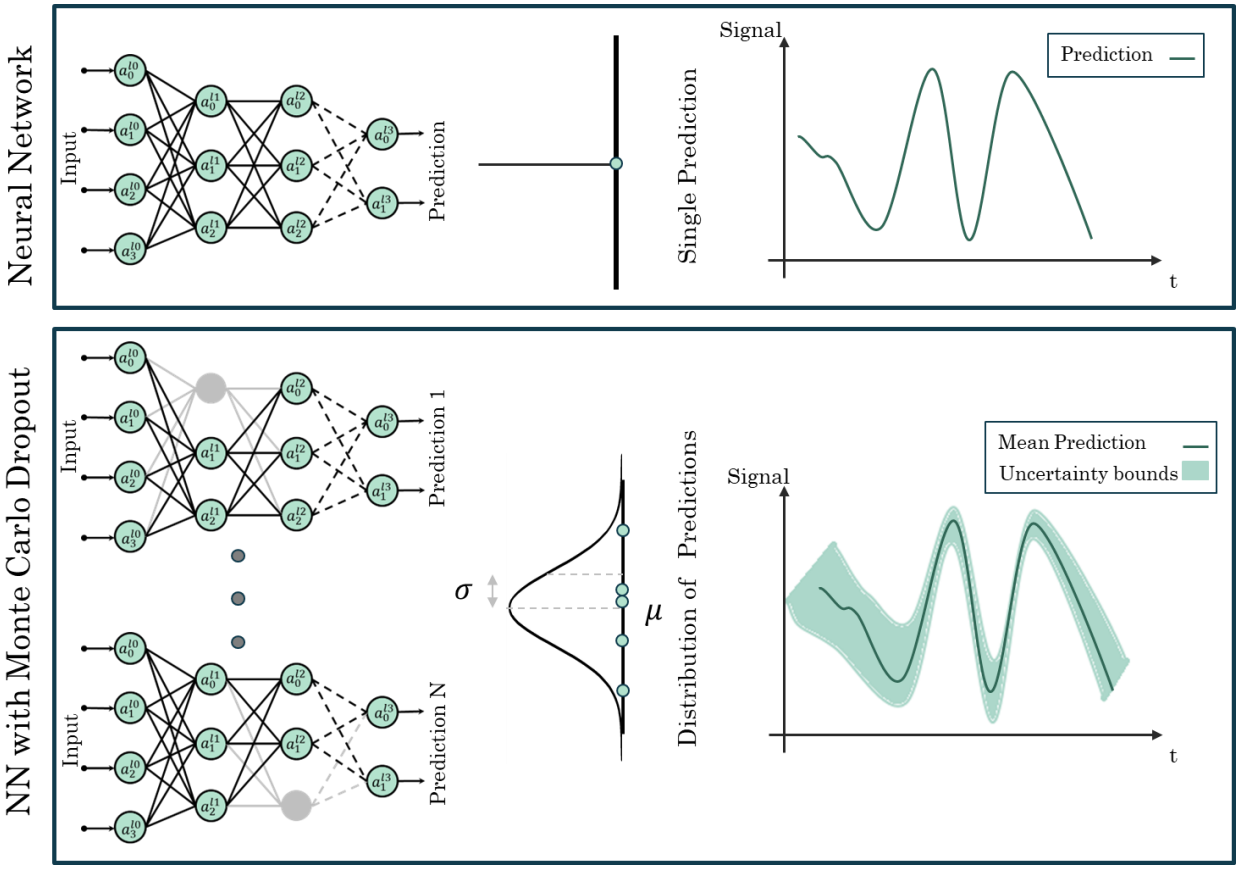


Figure 5.3 Monte Carlo Dropout scheme

5.3 Data

The data used in this study consists of two sources: simulated data and real measured data obtained from a 6 MW offshore wind turbine. Both sources are explained in what follows.

Simulated Data

The well-known simple Kaimal Spectrum is used in this study to generate wind spectrum, describing the frequency content of the wind speed. The formulation is taken from [131], which takes three parameters to calculate the spectrum: the height above the ground, the mean wind speed, and the standard deviation to model the turbulence. We are interested in the wind load at the hub height level and to have all different wind speeds for different regions; 7 wind speeds for

region 3 from 10 to 16 m/s and from 6 to 9 m/s wind speeds for region 2 are selected. The shear velocity is assumed to be 2 m/s. Then, the wind speed spectrums are used to generate wind load time histories at the hub height. A sample of the generated wind load is shown in Figure 5.4. This simulated wind load is then used as input on the wind turbine's finite element model, described in the next paragraph, to simulate structural responses.

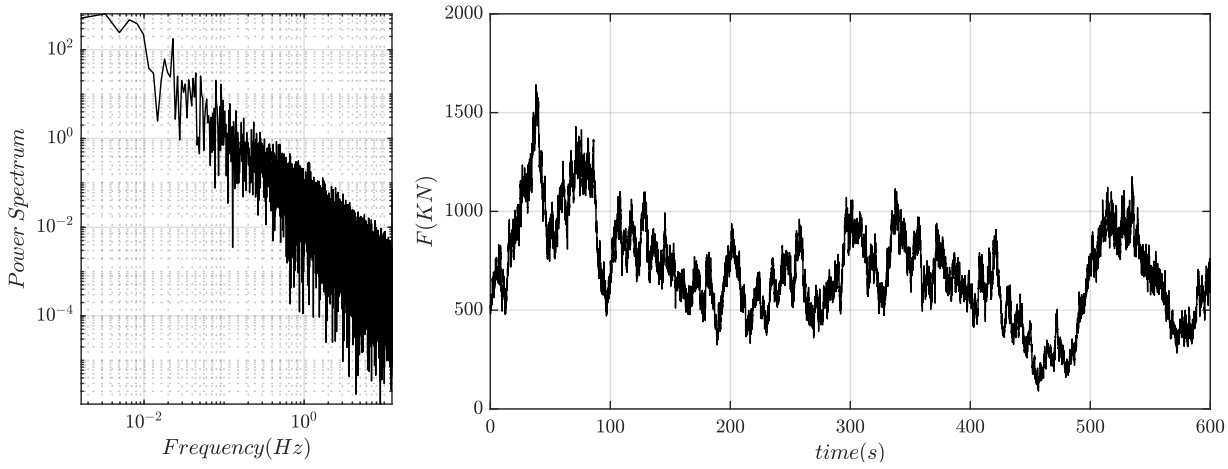


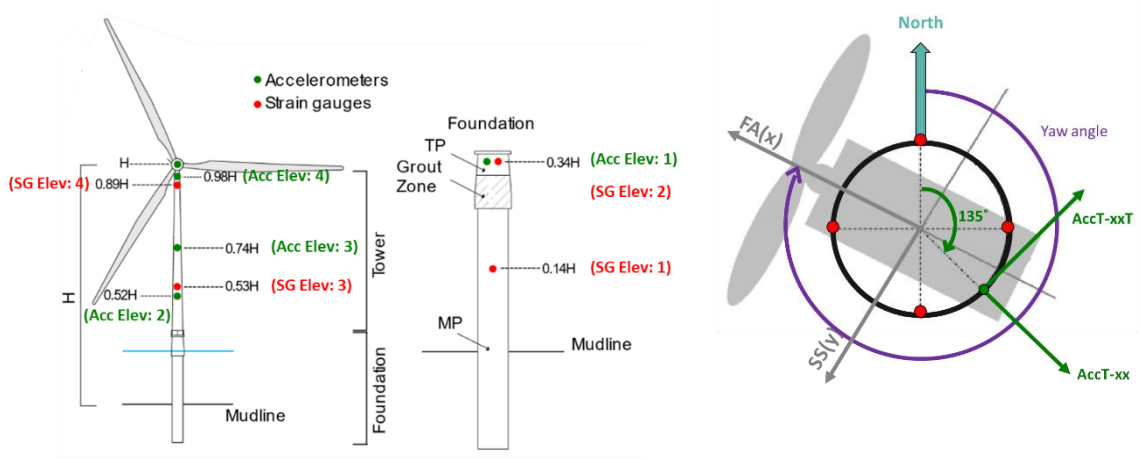
Figure 5.4 An example of the simulated wind load with mean wind speed = 13 m/s

A linear elastic FE model of this turbine is constructed using the structural analysis software OpenSees [72]. The model is built using the available structural design details of the tower and monopile, which consist of the material properties, tower geometry, foundation, and masses. Linear elastic-beam column elements with varying cross sections are employed to model the tower and foundation. Tower and monopile masses are lumped at their end nodes. The Rotor Nacelle Assembly (RNA) is modeled using a lumped mass and three mass moments of inertia at the center of gravity of the RNA at the top of the tower. Soil-structure interaction is modeled using a decomposed macro element stiffness matrix, separating lateral and rotational stiffnesses. Further details about the modeling of this OWT and its soil springs can be found in [77]. It should be noted

that the controller and the blades are not modeled, which results in an inconsistency between what the model can predict for the structural responses and what the structure would experience.

Real Measured Data

The available real data for this study is measured from an operating 6 MW monopile-supported wind turbine located in the North Sea. The turbine is instrumented with 8 perpendicular accelerometers across four levels, 16 strain gauges across four levels, and SCADA at the nacelle. Two weeks of continuous data are available, recorded at a frequency of 25Hz for vibration channels and 10Hz for SCADA. The strain gauges data are calibrated. Subsequently, they are used to calculate the resultant bending moments in two perpendicular directions (North and East) and reported in this format. The sensor layout is illustrated in Figure 5.5.



(a) (b)

Figure 5.5 Instrumentation layout a) Elevation View b) Top View

In the absence of direct input load measurements from the turbine's operational data, an additional step is required in the proposed framework. As a supervised learning approach, the framework requires measured responses and corresponding input loads for model training. To address this requirement, a recursive window-based Bayesian input estimator is employed [121].

This method, described in Chapter 4, enables the estimation of input loads based on measured structural responses. Subsequently, these predicted input loads are used in the dataset, providing training data.

5.4 Results

In this section, the outcome of this study is presented. The input features in this study are the measured moments in fore-aft (FA) and side-to-side (SS) directions, along with the measured acceleration in the same directions. The output of the network is the estimated input load of the structure in the FA direction, or in other words, the thrust load of the turbine. This section has three subsections. In the first one, the method is trained and validated for high wind speeds, which corresponds to region three of operation for the turbine. The next subsection extends the analysis to include multitask learning for regions two and three. In the last subsection, the uncertainty quantification of the predictions using the Monte Carlo dropout method is explored.

5.4.1 Region three

In the first step of training and testing the proposed framework, we focused on region three of the operational condition of the turbine. The two phases of this framework are explained in the following subsections.

Phase I training

A dataset of 700 minutes of data corresponding to 70 different wind loads was generated. These wind loads were derived from 7 different wind speeds, and 10 different seeds per wind speed were used for generating data. These generated wind loads have been utilized to simulate structural responses, forming a dataset for training and testing in phase I. The arrangement of the generated

loads ensures that the training data includes different scales of wind load. The data has been split into a 60-40 ratio for training and testing. An LSTM network with 6 hidden layers has been employed for this task. The first 3 layers contain 256 neurons each, and the last 3 layers contain 128 neurons each. Dropout layers with a dropout rate of 0.2 have been incorporated into the network. The Adam optimizer has been used with a learning rate of 0.0001, and the model has been trained for 30 epochs with a batch size of 64.

The results for phase I are shown in Figure 5.6 and Figure 5.7, where the vertical black line separates the training and testing parts of the data. Four different accuracy metrics are used to assess the performance of the phase I model: Root Mean Square Error (RMSE), Relative Root Mean Square Error (RRMSE), Mean Absolute Error (MAE), and Time Response Assurance Criterion (TRAC). Each metric is chosen for its ability to provide unique insights into the regression problem. RMSE measures the model's overall fit to the data, while MAE offers a robust alternative that is less sensitive to outliers. RRMSE, a normalized form of RMSE, proves useful when comparing different datasets, such as the different training phases in this problem. Lastly, TRAC measures how well the model captures the dynamic behavior of the time series. For a more detailed understanding of the trained model’s performance, the reader is referred to Table 5.1. Overall, the metrics indicate that the model demonstrates no signs of overfitting and performs well on unseen data. RRMSE values are also consistently low for training and testing datasets, suggesting a strong fit. While MAE and RMSE metrics reveal slightly higher values for the testing data, indicating the presence of more outliers compared to the training data, the model's performance remains robust across both sets. Yet, all these metrics are only achieved on simulated data.

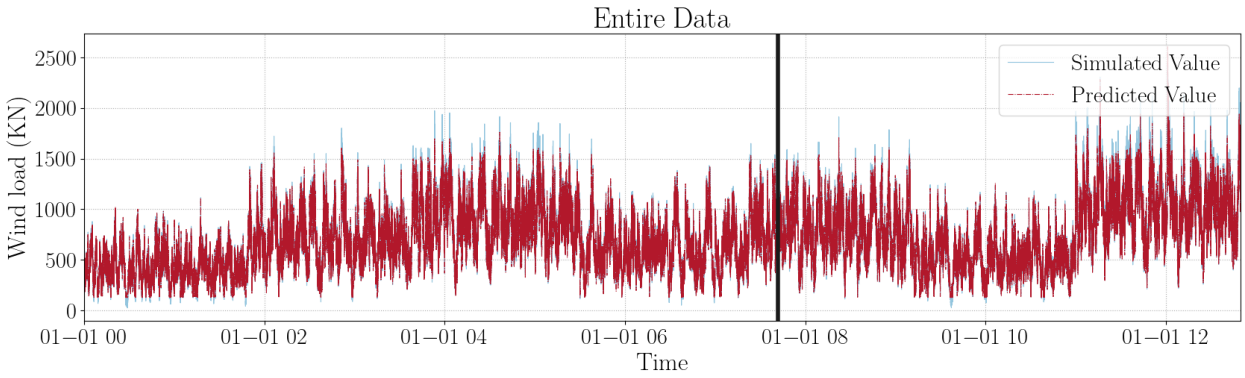


Figure 5.6 Phase I training and testing results

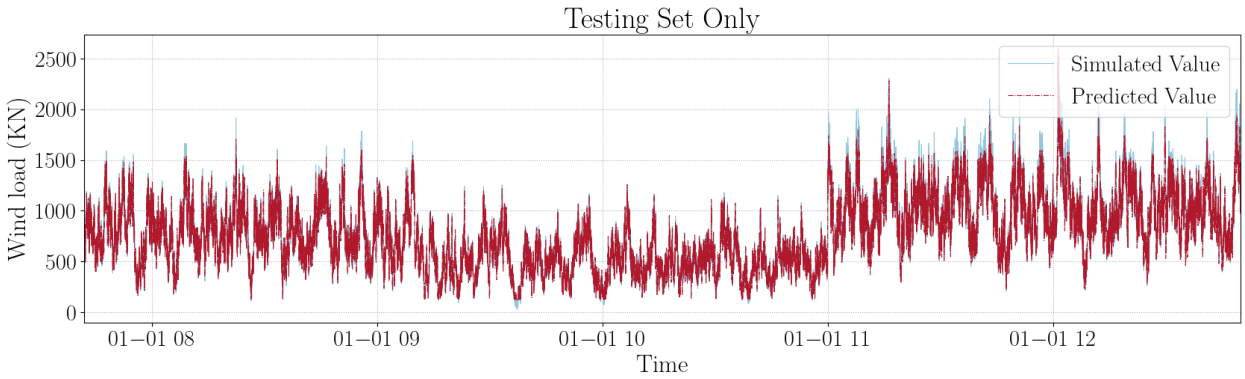


Figure 5.7 Zoomed in results in the testing section

Table 5.1 Accuracy metrics for training and testing sets for phase

	Training set	Testing set
RRMSE (%)	3.7%	3.5%
RMSE (KN)	26.94	30.55
MAE (KN)	20.96	23.29
TRAC	0.99	0.99

With phase I trained model yielding promising results on the simulated data, the next step is to assess the model's performance on real-world data. Figure 5.8 shows the prediction of phase I model when applied to real data. The accuracy metrics are also reported as: RRMSE = 22.5% RMSE=193.53 (KN) MAE = 169.1 7(KN) and TRAC =0.98. Notably, the TRAC value indicates

a high level of accuracy in capturing the dynamic behavior of the input load, which holds significant value for practical applications. However, it's observed from both the RMSE and MAE values that the mean thrust load deviates by approximately 200 KN. This deviation can be attributed to the fact that the numerical model does not include the controller responsible for regulating the turbine's thrust load under high wind speeds to prevent structural damage. As a result, the phase I trained model falls short in accurately predicting the mean thrust loads.

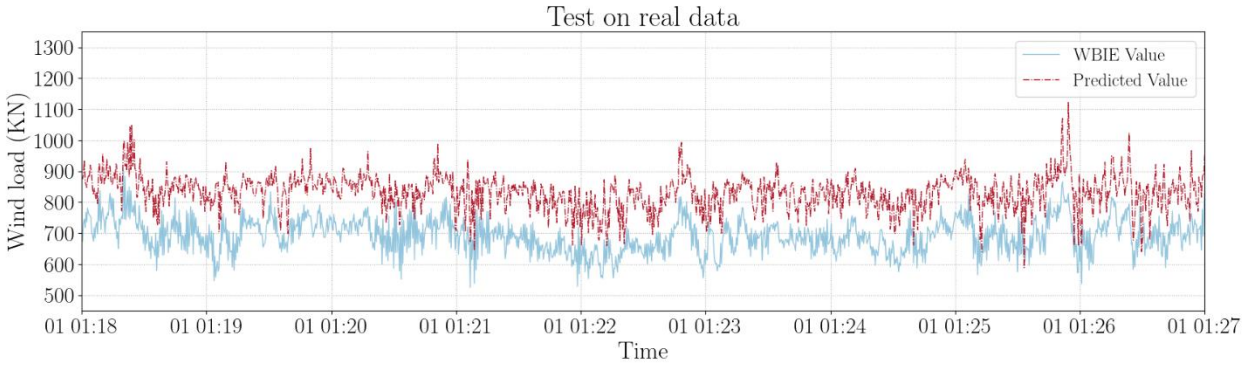


Figure 5.8 Phase I model prediction compared to real data

Phase II fine tuning

At this stage, 100 minutes of real data were utilized to fine-tune the network. Given the absence of measured input load data for this study, the output data for phase II training was obtained using the WBIE method. Consequently, 10 different datasets were selected from instances when the turbine operated at full capacity power production under high wind speeds. WBIE was then applied to these datasets, resulting in 10 sets of real wind load time histories for fine-tuning. As real data availability is limited, the training and testing split for this case is 80-20, ensuring enough training data is available for the model. To leverage transfer learning, the first three layers, assumed to have captured the general relationship between the structural responses and the input, were kept frozen.

Meanwhile, the last three layers were retrained using the measured data to capture nuanced details that were inadequately simulated and trained using the simulated data.

The Adam optimizer with a learning rate of 0.000001 was employed for optimization. This very low learning rate was chosen based on trial and error. The training was conducted over 200 epochs, with a stopping criterion based on the difference in validation loss between epochs. Training stopped when this difference fell below 0.0001 for 10 epochs. Given the limited availability of real data, attention was given to extracting as much detail as possible through transfer learning. Therefore, the first four layers of the network were kept frozen while the last two layers' weights were updated through fine-tuning. Also, the choice of learning rate and epochs was carefully tailored to maximize information extraction from the real data. Results are shown in Figure 5.9 and Figure 5.10. Table 5.2 summarizes the accuracy metrics of the model. After fine-tuning, the mean thrust load estimation showed significant improvement. The process of fine-tuning the pre-trained model with real-world data successfully bridged the gap between the FE model and the actual structure, addressing discrepancies observed in the initial Phase I results. This adjustment led to more accurate predictions, demonstrating the effectiveness of the transfer learning approach in enhancing model performance with limited real-world data. Although the metrics are slightly better for training results, testing results are not far off from training to call this case an overfitting case.

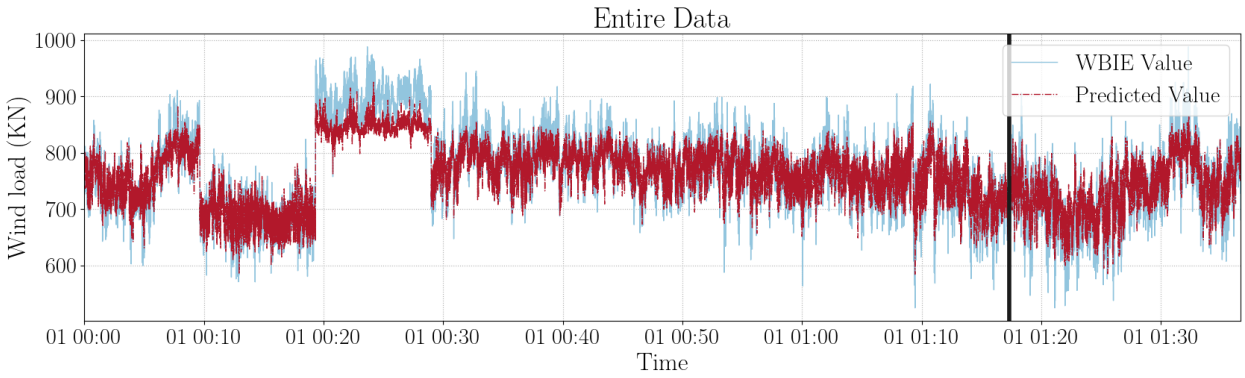


Figure 5.9 Phase II training and testing results

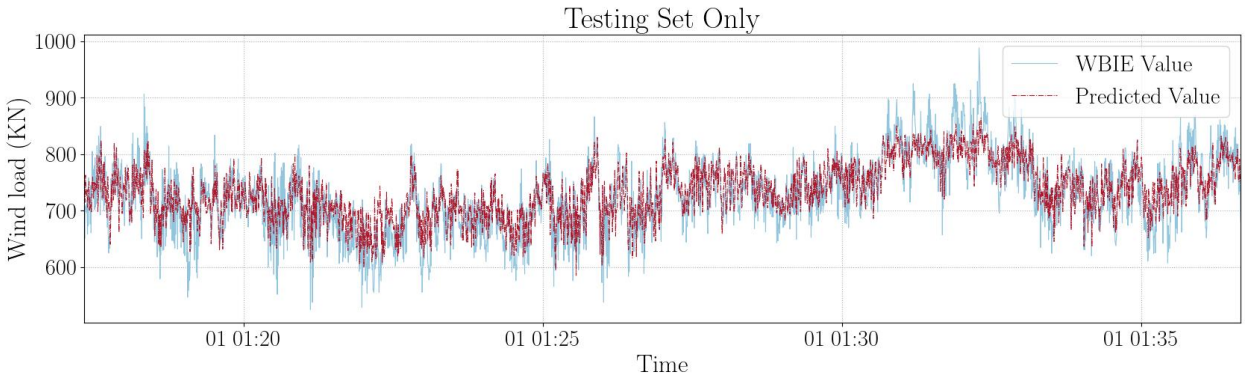


Figure 5.10 Zoomed in results in the testing section after fine-tuning

Table 5.2 Accuracy metrics for training and testing sets for phase II

	Training set	Testing set
RRMSE (%)	2.5%	4.3%
RMSE (KN)	19.78	31.21
MAE (KN)	15.30	24.62
TRAC	0.99	0.99

5.4.2 Multitask Learning

The multitask learning approach is applied to the data from regions two and three, which were simulated using the previously outlined methodology. Specifically, region three data remained consistent with previous simulations; just a reduced dataset was utilized compared to the single-

task approach. For region two, data were simulated with wind speeds ranging from 6 to 9 m/s, incremented at 0.5 m/s intervals. Each wind speed category comprised 10 sets of data, with each set spanning approximately 9 minutes.

In the multitask learning framework, the neural network layers were divided into two parts. The first half of the network focused on learning shared information across all tasks (regions), while the second half was dedicated to capturing region-specific nuances. Additionally, consistent with the transfer learning approach, the first three layers of the network were kept frozen to retain general information learned from previous phases. It's worth noting that the network architecture and hyperparameters remained the same as in the previous section.

The results for region three are shown in Figure 5.11 and Figure 5.12, where the training and testing datasets are visually separated by a vertical black line. The true values are shown with blue lines. Notably, the testing set shows superior performance compared to the training set. This is because the training dataset includes multiple datasets from different times to learn various conditions, resulting in larger errors. In contrast, the testing set, which covers less variability in wind speed and thrust load, shows smaller errors and better performance. While the multitask learning approach demonstrates slightly lower performance compared to the single-task approach for region three, this discrepancy can be attributed to the nature of the loss function in multitasking, which aims to minimize the sum of losses across tasks. Nevertheless, the benefit of having a single model capable of handling multiple tasks concurrently remains the same.

Moving to region two, the results are presented in Figure 5.13 and Figure 5.14. The training data is a collection of different datasets handpicked to reflect region 2 specifically. The jumps in the training data are coming from this. It's important to acknowledge that, in our scenario, where

the ground truth for outputs is derived rather than directly measured, any errors or assumptions inherent in this derivation process can impact the final results of the framework. Given the complexity of region two dynamics, influenced by factors such as wave interactions and dynamic controller adjustments, as well as potential sensor measurement errors, it's understandable that some discrepancies may arise in the training data. Nonetheless, despite these challenges, the framework demonstrates satisfactory performance. A summary of the accuracy metrics for multitask learning across both regions is provided in Table 5.3.

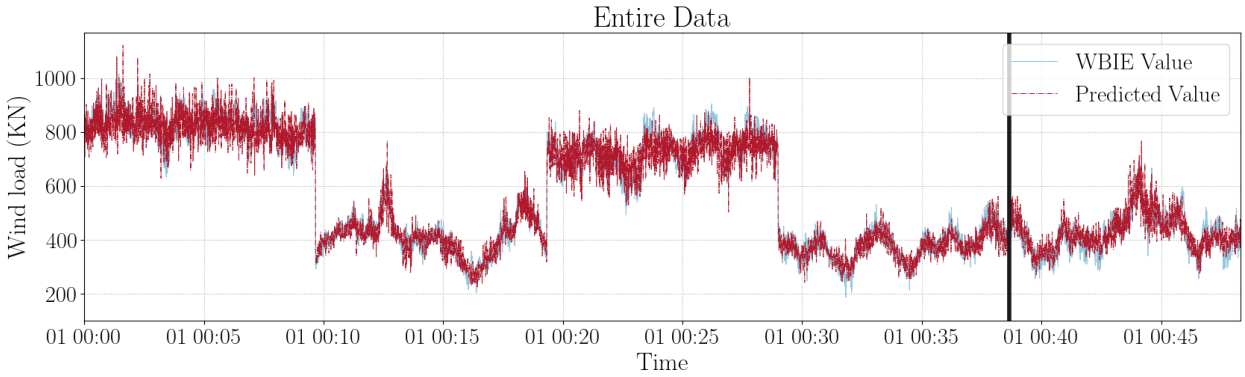


Figure 5.11 Region three results after fine-tuning using multitask learning

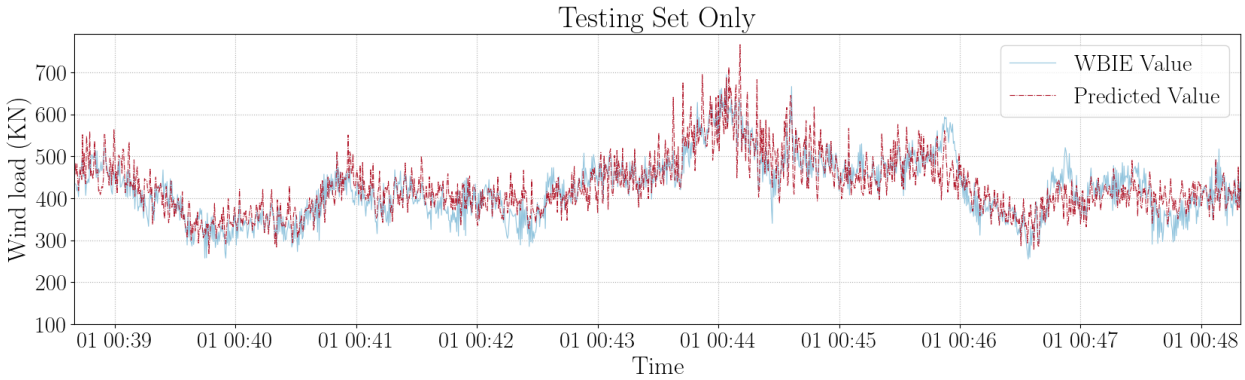


Figure 5.12 Zoomed in testing results for region three

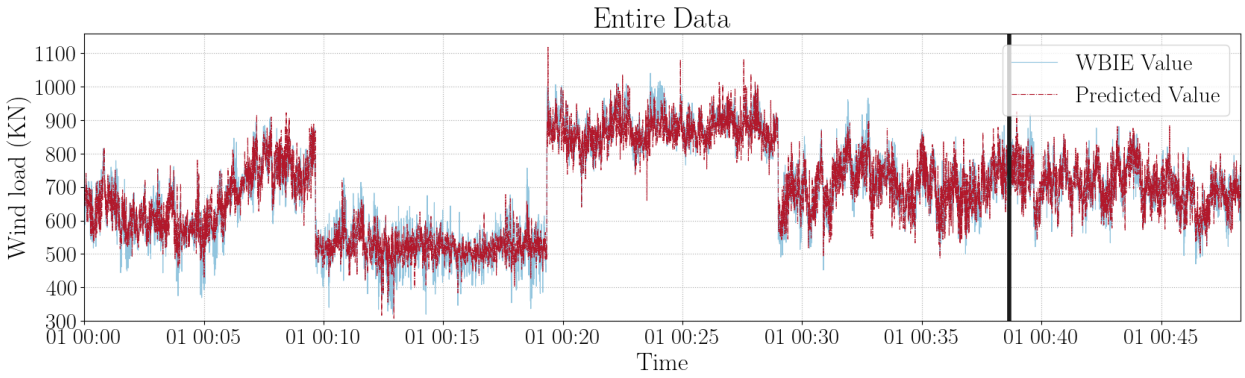


Figure 5.13 Region two results after fine tuning using multitask learning

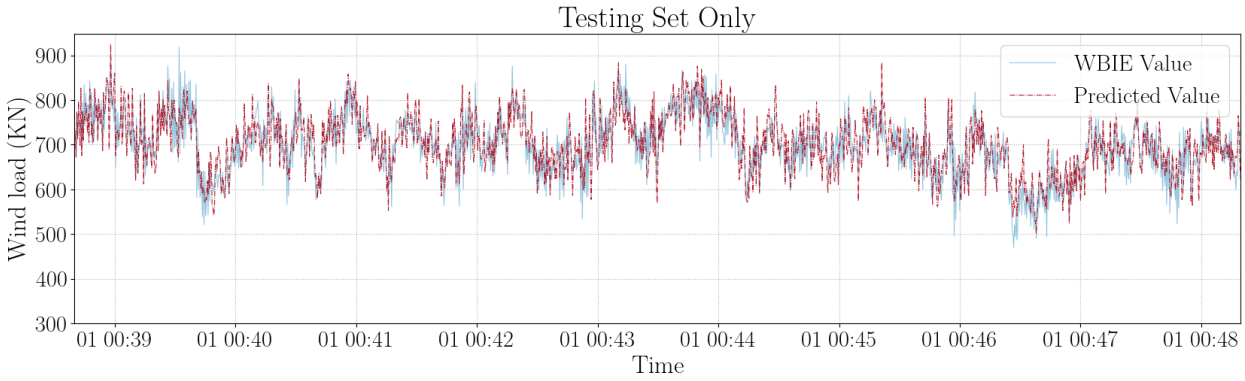


Figure 5.14 Zoomed in testing results for region two

Table 5.3 Multitask learning metrics for regions two and three

	Region 3		Region 2	
	Training set	Testing set	Training set	Testing set
RRMSE (%)	5.1	4.5	5.2	7.7
RMSE (KN)	35.76	31.39	32.55	33.24
MAE (KN)	28.34	25.03	24.26	26.03
TRAC	0.99	0.99	0.99	0.99

5.4.3 Uncertainty Quantification of the Neural Network

This section is focused on quantification of the uncertainty in the neural network model output by employing the Monte Carlo Dropout method. The importance of this process lies in its ability

to evaluate the reliability and robustness of the predictive models used. By quantifying uncertainty, a measure of confidence in the predictions made by the model is provided. The Monte Carlo Dropout method involves applying dropout during both the training and testing phases and performing multiple forward passes with different dropout masks, approximating the posterior distribution of model parameters and helping to estimate the uncertainty in the model's predictions.

As detailed in the methodology section, the Monte Carlo dropout method is extended to the inference phase, with 20 percent of neurons being randomly dropped in each layer during inference. Subsequently, 100 predictions are generated, from which the mean and standard deviation of predictions are computed for each time step. Figure 5.15 shows the uncertainty prediction results for region three. The uncertainty quantification findings align with observations from Figure 5.11, particularly the period between minutes 19 and 28, where predictions showed imperfections. Here, the uncertainty quantification reveals larger uncertainty during this timeframe, indicating less confidence in predictions compared to other data segments, including the testing set. A similar trend is evident in the results for region two, depicted in Figure 5.13 and Figure 5.16. In regions where multitask predictions were less accurate, Monte Carlo dropout analysis shows increased uncertainty, as depicted by the salmon-colored lines. This correlation underscores the utility of uncertainty quantification in identifying and characterizing areas of prediction ambiguity, thereby informing more nuanced interpretations of model performance and reliability.

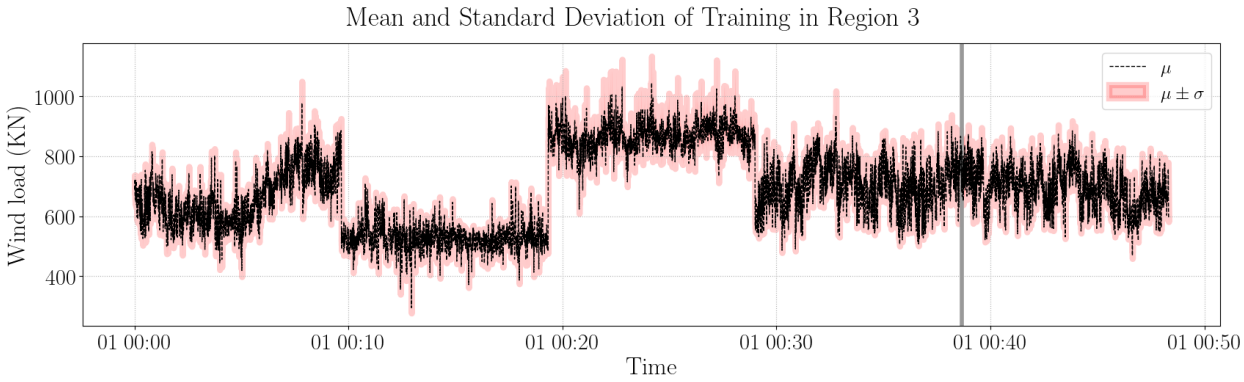


Figure 5.15 Uncertainty quantification of predictions in region three

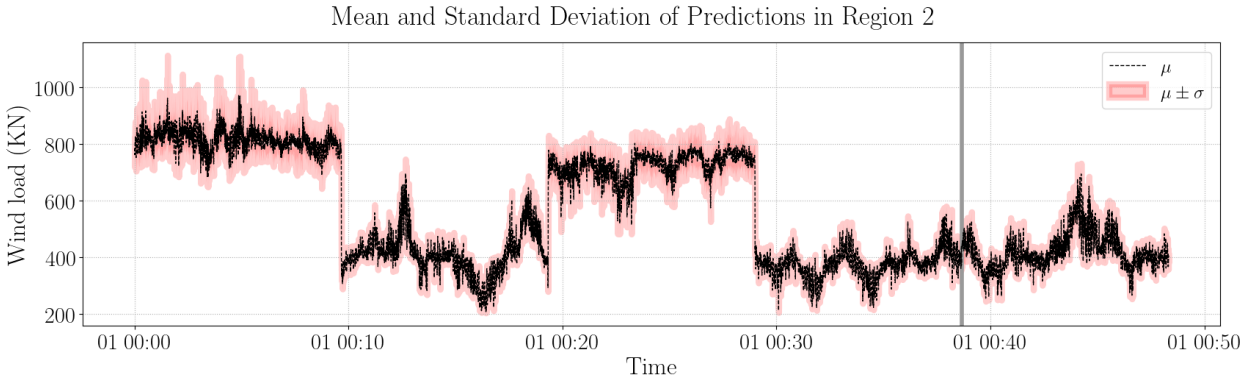


Figure 5.16 Uncertainty quantification of predictions in region two

This framework effectively addresses the computational expense associated with the initial physics-based model. The traditional physics-based method requires approximately 6000 seconds to process each 10-minute dataset. However, once the surrogate model is trained, it can predict the input load in mere seconds, demonstrating significant improvements in computational efficiency and scalability. One of the key advantages of using this surrogate model for input load estimation is the elimination of the need for continual tuning. In contrast to the physics-based model, which requires a level of tuning for each dataset, the surrogate model is trained once on a comprehensive dataset and can then be applied universally without further adjustments. This feature not only saves

time but also enhances the model’s robustness and reliability. Moreover, the model has substantial practical implications particularly for real-time load estimation.

5.5 Conclusions

The study successfully developed a hybrid surrogate model that integrates data-driven and physics-based approaches for input load estimation on offshore wind turbines. The surrogate model showcases an advancement in the field of load estimation methodologies. By leveraging a combination of data-driven and physics-based approaches, the model overcomes the computational expense associated with traditional physics-based methods while maintaining accuracy, highlighting the surrogate model's reliability. Through transfer learning, the model successfully adapts to real-world data, showcasing its versatility and practical applicability. Moreover, the multitask learning strategy accommodates the diverse behaviors observed across different operational regions of offshore wind turbines. By effectively capturing shared information while addressing region-specific intricacies, this approach enhances the model's adaptability. Additionally, the Monte Carlo Dropout method is used to quantify the uncertainty of trained neural network model by offering a measure of confidence in the model's predictions. Through quantifying uncertainties, the trained model's reliability and robustness are further substantiated, enabling nuanced interpretations of predictions, and enhancing applicability in practical settings. Overall, this hybrid surrogate input load estimation framework represents a significant advancement in offshore wind turbine load estimation. Its efficiency, accuracy, and scalability hold promise for facilitating real-time load estimation, thereby facilitating proactive maintenance strategies.

Acknowledgments

This study is partially supported by the National Science Foundation grant 2230630 and Massachusetts Clean Energy Center under the AmplifyMass program. A special acknowledgment goes to Finn Rüdinger and Ross McAdam at Ørsted for the technical support. The opinions, findings, and conclusions expressed in this document are those of the authors and do not necessarily represent the views of the sponsors and organizations involved in this project.

6 Chapter 6: Physics-Informed Machine Learning for Input Load Estimation

6.1 Introduction

In the field of offshore wind turbine input load estimation, three primary methodologies are utilized: physics-based, data-driven, and hybrid approaches. Each method has unique advantages and limitations, and they often complement each other in addressing complex challenges in load estimation. Physics-based models are grounded in fundamental principles such as aerodynamics, structural mechanics, and control theory. Data-driven methods, on the other hand, rely on statistical and machine learning techniques to infer relationships directly from the data, often bypassing the need for detailed physical modeling. The latter, are particularly powerful in scenarios where large amounts of data are available, enabling models to capture complex, nonlinear interactions that may be difficult to express through traditional physics-based approaches. Therefore, hybrid models that combine the strengths of both physics-based and data-driven methods can offer the best of both worlds, providing more accurate and robust load estimation by leveraging the underlying physics while also adapting to empirical data.

Physics-informed machine learning (PIML) is a powerful hybrid approach that combines fundamental physics principles with data-driven machine learning (ML) techniques to enhance model accuracy, interpretability, and reliability, particularly in scientific and engineering applications. While traditional data-driven models are effective at capturing complex patterns from large datasets, they often lack interpretability and may struggle to generalize beyond the training data, especially in domains governed by well-established physical laws. PIML addresses these

limitations by embedding domain-specific knowledge and constraints into the learning process, leveraging the strengths of both physics and ML. This integration has gained significant attention in recent years due to its ability to overcome challenges associated with purely physics-based and data-driven approaches. It is particularly relevant in fields where data is sparse, noisy, or expensive to obtain, such as offshore wind energy.

The approaches to incorporating physics into ML models can be categorized into two general groups: 1) direct physics integration techniques, 2) hybrid and data-enrichment strategies. Each group provides a unique way to embed, enhance, or guide the ML process using physical principles.

Direct Physics Integration Techniques involve directly incorporating physical laws into the model's structure or training process, making these laws a central part of the model's predictions. They integrate governing physical equations, often represented by partial differential equations (PDEs), directly into the loss function of a neural network. By minimizing the discrepancy between predicted and true physical behaviors, these methods ensure that the model inherently respects known physical principles such as conservation of mass or energy directly into the model's architecture or training process, ensuring that predictions remain physically plausible. This approach enhances the model's reliability and trustworthiness by ensuring that the learning process respects fundamental physical laws [117].

Within this first research domain, Raissi et al. developed Physics-Informed Neural Networks (PINNs) to solve both forward and inverse problems involving nonlinear PDEs [132]. Unlike traditional ML models, PINNs embed governing equations directly into the learning process, which reduces the need for large datasets and improves accuracy. This approach has been applied

across fields such as fluid dynamics, quantum mechanics, and reaction-diffusion systems, demonstrating its versatility in data-driven scientific computing. In other research, Zhu and colleagues introduced a framework that leverages physics constraints to improve surrogate modeling and uncertainty quantification in high-dimensional settings [133]. By embedding governing physical laws and minimizing the reverse Kullback-Leibler divergence, this approach extends PINNs by focusing on high-dimensional problems and capturing uncertainty in predictions without the need for labeled data.

Further advancements have been made in the field of wind turbine aerodynamic load estimation. Baisthakur and Fitzgerald developed a method to enhance the accuracy of aerodynamic load estimation by integrating fluid dynamics equations into neural networks [120]. This method allows the model to learn complex aerodynamic interactions and accurately predict loads on wind turbine blades, bypassing traditional methods that are often computationally intensive and simplified. Another application of direct physics integration is seen in the work of Haywood and Chatzi, who applied PINNs to predict future states of dynamic systems, integrating known physical laws into the neural network architecture. [134]. Their approach, validated with single degree of freedom (SDOF) systems, combines observation, physics, and initial-condition losses in the loss function, improving the accuracy and speed of predictions, which is crucial for structural health monitoring and real-time control.

In a recent research, a Physics-Guided Deep Neural Network was developed for structural damage detection, integrating finite element models with data-driven methods [135]. The model uses a cross-physics-data domain loss function, combining physical and feature discrepancy terms to align simulated and real-world data. This method improves damage localization accuracy,

overcoming challenges like data scarcity and noise. Another innovative approach is the Physics-guided Deep Markov Model which enhances the modeling of nonlinear dynamical systems under uncertainty by incorporating physics-based constraints [136]. The model uses a variational inference approach and physics-based regularization in its loss function, validated through simulations of dynamic systems like pendulums and fatigue crack growth.

A Physics-Informed Neural Ordinary Differential Equation (PINODE) framework has also been proposed for structural identification which integrates known physics with neural networks for structural identification [137]. The loss function includes a physics-informed term encoding governing differential equations and a data-driven term capturing discrepancies between the model and actual behavior. This allows accurate modeling and detection of potential damages in structures.

Hybrid and Data-Enrichment Strategies leverage the strengths of both physics-based and data-driven approaches by combining them in various ways. This group includes hybrid models, data augmentation, domain adaptation and Transfer Learning (TL). Hybrid models integrate physics-based models with ML models, allowing the physics model to provide a baseline understanding of the system while the ML model captures complex or unmodeled phenomena. This combination benefits from the robustness and interpretability of physics-based models and the pattern recognition capabilities of ML. Domain adaptation and transfer learning are similar in that they involve using pre-trained models from physics-based domains and adapting them to new data, ensuring consistency with underlying physical principles and enhancing applicability in real-world environments. Data augmentation uses simulated data from physics-based models to enrich the training dataset, which is particularly valuable when real-world data is scarce or expensive to

obtain. By exposing the ML model to a diverse range of conditions through simulations, it can improve its generalization capability.

Data augmentation plays a crucial role in enhancing model performance, especially in scenarios where real-world data is limited. Ritto and Rochinha integrated digital twin, physics-based modeling, and machine learning to create a robust training dataset through simulated damage scenarios [138]. This approach allowed for quick identification of structural damage in bar structures, enhancing safety and maintenance by enabling real-time decision-making. Similarly, in manufacturing, Peres et al. used synthetic images to address data scarcity in quality inspections of structural adhesives [139]. By generating diverse defect scenarios in a simulation environment, they improved the mean Average Precision of object detection models, demonstrating the cost-effectiveness and robustness of synthetic data augmentation in quality control.

Transfer learning techniques also contribute significantly to the field of hybrid strategies. In one study, researchers developed a deep transfer learning network combining numerical simulations and lab experiments to detect damage in steel frames with bolted connections [140]. This approach used minimal labeled data to adapt pre-trained models to new conditions, effectively enhancing damage detection under various scenarios. Another innovative application of transfer learning is seen in the work by Wang et al., who proposed a method for rolling bearing fault diagnosis using a deformable Convolutional Neural Network and a Deep Long Short-Term Memory network [141]. The method adapted to new working conditions with minimal labeled data, achieving high accuracy in varying operational environments, making it suitable for real-world applications.

Tronci et al. introduced a novel approach based on transfer learning and speaker recognition techniques for bridge damage detection, addressing data scarcity in unhealthy structural states [142]. By employing features traditionally used in speaker recognition the framework captured the vibration characteristics of both audio signals and structural responses. This use of cross-domain features improved the classification accuracy of damage scenarios. In another research Gaussian Processes were applied to predict offshore wind turbine bending moments, with physics awareness achieved through strategic feature selection and custom kernel design [143]. This ensured accurate modeling of wind turbine dynamics, enhancing generalizability and prediction accuracy.

This research introduces two novel frameworks based on physics-informed neural networks to tackle the complex inverse problem of estimating input loads in offshore wind turbines. These frameworks are uniquely crafted to function in scenarios where input loads are not directly observed, providing an innovative approach that circumvents the need to treat input loads as explicit inputs or outputs of the neural networks. Both frameworks utilize the concept of physics-informed learning by incorporating the equations of motion into their loss functions, which guarantees that the predictions comply with the basic physical laws that regulate the system. The first framework uses time as the input and measured displacements and accelerations as the outputs. In this setup, the input load is considered an additional parameter that can be learned. The network dynamically estimates the input load using a loss function that merges data loss, which accounts for discrepancies between predicted and actual values, with a physics-based loss that enforces the equations of motion. The second framework introduces an integrated network that uses two interlinked networks to predict input loads and dynamic responses. This framework includes a comprehensive loss function composed of three elements: equation of motion loss, differentiation law loss, and data loss. The equation of motion loss ensures that the predicted input

load complies with physical laws, while the differentiation law loss preserves the consistency of relationships between displacement, velocity, and acceleration. The data loss minimizes the differences between predicted and actual measurements, ensuring the accuracy of observed data.

This chapter of the thesis proposes a solution for the inverse problem of input load estimation in offshore wind turbines using physics-informed machine learning. The chapter explains details on the methodology, presenting two innovative frameworks based on PINNs. Following the methodology, the chapter provides extensive results and discussion, showcasing the effectiveness of these frameworks through various test cases, including physical space and modal space estimations with different data inputs. The chapter concludes by summarizing the key findings and discussing the potential future applications and improvements of these physics-informed approaches in the context of offshore wind turbine monitoring.

6.2 Methodology

In this research, two novel physics-informed neural network (PINN) frameworks are proposed to tackle the inverse problem of input load estimation in offshore wind turbines. These frameworks are designed specifically for scenarios where input loads are never directly measured, addressing the unique challenges of offshore environments.

6.2.1 Framework 1: Input Load as Learnable Parameter

The first framework is structured with time (t) as the primary input and measured displacements $\mathbf{u}_{meas}$ as the output of the network. In some cases, acceleration $\ddot{\mathbf{u}}_{meas}$ can also be used as an additional input to the network. Here, the input load is treated as an additional learnable parameter

alongside the neural network parameters, which include the weights and biases. This allows the network to estimate the input load dynamically during training.

The vector **ω** contains all the parameters of the network, including the weights and biases of the neural network layers, as well as the input load vector **F**, which is treated as a trainable parameter.

$$\boldsymbol{\omega} = [\mathbf{W}, \mathbf{b}, \mathbf{F}] \tag{6.1}$$

where **W** represents the weights of the network **b** represents the biases, and **F** represents the input load magnitude or vector depending on the problem we are solving. These parameters are updated during the training process through backpropagation, which adjusts the weights, biases, and input load to minimize the total loss.

The neural network $NN(t; \boldsymbol{\omega})$ takes the time *t* (and optionally acceleration $\ddot{\mathbf{u}}_{meas}$) as input and produces the predicted displacement $\hat{\mathbf{u}}$ as output. The loss function is defined as a combination of two key components:

1. **Data Loss** ($\mathcal{L}_{data}$)**:** This measures the discrepancy between the predicted and actual displacements, ensuring the network accurately reflects the observed data.

$$\mathcal{L}_{data}(\boldsymbol{\omega}) = \frac{1}{N_m}\sum_{i=1}^{N_m}(\|\hat{\mathbf{u}} - \mathbf{u}_{meas}\|^2) \tag{6.2}$$

where $NN(t; \boldsymbol{\omega}) \rightarrow (\hat{\mathbf{u}})$ and $(\hat{\mathbf{u}})$ is the predicted displacement output of the neural network, **ω** is a vector of the parameters of the network, and N_m is the number of measured outputs.

2. **Physics Loss** ($\mathcal{L}_{phys}$)**:** This enforces the equations of motion, allowing the network to treat the input load as a trainable parameter. By incorporating the equations of motion, the network

learns to adjust the input load parameter at each time step, aligning the model with physical principles and leading to more accurate input load estimations. The equation of motion can be represented as:

$$\mathbf{M\ddot{u}} + \mathbf{C\dot{u}} + \mathbf{Ku} = \mathbf{S}_{\boldsymbol{p}}\mathbf{F} \tag{6.3}$$

where $\ddot{\mathbf{u}} \in \mathbb{R}^{n_{DOF} \times n_t}$, $\dot{\mathbf{u}} \in \mathbb{R}^{n_{DOF} \times n_t}$ and $\mathbf{u} \in \mathbb{R}^{n_{DOF} \times n_t}$ are the nodal acceleration, velocity and displacements; $\mathbf{M}, \mathbf{C}, \mathbf{K} \in \mathbb{R}^{n_{DOF} \times n_{DOF}}$ are the mass, damping, and stiffness matrices of the structural system respectively, with n_{DOF} representing the number of degrees of freedom characterizing the system. $\boldsymbol{S_p} \in \mathbb{R}^{n_{DOF} \times n_f}$ is the Boolean selection matrix, n_f is the number of input loads and $\mathbf{F} \in \mathbb{R}^{n_f \times n_t}$ is the input load. The physics loss ensures that this relationship holds:

$$\mathcal{L}_{phys}(\boldsymbol{\omega}) = \frac{1}{N_m}\sum_{i=1}^{N_m}\left(\left\|\mathbf{M}\hat{\ddot{\mathbf{u}}} + \mathbf{C}\hat{\dot{\mathbf{u}}} + \mathbf{K}\hat{\mathbf{u}} - \mathbf{S}_{\mathbf{p}}\hat{\mathbf{F}}\right\|^2\right) \tag{6.4}$$

To calculate this loss, which requires predicted velocity $\hat{\dot{\mathbf{u}}}$ and acceleration $\hat{\ddot{\mathbf{u}}}$, time derivatives of displacement were computed using automatic differentiation through the "autograd.grad" function provided by PyTorch [144], which is used to compute gradients of tensors with respect to other tensors. This function implements reverse-mode differentiation. The total loss $\mathcal{L}_{total}$ is a weighted sum of the data and physics losses:

$$\mathcal{L}_{total}(\boldsymbol{\omega}) = \lambda_{data}\mathcal{L}_{data}(\boldsymbol{\omega}) + \lambda_{phys}\mathcal{L}_{phys}(\boldsymbol{\omega}) \tag{6.5}$$

where λ is a weighting factor balancing the two losses. A fixed weighting scheme might not adequately address the varying contributions of these losses, especially when their relative magnitudes differ significantly during training, which was the case in our work. To address this

issue, the weighting factor can be adaptively adjusted based on the changes in the gradients of the losses with respect to the model parameters, making the weights sensitive to how each loss evolves during training. The adjustment can be based on the magnitude of the gradients of each loss with respect to the network parameters $\boldsymbol{\omega}$. If the gradient of a particular loss component is large, it suggests that the loss is not well optimized, and the corresponding weight should be increased. Conversely, if the gradient is small, the weight can be decreased. This can be formulated as:

$$\lambda_{data} = \frac{\left\|\nabla_{\boldsymbol{\omega}} \mathcal{L}_{phys}(\boldsymbol{\omega})\right\|}{\|\nabla_{\boldsymbol{\omega}} \mathcal{L}_{data}(\boldsymbol{\omega})\| + \left\|\nabla_{\boldsymbol{\omega}} \mathcal{L}_{phys}(\boldsymbol{\omega})\right\| + \varepsilon} \tag{6.6}$$

$$\lambda_{phys} = \frac{\|\nabla_{\boldsymbol{\omega}} \mathcal{L}_{data}(\boldsymbol{\omega})\|}{\|\nabla_{\boldsymbol{\omega}} \mathcal{L}_{data}(\boldsymbol{\omega})\| + \left\|\nabla_{\boldsymbol{\omega}} \mathcal{L}_{phys}(\boldsymbol{\omega})\right\| + \varepsilon} \tag{6.7}$$

Where $\nabla_{\boldsymbol{\omega}} \mathcal{L}_{phys}(\boldsymbol{\omega})$ and $\nabla_{\boldsymbol{\omega}} \mathcal{L}_{data}(\boldsymbol{\omega})$ are the gradient of physics loss and data loss with respect to model parameters, respectively and ε is a small constant added to prevent division by zero. By normalizing these weights, we ensure that the sum of λ_{data} and λ_{phys} equals 1, ensuring a balanced contribution from both loss components

This gradient-based weighting approach allows for a more nuanced balancing between the physics loss and data loss, particularly useful when the relative importance of each loss changes during training. It also helps prevent any one loss from dominating, leading to a more stable and effective training process. This approach is particularly beneficial in complex, multi-objective optimization problems where the trade-offs between different objectives are not constant throughout training.

This loss function guides the optimization process, where the vector $\boldsymbol{\omega}$ is iteratively updated to minimize the total loss, ensuring that the model accurately fits the data while also following the rules of physics. The minimization is achieved through backpropagation, which adjusts the parameters $\boldsymbol{\omega} = [\mathbf{W}, \mathbf{b}, \mathbf{F}]$ by computing the gradients of the total loss with respect to each parameter:

$$\frac{\partial \mathcal{L}_{total}}{\partial \boldsymbol{\omega}} = \left[\frac{\partial \mathcal{L}_{total}}{\partial \mathbf{W}}, \frac{\partial \mathcal{L}_{total}}{\partial \mathbf{b}}, \frac{\partial \mathcal{L}_{total}}{\partial \mathbf{F}}\right] \tag{6.8}$$

These gradients are then used to update the parameters using an optimization algorithm (e.g., stochastic gradient descent):

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \eta \frac{\partial \mathcal{L}_{total}}{\partial \boldsymbol{\omega}} \tag{6.9}$$

Where η is the learning rate, controlling the step size of each update. This iterative process continues until the total loss $\mathcal{L}_{total}(\boldsymbol{\omega})$ converges to a minimum, optimizing the weights $\mathbf{W}$, biases $\mathbf{b}$, and the input load $\mathbf{F}$.

The framework is illustrated in the Figure 6.1, where the input is either time alone or as shown in some experiments, time combined with measured acceleration. In the results section, it will be discussed that measured acceleration was added as an input in some experiments to assess its impact on the accuracy and efficiency of input load estimation. This is depicted in Figure 6.1 with a dashed circle. The output of the neural network and the loss functions remain consistent across all experiments. The features were engineered in the hope that they would enhance the model's performance.

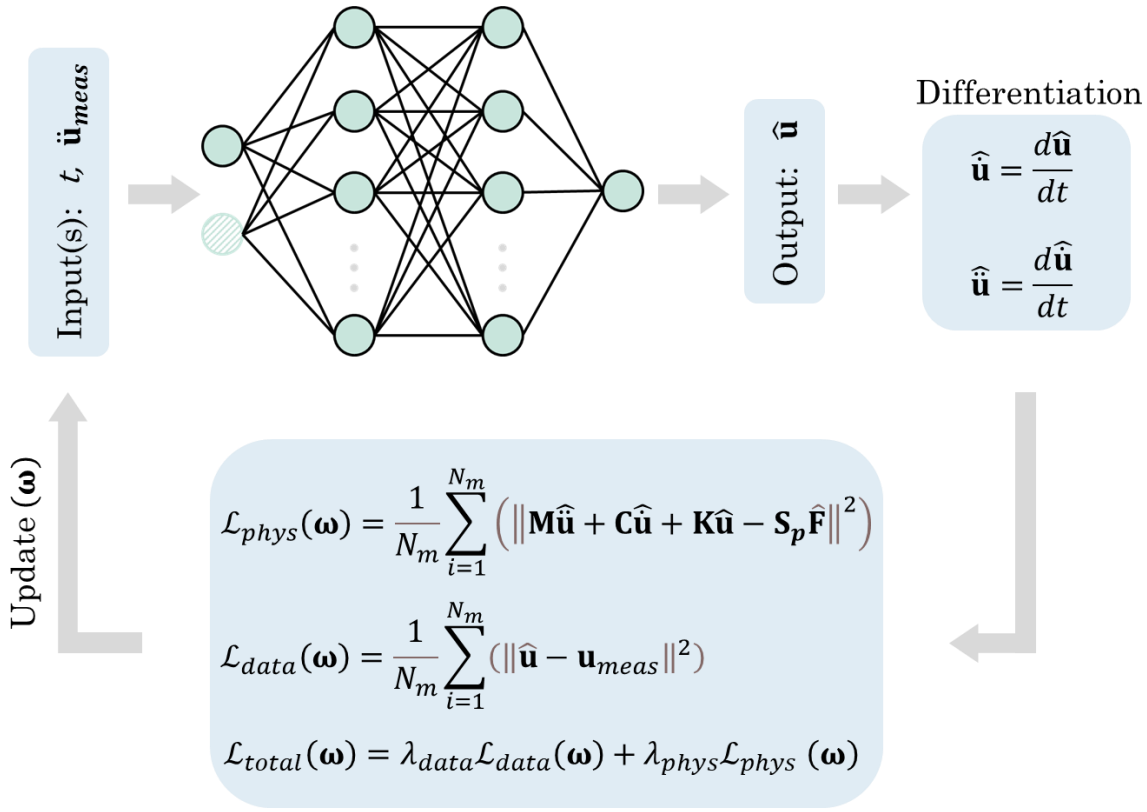


Figure 6.1 PINN framework 1

6.2.2 Model Order Reduction and Modal Space

The need for a reduced-order model arises from the inherent challenges of dealing with large systems, such as offshore wind turbines, which involve a large number of degrees of freedom. As the complexity of these models increases, so does the computational burden required to solve them. To overcome this challenge, model order reduction techniques can be employed, which simplify the model by reducing the number of state variables while preserving essential dynamics. The modal model is a specific type of model order reduction technique that is particularly useful for linear systems.

By transforming the problem into the modal domain, the behavior of the structure can be effectively captured using a smaller set of modes that represent the most significant dynamic

characteristics of the system. This approach takes advantage of the orthogonality of modes, a fundamental property in structural dynamics. The orthogonality of modes implies that each mode shape is independent of the others, meaning that the vibrational response in one mode does not influence the response in another. This allows the system's overall behavior to be described as a superposition of these independent modes. Using the orthogonality of modes, the displacements can be expressed in terms of modal coordinates $\mathbf{q} \in \mathbb{R}^{n_t \times n_{DOF}}$ and mode shapes $\mathbf{\Phi} \in \mathbb{R}^{n_{DOF} \times n_{DOF}}$.

$$\mathbf{u}(t) = \mathbf{\Phi q} \tag{6.10}$$

To create a reduced-order model, only the lower n_m modes is chosen. By focusing on these dominant n_m modes, the displacement, velocity, and acceleration of the system can be accurately approximated using the corresponding modal coordinates.

$$\mathbf{u}(t) \approx \mathbf{\Phi}_m \mathbf{q}_m, \dot{\mathbf{u}}(t) \approx \mathbf{\Phi}_m \dot{\mathbf{q}}_m, \ddot{\mathbf{u}} \approx \mathbf{\Phi}_m \ddot{\mathbf{q}}_m \tag{6.11}$$

where $\mathbf{\Phi}_m \in \mathbb{R}^{n_{DOF} \times n_m}$ is the reduced mode shape matrix and $\mathbf{q}_m(t) \in \mathbb{R}^{n_t \times n_m}$ represents the reduced modal coordinates. Substituting these approximations into the equation of motion:

$$\mathbf{M\Phi}_m \ddot{\mathbf{q}}_m + \mathbf{C\Phi}_m \dot{\mathbf{q}}_m + \mathbf{K\Phi}_m \mathbf{q}_m = \mathbf{S}_p \mathbf{F} \tag{6.12}$$

After some algebraic manipulation, the equation of motion simplifies to:

$$\ddot{\mathbf{q}}_m + (\mathbf{\Phi}_m^T \mathbf{M\Phi}_m)^{-1} \mathbf{C\Phi}_m \dot{\mathbf{q}}_m + (\mathbf{\Phi}_m^T \mathbf{M\Phi}_m)^{-1} \mathbf{K\Phi}_m \mathbf{q}_m = (\mathbf{\Phi}_m^T \mathbf{M\Phi}_m)^{-1} \mathbf{\Phi}_m^T \mathbf{S}_p \mathbf{F} \tag{6.13}$$

As a result, only n_m equations need to be solved, instead of n_{DOF}. This reduction in the number of equations significantly decreases the computational complexity of the model while retaining the essential dynamics of the system. Consequently, the model becomes more efficient and practical for applications where measuring all degrees of freedom is infeasible. In this approach, wherever the equations of motion are used in the loss functions, they are now replaced with the equations of

motion in the modal space. This means that instead of the network outputting displacements, it will output the modal coordinates. All calculations and optimizations are performed in the modal coordinates, and the results are then transformed back into the physical space.

6.2.3 Framework 2: Dual-Network Integrated Approach

The second framework employs an integrated network strategy, where the input load, which is not labeled or directly available as real data, is inferred through the model. This framework consists of two interconnected neural networks:

1. **First Network:** This network takes time (t), displacement $\mathbf{u}_{meas}$, and acceleration $\ddot{\mathbf{u}}_{meas}$ as inputs, and outputs the estimated input load $\mathbf{F}$ and velocity $\dot{\mathbf{u}}$:

$$NN_1(t, \mathbf{u}_{meas}, \ddot{\mathbf{u}}_{meas}; \boldsymbol{\omega_1}) \rightarrow \left(\widehat{\mathbf{F}}, \widehat{\dot{\mathbf{u}}}\right) \tag{6.14}$$

where $\widehat{\square}$ is the symbol for prediction and $\boldsymbol{\omega_1}$ is the first network parameter vector. Here, the input load is treated not as a trainable parameter but as a time history that is the network's output.

2. **Second Network:** This network receives the estimated input load and velocity $\left(\widehat{\mathbf{F}}, \widehat{\dot{\mathbf{u}}}\right)$ from the first network as inputs and outputs the predicted displacements and accelerations:

$$NN_2\left(\widehat{\mathbf{F}}, \widehat{\dot{\mathbf{u}}}; \boldsymbol{\omega_2}\right) \rightarrow \left(\widehat{\mathbf{u}}, \widehat{\ddot{\mathbf{u}}}\right) \tag{6.15}$$

where $\boldsymbol{\omega_2}$ is the second network parameter vector. To calculate loss, a complete forward pass through both networks is required. This process involves first running the inputs through the first network to obtain the estimated input load and velocity. These outputs are then fed into the second network to generate the predicted displacements and accelerations. The loss function in this framework comprises three components:

- **Equation of Motion Loss** ($\mathcal{L}_{eom}$)**:** This loss component ensures that the predicted input load satisfies the physical equations governing the system.

$$\mathcal{L}_{eom}(\boldsymbol{\omega_1}, \boldsymbol{\omega_2}) = \frac{1}{N_m}\sum_{i=1}^{m}\left(\left\|\mathbf{M}\,\hat{\ddot{\mathbf{u}}} + \mathbf{C}\hat{\dot{\mathbf{u}}} + \mathbf{K}\hat{\mathbf{u}} - \mathbf{S}_p\hat{\mathbf{F}}\right\|^2\right) \tag{6.16}$$

where predicted input and velocity is the output of the first network and predicted acceleration, and displacement are the outputs of the second network.

- **Differentiation Law Loss** ($\mathcal{L}_{diff}$)**:** This part of the loss function ensures that the predicted velocity and acceleration maintain their differential relationships with the displacements and velocity, respectively. This enforces the continuity and physical accuracy of the model across time steps.

$$\mathcal{L}_{diff}(\boldsymbol{\omega_1}, \boldsymbol{\omega_2}) = \frac{1}{N_m}\sum_{i=1}^{N_m}\left(\left\|\hat{\dot{\mathbf{u}}} - \frac{d\hat{\mathbf{u}}}{dt}\right\|^2 + \left\|\hat{\ddot{\mathbf{u}}} - \frac{d\hat{\dot{\mathbf{u}}}}{dt}\right\|^2\right) \tag{6.17}$$

Here the predicted acceleration and displacement relationship the predicted velocity is enforced.

- **Data Loss:** This measures the difference between the predicted displacements and accelerations from the second network and the actual measured values. The goal is to minimize this discrepancy, ensuring that the output of the second network closely matches the real-world measurements.

$$\mathcal{L}_{data}(\boldsymbol{\omega_1}, \boldsymbol{\omega_2}) = \frac{1}{N_m}\sum_{i=1}^{N_m}\left(\|\hat{\mathbf{u}} - \mathbf{u}_{meas}\|^2 + \left\|\hat{\ddot{\mathbf{u}}} - \ddot{\mathbf{u}}_{meas}\right\|^2\right) \tag{6.18}$$

The total loss in this framework is a weighted combination of these three losses, guiding the network to accurately estimate the input load while adhering to physical principles and observed data.

$$\mathcal{L}_{total}(\boldsymbol{\omega_1}, \boldsymbol{\omega_2}) = \lambda_{data}\mathcal{L}_{data}(\boldsymbol{\omega_1}, \boldsymbol{\omega_2}) + \lambda_{eom}\mathcal{L}_{eom}(\boldsymbol{\omega_1}, \boldsymbol{\omega_2}) + \lambda_{diff}\mathcal{L}_{diff}(\boldsymbol{\omega_1}, \boldsymbol{\omega_2}) \quad (6.19)$$

The second framework is illustrated in the Figure 6.2, as shown here it is necessary to perform a full forward pass through both networks to calculate the losses and update and weights.

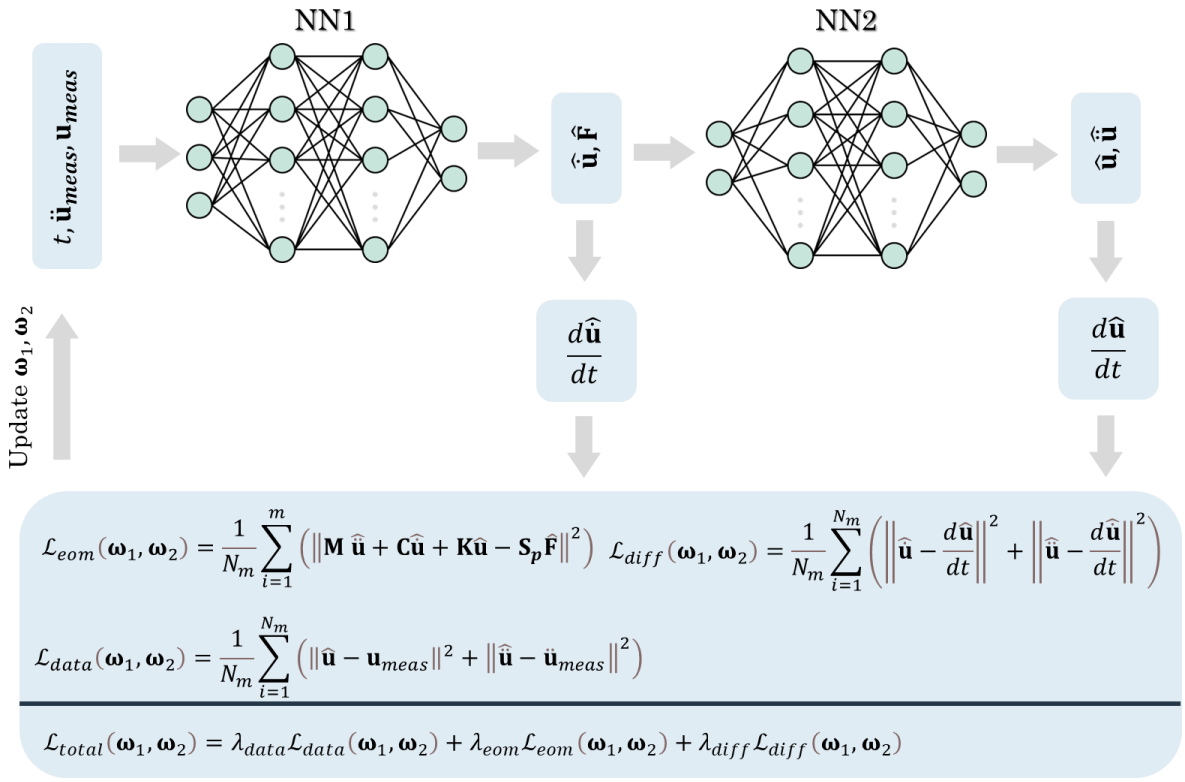


Figure 6.2 PINN framework 2

6.3 Validation Using a 3-DOF Model

In this section seven different sets of input load estimation problems are defined for a 3DOF shear building example and solved using the first and second frameworks proposed in the

methodology section. Then the input load estimation problem of an offshore wind turbine is explored.

In this study, a simple 3DOF numerical example is used to simulate data and test the proposed methodology. The mass and stiffness matrices of the system are defined as follows:

$$\mathbf{K} = \begin{bmatrix} 600 & -600 & 0 \\ -600 & 1800 & -1200 \\ 0 & -1200 & 3000 \end{bmatrix} \left(\frac{N}{m}\right) \qquad \mathbf{M} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 150 & 0 \\ 0 & 0 & 200 \end{bmatrix} (kg) \tag{6.20}$$

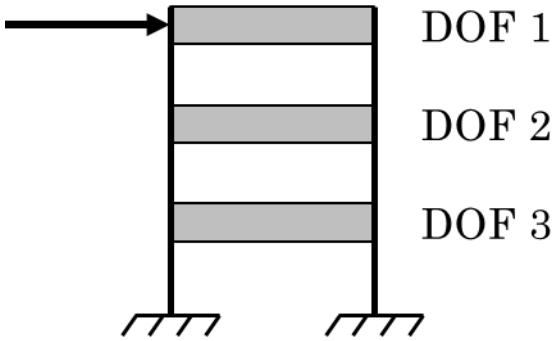


Figure 6.3 3DOF example

Figure 6.3 shows the 3DOF system and where the load is applied on it. The natural frequencies of the system, calculated from the eigenvalues of the system, are 0.23, 0.49, and 0.73 Hz, respectively. Rayleigh damping is applied to account for energy dissipation, with damping ratios set to 0.01 for the first mode and the second mode. The input load applied to the structure is a sinusoidal force with the following form:

$$\mathbf{F}(t) = p_0 \sin\left(\omega_f t\right) \tag{6.21}$$

where $p_0 = 100$ N is the amplitude of the force, and $\omega_f = 0.4$ rad/s is the frequency of the sinusoidal load. The structural response to the input load is simulated and the simulated response

is then polluted with zero-mean Gaussian white noise, with a standard deviation of 0.01. Simulated exact and noisy displacement responses are shown in Figure 6.4.

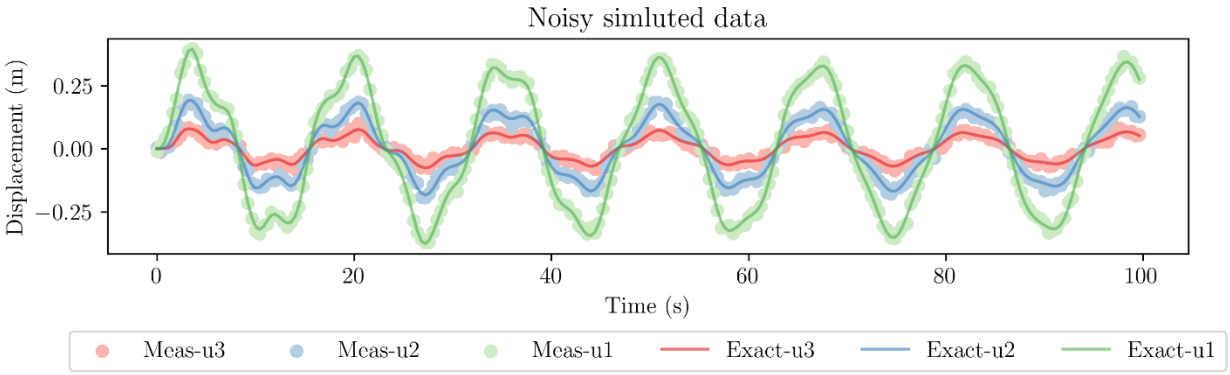


Figure 6.4 Simulated displacements for all DOFs

6.3.1 Problem 1: Framework 1, Physical space, estimation of the magnitude of the input load

In the first problem, a PINN was implemented to estimate the input load on the 3-DOF structural model. The network was designed as a fully connected feedforward neural network with three hidden layers, each containing 32 neurons. The input to the network was time, and the outputs were the displacements at all degrees of freedom (DOFs). The training process aimed to enforce the equations of motion by including them in the loss function, as outlined in the methodology section in the first framework. The shape of the input load was assumed to be sinusoidal, and the network was tasked with estimating the magnitude of this input load by introducing an additional trainable parameter within the model. Both the network parameters and p_0 were trained using the Adam optimizer, with learning rates set at 5e-2 for both. No dropout was applied during the training process, allowing the network to fully utilize all available data for learning.

After training for 20,000 epochs, the network successfully learned to predict the displacements across all DOFs, though with some minor discrepancies. The results are presented in Figure 6.5. The blue line represents the PINN's predicted displacements for each DOF, while the exact simulated responses are shown as the underlying reference. Although the predicted displacements generally match the exact responses well, there is a slight underperformance in capturing finer oscillations, especially at the peaks of the displacements. These nuances are mostly missed, leading to small deviations between the predicted and exact displacements. This is further reflected in the performance metrics shown in Table 6.1.

The RMSE and RRMSE are both low, with the RRMSE at just 1%, indicating a good overall match despite the missed oscillations. The TRAC values are slightly lower at 0.98 for the lower DOFs (2 and 3), where the displacements are smaller and capturing these finer details is more challenging. However, at the top node (DOF 1), the match is excellent. The predicted input load also shows a good alignment with the exact load applied to the structure. Since the shape of the input was predefined as sinusoidal, the TRAC was not calculated, as it is unnecessary in this context. Instead, the RMSE was calculated, showing a very small error of just 4 N, indicating that the input load's magnitude was estimated accurately.

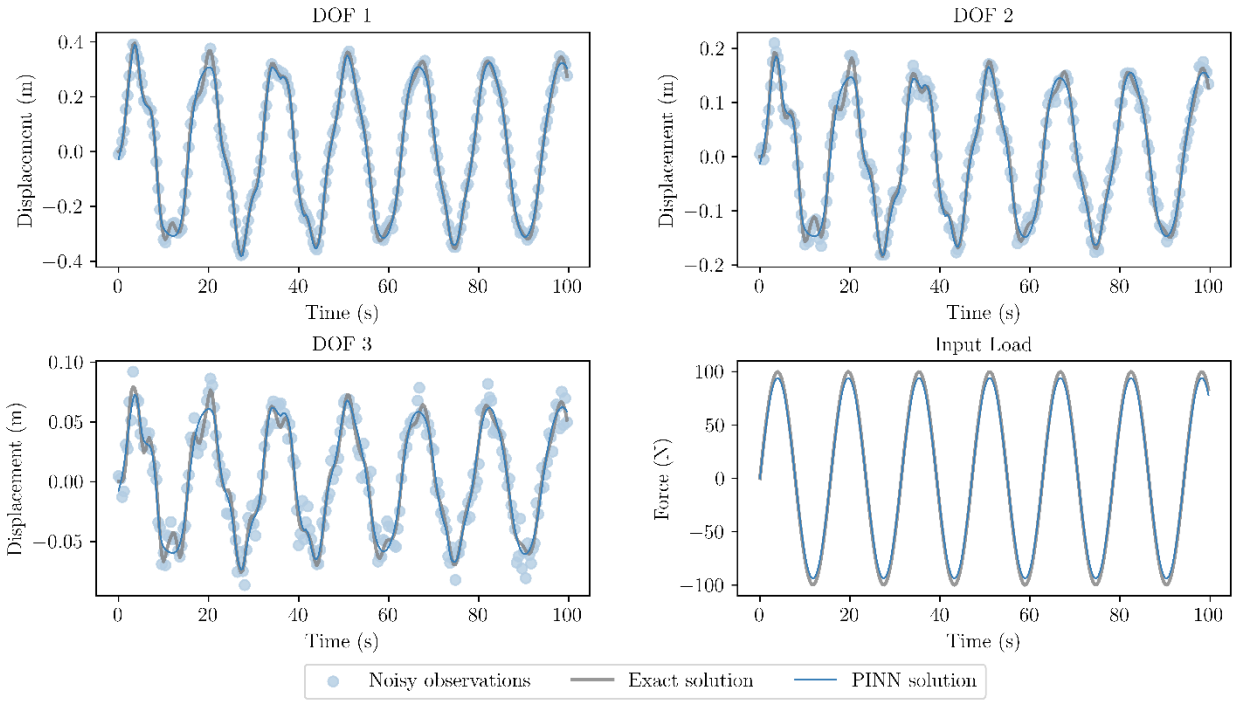


Figure 6.5 Estimated displacements and input load using PINN (p_0 as a trainable parameter) in physical space

Table 6.1 Accuracy metrics for the first problem

	RMSE	RRMSE (%)	TRAC
Displacements	(0.016, 0.012, 0.006)(m)	(7, 13, 25)	(0.995, 0.989, 0.982)
Input	4.3(N)	6	-

6.3.2 Problem 2: Framework 1, Physical space, estimation of the time history of the input load

In the second problem, we extend the PINN framework to estimate the entire time history of the input load on the 3-DOF structural model without assuming a predefined shape, as was done in the first problem. The network is structured similarly, with time as the input and displacements at all DOFs as the outputs. However, unlike the previous scenario where the input load was assumed to be sinusoidal, the shape of the input load here is entirely unknown. The goal is for the

network to learn this shape purely from the data, using additional trainable parameters that correspond to the full time history of the input load. To tackle this more complex problem, a more sophisticated neural network architecture was employed. The defined network is a fully connected feedforward neural network with four hidden layers, each containing 64 neurons. The network parameters were trained using the Adam optimizer, with a learning rate of 5e-3 for the PINN parameters and 5e-2 for the input time history. No dropout was applied during the training, allowing the network to fully leverage the data for learning.

After training for 20,000 epochs, the results demonstrated the network's ability to predict the displacements across all DOFs, although with slightly more error compared to the first problem. Figure 6.6 shows that the network predicts the displacements well, though with a bit more discrepancy than in the first problem as tabulated in Table 6.2. This increase in error is expected, given that the training process here also involved predicting the shape of the input load, making the task more challenging and susceptible to small errors that can manifest in both the predicted displacements and the input load. Even though the network was provided with no prior information about the shape of the input load, it managed to capture the general shape of the input load quite well as evidenced by the TRAC shown in Table 6.2, which is 0.989. The magnitude of the input load is also captured reasonably well, with an error of only 8 N, or approximately 25%. This is a good result, showing that the network was able to estimate the input load with a high degree of accuracy despite the increased complexity of the problem.

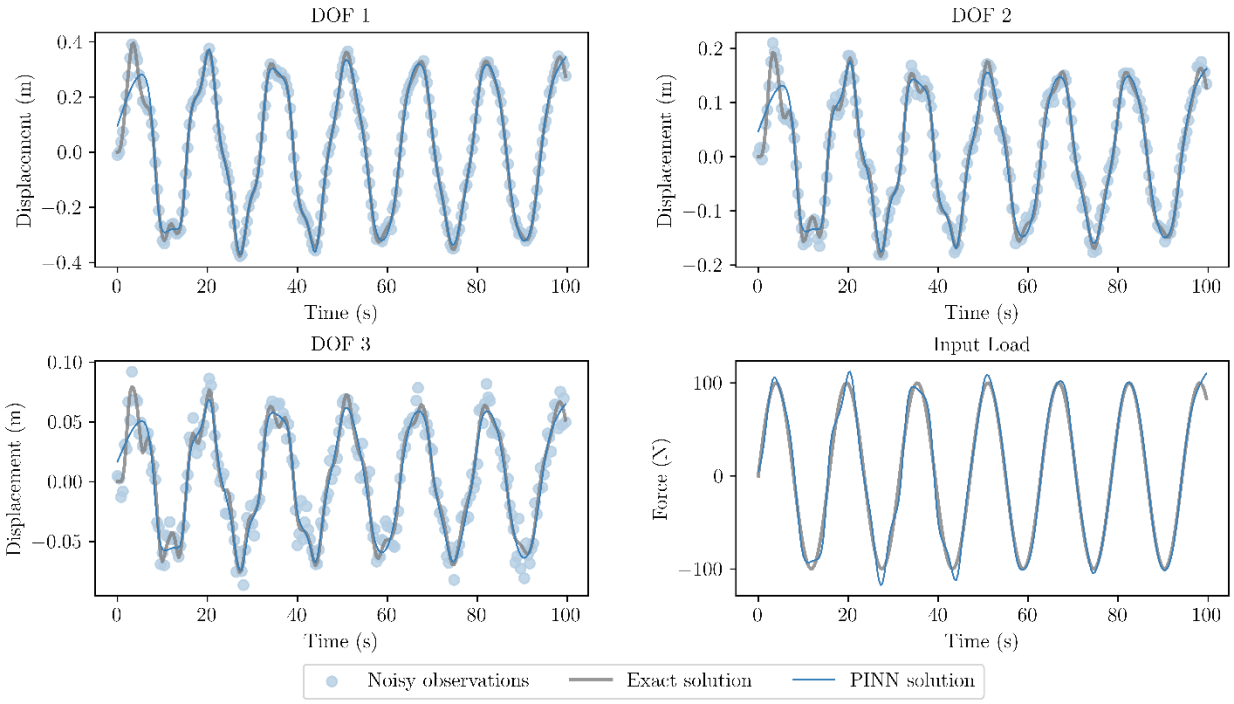


Figure 6.6 Estimated displacements and input load using PINN (F(t) as trainable parameters) in physical space

Table 6.2 Accuracy metrics for the second problem

	RMSE	RRMSE (%)	TRAC
Displacements	(0.028, 0.017, 0.008)(m)	(12, 17, 26)	(0.986, 0.978, 0.969)
Input	8.1(N)	11.3	0.989

6.3.3 Framework 1, Physical space, estimation of the time history of the input load with acceleration data

In the third problem, the PINN framework was extended by incorporating acceleration as an additional input alongside time to improve the estimation of the input load. The inclusion of acceleration data was motivated by the hypothesis that it could help better capture the shape of the input load, while displacement data could help accurately predict its magnitude. The goal was to leverage both acceleration and displacement information to enhance the network's ability to estimate the entire time history of the input load. To address this more complex task, the network

architecture was further expanded. The defined network is a fully connected feedforward neural network with five hidden layers, each containing 64 neurons. The network parameters were trained using the Adam optimizer, with learning rates of 5e-4 for the PINN parameters and 5e-2 for the input time history. No dropout was applied during the training process.

Figure 6.7 shows that the network successfully predicts the displacements, showing a good match with the actual measured values, similar to the previous problems. The addition of acceleration data, as expected, contributed to better capturing the higher-frequency components, especially in the higher DOFs where the displacements are larger. This improvement is evident in the metrics shown in Table 6.3 where both the RMSE and RRMSE have improved compared to the second problem for the first two DOFs. However, the inclusion of acceleration data also introduced some noise into the system, which manifests as slight deviations at the peaks of the input load. Despite this, the overall shape and average magnitude of the input load are still very well captured, much like in the previous problem. The better prediction of displacements can be attributed to the additional information provided by the acceleration data, which captures higher frequencies more effectively.

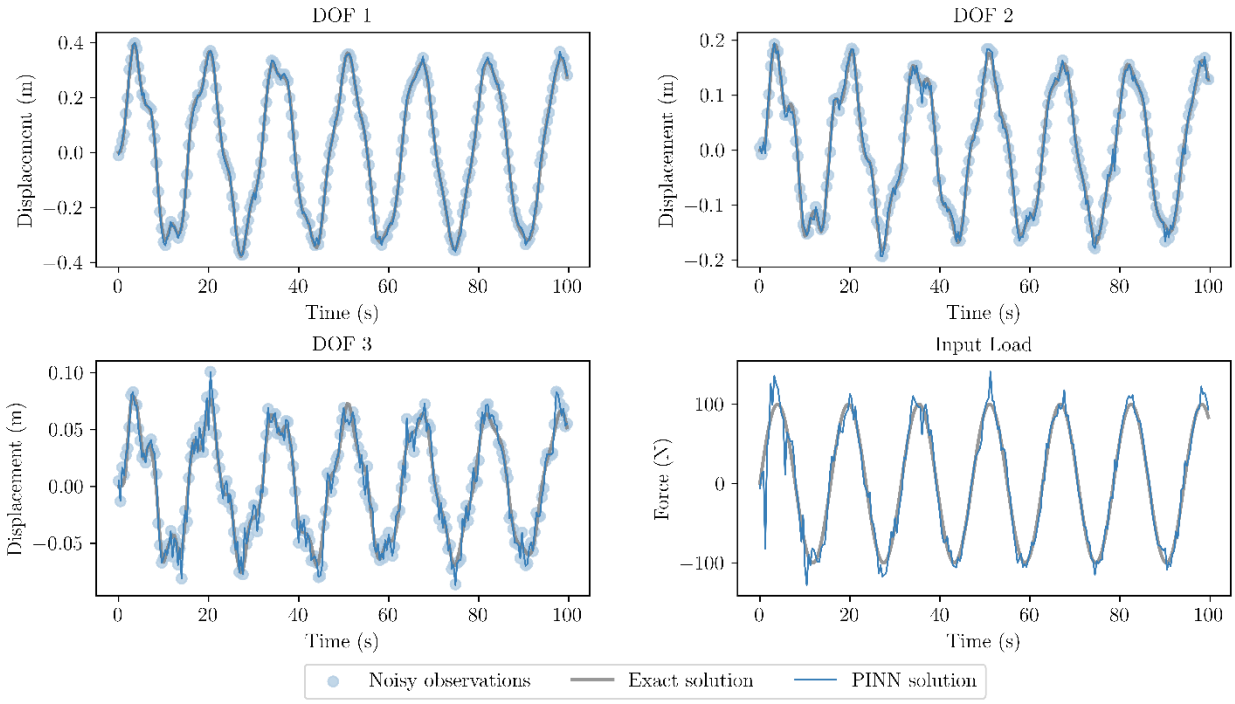


Figure 6.7 Estimated displacements and input load using PINN (F(t) as trainable parameters) using acceleration data in physical space

Table 6.3 Accuracy metrics for the third problem

	RMSE	RRMSE (%)	TRAC
Displacements	(0.011, 0.010, 0.010)(m)	(5, 9, 21)	(0.998, 0.992, 0.959)
Input	15.3(N)	21	0.959

6.3.4 Problem 4: Framework 1, Modal space, estimation of the magnitude of the input load

In the fourth problem, a more realistic scenario was considered where not all degrees of freedom (DOFs) were measured. Specifically, only two out of the three DOFs were measured. To address this, a transition to the modal space was employed, as detailed in the methodology. Two dominant modes were selected to represent the system, allowing the problem to be simplified while retaining the critical dynamics of the structure. PINN was utilized, with time as the input and the modal

coordinates ($\mathbf{q}_1$ $\mathbf{q}_2$) as the outputs for the two selected modes. The loss function was defined by enforcing the equations of motion in the modal space, ensuring adherence to the physical constraints of the system.

The shape of the input load was assumed to be sinusoidal, similar to the first problem, and the network was tasked with estimating the magnitude p_0 of the input load. A fully connected feedforward network with five hidden layers, each containing 128 neurons, was defined for this purpose. The learning rate was set to 5e-3 for network parameters and 5e-2 for p_0 , and the Adam optimizer was used for training. A dropout rate of 10% was applied to prevent overfitting during the training process.

As shown in Figure 6.8, after training the network for 20,000 epochs, the results demonstrated accurate predictions of the displacements and the input load magnitude. The accuracy in displacement estimation suggests that the dynamics in modal space were effectively captured, with the reduced number of measured DOFs. However, as expected, some higher dynamics are missing due to the omission of the third mode, which would have been captured in the full model. This is reflected in the performance metrics shown in Table 6.4, where the displacements exhibit higher RMSE and RRMSE values, along with lower TRAC values, compared to the first problem. Despite this, the additional error introduced by omitting the third mode is not significant enough to undermine the overall effectiveness of the model. The magnitude of the input load is still very well captured, which was the primary objective.

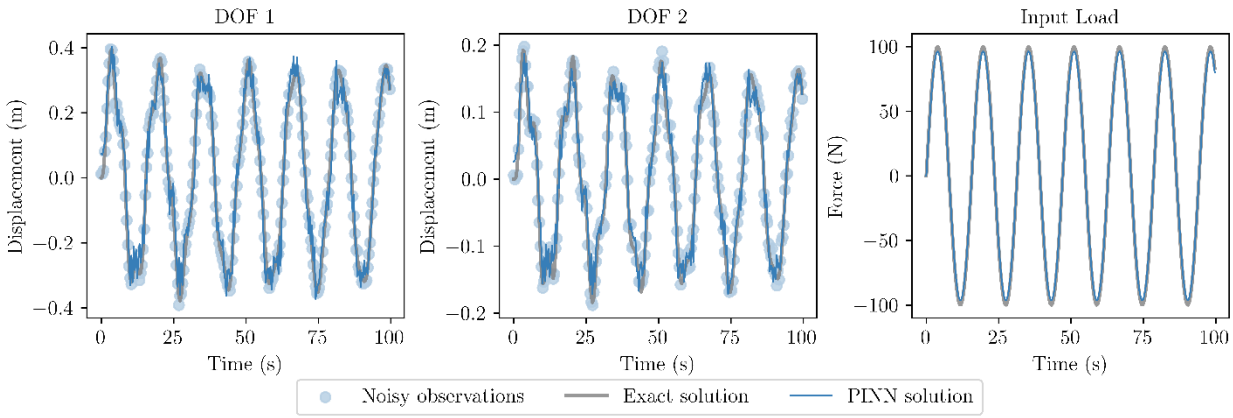


Figure 6.8 Estimated displacements and input load using PINN (p_0 as a trainable parameter) in modal space

Table 6.4 Accuracy metrics for the fourth problem

	RMSE	RRMSE (%)	TRAC
Displacements	(0.033, 0.019)(m)	(14, 17)	(0.982, 0.972)
Input	0.9(N)	1	-

6.3.5 Problem 5: Framework 1, Modal space, estimation of the time history of the input load

In the fifth problem, the PINN framework was extended to estimate the entire time history of the input load, similar to the second problem. However, unlike the second problem, where the input load was estimated in physical space, this time the estimation was performed in the modal space, similar to the approach in the fourth problem. The input to the network was time, and the outputs were the first two modal coordinates. The shape of the input load was not predefined, requiring the network to learn this shape purely from the data by treating the input time history as trainable parameters. The equations of motion were enforced in the modal space to ensure the network adhered to the system's physical constraints. The defined network architecture consisted of six hidden layers, each containing 128 neurons. The network parameters were trained using the

Adam optimizer, with learning rates set to 5e-3 for the PINN parameters and 5e-2 for the input time history. A dropout rate of 10% was applied to prevent overfitting during the training process.

After training, the results, as shown in Figure 6.9, demonstrate that the displacements were predicted well, although with slightly more error compared to the previous problem (where the input load was sinusoidal). This increase in error is expected due to the added complexity of estimating both the shape and magnitude of the input load in the modal space. Despite this, the network managed to capture both the shape and magnitude of the input load effectively. The performance metrics, which are presented in Table 6.5, indicate that while the displacements were predicted accurately.

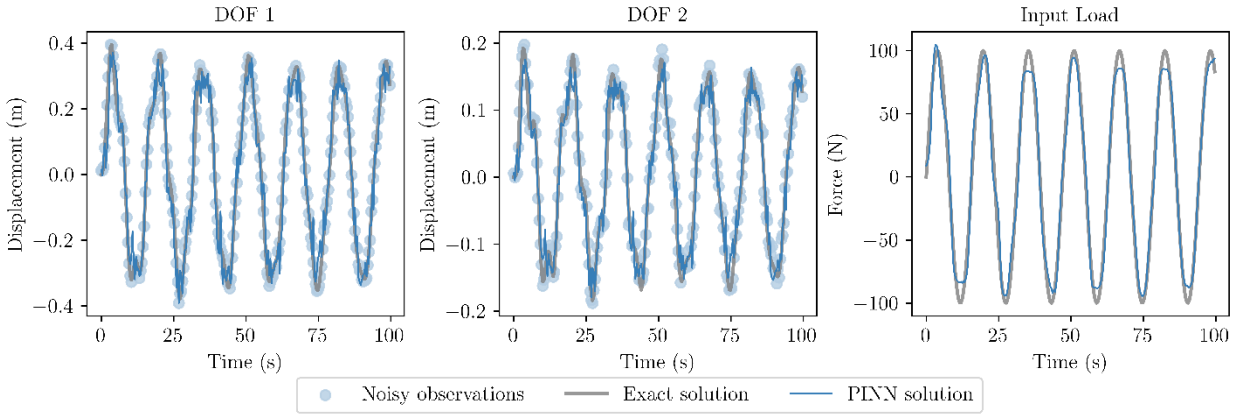


Figure 6.9 Estimated displacements and input load using PINN (F(t) as trainable parameters) in modal space

Table 6.5 Accuracy metrics for the fifth problem

	RMSE	RRMSE (%)	TRAC
Displacements	(0.036, 0.021)(m)	(15, 19)	(0.976, 0.965)
Input	8.9(N)	12.5	0.984

6.3.6 Problem 6: Framework 1, Modal space, estimation of the time history of the input load with acceleration data

In the sixth problem, the PINN framework was further extended by incorporating acceleration as an additional input alongside time to improve the estimation of the input load, similar to the third problem but this time in the modal space. The network architecture consisted of six hidden layers, each containing 128 neurons. The network parameters were trained using the Adam optimizer, with learning rates set to 5e-3 for the PINN parameters and 5e-2 for the input time history. A dropout rate of 10% was applied to help prevent overfitting during the training process. The results, as shown in Figure 6.10, demonstrate that the network was able to predict the modal displacements well with some minor discrepancies. This deviation is likely due to the added complexity of estimating the input load while also considering the influence of acceleration data, which introduces higher-frequency dynamics into the model. Table 6.6 presents the metrics for this problem.

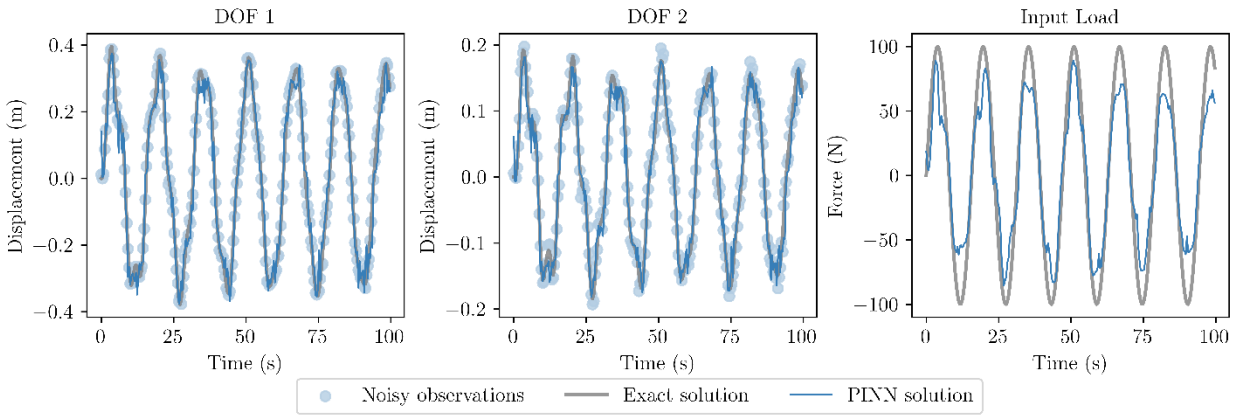


Figure 6.10 Estimated displacements and input load using PINN (F(t) as trainable parameters) using acceleration data in modal space

Table 6.6 Accuracy metrics for the sixth problem

	RMSE	RRMSE (%)	TRAC
Displacements	(0.040, 0.019)(m)	(17, 17)	(0.972, 0.971)
Input	23.1(N)	32.5	0.959

6.3.7 Framework 2, Physical space, estimation of the time history of the input load

In this problem, a second framework was developed for estimating the input load on the 3DOF structural model. Unlike the previous problems where the focus was solely on estimating the input load, this framework extends the approach by also attempting to reconstruct the displacements and accelerations using the estimated input load. This dual objective is particularly useful, as it provides a way to validate the accuracy of the estimated input load indirectly, by checking how well it can reproduce the measured displacements and accelerations. The framework operates by first estimating the input load through a neural network, similar to the previous problems. However, in this framework, a second neural network is used to reconstruct the displacements and accelerations based on the estimated input load. The loss functions are designed to connect both frameworks, allowing the reconstructed displacements and accelerations to be compared with the measured values. This approach is valuable because it offers a method to assess the accuracy of the input load estimation in cases where the real measured input load is not available. By comparing the reconstructed displacements and accelerations with the measured ones, we can gauge the effectiveness of the input load estimation.

In Figure 6.11 the reconstructed displacements and accelerations are shown as black lines, while the measured ones are depicted as thicker light blue lines. The first row presents the displacements for all nodes, and the second row shows the accelerations for all DOFs. The results indicate a very good match between the reconstructed and measured values, sometimes even outperforming

previous experiments. This suggests that the input load estimated by the first neural network captures the dynamic behavior of the system effectively, also obvious from the estimated and true input load plot. Although there is still room for improvement, the results are promising, as reflected in the metrics presented in Table 6.7.

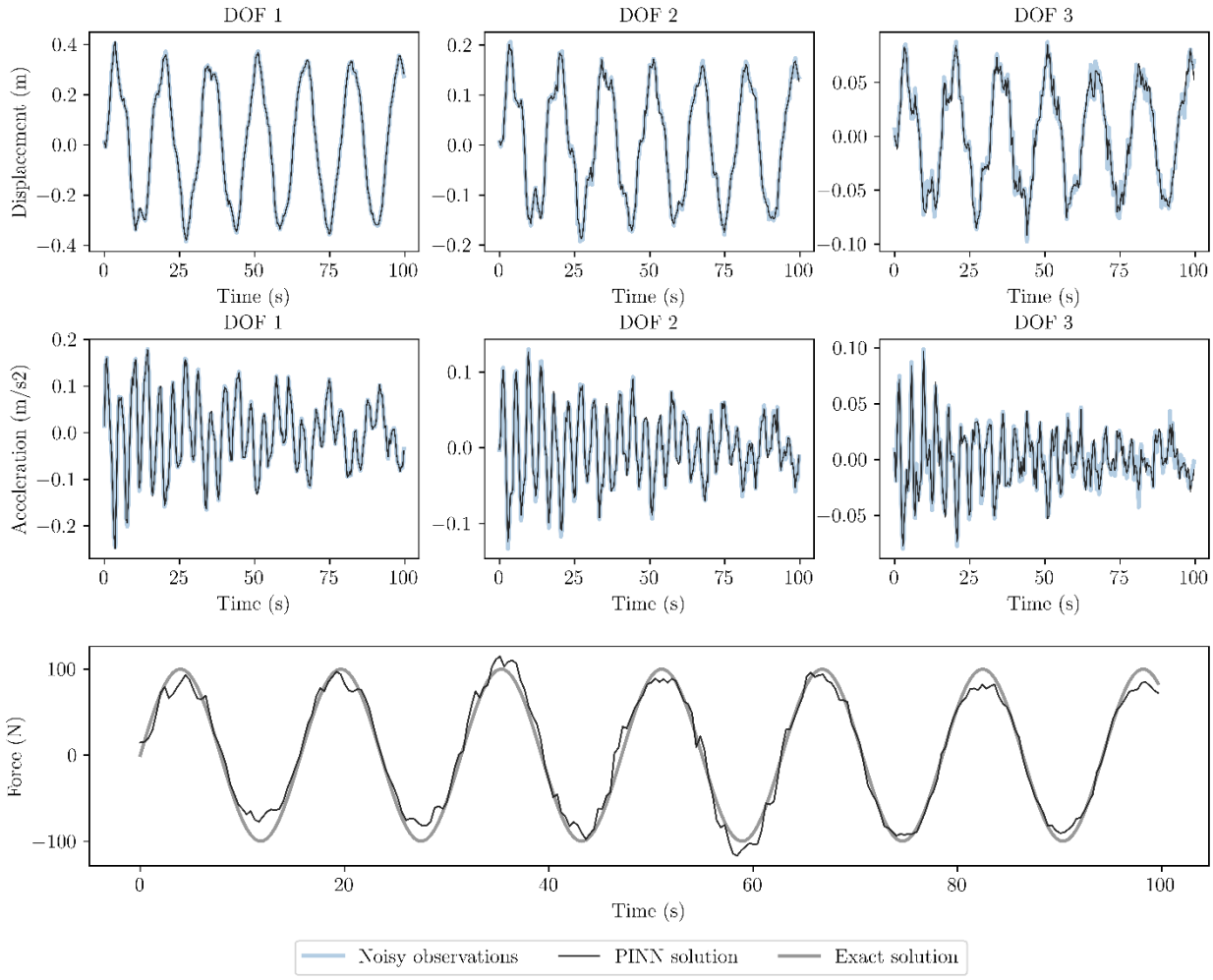


Figure 6.11 Estimated displacements and input load using the second framework

Table 6.7 Accuracy metrics for the seventh problem

	RMSE	RRMSE (%)	TRAC
Displacements	(0.003, 0.005, 0.005) (m)	(1, 5, 12)	(1, 0.998, 0.986)

Accelerations	(0.002, 0.004, 0.004)(m/s^2)	(3, 8, 16)	(0.999, 0.994, 0.973)
Input	11.8(N)	16	0.974

6.3.8 Discussion of the problems

The first three problems, solved in physical space, demonstrated accurate results; however, they required all degrees of freedom (DOFs) to be measured, which is impractical for real, large-scale structures. Accuracy metrics from these problems, such as the RMSE value for the displacements, highlight the model's precision. Nonetheless, transitioning to modal space (Problems 4-6) is preferred because it reduces the complexity of needing to measure all DOFs. In modal space, the RMSE for displacement predictions remained relatively low—though slightly higher than in physical space, which is expected—indicating effective performance even with fewer measurements.

Among the modal space problems, letting the network determine the shape of the input load without prior assumptions (as in Problems 5 and 6) is advantageous. This approach is closer to reality since the shape of the input is not easily accessible and leads to effective load shape prediction, as evidenced by the RMSE values. However, the inclusion of acceleration data (as in Problems 3 and 6) introduced noise, which negatively impacted prediction accuracy. For instance, in Problem 6, the RMSE for input load estimation increased to 23.1 N compared to 8.9 N in Problem 5, confirming that adding acceleration data tends to degrade accuracy by introducing higher-frequency dynamics that the model finds challenging to capture accurately.

Problem 7 introduces a dual-network approach in physical space, which simultaneously estimates input loads and reconstructs dynamic responses. While this method is currently applied in physical space and awaits transition to modal space, it showed promising results. The RMSE

values for displacement predictions were remarkably low, suggesting high accuracy. This integrated approach, by validating estimated input loads through reconstructed outputs, offers a comprehensive solution that balances accuracy and computational efficiency, potentially outperforming other frameworks, especially in complex scenarios.

6.3.9 Wind turbine problem

6.3.9.1 Input load magnitude estimation

A linear elastic FE model of the same 6MW turbine as previous chapters (54 and 5) is built in MATLAB. The model is built using the available structural design details of the tower and monopile, which consist of the material properties, tower geometry, foundation, and masses. The first four natural frequencies of the structure are: 0.22, 0.23, 0.95, 0.99 Hz. Rayleigh damping was assumed for the system, with Rayleigh damping coefficients set to 0.0059 for the first bending mode and 0.0072 for the second bending mode. A sinusoidal input load was applied to the structure, with the magnitude of 1e5 N and frequency of 0.4 Hz, and the PINN framework was utilized to estimate both the magnitude and the entire time history of the input load. In this framework, time was fed as the input to the network, while the displacements at 4 key elevations on the turbine structure were used as the outputs. The modal space formulation was used here. The first four modes of the wind turbine were utilized in the analysis, with the network architecture consisting of five hidden layers and 128 neurons per layer. The learning rate was set at 5e-3 for the PINN parameters, while a learning rate of 5e-2 was applied to the input load. A 10 % dropout rate is applied to the network.

During the initial trials, it was observed that the input load, with a magnitude on the order of 1e5, was too large for the network to accurately estimate. To address this challenge, a

normalization approach was adopted. The input load was divided by a constant factor to reduce its magnitude, thereby making it more manageable for the network to predict. The normalization factor was chosen based on the average value of the mass matrix, which served to align the magnitude of the input load with the other terms in the equation of motion. This adjustment ensured that all components of the equation were within the same order of magnitude, facilitating more effective learning. A normalization factor of 1e3 was selected for this purpose. Figure 6.12 and Table 6.8 show results for the wind turbine problem. It could be further improved with hyperparameter tuning and more complex networks.

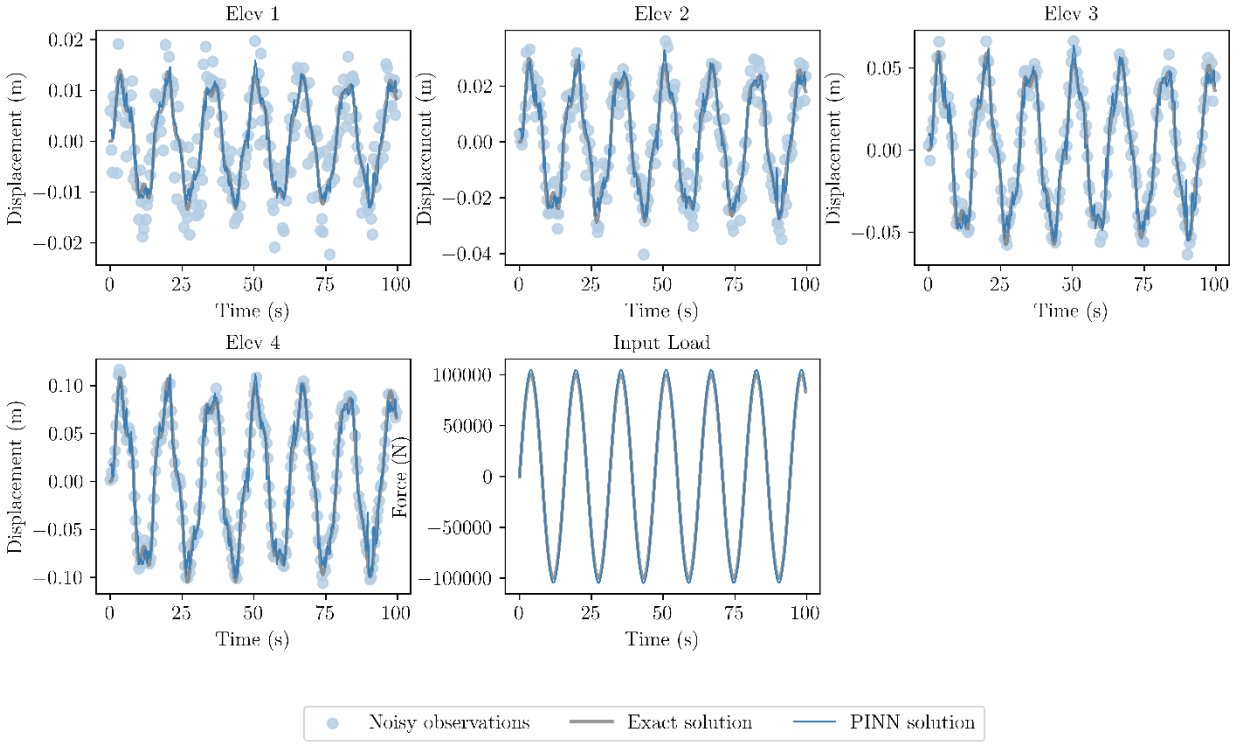


Figure 6.12 Estimated displacements and input load of wind turbine using PINN (p_0 as trainable parameter) in modal space

Table 6.8 Accuracy metrics for the turbine when only the magnitude of the input load is estimated

	RMSE	RRMSE (%)	TRAC
Displacements	(0.001, 0.003, 0.006,0.011)(m)	(18, 18, 18,18)	(0.967, 0.968, 0.969,0.969)
Input	3455(N)	5	-

6.3.9.2 Input load time history estimation

In this problem, the focus was on estimating the entire time history of the input load on a wind turbine structure, similar to the second and fifth problems previously addressed, with the estimation conducted in the modal space. The challenge here was to estimate the input load without assuming any predefined shape, allowing the network to learn the time history purely from the data. FCN was used with five hidden layers, each containing 128 neurons. The network parameters were trained using the Adam optimizer, with learning rates set at 5e-4 for the PINN parameters and 5e-2 for the input load time history. To prevent overfitting, a dropout rate of 10% was applied during the training process. Additionally, a learning rate scheduler was introduced to reduce the learning rate after 15,000 steps. This was necessary because the initial learning rate was set higher to allow the network to make significant adjustments in the early stages of training, bringing the predictions into a reasonable range before fine-tuning.

The results, as shown in Figure 6.13, demonstrate that the network was able to predict the displacements well, though with more error compared to the previous problem where only the magnitude of the input load was estimated. This increase in error is expected given the added complexity of estimating both the shape and magnitude of the input load simultaneously. The network managed to capture the general shape and magnitude of the input load effectively. The performance metrics, presented in Table 6.9, reflect these findings. The ability to estimate the full

time history of the input load, rather than just the magnitude, represents a significant advancement in the application of PINN frameworks.

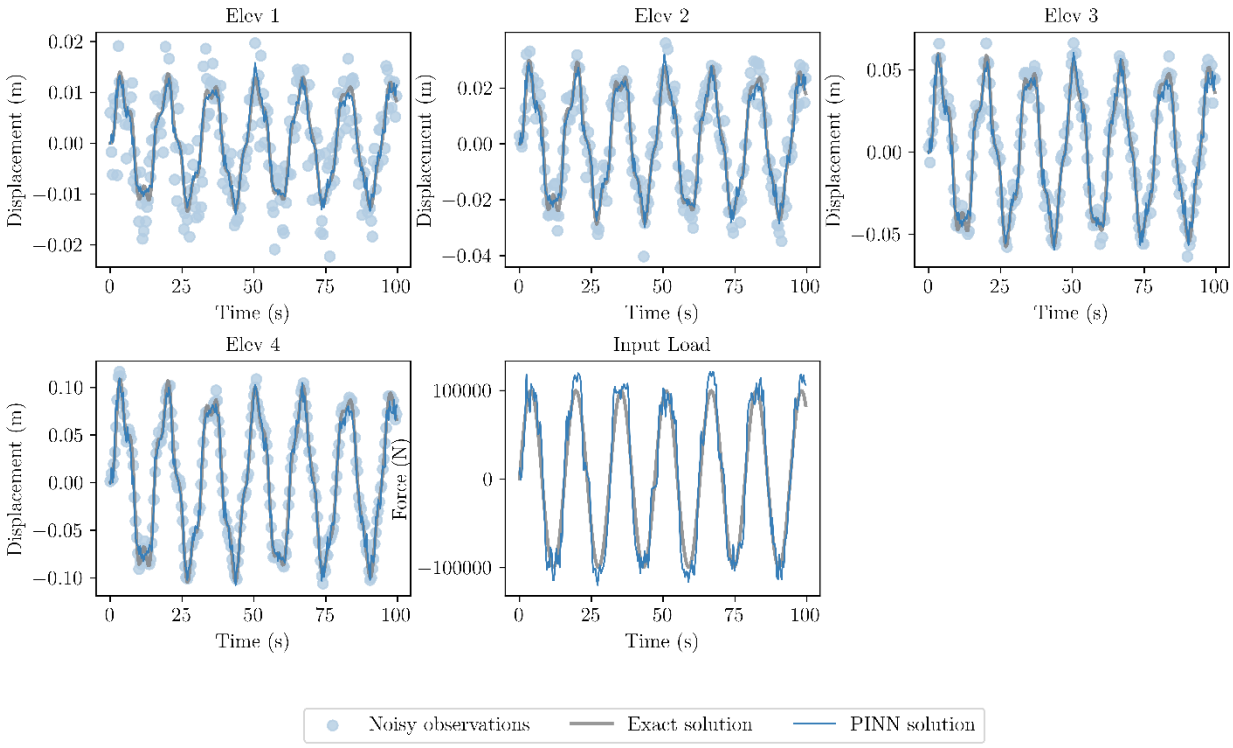


Figure 6.13 Estimated displacements and input load of wind turbine using PINN (F(t) as trainable parameters) in modal space

Table 6.9 Accuracy metrics for the turbine when only the time history of the input load is estimated

	RMSE	RRMSE (%)	TRAC
Displacements	(0.001, 0.003, 0.005,0.009)(m)	(16, 15, 15,15)	(0.976, 0.977, 0.978,0.979)
Input	19778 (N)	28	0.952

The first set of problems relied on the measurement of all DOFs, a requirement that is shared by the second framework. However, by transitioning to modal space, where the equations of motion could be enforced in a decoupled manner, this requirement was effectively reduced. This transition not only improved computational efficiency but also addressed the practical limitations

of measuring all DOFs in large structures, such as wind turbines. Future work will focus on adapting the second framework to operate in modal space, with plans to test this approach on the turbine model.

The promising results achieved on the 3DOF model, and subsequently on the wind turbine model, highlight the effectiveness of these frameworks, whether estimating the magnitude or the full time history of the input load. Without the integration of physical knowledge, purely data-driven frameworks would struggle to estimate input loads, as they typically focus on matching available data and cannot be trained to estimate parameters for which no direct measurements exist. One of the challenges encountered (in the wind turbine problem) was the need for normalization when dealing with large input magnitudes. This challenge was addressed in the first framework by introducing a normalization factor based on the mass matrix, ensuring that the network could effectively handle large inputs. This normalization approach could also be further developed and applied within the second framework to enhance its robustness.

6.4 Conclusions

The research presented in this chapter introduces two innovative physics-informed neural network frameworks for the estimation of input loads in offshore wind turbines. These frameworks effectively address the critical challenge of estimating input loads in scenarios where direct measurement is impractical, thereby advancing the capabilities of real-time structural health monitoring and control in offshore environments. The first framework demonstrated its robustness by dynamically estimating input loads as a learnable parameter while adhering to the equations of motion. The second framework further enhanced this approach by incorporating a dual-network

architecture, allowing for the simultaneous estimation of input loads and reconstruction of dynamic responses.

The application of these frameworks to both simplified 3DOF models and more complex wind turbine models highlighted their versatility and accuracy. The use of modal space transformation was particularly beneficial in reducing computational complexity while maintaining the fidelity of the system's dynamics, thus making these frameworks suitable for large-scale structures like wind turbines. Despite the inherent challenges, such as the need for normalization when dealing with large input magnitudes, the proposed methods provided reliable load estimations even with limited measurement data.

This research underscores the potential of physics-informed machine learning in enhancing the accuracy and efficiency of input load estimation in offshore wind turbines. Future work will focus on further refining these frameworks, including adapting the second framework for use in modal space and applying these methods to full-scale offshore wind turbine models and real data to assess their performance in real-world scenarios.

7 Conclusions and Future Work

This dissertation has presented a comprehensive exploration of methodologies for enhancing the structural health monitoring of offshore wind turbines focusing on optimal sensor placement, minimal sensor placement, physics-based input load estimation, hybrid and physics-informed machine learning models. The key findings across the chapters have advanced our understanding and provided practical solutions for the effective monitoring and maintenance of OWTs.

Chapter 2 introduced a sophisticated framework for Optimal Sensor Placement grounded in information theory. This framework was developed to optimize the placement of sensors for parameter estimation and strain history prediction on OWTs, while also considering the costs associated with sensor installation. The study applied this framework to a realistic numerical model of an offshore wind turbine, demonstrating that strategic sensor placement can drastically reduce uncertainties in structural parameter estimation and improve the accuracy of strain predictions. This chapter highlighted the importance of balancing information gain with cost-effectiveness, offering a valuable tool for industry professionals tasked with the SHM of large-scale OWTs.

Chapter 3 explored the application of a theoretical observability framework to assess the observability of the system under study. The primary objective was to determine whether a minimal yet effective sensor network could capture the necessary structural information for SHM. While this chapter provided a valuable application of the observability framework to the specific context of wind turbines. The results from this analysis offer insights into the design of sensor networks. This work lays the groundwork for more in-depth studies, with plans to further refine these methods and extend their application to differentiate between various types of loads, such as wind and wave loads, which are critical for the detailed monitoring of OWTs.

Chapter 4 presented a novel recursive window-based Bayesian estimator designed to address the challenge of input load estimation using output-only measurements. This estimator was validated through both numerical simulations and real-world data from an OWT in the North Sea. The study found that this estimator significantly outperforms traditional methods like the augmented Kalman filter, and linear regression. By enabling more accurate real-time estimation of input loads and virtual sensing of unmeasured responses, this work enhances the capability of digital twin models for OWTs, facilitating more reliable and proactive maintenance strategies.

Chapter 5 introduced an innovative hybrid surrogate model that combines data-driven and physics-based approaches to estimate input loads in real-time. This model was carefully designed to overcome the challenges posed by the limited availability of experimental data, which is a common issue in the offshore wind industry. By employing transfer learning and multitask learning techniques, the model was first trained on synthetic data generated from physics-based simulations and then fine-tuned using real-world data. This hybrid approach not only reduced computational costs but also improved the model's robustness and accuracy across various operational conditions. Additionally, the use of Monte Carlo dropout for uncertainty quantification provided a deeper insight into the reliability of the model's predictions, making it a valuable tool for advancing SHM practices in offshore wind energy.

Chapter 6 explored the application of physics-informed neural networks to the complex problem of input load estimation in scenarios where direct load measurements are not available. Two innovative frameworks were developed, both incorporating the fundamental equations of motion into the loss functions of the neural networks to ensure that the model’s predictions adhered to the physical laws governing the system. The first framework treated the input load as an

additional trainable parameter, dynamically adjusting it based on observed data. The second framework employed a dual-network approach, where interconnected networks worked together to predict both the input loads and the dynamic responses of the structure. These frameworks were validated on multi-degree-of-freedom systems, demonstrating their potential for application to real-world OWTs. This chapter highlights the promising future of integrating machine learning with physics-based models to solve inverse problems in engineering.

Future work

- Given the challenge of obtaining real input load data through direct measurement, future research could explore the development of unsupervised or semi-supervised machine learning models that do not require extensive labeled datasets. These models could infer loads based on limited data, making them more practical for industry use.
- Expanding the current PINNs to incorporate multi-physics models, such as coupling fluid dynamics with structural mechanics, could provide a more comprehensive understanding of how different environmental factors impact offshore wind turbine structures.
- Future work could explore the application of transfer learning techniques to adapt input load estimation models developed for one type of turbine to others with minimal retraining. This would make the models more versatile and easier to deploy across different turbine platforms.
- Incorporating Bayesian inference methods into load estimation models could provide a probabilistic understanding of the uncertainties associated with the estimated loads.
- Research could explore the use of high-frequency data from sensors to improve the temporal resolution of input load estimations. This would allow for the detection of short-

duration, high-intensity events (such as gusts or waves) that could impact the structural integrity of the turbine.

- Machine learning techniques could be employed to detect anomalies or irregularities in load estimations that may indicate potential issues with the turbine or the monitoring system itself. This could help in early identification of problems, preventing damage before it occurs.
- Future studies could investigate the integration of input load estimation models with predictive analytics tools to forecast future loading events. This would enable proactive maintenance strategies, allowing operators to anticipate and mitigate potential issues before they arise.

8 References

1. IEA. Offshore Wind Outlook 2019: International Energy Agency,

; 2019. Available from: https://www.iea.org/reports/offshore-wind-outlook-2019.

2. Adedipe O, Brennan F, Kolios A. Corrosion fatigue load frequency sensitivity analysis. Marine Structures. 2015;42:115-36.

3. Brennan F, Tavares I. Fatigue design of offshore steel mono-pile wind substructures. Proceedings of the Institution of Civil Engineers-Energy. 2014;167(4):196-202.

4. Moaveni B. System and damage identification of civil structures: UC San Diego; 2007.

5. Guidelines for Structural Health Monitoring for Offshore Wind Turbine Towers & Foundations. 2017. Report No.: 16-1036.

6. Behmanesh I, Moaveni B. Probabilistic identification of simulated damage on the Dowling Hall footbridge through Bayesian finite element model updating. Structural Control and Health Monitoring. 2015;22(3):463-83.

7. Beck JL, Katafygiotis LS. Updating models and their uncertainties. I: Bayesian statistical framework. Journal of Engineering Mechanics. 1998;124(4):455-61.

8. Beck JL, Au S-K. Bayesian updating of structural models and reliability using Markov chain Monte Carlo simulation. Journal of engineering mechanics. 2002;128(4):380-91.

9. Beck JL, Yuen K-V. Model selection using response measurements: Bayesian probabilistic approach. Journal of Engineering Mechanics. 2004;130(2):192-203.

10. Behmanesh I, Moaveni B, Papadimitriou C. Probabilistic damage identification of a designed 9-story building using modal data in the presence of modeling errors. Engineering Structures. 2017;131:542-52.

11. Sohn H, Law KH. A Bayesian probabilistic approach for structure damage detection. Earthquake engineering & structural dynamics. 1997;26(12):1259-81.

12. Song M, Yousefianmoghadam S, Mohammadi M-E, Moaveni B, Stavridis A, Wood RL. An application of finite element model updating for damage assessment of a two-story reinforced concrete building and comparison with lidar. Structural Health Monitoring. 2018;17(5):1129-50.

13. Yuen KV, Beck JL, Au SK. Structural damage detection and assessment by adaptive Markov chain Monte Carlo simulation. Structural Control and Health Monitoring. 2004;11(4):327-47.

14. CAPELLARI G. Optimal design of sensor networks for structural health monitoring: Politecnico Di Milano; 2018.

15. Papadimitriou C. Optimal sensor placement methodology for parametric identification of structural systems. Journal of sound and vibration. 2004;278(4-5):923-47.

16. Heo G, Jeon J. An experimental study of structural identification of bridges using the kinetic energy optimization technique and the direct matrix updating method. Shock and Vibration. 2016;2016.

17. Heo G, Wang M, Satpathi D. Optimal transducer placement for health monitoring of long span bridge. Soil dynamics and earthquake engineering. 1997;16(7-8):495-502.

18. Imamovic N. Model validation of large finite element model using test data 1998.

19. Kammer DC. Sensor placement for on-orbit modal identification and correlation of large space structures. Journal of Guidance, Control, and Dynamics. 1991;14(2):251-9.

20. Kammer DC. Optimal sensor placement for modal identification using system-realization methods. Journal of Guidance, Control, and Dynamics. 1996;19(3):729-31.

21. Kirkegaard PH, Brincker R. On the optimal location of sensors for parametric identification of linear structural systems. Mechanical Systems and Signal Processing. 1994;8(6):639-47.

22. Udwadia FE. Methodology for optimum sensor locations for parameter identification in dynamic systems. Journal of engineering mechanics. 1994;120(2):368-90.

23. Yang C, Lu Z. An interval effective independence method for optimal sensor placement based on non-probabilistic approach. Science China Technological Sciences. 2017;60(2):186-98.

24. Heredia-Zavoni E, Esteva L. Optimal instrumentation of uncertain structural systems subject to earthquake ground motions. Earthquake engineering & structural dynamics. 1998;27(4):343-62.

25. Heredia-Zavoni E, Montes-Iturrizaga R, Esteva L. Optimal instrumentation of structures on flexible base for system identification. Earthquake engineering & structural dynamics. 1999;28(12):1471-82.

26. Flynn EB, Todd MD. A Bayesian approach to optimal sensor placement for structural health monitoring with application to active sensing. Mechanical Systems and Signal Processing. 2010;24(4):891-903.

27. Shannon CE. A mathematical theory of communication. The Bell system technical journal. 1948;27(3):379-423.

28. Papadimitriou C, Beck JL, Au S-K. Entropy-based optimal sensor location for structural model updating. Journal of Vibration and Control. 2000;6(5):781-800.

29. Yuen K-V, Katafygiotis LS, Papadimitriou C, Mickleborough NC. Optimal sensor placement methodology for identification with unmeasured excitation. J Dyn Sys, Meas, Control. 2001;123(4):677-86.

30. Metallidis P, Verros G, Natsiavas S, Papadimitriou C. Fault detection and optimal sensor location in vehicle suspensions. Journal of vibration and control. 2003;9(3-4):337-59.

31. Papadimitriou C. Pareto optimal sensor locations for structural identification. Computer methods in applied mechanics and engineering. 2005;194(12-16):1655-73.

32. Papadimitriou C. Optimal Sensor Placement for Response Reconstruction in Structural Dynamics. Model Validation and Uncertainty Quantification, Volume 3: Springer; 2020. p. 205-10.

33. Lindley DV. On a measure of the information provided by an experiment. The Annals of Mathematical Statistics. 1956:986-1005.

34. Kullback S, Leibler RA. On information and sufficiency. The annals of mathematical statistics. 1951;22(1):79-86.

35. Papadimitriou C, Lombaert G. The effect of prediction error correlation on optimal sensor placement in structural dynamics. Mechanical Systems and Signal Processing. 2012;28:105-27.

36. Nabiyan MS, Khoshnoudian F, Moaveni B, Ebrahimian H. Mechanics-based model updating for identification and virtual sensing of an offshore wind turbine using sparse measurements. Structural Control and Health Monitoring. 2021;28(2):e2647.

37. Papadimitriou C, Fritzen CP, Kraemer P, Ntotsios E. Fatigue predictions in entire body of metallic structures from a limited number of vibration sensors using Kalman filtering. Structural Control and Health Monitoring. 2011;18(5):554-73.

38. Henkel M, Häfele J, Weijtjens W, Devriendt C, Gebhardt C, Rolfes R. Strain estimation for offshore wind turbines with jacket substructures using dual-band modal expansion. Marine Structures. 2020;71:102731.

39. Iliopoulos A, Shirzadeh R, Weijtjens W, Guillaume P, Van Hemelrijck D, Devriendt C. A modal decomposition and expansion approach for prediction of dynamic responses on a monopile offshore wind turbine using a limited number of vibration sensors. Mechanical Systems and Signal Processing. 2016;68:84-104.

40. Iliopoulos A, Weijtjens W, Van Hemelrijck D, Devriendt C. Fatigue assessment of offshore wind turbines on monopile foundations using multi-band modal expansion. Wind Energy. 2017;20(8):1463-79.

41. Skafte A, Kristoffersen J, Vestermark J, Tygesen UT, Brincker R. Experimental study of strain prediction on wave induced structures using modal decomposition and quasi static Ritz vectors. Engineering Structures. 2017;136:261-76.

42. Tarpø M, Nabuco B, Georgakis C, Brincker R. Expansion of experimental mode shape from operational modal analysis and virtual sensing for fatigue analysis using the modal expansion method. International Journal of Fatigue. 2020;130:105280.

43. Häfele J, Hübler C, Gebhardt CG, Rolfes R. A comprehensive fatigue load set reduction study for offshore wind turbines with jacket substructures. Renewable Energy. 2018;118:99-112.

44. Abdullah MM, Richardson A, Hanif J. Placement of sensors/actuators on civil structures using genetic algorithms. Earthquake engineering & structural dynamics. 2001;30(8):1167-84.

45. Bedrossian H, Masri S. Optimal placement of sensors and shakers for modal identification. Computational Stochastic Mechanics, Millpress, Rotterdam. 2003;5357.

46. Papadimitriou C, editor Applications of genetic algorithms in structural health monitoring. Proc 5th World Congress on Computational Mechanics, Vienna, Austria; 2002.

47. Yao L, Sethares WA, Kammer DC. Sensor placement for on-orbit modal identification via a genetic algorithm. AIAA journal. 1993;31(10):1922-8.

48. Kullaa J. Bayesian virtual sensing for full-field dynamic response estimation. Procedia engineering. 2017;199:2126-31.

49. Kullaa J. Bayesian virtual sensing in structural dynamics. Mechanical Systems and Signal Processing. 2019;115:497-513.

50. Ercan T, Papadimitriou C. Optimal Sensor Placement for Reliable Virtual Sensing Using Modal Expansion and Information Theory. Sensors. 2021;21(10):3400.

51. McKenna F. OpenSees: a framework for earthquake engineering simulation. Computing in Science & Engineering. 2011;13(4):58-66.

52. Bhattacharya S. Design of foundations for offshore wind turbines: Wiley Online Library; 2019.

53. Kalman RE, editor On the general theory of control systems. Proceedings first international conference on automatic control, Moscow, USSR; 1960.

54. Kalman RE. Mathematical description of linear dynamical systems. Journal of the Society for Industrial and Applied Mathematics, Series A: Control. 1963;1(2):152-92.

55. Hermann R, Krener A. Nonlinear controllability and observability. IEEE Transactions on automatic control. 1977;22(5):728-40.

56. Chow W-L. Über systeme von liearren partiellen differentialgleichungen erster ordnung. Mathematische Annalen. 1940;117(1):98-105.

57. Hermann R, editor On the accessibility problem in control theory. International Symposium on Nonlinear Differential Equations and Nonlinear Mechanics; 1963: Elsevier.

58. Haynes G, Hermes H. Nonlinear controllability via Lie theory. SIAM Journal on Control. 1970;8(4):450-60.

59. Brockett RW. System theory on group manifolds and coset spaces. SIAM Journal on control. 1972;10(2):265-84.

60. Sussmann HJ, Jurdjevic V. Controllability of nonlinear systems. Journal of Differential Equations. 1972;12.

61. Diop S, Fliess M, editors. Nonlinear observability, identifiability, and persistent trajectories. [1991] Proceedings of the 30th IEEE Conference on Decision and Control; 1991: IEEE.

62. Diop S, Wang Y, editors. Equivalence between algebraic observability and local generic observability. Proceedings of 32nd IEEE Conference on Decision and Control; 1993: IEEE.

63. Anguelova M. Observability and identifiability of nonlinear systems with applications in biology: Chalmers Tekniska Hogskola (Sweden); 2007.

64. Glad S, Ljung L, editors. Model structure identifiability and persistence of excitation. 29th IEEE Conference on Decision and Control; 1990: IEEE.

65. Ljung L, Glad T. On global identifiability for arbitrary model parametrizations. automatica. 1994;30(2):265-76.

66. Sedoglavic A, editor A probabilistic algorithm to test local algebraic observability in polynomial time. Proceedings of the 2001 international symposium on Symbolic and algebraic computation; 2001.

67. Chatzis MN, Chatzi EN, Smyth AW. On the observability and identifiability of nonlinear structural and mechanical systems. Structural Control and Health Monitoring. 2015;22(3):574-93.

68. Shi X, Williams M, Chatzis M. A robust algorithm to test the observability of large linear systems with unknown parameters. Mechanical Systems and Signal Processing. 2021;157:107633.

69. Maes K, Chatzis M, Lombaert G. Observability of nonlinear systems with unmeasured inputs. Mechanical Systems and Signal Processing. 2019;130:378-94.

70. Shi X, Chatzis M. An efficient algorithm to test the observability of rational nonlinear systems with unmeasured inputs. Mechanical Systems and Signal Processing. 2022;165:108345.

71. Maes K, Chatzis M, Vandebril R, Lombaert G. Observability of modally reduced order models with unknown parameters. Mechanical Systems and Signal Processing. 2021;146:106993.

72. Mazzoni S, McKenna F, Scott MH, Fenves GL. OpenSees command language manual. Pacific earthquake engineering research (PEER) center. 2006;264(1):137-58.

73. Igwemezie V, Mehmanparast A, Kolios A. Current trend in offshore wind energy sector and material requirements for fatigue resistance improvement in large wind turbine support structures–A review. Renewable and Sustainable Energy Reviews. 2019;101:181-96.

74. Nozari A, Kuchma D, Hines E, editors. Effect of wind directionality on fatigue life of monopile support structures for offshore wind turbines. Journal of Physics: Conference Series; 2020: IOP Publishing.

75. Johansen SS, Nejad AR, editors. On digital twin condition monitoring approach for drivetrains in marine applications. International Conference on Offshore Mechanics and Arctic Engineering; 2019: American Society of Mechanical Engineers.

76. Moghadam FK, Nejad AR. Online condition monitoring of floating wind turbines drivetrain by means of digital twin. Mechanical Systems and Signal Processing. 2022;162:108087.

77. Moynihan B, Mehrjoo A, Moaveni B, McAdam R, Rüdinger F, Hines E. System identification and finite element model updating of a 6 MW offshore wind turbine using vibrational response measurements. Renewable Energy. 2023;219:119430.

78. Dimitrov N, Kelly MC, Vignaroli A, Berg J. From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases. Wind Energy Science. 2018;3(2):767-90.

79. Jacquelin E, Bennani A, Hamelin P. Force reconstruction: analysis and regularization of a deconvolution problem. Journal of sound and vibration. 2003;265(1):81-107.

80. Mendez Reyes H, Kanev S, Doekemeijer B, van Wingerden J-W. Validation of a lookup-table approach to modeling turbine fatigue loads in wind farms under active wake control. Wind Energy Science. 2019;4(4):549-61.

81. Evans M, Han T, Shuchun Z, editors. Development and validation of real time load estimator on Goldwind 6 MW wind turbine. Journal of Physics: Conference Series; 2018: IOP Publishing.

82. Schröder L, Dimitrov NK, Verelst DR, Sørensen JA, editors. Wind turbine site-specific load estimation using artificial neural networks calibrated by means of high-fidelity load simulations. Journal of Physics: Conference Series; 2018: IOP Publishing.

83. Kalman RE. A new approach to linear filtering and prediction problems. 1960.

84. Mehlan FC, Nejad AR, Gao Z. Digital twin based virtual sensor for online fatigue damage monitoring in offshore wind turbine drivetrains. Journal of Offshore Mechanics and Arctic Engineering. 2022;144(6):060901.

85. Maes K, De Roeck G, Lombaert G, Iliopoulos A, Van Hemelrijck D, Devriendt C, et al., editors. Continuous strain prediction for fatigue assessment of an offshore wind turbine using Kalman filtering techniques. 2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings; 2015: IEEE.

86. Noppe N, Tatsis K, Chatzi E, Devrient C, Weijtjens W, editors. Fatigue stress estimation of offshore wind turbine using a Kalman filter in combination with accelerometers. Proceedings of International Conference on Noise and Vibration Engineering (ISMA 2018), International Conference on Uncertainty in Structural Dynamics (USD 2018); 2018: KU Leuven, Department of Mechanical Engineering.

87. Gillijns S, De Moor B. Unbiased minimum-variance input and state estimation for linear discrete-time systems. Automatica. 2007;43(1):111-6.

88. Lourens E, Reynders E, De Roeck G, Degrande G, Lombaert G. An augmented Kalman filter for force identification in structural dynamics. Mechanical systems and signal processing. 2012;27:446-60.

89. Van der Male P, Lourens E, editors. Operational vibration-based response estimation for offshore wind lattice structures. Structural Health Monitoring and Damage Detection, Volume 7: Proceedings of the 33rd IMAC, A Conference and Exposition on Structural Dynamics, 2015; 2015: Springer.

90. Maes K, Iliopoulos A, Weijtjens W, Devriendt C, Lombaert G. Dynamic strain estimation for fatigue assessment of an offshore monopile wind turbine using filtering and modal expansion algorithms. Mechanical Systems and Signal Processing. 2016;76:592-611.

91. Branlard E, Giardina D, Brown CS. Augmented Kalman filter with a reduced mechanical model to estimate tower loads on a land-based wind turbine: a step towards digital-twin simulations. Wind Energy Science. 2020;5(3):1155-67.

92. Tatsis K, Dertimanis VK, Papadimitriou C, Lourens E, Chatzi E. A general substructure-based framework for input-state estimation using limited output measurements. Mechanical Systems and Signal Processing. 2021;150:107223.

93. Azam SE, Chatzi E, Papadimitriou C. A dual Kalman filter approach for state estimation via output-only acceleration measurements. Mechanical systems and signal processing. 2015;60:866-86.

94. Azam SE, Chatzi E, Papadimitriou C, Smyth A. Experimental validation of the Kalman-type filters for online and real-time state and input estimation. Journal of vibration and control. 2017;23(15):2494-519.

95. Naets F, Croes J, Desmet W. An online coupled state/input/parameter estimation approach for structural dynamics. Computer methods in applied mechanics and engineering. 2015;283:1167-88.

96. Liu L, Su Y, Zhu J, Lei Y. Data fusion based EKF-UI for real-time simultaneous identification of structural systems and unknown external inputs. Measurement. 2016;88:456-67.

97. Ebrahimian H, Astroza R, Conte JP, Papadimitriou C. Bayesian optimal estimation for output-only nonlinear system and damage identification of civil structures. Structural Control and Health Monitoring. 2018;25(4):e2128.

98. Song M, Moaveni B, Ebrahimian H, Hines E, Bajric A. Joint parameter-input estimation for digital twinning of the Block Island wind turbine using output-only measurements. Mechanical Systems and Signal Processing. 2023;198:110425.

99. Chatzi EN, Smyth AW. The unscented Kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures. 2009;16(1):99-123.

100. Astroza R, Ebrahimian H, Conte JP. Material parameter identification in distributed plasticity FE models of frame-type structures using nonlinear stochastic filtering. Journal of Engineering Mechanics. 2015;141(5):04014149.

101. Astroza R, Ebrahimian H, Li Y, Conte JP. Bayesian nonlinear structural FE model and seismic input identification for damage assessment of civil structures. Mechanical Systems and Signal Processing. 2017;93:661-87.

102. Song M, Astroza R, Ebrahimian H, Moaveni B, Papadimitriou C. Adaptive Kalman filters for nonlinear finite element model updating. Mechanical Systems and Signal Processing. 2020;143:106837.

103. Valikhani M, Jahangiri V, Ebrahimian H, Moaveni B, Liberatore S, Hines E. Inverse modeling of wind turbine drivetrain from numerical data using Bayesian inference. Renewable and Sustainable Energy Reviews. 2023;171:113007.

104. Wei D, Li D, Jiang T, Lyu P, Song X. Load identification of a 2.5 MW wind turbine tower using Kalman filtering techniques and BDS data. Engineering Structures. 2023;281:115763.

105. Muto K, Namura N, Ukei Y, Takeda N, editors. Model-based load estimation for wind turbine blade with Kalman filter. 2019 8th International Conference on Renewable Energy Research and Applications (ICRERA); 2019: IEEE.

106. Branlard E, Jonkman J, Brown C, Zang J. A digital-twin solution for floating offshore wind turbines validated using a full-scale prototype. Wind Energy Science Discussions. 2023;2023:1-34.

107. Dimitrov N, Natarajan A, editors. 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; 2019: IOP Publishing.

108. McInerney M, Dhawan AP, editors. Use of genetic algorithms with backpropagation in training of feedforward neural networks. IEEE international conference on neural networks; 1993: IEEE.

109. Boser BE, Guyon IM, Vapnik VN, editors. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory; 1992.

110. Williams CK, Rasmussen CE. Gaussian processes for machine learning: MIT press Cambridge, MA; 2006.

111. Jiang H, Wang H, Vaz MA, Bai X. Research on dynamic response prediction of semi-submersible wind turbine platform in real sea test model based on machine learning. Applied Ocean Research. 2024;142:103808.

112. Singh D, Dwight RP, Laugesen K, Beaudet L, Viré A, editors. Probabilistic surrogate modeling of offshore wind-turbine loads with chained Gaussian processes. Journal of Physics: Conference Series; 2022: IOP Publishing.

113. Neshat M, Nezhad MM, Abbasnejad E, Mirjalili S, Tjernberg LB, Garcia DA, et al. A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the Lillgrund offshore wind farm. Energy conversion and management. 2021;236:114002.

114. Haensch A, Tronci EM, Moynihan B, Moaveni B. Regularized hidden Markov modeling with applications to wind speed predictions in offshore wind. Mechanical Systems and Signal Processing. 2024;211:111229.

115. Black IM, Cevasco D, Kolios A, editors. Deep neural network hard parameter multi-task learning for condition monitoring of an offshore wind turbine. Journal of Physics: Conference Series; 2022: IOP Publishing.

116. Yin X, Zhao X. Big data driven multi-objective predictions for offshore wind farm based on machine learning algorithms. Energy. 2019;186:115704.

117. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nature Reviews Physics. 2021;3(6):422-40.

118. Hao Z, Liu S, Zhang Y, Ying C, Feng Y, Su H, et al. Physics-informed machine learning: A survey on problems, methods and applications. arXiv preprint arXiv:221108064. 2022.

119. Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics–informed neural networks: Where we are and what's next. Journal of Scientific Computing. 2022;92(3):88.

120. Baisthakur S, Fitzgerald B. Physics-Informed Neural Network surrogate model for bypassing Blade Element Momentum theory in wind turbine aerodynamic load estimation. Renewable Energy. 2024;224:120122.

121. Mehrjoo A, Tronci EM, Moynihan B, Moaveni B, Rüdinger F, McAdam R, et al. Recursive Bayesian Estimation of Wind Load on a Monopile-Supported Offshore Wind Turbine Using Output-Only Measurements Mechanical Systems and Signal Processing. 2024 (Under Review).

122. Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. Journal of Big data. 2016;3:1-40.

123. Pan SJ, Yang Q. A survey on transfer learning. IEEE Transactions on knowledge and data engineering. 2009;22(10):1345-59.

124. Bull L, Gardner P, Gosliga J, Rogers T, Dervilis N, Cross E, et al. Foundations of population-based SHM, Part I: Homogeneous populations and forms. Mechanical systems and signal processing. 2021;148:107141.

125. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735-80.

126. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. Biometrika. 1986;71(599-607):6.

127. Jordan MI. Serial order: A parallel distributed processing approach. Advances in psychology. 121: Elsevier; 1997. p. 471-95.

128. Caruana R. Multitask learning. Machine learning. 1997;28:41-75.

129. Zhang Y, Yang Q. A survey on multi-task learning. IEEE transactions on knowledge and data engineering. 2021;34(12):5586-609.

130. Gal Y, Ghahramani Z, editors. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. international conference on machine learning; 2016: PMLR.

131. Branlard E. Generation of time series from a spectrum. Risø, National laboratory for sustainable energy DTU. 2010.

132. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics. 2019;378:686-707.

133. Zhu Y, Zabaras N, Koutsourelakis P-S, Perdikaris P. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. Journal of Computational Physics. 2019;394:56-81.

134. HAYWOOD-ALEXANDER M, CHATZI E. Physics-Informed Neural Networks for One-Step-Ahead Prediction of Dynamical Systems. STRUCTURAL HEALTH MONITORING 2023. 2023.

135. Huang Z, Yin X, Liu Y. Physics-guided deep neural network for structural damage identification. Ocean Engineering. 2022;260:112073.

136. Liu W, Lai Z, Bacsa K, Chatzi E. Physics-guided Deep Markov Models for learning nonlinear dynamical systems with uncertainty. Mechanical Systems and Signal Processing. 2022;178:109276.

137. Lai Z, Mylonas C, Nagarajaiah S, Chatzi E. Structural identification with physics-informed neural ordinary differential equations. Journal of Sound and Vibration. 2021;508:116196.

138. Ritto T, Rochinha F. Digital twin, physics-based model, and machine learning applied to damage detection in structures. Mechanical Systems and Signal Processing. 2021;155:107614.

139. Peres RS, Guedes M, Miranda F, Barata J. Simulation-based data augmentation for the quality inspection of structural adhesive with deep learning. IEEE Access. 2021;9:76532-41.

140. Bao N, Zhang T, Huang R, Biswal S, Su J, Wang Y. A Deep Transfer Learning Network for Structural Condition Identification with Limited Real-World Training Data. Structural Control and Health Monitoring. 2023;2023(1):8899806.

141. Wang Z, Liu Q, Chen H, Chu X. A deformable CNN-DLSTM based transfer learning method for fault diagnosis of rolling bearing under multiple working conditions. International Journal of Production Research. 2021;59(16):4811-25.

142. Tronci EM, Beigi H, Feng MQ, Betti R. A transfer learning SHM strategy for bridges enriched by the use of speaker recognition x-vectors. Journal of Civil Structural Health Monitoring. 2022;12(6):1285-98.

143. Moynihan B, Tronci EM, Hughes MC, Moaveni B, Hines E. Virtual sensing via Gaussian Process for bending moment response prediction of an offshore wind turbine using SCADA data. Renewable Energy. 2024;227:120466.

144. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, et al. Automatic differentiation in pytorch. 2017.