DISSERTATION

SIMULATION, OPTIMAL CONTROL, AND CONTROL CO-DESIGN OF WIND AND MARINE TURBINES USING DERIVATIVE FUNCTION SURROGATE MODELS

Submitted by

Athul Krishna Sundarrajan

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2025

Doctoral Committee:

Advisor: Daniel R. Herber

James Cale
Steven Conrad
Gaofeng Jia

ABSTRACT

SIMULATION, OPTIMAL CONTROL, AND CONTROL CO-DESIGN OF WIND AND MARINE TURBINES USING DERIVATIVE FUNCTION SURROGATE MODELS

Researchers in industry and academia are investigating different approaches to identify cost-effective, system-level optimal designs of wind and marine turbines, which would lead to wide-scale adoption of these technologies. Central to these efforts are open-source computational modeling tools and reference models of these systems. Researchers typically propose improvements to the reference models, implement these changes using the modeling tools, and compare the performance to the baseline performance. As a part of the design process, researchers have investigated the impact the controls have on the performance of these systems. A controller is necessary to ensure that these systems can produce power and minimize loads in offshore environments. In addition to the controllers, researchers are utilizing design optimization studies to efficiently explore the design space associated with these systems, including identification of optimal designs and non-intuitive trade-offs. Additionally, control co-design (CCD) studies, where both the physical aspects and the controller are optimized simultaneously, have been identified as a potential pathway to minimize the cost of energy associated with these systems.

Since wind and marine turbines are at different technology readiness levels, the key focus of the research being carried out for them is different. A key requirement for marine turbines is to develop open-source modeling tools and reference models. Recently, OpenFAST, an open-source tool used for wind turbines, has been extended for the simulation of marine turbines. An open-source reference model of a floating marine turbine has been developed as well. However, to effectively simulate this model using OpenFAST, a controller is required. For wind turbines, there are multiple open-source reference models available. A key research focus is to formulate and solve multi-objective optimization studies with a focus on the controller, to explore the trade-offs

between multiple conflicting objectives. To aid this, design tools built on OpenFAST have been developed.

However, a main drawback of utilizing OpenFAST for design optimization studies is the computational cost. Depending on the system, it can take anywhere between 10 minutes to 7 hours to simulate OpenFAST. Design optimization studies require several hundred function evaluations to identify the optimal design. Therefore, utilizing OpenFAST directly can be ineffective. The core contribution of this dissertation is to address this issue.

To address this issue, different data-driven low-fidelity or surrogate modeling approaches are explored. A variety of approaches are considered, ranging from system identification approaches to recurrent neural networks. In addition to these, an approach called a derivative function surrogate model (DFSM), which has been utilized in recent studies to approximate the dynamic response of wind turbines, is also explored. A novel approach is developed to construct a DFSM that could be used for simulation and control optimization studies. Low-fidelity models using these approaches are constructed, and the trade-offs associated with each approach are explored. Three factors inform the trade-offs between these models. The simulation time using these models, how extendable these modeling approaches are towards modeling the system response for different wind speeds, and the variance in response of these models for different signals. Results show that the proposed DFSM approach balances computational time and model accuracy better than the system identification and deep learning based models.

The DFSM approach is tested for five different systems, which include four different floating wind turbine systems and a marine turbine system. The DFSM results in nearly 12 to 700 times speedup in terms of simulation time, in comparison to the high fidelity model across these five systems. Then, different case studies are carried out that explore the efficacy of the DFSM for optimal control and multi-objective CCD studies. The key takeaways from these studies can be summarized as follows. From the open-loop optimal control study carried out for a marine turbine system, it was found that for a 5% increase in the generated power, there is a corresponding 15% increase in the tower base fatigue loads. Using a multi-fidelity optimization approach for closed-loop op-

timal control studies, it is possible to identify a set of controller parameters that are within 0.5% of the high-fidelity maxima, with nearly a quarter of the number of high-fidelity function calls. And finally, a multi-objective CCD study is carried out to identify plant and controller parameters that balance conflicting objectives of tower-base fatigue load minimization and improving power quality for floating wind turbines. The study considers a single plant variable, namely the column spacing of the semisubmersible platform, and the controller parameters correspond to the above-rated blade pitch controller. A nested optimization approach is used to solve this problem, and for every iteration of the inner loop, a multi-objective controller optimization problem is solved. The identified point has just 2% higher fatigue damage loading than the optimal case, while just having 15% higher value of power quality metric.

ACKNOWLEDGEMENTS

The doctoral journey that has resulted in this document would not have been possible without the encouragement and support of my advisor, Dr. Daniel Herber. I have greatly benefited from his thoughtful and methodical approach to problem solving, which has taught me to break down complex challenges into manageable steps. His emphasis on steady progress over unattainable perfection has shaped how I approach research and work in general, helping me focus on momentum and continuous improvement. His commitment to high standards, coupled with an encouraging and solution-oriented attitude, made this journey both rigorous and rewarding.

This work has also been enriched by the support and mentorship of my collaborators at the National Renewable Energy Laboratory (NREL). I am especially grateful to Dr. Daniel Zalkind, Dr. Hannah Ross, Dr. Toan Thanh Tran, and Will Wiley for hosting me during my internship and for their patience in answering countless questions—both big and small. I would also like to thank the members of my doctoral committee—Dr. Jim Cale, Dr. Steven Conrad, and Dr. Gaofeng Jia—for their valuable advice, constructive feedback, and guidance.

I am thankful to the faculty of the Department of Systems Engineering, particularly Dr. Thomas Bradley and Dr. Tim Coburn, for their mentorship and for shaping my growth as a researcher. A special thanks goes to Dr. Erika Gallegos and her data visualization class—many of the concepts I learned there found their way into the countless figures in this dissertation. I am also grateful to our graduate advisors, Dr. Debra Dandaneau and Ingrid Bridge, for their steady support and guidance in navigating the less-visible but equally important administrative side of graduate school.

To the members of the Herber Research Group—Dr. Saeed Azad, Roberto Vercellino, Temitope Adeniji, James Wheaton, Shawn Miller, and Zirradin Gulumjanli—thank you for the countless hours spent discussing research, sharing ideas, and laughing through the challenges. To my friends—Sanket, Muhundan, Parneeth, Ambika, Projith, Yixuan, Emily Ku, Siddhesh, Devika, Ananth, Abbi, Venky—thank you for your unwavering support, much-needed distractions, and encouragement. I am grateful to Cibi and Sharayah for opening their home to me and giving me a

place to crash during this final stretch. And to Milena—thank you for your patience, understanding, and constant encouragement.

Finally, to my family and especially my parents—thank you for your unconditional love, constant encouragement, and the sacrifices you have made for me. This journey would not have been possible without you, and I owe everything to your support.

A significant portion of the work in this dissertation was carried out on the ASHA cluster provided by the Walter Scott Jr. College of Engineering. Some of the research was also performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy, located at the National Renewable Energy Laboratory.

DEDICATION

We shall not cease from exploration

And the end of all our exploring

Will be to arrive where we started

And know the place for the first time.

–T.S. Eliot

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Wind Turbines

The use of wind turbines has proven to be an effective way to produce clean electricity and reduce global dependence on fossil fuels [1]. The maturation of this technology has led to the widespread adoption of wind energy across the world. Currently, over 5% of the global electricity demand is met by power produced from wind turbines [2]. In some countries, such as Denmark, nearly 58% of electricity is generated from wind turbines, while countries like Spain, Germany, and the Netherlands produce more than 25% of their power from wind energy [3]. Nearly 29 countries use wind turbines to satisfy at least 10% of their electricity demand.

Typically, wind turbines have been deployed in onshore environments, in strategic areas with steady winds. However, due to the rising demand for constant, high-quality power, there is a corresponding increase in the demand for wind energy. This has led to extensive investigation into avenues for maximizing wind energy production. Examples include airborne wind energy systems [4], hybrid systems such as wind-solar [5] or wind-hydrogen [6], and especially offshore wind energy systems [7–9]. The focus of this dissertation is on offshore wind turbines. The key motivation behind offshore wind turbines is to harvest the consistently high winds found in offshore environments. Currently, a total of 64.3 GW of offshore wind capacity has been installed worldwide [10]. The majority of offshore wind turbines have been deployed in water depths of up to 55 meters in a *fixed-bottom* configuration, where the turbine is installed on a foundation anchored to the ocean floor.

1.1.1 Floating Offshore Wind Turbines

Even higher wind speeds are present in deep offshore environments, which can be harvested to meet a greater portion of the rising global energy demand. In these deeper waters, the mate-

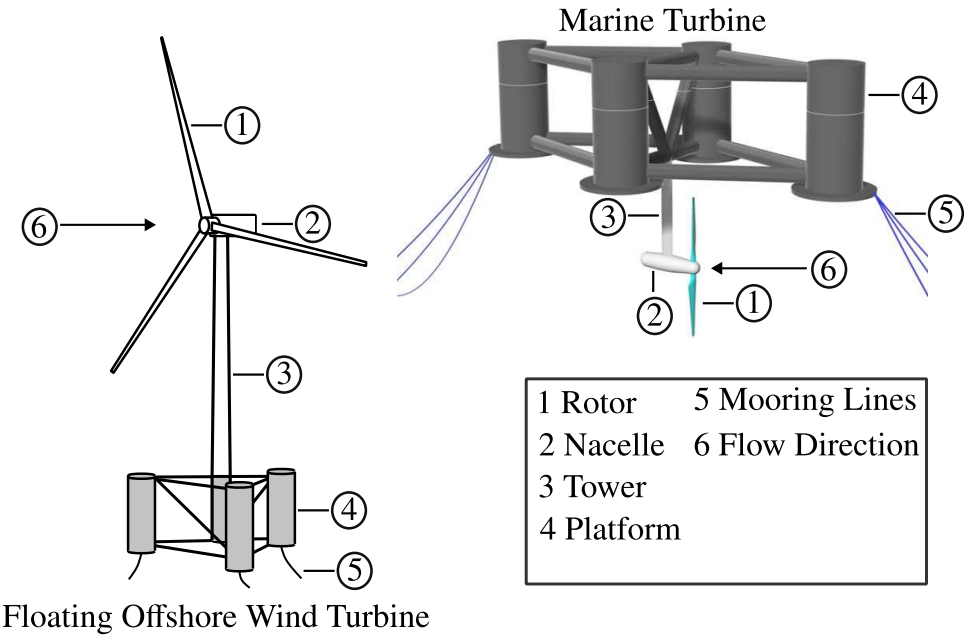


Figure 1.1: Common ontology of floating offshore wind turbine (FOWT) and floating marine turbine system. *Illustration courtesy of NREL.*

rial, installation, and maintenance costs of a fixed supporting foundation would be prohibitively high, making the turbine nonviable. Therefore, to effectively deploy wind turbines in deeper offshore environments, a different support structure is required. Recently, floating offshore wind turbines (FOWTs) have been identified as a promising renewable technology capable of harvesting wind energy in deep-sea environments [9, 11, 12]. The potential advantages of FOWTs have generated significant interest from both academia and industry in identifying suitable designs and architectures that make these technologies cost-effective [13]. Several companies are currently testing different prototypes to assess their practical viability [14–16].

1.2 Marine Turbines

Marine hydrokinetic turbines, or *marine turbines* for short, are similar to wind turbines, but instead of converting the kinetic energy in wind, they convert the kinetic energy in flowing water—such as in riverine and tidal environments—into electrical energy [13,17]. Similar to wind turbines, they have specialized blades that rotate when water flows over them, driving a generator to pro-

duce electricity. A recent study estimated that 2300 TWh/yr of energy is available from tidal and riverine currents in the United States [18]. A significant portion of this energy is located in remote regions such as Alaska and Hawaii. These remote communities still rely on diesel generators for reliable electricity generation. By harnessing the energy available in tidal and riverine currents, it is possible to reduce the use of fossil fuels in these areas [19].

However, marine turbines are not as widespread as wind turbines and have a lower rate of adoption [12,20]. This is due to several factors associated with the life cycle of these systems [21]. Typically, for the construction and deployment of both wind and marine turbines, the first step is to estimate the amount of energy that can be harvested from a specific site [22]. For wind turbines, these assessments can be performed using widely available weather data. However, for marine turbines, suitable sites must be identified specifically for turbine installation. At feasible sites, the average flow speed must exceed 2 m/s for the project to be viable [23]. Additionally, sites located in straits or inlets are preferred since the tidal water flow is funneled through narrow passages, resulting in higher flow speeds and predictable bidirectional flow. Once a site is identified, the flow characteristics at that site must be estimated [24]. Furthermore, the ocean floor must be mapped to determine whether a turbine can be mounted there. Because tidal and riverine environments are harsher, they pose unique challenges for the design, deployment, and maintenance of marine turbines. As these systems operate in a denser fluid, the blades and hydrofoils must be designed to withstand the loads experienced during operation [13, 17, 25]. Additionally, the blades must be designed and the turbine operated in a manner that prevents cavitation [26]. The turbine must also be designed to minimize biofouling and sediment deposition. Nevertheless, the potential benefits of this technology have motivated researchers to address these challenges and develop methods to reduce the cost of energy for such systems.

Several studies and white papers have investigated the critical needs and proposed solutions that could lead to the widespread adoption of these systems. These investigations examine all aspects of the system life cycle, including resource characterization and assessment [27], conceptual design and control [10], digitalization for operation and maintenance [28], and power quality

assessment [29]. The focus of this dissertation is on conceptual design and control, and on analyzing how these areas influence the performance of such systems. These studies aim to identify what design and control changes can be implemented and how these changes affect system performance.

Typically, such studies begin with a baseline system to understand its performance limits. Researchers then identify design or control improvements to overcome these limitations. For example, in Refs. [30, 31], the authors analyzed the performance of an FOWT and found that the interaction between platform dynamics and system control poses specific challenges that must be addressed to minimize the cost of energy. To address this, they proposed and investigated the design and control of a platform actuator system. Similarly, in Ref. [32], the authors aimed to reduce the levelized cost of energy (LCOE) by 25% compared to the baseline system. To achieve this ambitious target, they proposed several innovations, including using a two-bladed rotor, adopting a downwind concept, and increasing the generator size.

There are three essential components for carrying out conceptual design and control studies: detailed modeling tools, reference models, and, more recently, optimization studies. These three components are discussed briefly in the following sections.

1.3 Modeling Tools

Since physically implementing design and control changes on real-world systems is not feasible, researchers utilize computational modeling tools to understand their impact. A variety of model types are used for such investigations. These models can be categorized based on their fidelity level, ranging from simple low-fidelity models that employ simplified physics [33–36], to detailed high-fidelity models that use computational fluid dynamics (CFD) to capture flow behavior and finite element analysis (FEA) to quantify the resulting loads on system components [37–42]. However, for design and control investigations, models at the extremities of the fidelity spectrum are typically not used. To test system performance, it is necessary to simulate operation under different environmental conditions [43], which can require several hundred to several thousand simulations. These extreme models do not scale well for this purpose: simplified models fail to

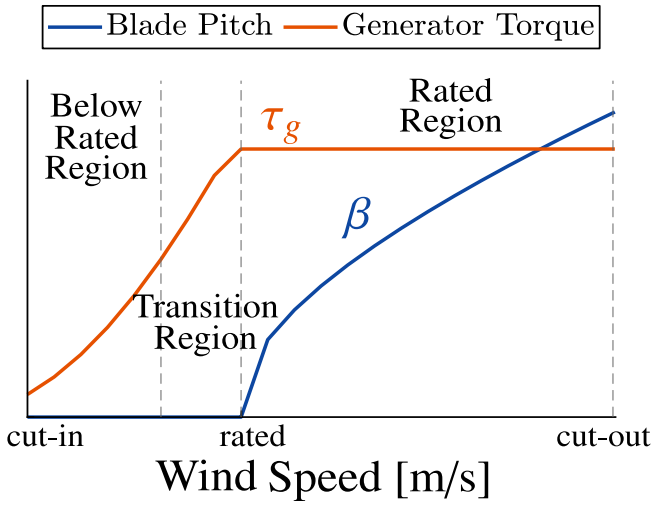


Figure 1.2: Generator torque and blade pitch vs. wind speed.

capture subsystem interactions accurately, while CFD/FEA simulations are too computationally expensive for such large simulation sets.

To balance these trade-offs, researchers and engineers often employ mid-fidelity models. These models capture the physics of the different subsystems with reasonable accuracy while using simplified assumptions and modeling approaches to remain computationally tractable. Such models are often referred to as *aero-servo-hydro-elastic* models and are widely used to study the effects of design and control changes. Aero-servo-hydro-elastic modeling tools comprise multiple modules and solvers that represent the aerodynamics, hydrodynamics, and structural (elastic) responses of wind turbines. A key advantage of these models is that they allow researchers to study the effect of the controller on the system response, which is difficult to assess using simplified models. Using these computational tools, it is possible to explore the performance and trade-offs of a wide range of architectures and designs.

In addition to these modeling tools, open-source reference models play an important role in advancing these technologies. Such models are made available to the research community, enabling researchers to implement and test new designs or control innovations. By applying these innovations to open-source reference models, researchers can demonstrate their effectiveness by comparing them against baseline designs.

1.3.1 Need for a Controller

It is impossible to assess the performance of any wind or marine turbine system without understanding the influence of its controller. This is especially true for floating wind and marine turbines. Due to the stochastic nature of the offshore environment, a controller is necessary to extract maximum power and reduce loads across the entire range of environmental conditions for both wind and marine turbines. There are two general types of control approaches used for these systems: open-loop and closed-loop control. Open-loop control seeks to identify a control input without incorporating system output feedback into the control response. In contrast, closed-loop control uses system outputs to compute the control response.

In practice, closed-loop controllers are typically used for wind turbine operation [33, 44]. For wind turbines, there are two primary control variables: the generator torque (τ_g) and the blade pitch angle (β). The operating range between the cut-in and cut-out wind speeds is typically subdivided into two regions: below-rated and rated. Different control objectives and variables are applied in these regions. In the below-rated region, τ_g is the primary control variable, while in the rated region, β is the primary control variable, as illustrated in Fig. 1.2. For marine turbines, the control schemes are not as standardized as those used for wind turbines. However, both of these control variables are also considered for marine turbines. A more detailed discussion of the different control schemes for marine turbines is provided in Chap. 3.

1.4 Design Optimization

Traditionally, system parameters in these designs were adjusted manually to identify optimal configurations. However, because the associated design space—that is, the space of all possible variable combinations—can be highly nonlinear, manual tuning is not an effective strategy. Moreover, enumerating all possible combinations can result in millions of candidate designs, making exhaustive exploration infeasible. Recently, researchers have begun using design optimization studies to efficiently explore the design space of these systems. Such methods have proven effec-

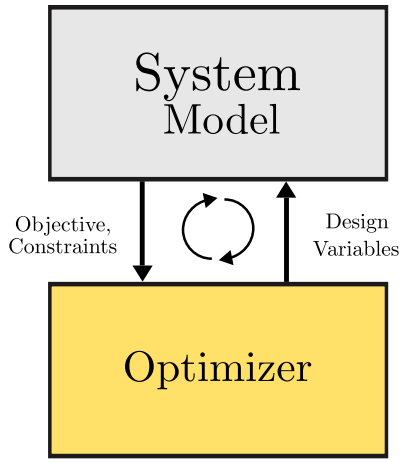


Figure 1.3: Illustration of a design optimization problem.

tive in identifying optimal designs and have been successfully applied to the design of numerous systems.

Design optimization studies aim to identify a set of system parameters, known as design variables, that minimize or maximize specific performance metrics subject to certain constraints. These studies couple a computational model of the system's response with a nonlinear programming solver to determine optimal parameters, as illustrated in Fig. 1.3. A representative formulation of a design optimization problem is:

$$\begin{aligned} \text{Design variables:} \quad & \text{[Blade length, Platform diameter]} && \text{(1.1a)} \\ \text{Minimize:} \quad & \text{Cost of energy} && \text{(1.1b)} \\ \text{Subject to:} \quad & \text{Minimum generated power} && \text{(1.1c)} \\ & \text{Platform motion} && \text{(1.1d)} \end{aligned}$$

In the remainder of this dissertation, this notational structure in Prob. (1.1) will be used to represent optimization problem formulations.

Two distinct sets of design variables are considered for wind and marine turbines. The first, referred to as plant variables, are associated with the physical characteristics of the turbine—such as tower dimensions, platform geometry, mooring line length or thickness, blade length, and airfoil

shape. The second, control variables, relate to the controller, such as control trajectories or closed-loop controller parameters. Different solution strategies are required to address open-loop and closed-loop optimization problems, as discussed in Chap. 2.

Because the dynamics of wind and marine turbines strongly depend on both plant and control variables, the coupling between them must be considered in design optimization studies [32,33,45]. The optimal control settings depend on the optimal plant configuration, and vice versa [11,46]. To address this interdependence, control co-design (CCD) has emerged as a paradigm that simultaneously optimizes the plant and controller to achieve system-level optimal designs. Recently, CCD has been increasingly applied to identify designs that can accelerate the large-scale deployment of renewable energy systems [9, 11, 47].

Several classes of algorithms exist for solving design optimization problems [48]. Gradient-based approaches, which rely on derivatives of the objectives and constraints with respect to design variables, are efficient and scalable when accurate sensitivities are available [49]. However, for complex systems with noisy or discontinuous design spaces, such methods can become inefficient. In these cases, gradient-free approaches—such as genetic algorithms [50, 51], particle swarm optimization [52], or simulated annealing [53]—are more appropriate, though they typically require a larger number of function evaluations. Hybrid approaches like COBYLA [54] offer a compromise by locally linearizing nonlinear objectives and constraints, enabling efficient optimization without explicit gradients. Consequently, COBYLA is commonly employed for CCD and design optimization of wind and marine turbines [55].

Since floating offshore wind turbines (FOWTs) and marine turbines are at different technology readiness levels (TRLs), their design requirements differ. FOWTs currently have TRLs between 8 and 9 [56], while floating marine turbines are estimated to be between TRLs 5 and 6 based on [57, 58]. Furthermore, the availability of open-source modeling tools and reference models varies between these two technologies. The following sections summarize their current state and outline potential research directions.

1.5 Floating Offshore Wind Turbine Design

The standard onshore configuration of a three-bladed, upwind turbine with combined generator torque and blade pitch control served as the foundation for offshore wind turbine development. Early FOWT design efforts primarily focused on the supporting platform and mooring systems [59, 60]. The floating platform and mooring system were identified as the dominant cost drivers, as the platform constitutes a significant portion of the overall system cost [7, 61, 62]. In addition to cost, the type of support structure has a strong influence on the overall system dynamics.

Several computational tools have been developed to model FOWTs, including OpenFAST [63], Bladed [64], and HAWC2 [65]. These are examples of aero-servo-hydro-elastic modeling tools that enable coupled dynamic analysis of the turbine system. In parallel, a series of open-source reference turbine models—such as the NREL 5 MW [66], DTU 10 MW [67], IEA 15 MW [68], and IEA 22 MW [69]—have been developed and made available to the research community. A key insight from these modeling efforts is that the additional degrees of freedom introduced by platform motion significantly affect the turbine's dynamic response. These motions are therefore critical plant and control design drivers [30,59,61]. A well-known example is the negative damping problem, which amplifies tower loads and adversely impacts power generation [33]. Due to the inherent coupling between the controller, dynamics, and plant, several researchers have emphasized the importance of using control co-design (CCD) to achieve system-level optimal designs. CCD studies have been conducted for various platform types [70–74], consistently demonstrating that controller design and performance strongly influence the optimal plant configuration and overall system objectives.

When designing and analyzing complex systems such as wind turbines, multiple performance metrics must be evaluated and compared [75]. In many cases, these metrics exhibit inherent trade-offs—for instance, maximizing power production often leads to increased structural loads [70]. As a result, an optimization study formulated to maximize power alone may not yield an optimal system design. Nonetheless, many recent design optimization studies focus on a single performance metric, optimizing it while constraining other conflicting objectives. To identify true system-level

optima, it is essential to understand the trade-offs among competing performance metrics. Single-objective optimization approaches, while useful for specific design goals, are therefore insufficient for exploring these trade-offs comprehensively.

Understanding such trade-offs is particularly critical in the context of both open-loop and closed-loop control studies, since the controller mediates the competing objectives of maximizing energy capture and minimizing structural loads [76–78]. Analyzing these trade-offs within a CCD framework enables researchers to quantify the influence of plant and control variables on performance outcomes. Beyond the academic motivation, feedback from engineers in original equipment manufacturers (OEMs) highlights the practical need for tools that can explore trade-offs between different performance metrics rather than simply identify a single "optimal" design. Therefore, a key requirement for modern wind turbine design is the development of methods and tools capable of formulating and solving multi-objective optimization problems to better understand and navigate these trade-offs.

1.6 Marine Turbine Design

Identifying cost-optimal designs is essential for enabling the widespread adoption of marine energy technologies [13]. Exploring multiple designs and comparing their performance allows designers and manufacturers to understand underlying trade-offs and identify configurations that best balance competing considerations. However, current modeling and design practices for marine turbines do not facilitate efficient design space exploration [57,79–81]. Typically, lower-order models are developed for specific designs to estimate loads and performance [57,79–81]. In parallel, scaled experimental models are fabricated and tested for these particular designs when feasible [80–84]. These studies provide valuable insights into the performance of individual concepts. However, it remains unclear whether these models and experimental setups can be readily extended to other designs or architectures. The availability of open-source modeling and design tools could greatly accelerate the identification of standard architectures by enabling researchers to efficiently evaluate the feasibility and performance of a wide range of designs [47].

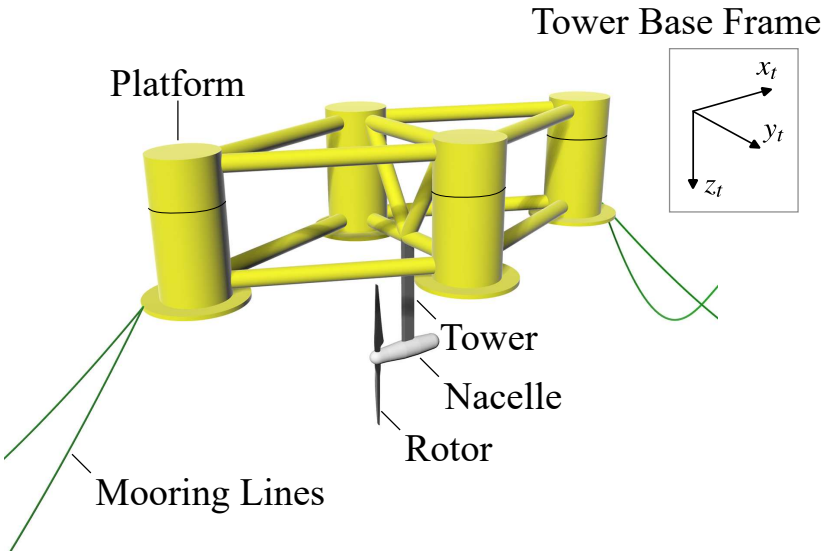


Figure 1.4: RM1 Quad (illustration courtesy of NREL).

1.6.1 OpenFAST's New Features for Marine Turbines

There are only a limited number of open-source tools capable of modeling marine turbines. OpenFAST, an open-source modeling framework for horizontal-axis wind turbines developed by the National Renewable Energy Laboratory (NREL), has been widely adopted to explore wind turbine design spaces and assess performance. Recently, OpenFAST has been extended to simulate axial-flow marine turbines, as described in Ref. [63]. These extensions include the addition of buoyant forces on the blade sections, hub, nacelle, and tower, as well as the ability to model turbines operating below sea level. These new features enable OpenFAST to be used for marine turbine modeling, allowing researchers and developers in both academia and industry to better predict turbine performance and structural loads. Furthermore, they facilitate the advancement of marine energy technology through control co-design optimization studies [47]. Using OpenFAST, the performance and feasibility of different marine turbine configurations can be explored efficiently. At present, OpenFAST supports only horizontal-axis turbines; therefore, the focus of this dissertation will also be restricted to horizontal-axis marine turbines.

In addition to OpenFAST, researchers have introduced a set of open-source reference models for various types of tidal energy converters [13], titled RM1 through RM4, with RM standing

for reference model. Of these, RM1 is a bottom-fixed axial-flow marine turbine. Recently, the RM1 rotor has been updated to include a floating platform, resulting in the floating RM1 Quad (F-RM1-Q) model, introduced in Ref. [85] and shown in Fig. 1.4. The RM1 model described in Refs. [13, 17] features a twin-rotor configuration that is fixed to the seabed. Since OpenFAST currently does not support twin-rotor systems, the F-RM1-Q includes only a single rotor. This model was designed by NREL to test and demonstrate new features added to the OpenFAST code.

The supporting structure for the F-RM1-Q is an asymmetric semisubmersible platform consisting of four main columns connected by a series of smaller-diameter cross members. Heave plates are attached to the bottom of the main columns to dampen platform motion. The rotor-nacelle assembly (RNA) is mounted at the center of the platform via a supporting tower structure. The physical rotor system described in Refs. [13, 17] is used with this floating platform. The mooring system is also asymmetric, providing the large surge stiffness needed to counteract turbine thrust through six catenary lines. Further details on the F-RM1-Q specifications are provided in Ref. [85].

To effectively simulate a marine turbine using OpenFAST, a controller is required. However, there is currently limited availability of open-source controllers that can be easily tuned and applied to marine turbines. Therefore, a key research need for marine turbine development is the creation of a controller that can be used for simulation, analysis, and control co-design studies.

1.7 Addressing Computational Expense

To perform system-level design studies, models that accurately capture the dynamics of wind and marine turbines are required. As system complexity and model fidelity increase, such simulations become increasingly computationally expensive [86, 87]. A direct consequence of capturing detailed dynamic responses is the high computational cost associated with evaluating these systems. For instance, simulating a wind turbine using OpenFAST can take between 7 minutes and 1 hour, whereas simulating a marine turbine may require 3–7 hours. This computational burden can render the direct use of high-fidelity models inefficient for the following applications.

Estimating system performance

To evaluate performance metrics for wind and marine turbines, the dynamic model must be simulated for several hundred design load cases (DLCs), corresponding to different environmental conditions and operating scenarios as specified by IEC standards [43]. Since the simulation time for a single DLC can range from minutes to hours, evaluating a system over all relevant DLCs incurs a significant computational cost in terms of CPU hours.

Use in design optimization studies

Design optimization requires evaluating the dynamic system hundreds or even thousands of times to identify optimal physical designs or control laws. Furthermore, the software architecture of many simulation tools may not allow direct access to all variables of interest. State variables, for example, are often internal quantities dependent on the simulation inputs. Engineers also need to understand how variations in the system's physical parameters affect its dynamic response.

Gradient-based solvers are commonly employed to address these challenges efficiently. However, many simulation tools do not provide the accurate gradient information required by such solvers. Finite-difference approximations can be used to estimate gradients, but these estimates are often noisy, insensitive to small perturbations in design variables, and computationally expensive. As a result, engineers frequently resort to gradient-free solvers, such as genetic algorithms (GA), which require hundreds of function evaluations, or to methods like COBYLA, which do not scale well with increasing numbers of design variables [88]. These computational challenges are further compounded in multi-objective optimization studies, which typically involve solving multiple optimization subproblems.

Computationally inexpensive models that can accurately capture key dynamic quantities, such as system states and performance metrics, are highly valuable for efficient analysis [86, 89, 90]. These approximations are commonly referred to as low-fidelity or surrogate models due to their limited scope in representing the full system response. The importance of low-fidelity models in applications involving computationally expensive high-fidelity simulations has been extensively

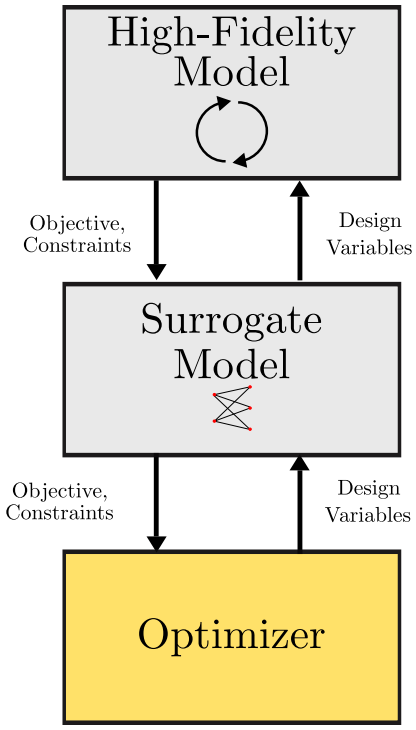


Figure 1.5: Illustration of a surrogate model-based design optimization approach to approximate high-fidelity model response using a surrogate model.

discussed in the design optimization literature. An illustration of the relationship between the high-fidelity model, surrogate model, and optimizer is shown in Fig. 1.5.

In this dissertation, we do not propose novel concepts to reduce the cost of energy associated with wind or marine turbines. Instead, the core contribution is more fundamental: the development of data-driven low-fidelity models derived from high-fidelity simulations that balance computational cost and accuracy. These models can be used to estimate system performance and to solve optimal control problems for a variety of systems.

Several approaches exist in the literature for constructing such low-fidelity models, ranging from classical system-identification techniques to modern deep learning methods. In addition, derivative function surrogate models (DFSMs) have been applied to model wind turbine responses and to support design optimization studies [86]. All of these approaches can be employed to construct low-fidelity models suitable for design optimization and optimal control applications.

Consequently, it is important to compare these methods and understand the trade-offs associated with each.

Derivative Function Surrogate Model (DFSM)

A primary focus of this dissertation is the DFSM approach and the resulting surrogate models. DFSM was first explored in Ref. [86] to approximate the dynamic response of wind turbines for use in open-loop optimal control studies. Fundamentally, the method approximates the system response as a continuous-time state-space model. The surrogate model predicts the state derivatives as a function of inputs such as controls, states, and plant variables. Such a model is essential for open-loop optimal control studies, as certain numerical methods used to solve these problems require this derivative information. The solution approaches are discussed in detail in Chap. 2. In Ref. [86], the authors constructed a surrogate model that predicts state derivatives based on the system states and key plant variables. Once constructed, this model could be employed for control co-design (CCD) studies.

The DFSM approach has since been adopted and refined for other systems, as in Refs. [87, 91, 92], with wind turbines being the primary use case. Extending DFSM to marine turbines is a straightforward adaptation, which is carried out in this dissertation. In principle, DFSM can be applied to multiple systems where a control-oriented dynamic model is required, including vertical-axis wind turbines, battery systems, building thermal systems, wind farms, robotic systems, and wave energy converters [76, 89, 93–97].

However, applying DFSM to new systems requires expanding the construction procedure. Previous studies assume that the high-fidelity model can be evaluated for a given set of input variables to construct the surrogate. This assumption does not hold for many systems. Therefore, in this dissertation, a novel approach is presented to construct DFSMs using input-output time series data. The approach is application-agnostic, meaning it can be extended to multiple systems wherever input-output time series data is available.

1.8 Research Questions

Based on the discussion in the previous sections, the research goals and tasks addressed in this dissertation are as follows:

(RQ1) Develop an easy-to-tune controller for closed-loop control of marine turbines.

(RQ2) Identify and compare different approaches for constructing low-fidelity models of wind and marine turbines and select the most suitable approach.

(RQ3) Demonstrate the efficacy of the low-fidelity modeling approach in predicting the performance of various wind and marine turbine systems.

(RQ4) Demonstrate the efficacy of the low-fidelity model for open-loop and closed-loop optimal control studies of wind and marine turbines.

(RQ5) Demonstrate the efficacy of the low-fidelity model for multi-objective and control co-design (CCD) studies of wind turbines.

The remainder of the dissertation is organized as follows: Chapter 2 discusses the computational tools, optimal control and CCD problem formulations, solution strategies, and the different low-fidelity modeling approaches employed in this dissertation. Chapter 3 focuses on the development of an open-source controller for marine turbines. Chapter 4 details the construction and comparison of the various low-fidelity modeling approaches. Chapter 5 demonstrates how the low-fidelity modeling approach addresses the research tasks outlined in RQ3–RQ5. Finally, Chapter 6 summarizes the key contributions of this dissertation and outlines directions for future research.

Chapter 2

Background

This chapter provides an overview of the different concepts and tools used in this dissertation[1]. The chapter starts by discussing the different types of dynamic design optimization problems that can be formulated and their solution strategies in Sec. 2.1. Then, Secs. 2.2–2.4 discuss the different software tools that are utilized as part of this dissertation. In Sec. 2.5, we discuss the formulations for multi-objective optimization problems and strategies for solving them. Finally, in Sec. 2.6, an overview of the different surrogate/low-fidelity modeling approaches that can be used to formulate and solve these problems is outlined.

2.1 Optimal Control and Control Co-Design Problem Formulations

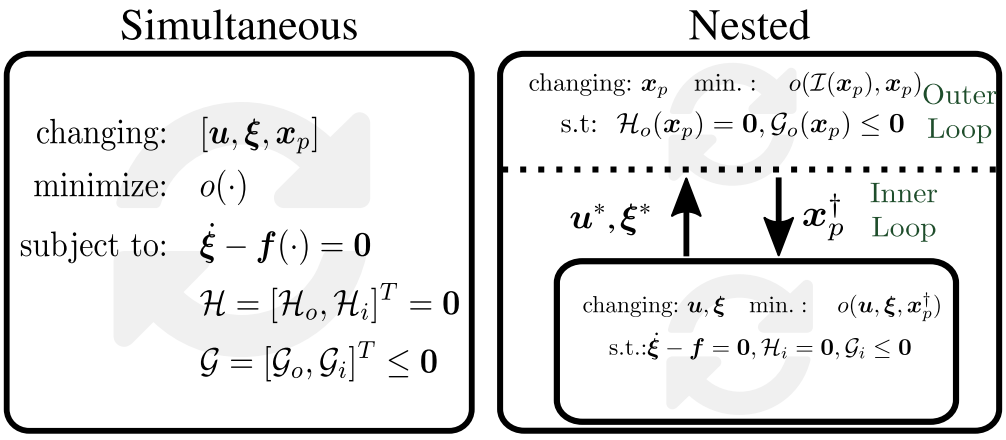


Figure 2.1: Two different CCD coordination strategies.

There are two widely used coordination strategies to solve CCD problems, namely, the simultaneous and nested methods [100]. In the simultaneous method, the CCD problem is formulated

[1]Parts of this chapter have been adapted from Refs. [98, 99].

as a single dynamic optimization problem, as shown in Fig. 2.1, and solved. In the nested strategy, the CCD problem is reformulated as a bi-level optimization problem, where the outer loop solves for the optimal plant design, while the inner loop solves for the optimal controls and states. In addition to CCD studies, we are also interested in formulating and solving optimal control studies. As mentioned previously, we are interested in solving both open-loop and closed-loop optimal control problems. Several CCD problems have been formulated as open-loop optimal control problems [100–102]. The problem formulation for open-loop and closed-loop optimal control and CCD problems is discussed in the following subsections.

2.1.1 Simultaneous CCD Formulation

The simultaneous CCD problem is generally formulated as:

$$\begin{aligned}
&\text{changing:} && [\boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p] && \text{(2.1a)}\\
&\text{minimize:} && o = \mathcal{M}(\boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f), \boldsymbol{x}_p) + \int_{t_0}^{t_f} \mathcal{L}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p) dt && \text{(2.1b)}\\
&\text{subject to:} && \dot{\boldsymbol{\xi}}(t) - \boldsymbol{f}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p) = \boldsymbol{0} && \text{(2.1c)}\\
& && \boldsymbol{\mathcal{P}_h}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p) = \boldsymbol{0} && \text{(2.1d)}\\
& && \boldsymbol{\mathcal{P}_g}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p) \leq \boldsymbol{0} && \text{(2.1e)}\\
& && \boldsymbol{\mathcal{B}_h}(t_0, t_f, \boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f)) = \boldsymbol{0} && \text{(2.1f)}\\
& && \boldsymbol{\mathcal{B}_g}(t_0, t_f, \boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f)) \leq \boldsymbol{0} && \text{(2.1g)}\\
&\text{where:} && \boldsymbol{y} = \boldsymbol{g}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p)
\end{aligned}$$

where $t \in [t_0, t_f]$ is the defined time horizon, $\boldsymbol{\xi}$ are the states, $\boldsymbol{u}$ are the controls, $\boldsymbol{y}$ are the system outputs, and $\boldsymbol{x}_p$ are the plant variables. The term $\mathcal{M}(\boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f), \boldsymbol{x}_p)$ is the Mayer term or terminal cost. Subsequently, the term $\mathcal{L}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p)$ is the Lagrange term or running cost. An example of this term is maximizing the power generated over the time horizon. $\boldsymbol{f}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p)$ is the dynamic function or the state-derivative function that describes how the states evolve over time given the inputs, and $\boldsymbol{g}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p)$ describes how the time-dependent outputs evolve. $\boldsymbol{\mathcal{H}} = \{\boldsymbol{\mathcal{P}_h}, \boldsymbol{\mathcal{B}_h}\}$ are

the sets of equality path and boundary constraints (e.g., initial values of the states or control variables), and $\boldsymbol{\mathcal{G}} = \{\boldsymbol{\mathcal{P}_g}, \boldsymbol{\mathcal{B}_g}\}$ are the sets of inequality path and boundary constraints (e.g., simple state and control bounds, limitations on the tower bending moment).

2.1.2 Nested CCD Formulation

The nested CCD problem is formulated as:

$$\begin{aligned}
\text{changing:} \quad & \boldsymbol{x}_p && \text{(2.2a)}\\
\text{minimize:} \quad & o\left(\mathcal{I}\left(\boldsymbol{x}_p\right), \boldsymbol{x}_p\right) && \text{(2.2b)}\\
\text{subject to:} \quad & \boldsymbol{\mathcal{H}}_o(\boldsymbol{x}_p) = \mathbf{0} && \text{(2.2c)}\\
& \boldsymbol{\mathcal{G}}_o(\boldsymbol{x}_p) \leq \mathbf{0} && \text{(2.2d)}\\
\text{where:} \quad & \mathcal{I}\left(\boldsymbol{x}_p\right) = \underset{\boldsymbol{\xi},\boldsymbol{u}}{\arg\min}\left(\text{Prob. (2.3) with } \boldsymbol{x}_p\right) && \text{(2.2e)}
\end{aligned}$$

where $\mathcal{I}\left(\boldsymbol{x}_p\right)$ is the solution to the following inner-loop problem:

$$\begin{aligned}
\text{changing:} \quad & [\boldsymbol{u}, \boldsymbol{\xi}] && \text{(2.3a)}\\
\text{minimize:} \quad & o\left(\boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p^{\dagger}\right) && \text{(2.3b)}\\
\text{subject to:} \quad & \dot{\boldsymbol{\xi}} - \boldsymbol{f}\left(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p^{\dagger}\right) = \mathbf{0} && \text{(2.3c)}\\
& \boldsymbol{\mathcal{H}}_i\left(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p^{\dagger}, \boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f)\right) = \mathbf{0} && \text{(2.3d)}\\
& \boldsymbol{\mathcal{G}}_i\left(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p^{\dagger}, \boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f)\right) \leq \mathbf{0} && \text{(2.3e)}\\
\text{where:} \quad & \boldsymbol{y} = \boldsymbol{g}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p^{\dagger})
\end{aligned}$$

where $\boldsymbol{x}_p^{\dagger}$ is the candidate plant design from the outer loop, and subscripts 'o' and 'i' refer to outer-loop and inner-loop, respectively.

2.1.3 Closed-Loop Optimal Control based CCD

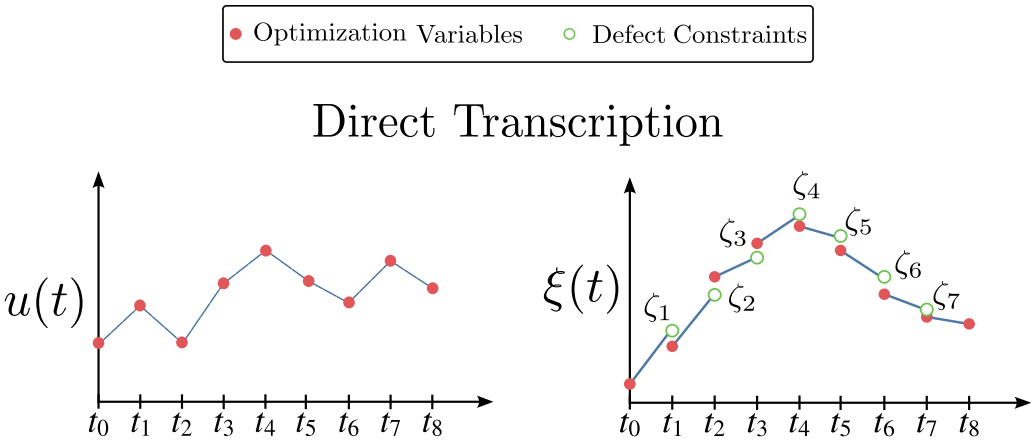
In this dissertation, we consider a closed-loop CCD problem, where the control parameters and the plant variables are the key decision variables. The resulting closed-loop optimal control-based CCD problem can be formulated as follows:

$$
\begin{aligned}
\text{changing:} \quad & [\boldsymbol{x}_c, \boldsymbol{x}_p] && \text{(2.4a)}\\
\text{minimize:} \quad & o(\boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p) && \text{(2.4b)}\\
\text{subject to:} \quad & \boldsymbol{\mathcal{H}}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p, \boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f)) = \boldsymbol{0} && \text{(2.4c)}\\
& \boldsymbol{\mathcal{G}}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{y}, \boldsymbol{x}_p, \boldsymbol{\xi}(t_0), \boldsymbol{\xi}(t_f)) \leq \boldsymbol{0} && \text{(2.4d)}\\
\text{where:} \quad & \boldsymbol{u} = \boldsymbol{C}(\boldsymbol{x}_c, \boldsymbol{\xi})\\
& \boldsymbol{\xi} = \int_{t_0}^{t_f} \dot{\boldsymbol{\xi}}(t)\mathrm{d}t = \int_{t_0}^{t_f} \boldsymbol{f}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p)\mathrm{d}t\\
& \boldsymbol{y} = \boldsymbol{g}(t, \boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{x}_p)
\end{aligned}
$$

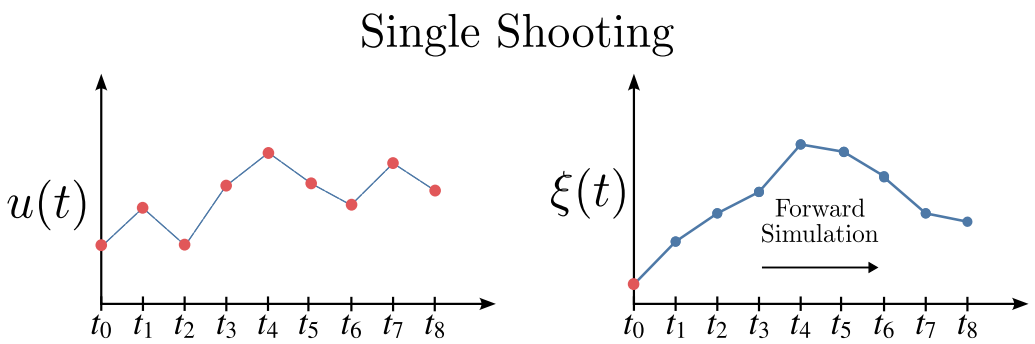
where $\boldsymbol{x}_u$ are the controller parameters, $\boldsymbol{C}(\cdot)$ is the controller and $\mathcal{O}$ is the objective function. In this formulation, both the optimization variables $\boldsymbol{x}_c, \boldsymbol{x}_p$ are static variables, making this a regular optimization problem. Therefore, a specialized solution strategy is not required to solve the given problem, and commonly available optimization solvers can be used. The closed-loop CCD problem shown in Eq. 2.4 can be solved using both simultaneous and nested formulations.

2.1.4 Solution Strategies

Depending on the type of control study, different problem formulations, design variables, and solution strategies must be considered, as shown in the previous section. A prevalent numerical approach for both kinds of optimal control problems (open- and closed-loop) is direct methods such as the single-shooting (or simulation-based) and direct transcription (DT) approaches, shown in Fig. 2.2. Open-loop optimal control problems are also referred to as infinite-dimensional problems, as the solution for $\boldsymbol{u}(t)$ is required for infinitely many points in the time horizon $t \in [t_0, t_f]$. Both the DT and shooting approaches convert this infinite-dimensional problem into a finite-dimensional



(a) A visual representation of the direct transcription approach.



(b) A visual representation of the single shooting approach.

Figure 2.2: Difference between the single-shooting and direct transcription methods.

one. This is done by discretizing certain continuous signals over a finite number of points in a time grid [102].

For open-loop optimal control problems, the differences between the shooting and DT approaches primarily lie in which discretized signals are optimized and how the state dynamics are enforced. The shooting approach directly optimizes the discretized open-loop controls. The DT approach directly optimizes both the discretized open-loop controls and states as independent optimization variables while enforcing the system dynamics through constraints in the optimization problem [103]. The steps involved in transforming the infinite-dimensional problem into a finite-dimensional one using the DT and shooting approaches are explained in the following subsections. Additionally, a visual representation of these approaches is presented in Fig. 2.2.

2.1.5 Direct Transcription

Consider a simple problem with one state and control variable with simple dynamics:

$$\text{changing:} \quad \boldsymbol{x}(t) = [u(t), \xi(t)] \tag{2.5a}$$

$$\text{minimize:} \quad \int_{t_0}^{t_f} u(t)^2 \mathrm{d}t \tag{2.5b}$$

$$\text{subject to:} \quad \dot{\xi}(t) = \xi(t) + u(t) \tag{2.5c}$$

$$u(t_f) = 1 \tag{2.5d}$$

$$u_{\min} \leq u(t) \leq u_{\max} \tag{2.5e}$$

$$\xi_{\min} \leq \xi(t) \leq \xi_{\max} \tag{2.5f}$$

The different steps associated with the DT approach to transform this infinite-dimensional problem into a finite-dimensional one are explained below.

1. Discretizing the time horizon and optimization variables.

The first step in the process is discretizing the time horizon. Suppose the time horizon for this problem is between $t \in [t_0, t_f]$. This time horizon is discretized using a time step $\Delta = t_1 - t_0$ into $n_t = 5$ points as:

$$\boldsymbol{t} = [t_0, t_1, t_2, t_3, t_f]^T$$

Next, the state and control trajectories ($[u(t), \xi(t)]$) are also discretized on this mesh. Discretizing each variable for $n_t = 5$ points gives a total of ten points at which $u(t)$ and $\xi(t)$ must be identified:

$$\boldsymbol{U} = [u_0, u_1, u_2, u_3, u_f]$$

$$\boldsymbol{\Xi} = [\xi_0, \xi_1, \xi_2, \xi_3, \xi_f]$$

$$\boldsymbol{X} = [\boldsymbol{U}, \boldsymbol{\Xi}]^T$$

2. Constructing the defect constraints.

The next step is to reformulate the dynamic equation shown in Eq. (2.5c) using these discretized variables. To do this, different single-step or multi-step approaches that are used to numerically solve ODEs can be applied. In this example, we utilize the Euler forward method to discretize the defect constraints. Recall that the Euler forward rule can be used to approximate an ODE as:

$$\xi_{k+1} = \xi_k + \Delta \times f(\xi_k, u_k)$$
$$\zeta_{k+1} = \xi_{k+1} - \xi_k - \Delta \times f(\xi_k, u_k)$$
$$\zeta_{k+1} = 0$$

Using this rule and the discretized control and state variables, the dynamic equation can be specified as constraints:

$$\zeta_1 = \xi_1 - \xi_0 - \Delta(\xi_0 + u_0) = 0$$
$$\zeta_2 = \xi_2 - \xi_1 - \Delta(\xi_1 + u_1) = 0$$
$$\zeta_3 = \xi_3 - \xi_2 - \Delta(\xi_2 + u_2) = 0$$
$$\zeta_f = \xi_f - \xi_3 - \Delta(\xi_3 + u_3) = 0$$

This set of equations can be combined, reformulated, and represented as:

$$\begin{bmatrix} -\Delta u_0 & 0 & 0 & 0 & 0 & -(1+\Delta)\xi_0 & \xi_1 & 0 & 0 & 0 \\ 0 & -\Delta u_1 & 0 & 0 & 0 & 0 & -(1+\Delta)\xi_1 & \xi_2 & 0 & 0 \\ 0 & 0 & -\Delta u_2 & 0 & 0 & 0 & 0 & -(1+\Delta)\xi_2 & \xi_3 & 0 \\ 0 & 0 & 0 & -\Delta u_3 & 0 & 0 & 0 & 0 & -(1+\Delta)\xi_3 & \xi_f \end{bmatrix}$$

$$\boldsymbol{\zeta} \times [\boldsymbol{U}, \boldsymbol{\Xi}]^T = \mathbf{0}$$

3. Objective function.

Following the dynamics, we discretize the objective function shown in Eq. (2.5b) using the Euler forward-quadrature also. Using this quadrature rule, the objective function value can be calculated as:

$$\int_{t_0}^{t_f} u^2 dt = \sum_{i=0}^{n_t-1} \boldsymbol{U}^2[i]\Delta$$

where i refers to the i-th index. For the given time mesh and the discretized variables, this equation can also be represented as:

$$O = \boldsymbol{U}^T \boldsymbol{H} \boldsymbol{U}$$

$$\boldsymbol{H} = \begin{bmatrix} \Delta & 0 & 0 & 0 & 0 \\ 0 & \Delta & 0 & 0 & 0 \\ 0 & 0 & \Delta & 0 & 0 \\ 0 & 0 & 0 & \Delta & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4. Additional constraints.

The last step in the DT approach is to discretize the constraints. The constraints on u, ξ can be directly added as follows:

$$\boldsymbol{U}_{\min} = \begin{bmatrix} u_{\min}, u_{\min}, u_{\min}, u_{\min}, 1 \end{bmatrix} \qquad \boldsymbol{U}_{\max} = \begin{bmatrix} u_{\max}, u_{\max}, u_{\max}, u_{\max}, 1 \end{bmatrix}$$

$$\boldsymbol{\Xi}_{\min} = \begin{bmatrix} \xi_{\min}, \xi_{\min}, \xi_{\min}, \xi_{\min}, \xi_{\min} \end{bmatrix} \qquad \boldsymbol{\Xi}_{\max} = \begin{bmatrix} \xi_{\max}, \xi_{\max}, \xi_{\max}, \xi_{\max}, \xi_{\max} \end{bmatrix}$$

The constraint in Eq. (2.5d) can be addressed directly when the trajectories are discretized.

5. Transcribed problem.

Utilizing the DT approach, the problem presented in Prob. 2.5 can be transcribed as:

$$\begin{aligned}
\text{changing:} \quad & \boldsymbol{X} = [\boldsymbol{U}, \boldsymbol{\Xi}] && \text{(2.6a)}\\
\text{minimize:} \quad & \boldsymbol{U}^T \boldsymbol{H} \boldsymbol{U} && \text{(2.6b)}\\
\text{subject to:} \quad & \boldsymbol{\zeta} \boldsymbol{X} = \boldsymbol{0} && \text{(2.6c)}\\
& \boldsymbol{U}_{\min} \leq \boldsymbol{U} \leq \boldsymbol{U}_{\max} && \text{(2.6d)}\\
& \boldsymbol{\Xi}_{\min} \leq \boldsymbol{\Xi} \leq \boldsymbol{\Xi}_{\max} && \text{(2.6e)}
\end{aligned}$$

Notice how the linear-quadratic structure of the problem in Prob. (2.5) can be preserved when applying the direct transcription method. For this reason, DT methods are also referred to as order-maintaining methods, where the infinite-dimensional problem and the transcribed finite-dimensional problem both have the same order. The problem posed in Prob. (2.6) is a quadratic optimization problem. This problem can be solved using commercially available nonlinear programming solvers. A key reason why these problems can be solved efficiently is the availability of high-quality derivative information for both the objectives and the constraints. Because the objective is quadratic and the constraints are linear with respect to the design variables considered, the Hessian of the objective function is $\boldsymbol{H}$, and the gradient of the constraints is represented by the $\boldsymbol{\zeta}$ system matrices shown in Eqs. (2.6b, 2.6c), respectively.

In this example, a total of $n_t = 5$ points were used to demonstrate the steps involved. However, when formulating and solving these problems using the DT approach, a higher value of n_t is typically employed. Increasing n_t implies discretizing the problem using smaller time steps, which helps minimize errors in the system dynamics. In this example, the Euler forward method was used to construct the defect constraints. However, higher-order single-step methods such as the trapezoidal rule, Runge–Kutta methods, or Hermite–Simpson schemes can be employed to achieve greater accuracy. More details regarding these approaches can be found in Refs. [104–106]. In

addition to single-step approaches, pseudospectral methods have also been used to discretize and solve optimal control problems.

Additionally, a simple linear problem was used here to demonstrate the steps involved. However, when formulating and solving these problems for complex systems, the dynamics are often nonlinear. The transcribed problem for such cases will also be nonlinear. These problems are more challenging to solve and require advanced solvers. Beyond the need for advanced solvers, the quality of derivative information available for the objective and constraints also affects the convergence of these problems, as shown in Ref. [101].

2.1.6 Single Shooting Approach

For the optimal control problem shown in Prob. (2.5) we show the steps involved with single-shooting approach. We explain the approach when the control trajectory and the state value at t_0 are considered as the only optimization variables:

$$\text{changing:} \quad \boldsymbol{x} = [u(t), \xi(t_0)] \tag{2.7a}$$

$$\text{minimize:} \quad \int_{t_0}^{t_f} u(t)^2 \mathrm{d}t \tag{2.7b}$$

$$\text{subject to:} \quad \dot{\xi}(t) = \xi(t) + u(t) \tag{2.7c}$$

$$u(t_f) = 1 \tag{2.7d}$$

$$u_{\min} \leq u(t) \leq u_{\max} \tag{2.7e}$$

$$\xi_{\min} \leq \xi(t) \leq \xi_{\max} \tag{2.7f}$$

So only $u(t)$ is discretized, unlike the DT approach where both $u(t)$ and $\xi(t)$ are discretized.

1. Discretizing the time horizon and optimization variables.

The first step in the process is discretizing the time horizon. Using the same step size $\Delta = t_1 - t_0$ for $n_t = 5$ points ,the time horizon and the control trajectory are discretized as:

$$\boldsymbol{t} = [t_0, t_1, t_2, t_3, t_f]^T$$
$$\boldsymbol{U} = [u_0, u_1, u_2, u_3, u_f]^T$$

2. Evaluating the objective function and constraints.

The simulated system dynamics and controls are then used to evaluate the objective function and constraints.

$$\int_{t_0}^{t_f} u^2 dt = \sum_{i=0}^{n_t-1} \boldsymbol{U}^2[i]\Delta$$
$$\boldsymbol{U}_{\min} = \left[u_{\min}, u_{\min}, u_{\min}, u_{\min}, 1\right]$$
$$\boldsymbol{U}_{\max} = \left[u_{\max}, u_{\max}, u_{\max}, u_{\max}, 1\right]$$

3. Evaluating the dynamics.

Unlike the DT approach, where the dynamics can be enforced as defect constraints, an ODE solver is required to evaluate the dynamics for the shooting approach. The ODE solver is started from the point $\xi(t_0)$ identified by the optimizer The dynamics Eq. (2.5c) are evaluated for a discretized control trajectory. The control values at intermediate points are often obtained through interpolation.

$$\xi(t) = \xi(t_0) + \int_{t_0}^{t_f} (\xi(t) + u(t))\mathrm{d}t$$

Using the evaluated dynamics the constraint in Eq. (2.7f) is evaluated.

5. Transcribed problem.

The transcribed problem can be represented as:

$$\text{changing:} \quad \boldsymbol{X} = [\boldsymbol{U}, \xi(t_0)] \tag{2.8a}$$

$$\text{minimize:} \quad \sum_{i=0}^{n_t-1} \boldsymbol{U}^2[i]\Delta \tag{2.8b}$$

$$\text{subject to:} \quad \boldsymbol{U}_{\min} \leq \boldsymbol{U} \leq \boldsymbol{U}_{\max} \tag{2.8c}$$

$$\xi_{\min} \leq \xi(t) \leq \xi_{\max} \tag{2.8d}$$

$$\text{where:} \quad \xi(t) = \xi(t_0) + \int_{t_0}^{t_f} (\xi(t) + u(t))\mathrm{d}t \tag{2.8e}$$

It is clear from the solution approaches that shooting-based methods are easier to implement (i.e., they can utilize basic input/output dynamic modeling paradigms) with repeated simulations of the dynamics for different values explored by the optimizer. However, for open-loop optimal control problems, shooting methods can have convergence issues and struggle to effectively handle additional path constraints (e.g., Eq. (2.7f)) when they are added to the problem [103]. On the other hand, DT methods can handle path constraints efficiently [100, 101]. For the same problem and discretization scheme, however, the problem size for DT approaches is generally larger. Because of the way the problem is set up, to ensure that the dynamics specified as constraints are satisfied, the optimizer needs direct access to the dynamic model or the derivative function ($\dot{\xi}(t) = f(\cdot)$). When using simulation tools like OpenFAST, this information is not directly available and is often inaccessible. Alternative models must therefore be constructed that can encode the derivative information.

Because of the ease of implementation of shooting-based approaches, they are widely utilized for closed-loop controller optimization problems, where a computational modeling tool must be simulated to obtain the relevant performance metrics. It is straightforward to wrap the simulation tool with an optimizer to carry out these optimization studies. The simulation tool does not need to be modified, nor does its software need to be altered, to perform these closed-loop optimal control

studies. For this reason, the shooting method is widely utilized to solve closed-loop optimal control applications, including cases where OpenFAST is used as the simulation model [33, 88].

We utilize the WEIS toolbox, available in Ref. [107], to carry out these design optimization studies. WEIS is developed using the OpenMDAO framework and can therefore be paired with an optimizer to perform these studies. It also contains several modules that help with simulating the system, setting up the controller, and post-processing results. In addition to these modules, WEIS also includes DTQPy, a limited Python implementation of the DTQP toolbox available in Ref. [108], which can be used to formulate and solve linear-quadratic dynamic optimization problems using the DT approach.

2.2 WEIS Toolbox

The motivation for the methods and applications developed and discussed in this dissertation is to contribute to the functionality of a state-of-the-art wind and marine turbine design tool. In particular, the methods discussed in this dissertation are meant to contribute to the WEIS (Wind Energy with Integrated Servo-control) toolbox. WEIS is an open-source tool developed primarily by the National Renewable Energy Laboratory (NREL) to enable CCD of wind and marine turbines [9, 47]. It is built on OpenFAST [63], an aero-servo-hydro-dynamic solver that can be used to evaluate the dynamic response of wind and marine turbines. WEIS can be used to perform a wide variety of studies, ranging from analysis and simulations to optimal control and full-fledged CCD studies.

To facilitate this wide range of studies, WEIS utilizes different models of varying fidelity levels. Figure 2.3 shows a brief overview of the WEIS toolbox. At the lowest fidelity, WEIS uses a frequency-domain model called RAFT to approximate the dynamic response [109]. The rest of the dynamic models used in WEIS are time-domain models. Among these, there are two mid-fidelity time-domain models, while the highest-fidelity model is OpenFAST. Both mid-fidelity models are derived from OpenFAST. The first mid-fidelity model is obtained by linearizing the OpenFAST model. This model has been used for several design optimization studies [98, 110, 111]. The

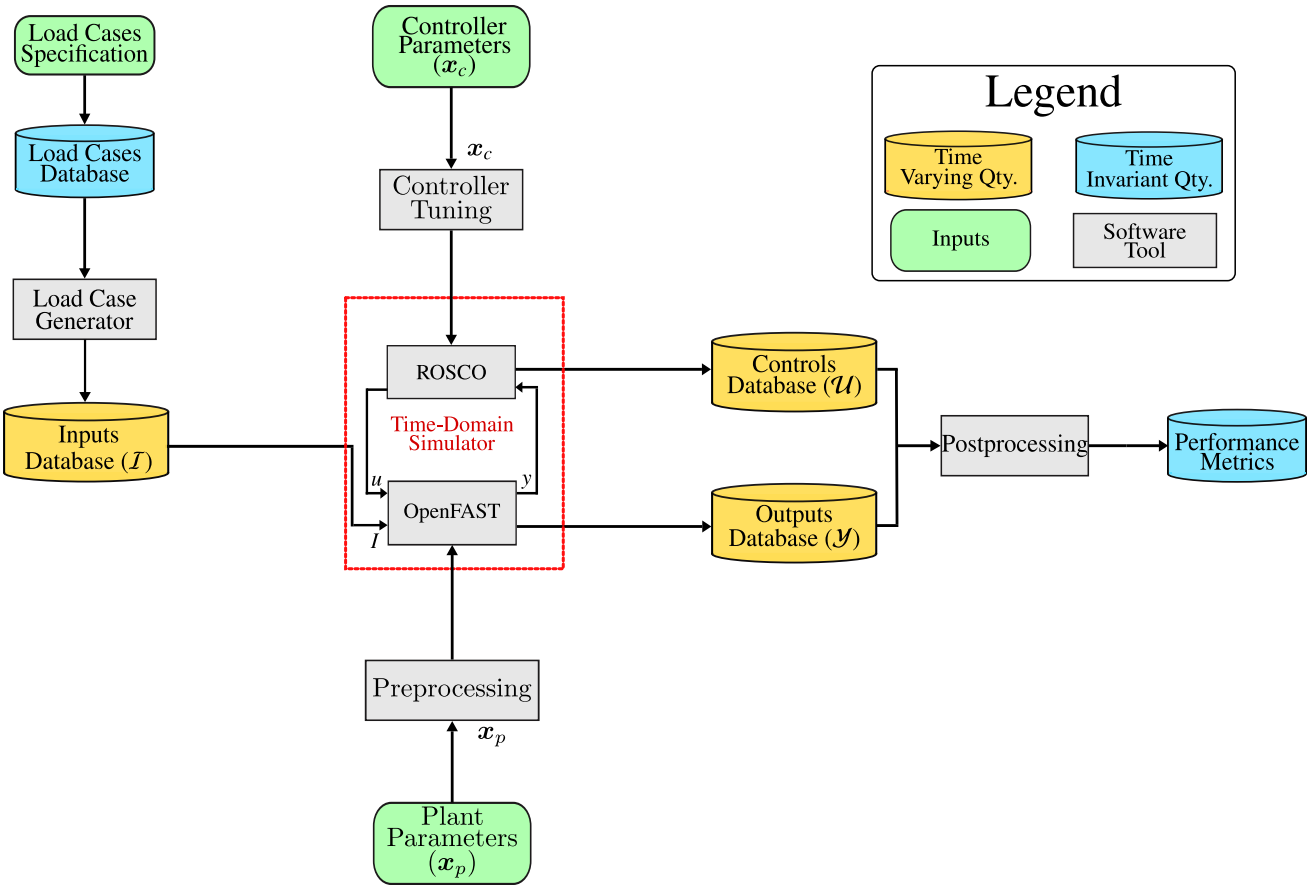


Figure 2.3: A high-level workflow diagram illustrating the different components associated with simulating a wind or marine turbine system.

second type of mid-fidelity model is data-driven. The goal of this dissertation is to identify and explore different approaches that can be used to construct these models.

The core functionality of the WEIS toolbox is built on OpenFAST. To effectively use OpenFAST to perform simulations, a controller is required. In this dissertation, the open-source ROSCO controller is used [33]. A high-level workflow diagram illustrating the different steps and software components associated with simulating a wind or marine turbine model using OpenFAST is shown in Fig. 2.3. At the heart of the WEIS toolbox are OpenFAST [110] and the open-source controller ROSCO introduced in Ref. [33]. To understand Fig. 2.3 and the motivation behind this work, it is necessary to understand how these core components are set up and operate.

2.3 OpenFAST

OpenFAST is an open-source modeling tool that is extensively used to simulate the dynamic response of fixed or floating wind turbine systems to various wind and wave excitations. Recently, OpenFAST has been extended to support the simulation of axial-flow marine turbines and is available at Ref. [63]. The addition of buoyant forces on the blade sections, hub, nacelle, and tower, along with the ability to model turbines below sea level, enables the use of OpenFAST for marine turbine simulations. These new features can help marine energy developers, including both industry and academia, better predict marine turbine performance and loads, as well as advance this technology through control co-design optimization [47].

Given a description of the different components, such as the rotor, generator, tower, platform, and mooring lines, along with their properties, OpenFAST can predict the aerodynamic and hydrodynamic loads, generator performance, and the displacements, velocities, and reaction loads of the turbine components for the specified wind and wave excitation signals. OpenFAST can be broken down into a set of interconnected modules that contain the relevant physics needed to model the aerodynamic, hydrodynamic, generator, and structural reactions [110]. OpenFAST formulates the dynamics of the turbine as a set of differential-algebraic equations (DAEs) and solves these DAEs to predict the system dynamics. Because OpenFAST must resolve the underlying physics of the different modules individually and account for the coupling between modules, simulations can be computationally expensive.

2.4 ROSCO

2.4.1 ROSCO Overview

ROSCO is a reference open-source controller developed by NREL for the control of fixed and floating wind turbines [33] and is available at Ref. [112]. ROSCO uses a simplified, first-order model along with the C_p surface of the given turbine to estimate how ω_g varies with the two

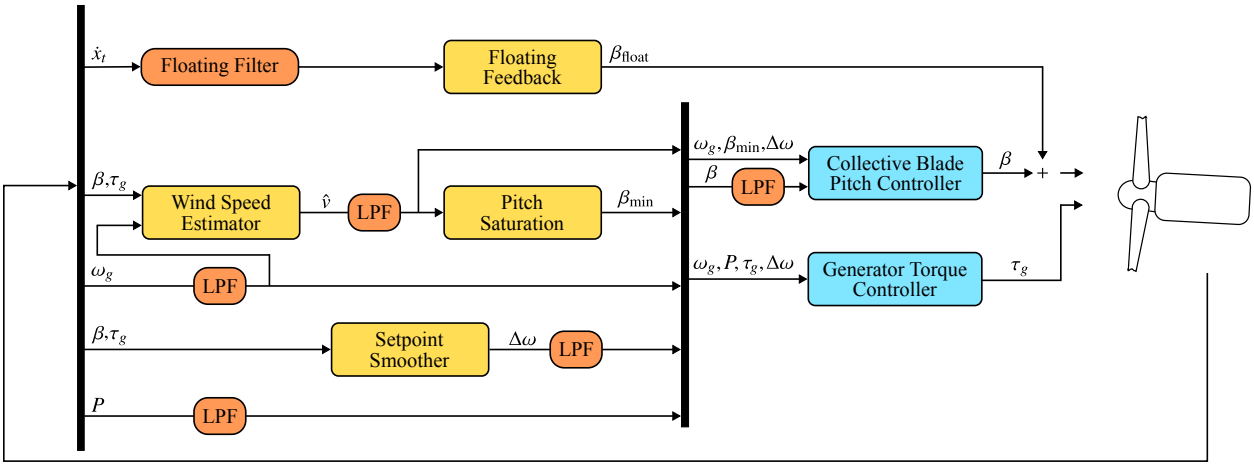


Figure 2.4: A block diagram summarizing the key components of the ROSCO toolbox. The main feedback variables are the generated power (P), the generator speed (ω_g), and the tower-top velocity ($\dot{x}_t$). The generator torque controller is summarized in Sec. 2.4.2, and the blade pitch controller and the floating feedback are summarized in Sec. 2.4.3.

primary control variables τ_g and β:

$$\dot{\omega}_g = \frac{N_g}{J}\left(\tau_a - N_g \tau_g \eta_{\text{gb}}\right) \tag{2.9a}$$

$$\tau_a = \frac{1}{2}\rho A_r \frac{C_p(\lambda,\beta)}{\omega_r} v^3 \tag{2.9b}$$

where N_g is the gearbox ratio, $A_r = \pi R^2$ is the swept area and R is the radius of the blades, v is the current (or wind) speed, J is the rotor inertia, ρ is the density of the fluid, η_{gb} is the combined generator and gearbox efficiency, C_p is the power coefficient expressed as a function of β and the tip speed ratio (λ), and ω_r and τ_a are the rotor speed and aerodynamic torque, respectively. Let $\boldsymbol{X}$ denote the variables associated with Eqs. (2.9a) and (2.9b):

$$\boldsymbol{X} = [N_g, J, \tau_a, \eta_{\text{gb}}, \rho, A_r, C_p, \omega_r, \lambda] \tag{2.10}$$

Two different control loops, both using a reference tracking proportional-integral (PI) architecture, are developed for τ_g (for the below-rated region) and β (for the above-rated region). The generator speed (ω_g) is the main feedback variable for both control loops. A set-point smoother is used to provide a smooth transition between these two controllers in the near-rated region (~ 2.0

m/s). The closed-loop system for both controllers and the corresponding proportional ($\boldsymbol{k}_p$) and integral ($\boldsymbol{k}_i$) gain schedules are derived using first-order linearizations of Eqs. (2.9a) and (2.9b). A block diagram of the ROSCO controller is shown in Fig. 2.4.

The input to this closed-loop system is the generator speed error, measured as:

$$-\Delta\omega_g = \omega_{g,\text{ref}} - \omega_g \tag{2.11}$$

where $\omega_{g,\text{ref}}$ is the reference generator speed, and the output is the control response $\Delta\tau_g$ or $\Delta\beta$. The resulting closed-loop system between $\Delta\tau_g$ or $\Delta\beta$ and $\Delta\omega_g$ is a second-order system with a response characterized by its natural frequency (ω_{des}) and damping ratio (ζ_{des}) [113]. The gains $\boldsymbol{k}_p$ and $\boldsymbol{k}_i$ for both the control systems subsequently depend on $\boldsymbol{x}_c = [\omega_{\text{des}}, \zeta_{\text{des}}]$ and the variables in $\boldsymbol{X}$. Briefly, ω_{des} affects how quickly the controller can respond to a change in the reference, and ζ_{des} affects the oscillatory response of the system. The desired controller performance can be achieved by selecting the appropriate values of $\boldsymbol{x}_c$. Please refer to Ref. [33] for a thorough discussion of the control loops and the tuning process. Sections 2.4.2 and 2.4.3 provide a brief overview of the $\omega_{g,\text{ref}}$ used for the below- and above-rated controllers and their associated gains.

2.4.2 Below-Rated Generator Torque Control

The generator torque is the main control variable used in the below-rated region. A commonly used control goal in the below-rated region is to track the $\omega_{g,\text{ref}}$ value that keeps the turbine operating at its optimal tip-speed ratio, λ_{opt}:

$$\omega_{g,\text{ref}} = N_g \frac{\lambda_{\text{opt}} \hat{v}}{R} \tag{2.12}$$

where $\hat{v}$ is the estimated rotor-effective current speed. Current speeds cannot be accurately measured at the turbine, so an estimator is required. Modeling uncertainties can lead to biases in the current speed estimate. To remove the speed estimator from the control loop, we use a new $\omega_{g,\text{ref}}$ value that can be obtained as follows from the relations for power generated (P) and τ_g, discussed

in Refs. [33, 44]:

$$\tau_g = K\omega_g^2 \quad \text{and} \quad \tau_g = \frac{P}{\omega_g} \tag{2.13}$$

Equating these two, we get:

$$\omega_{g,\text{ref}} = \omega_g = \left[\frac{P}{K}\right]^{\frac{1}{3}} \tag{2.14}$$

where K is given by:

$$K = \frac{1}{2}\rho\pi R^5 \frac{C_{p,\max}}{\lambda_{\text{opt}}} \frac{1}{N_g^3 \eta_{\text{gb}}} \tag{2.15}$$

This reference does not depend on $\hat{v}$ and doesn't adversely affect the performance obtained using Eq. (2.12). As shown in Fig. 2.4, the generator power signal is passed through a low-pass filter before being used to calculate $\omega_{g,\text{ref}}$. The values of ω_{vs} and ζ_{vs}[2] are then selected as $\boldsymbol{x}_{c,\text{vs}} = [0.7, 0.7]$ for marine turbines, and $[0.12, 0.85]$ for wind turbines to derive $\boldsymbol{k}_{p,\text{vs}}$ and $\boldsymbol{k}_{i,\text{vs}}$. As opposed to a gain schedule for $\boldsymbol{k}_{p,\text{vs}}$ and $\boldsymbol{k}_{i,\text{vs}}$ for different values v, the values corresponding to the rated flow speed $v = v_{\text{rated}}$ are used to simplify controller implementation without affecting the performance.

2.4.3 Above-Rated Blade Pitch Control

The control goal in the above-rated region is to vary β to track the rated generator speed $\omega_{g,\text{ref}} = \omega_{g,\text{rated}}$ for the turbine. Similar to the below-rated controller, ω_{pc} and ζ_{pc} are selected to derive the corresponding gains $\boldsymbol{k}_{p,\text{pc}}$ and $\boldsymbol{k}_{i,\text{pc}}$. The performance of the turbine in the above-rated region is critical because the majority of the power is generated in this region, and, correspondingly, the loads on the turbine are higher. Subsequently, the performance of the blade pitch controller must be optimized to meet the design goals. To improve the performance of the blade pitch controller, multiple values of $\boldsymbol{\omega}_{\text{pc}}$ and $\boldsymbol{\zeta}_{\text{pc}}$ for different values of $\boldsymbol{v}$ can be selected [88]. In addition to this

[2]Where the subscript "vs" refers to variable speed torque control, and subsequently, "pc" refers to pitch control.

control loop, for floating turbines, a feedback term is included to address the "negative-damping" problem.

Negative-damping problem.

The negative-damping problem is a known issue for floating turbines that use blade pitch control in the above-rated region [59, 114, 115]. Briefly, this phenomenon occurs as a result of the coupling between the tower motion and the blade pitch. Consider the case when the turbine is pitching forward. This forward motion increases the relative current speed faced by the turbine, thereby increasing the generator/rotor speed. The blade pitch controller then seeks to keep the generator speed at its rated value. When the response of the blade pitch controller (ω_{pc}) is faster than the tower motion (ω_{tower}), the relative thrust experienced by the turbine is reduced, causing the turbine to pitch forward even more, and vice versa when the turbine is pitching backward. This causes the tower fore-aft and platform pitching motion to be undamped. This subsequently increases the tower-base loads and affects ω_g regulation. To counteract this, the tower-top velocity $(\dot{x}_t)$ is filtered and proportionally fed back to the blade pitch controller to dampen the pitching motion. A combination of first-order high-pass, second-order low-pass, and notch filters is applied to the tower-top velocity signal. The corner frequencies of the high-pass and low-pass filters are at 0.01 rad/s and the fore-aft natural frequency of the platform ω_{ptfm}, respectively. The corresponding proportional gain is represented by $k_{\beta,\text{float}}$. This gain can either be derived using a generic tuning procedure in ROSCO or can be specified by the user.

In addition to these control loops, ROSCO has a number of modules and filters that provide additional functionalities. The reader can refer to Refs. [33, 44] for a detailed overview of the first-order model, tuning procedure, and the additional modules used in ROSCO. The controller's performance can be improved by selecting appropriate values of ω and ζ. Typically, for a given turbine model and a set of DLCs, an optimization problem is set up to find values of $\boldsymbol{x}_c = [\omega, \zeta]^{\text{T}}$ that maximize or minimize specific performance metrics, which are calculated by simulating the turbine model using OpenFAST. This process is shown in Fig. 2.3. Solving this optimization problem using OpenFAST simulations can be challenging for the reasons outlined in Chap. 1.

Therefore, key focus of this work will be on constructing a low-fidelity model that can predict the performance metrics for expensive simulations and accelerate the design optimization process. With the core elements of Fig. 2.3 explained, Sec. 2.4.4 will briefly outline the rest of the elements of Fig. 2.3.

2.4.4 Process Workflow

The process starts with the specification of the load cases, which characterize the wind speed and wave elevation signals that act as key inputs ($\mathcal{I}$) to the simulation. These specifications usually correspond to the metocean (meteorological–oceanographic) conditions at the site where the turbine is located, such as the average wind speed ($\bar{w}$), the turbulence intensity of the wind speed signal (i.e., the ratio of the standard deviation to the mean), and the height and period of the incoming waves. These specifications are passed to the load case generator, which produces the time series of the wind/current speed and wave elevation signals. Next, the controller parameters specified by the user are used to tune the controller. In parallel, the plant specifications, such as rotor dimensions, platform size, tower height and thickness, etc., are used to calculate system properties including platform mass, inertia, and tower frequency. These quantities are required to simulate the system using OpenFAST.

The next step is to carry out the simulations for these load cases using the controller and the plant model. OpenFAST passes the instantaneous values of key feedback variables, such as ω_g, to the controller, and the controller calculates the corresponding instantaneous values of τ_g and β back to OpenFAST. The results of these simulations, which are the time series of key inputs, controls, and outputs, are stored and passed on to the postprocessing tool, which calculates the key performance metrics. The WEIS toolbox is a collection of scripts and tools that can be used to perform this entire process. The workflow shown in Fig. 2.3 can be used to carry out a controller optimization problem for OpenFAST and ROSCO. This updated workflow is illustrated in Fig. 2.5.

2.5 Multi-Objective Optimization Problems

There are several ways to explore the trade-offs between conflicting performance objectives. Multi-objective optimization studies offer a rigorous approach for this purpose [48]. These studies aim to minimize multiple, often conflicting, objectives simultaneously. While the outcome of a single-objective optimization study is a single optimal vector of variables that minimizes a given performance metric subject to constraints, the outcome of a multi-objective study is a set of solutions known as the *Pareto front*. The extremities of the Pareto front correspond to the minima or maxima of the individual performance objectives, and the points along the front represent *non-dominated solutions*. A non-dominated solution is one that is not outperformed by any other solution in all objectives.

There are broadly three approaches to solving a multi-objective optimization problem: the weighted-objective method, the epsilon-constraint method, and specialized solvers that handle multiple objectives directly [48, 116]. The first and most straightforward approach is the weighted-objective method. In this method, the multi-objective problem is reformulated as a single-objective optimization problem by assigning different weights to each objective, such that the weights sum to one. Each subproblem is then solved to obtain a set of solutions, which is post-processed to identify the non-dominated solutions. The second approach, called the epsilon-constraint method, also reformulates the problem as a single-objective problem. Typically, one objective is selected as the primary objective, while the others are treated as constraints with a series of allowable values; an optimization problem is then solved for each set of constraint values. The third approach involves using specialized solvers, such as NSGA-II, to solve the multi-objective problem directly without reformulating it as a single-objective problem [51]. Each approach has its own set of trade-offs. Consider the following multi-objective optimization problem:

$$\begin{aligned} \text{changing:} \quad & \boldsymbol{x} \\ \text{minimize:} \quad & \boldsymbol{o} = [o_1, o_2] \\ \text{subject to:} \quad & \boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{\max} \end{aligned}$$

Using the weighted-objective method, this multi-objective optimization problem is reformulated and solved as:

$$
\begin{aligned}
\text{for} \quad & \boldsymbol{w} = [w_1, w_2, \cdots, w_n] \\
& \text{changing:} \quad \boldsymbol{x} \\
& \text{minimize:} \quad o = w_i o_1 + (1 - w_i) o_2 \\
& \text{subject to:} \quad 0 \leq w_i \leq 1 \\
& \qquad\qquad\quad \boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{\max} \\
& \text{where} \quad i = [1, 2, \cdots, n]
\end{aligned}
$$

Using the epsilon-constraint method, this multi-objective optimization problem is reformulated and solved as:

$$
\begin{aligned}
\text{for} \quad & \boldsymbol{o}_2 = [o_{2,1}, o_{2,2}, \cdots, o_{2,n}] \\
& \text{changing:} \quad \boldsymbol{x} \\
& \text{minimize:} \quad o_1 \\
& \text{subject to:} \quad o_2 = o_{2,i} \\
& \qquad\qquad\quad \boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{\max} \\
& \text{where} \quad i = [1, 2, \cdots, n]
\end{aligned}
$$

In this dissertation, we consider the weighted-objective method to solve the multi-objective problem. The epsilon-constraint method and GA-based approaches are not chosen for the following reasons:

- The main drawback of specialized solvers, such as NSGA-II, lies in software constraints. For each value of $\boldsymbol{x}$, multiple load cases must be simulated to calculate the relevant performance metrics. Since GA methods are population-based and require evaluating several hundred $\boldsymbol{x}$ values per iteration, they must be properly parallelized to be computationally efficient.

However, due to the way ROSCO and OpenFAST are structured, only one set of $\boldsymbol{x}$ values can be evaluated at a time. Parallelizing these runs would require considerable modifications to the implementation of WEIS, OpenFAST, and ROSCO, which is beyond the scope of this study. Therefore, specialized solver-based approaches are not considered.

- The epsilon-constraint method does not suffer from the parallelization issue. However, considerable preprocessing is required to formulate and solve the problem effectively. This preprocessing involves identifying the optimal values of o_1 and o_2, and generating a series of points for $\boldsymbol{o}_2$ at which the multi-objective problem can be solved. Consequently, automating the design optimization process can be challenging. In this study, multi-objective optimization problems are formulated and solved for several different systems, as will be explained later. Therefore, using the epsilon-constraint method for each system would be time-consuming.

The weighted-objective method does not suffer from these drawbacks; hence, it is utilized to solve the multi-objective problems in this study.

2.6 Surrogate Modeling Methodology

Different surrogate modeling approaches have been utilized to address the two drawbacks outlined in Chap. 1. Using the elements presented in Fig. 2.5, these surrogate modeling approaches can be visualized as shown in Fig. 2.6. The first approach to constructing surrogate models is to map key performance metrics like fatigue damage or AEP, given load case specifications like average wind speed, wave elevation, turbulence intensity, etc, or their time series [117]. This approach is shown in Fig. 2.6a. A variety of different data-fitting approaches (e.g., artificial neural networks, Gaussian process models, recurrent neural networks, etc.) have been used to construct these surrogate models. Although some of these studies have used time-varying quantities as inputs to the surrogate models, the outputs predicted are time-invariant quantities.

The second set of approaches, which has been widely used in surrogate-assisted design optimization, involves constructing a surrogate model to predict the performance metrics given the

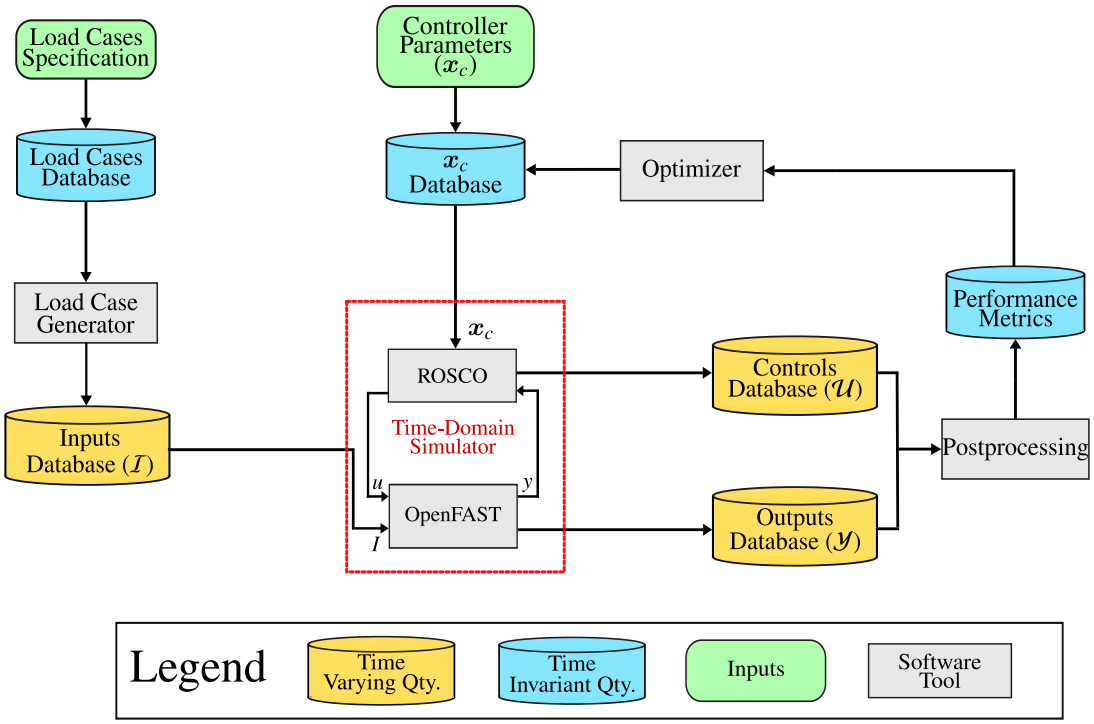
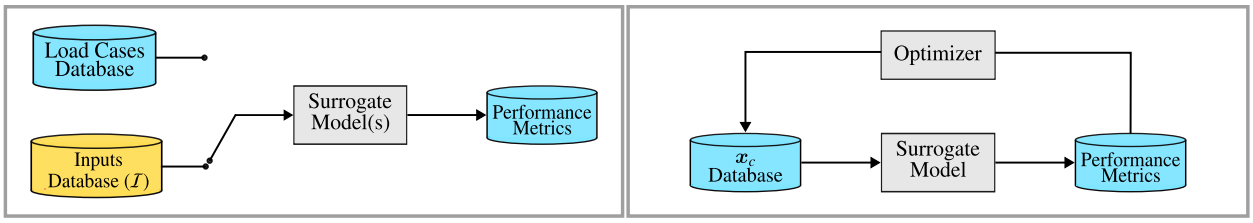


Figure 2.5: A high-level workflow diagram illustrating an optimization problem to identify the controller parameters ($\boldsymbol{x}_c$) for ROSCO using a shooting approach.

design variables in question [118,119]. This approach is shown in Fig. 2.6b. Typically, a set of load cases is identified, and a sampling scheme for the design variables is created. Then, the system is simulated to get the values of the key performance metrics at these sample points. A surrogate is then trained to predict the performance metrics given the design variables as the inputs, and is used in the design optimization process.

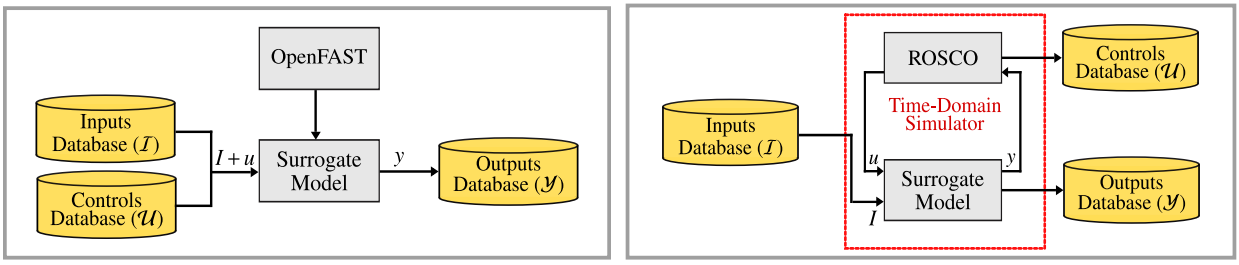
It is straightforward to combine these two surrogate modeling approaches to create a low-fidelity model that can be used for these two applications. However, there are some drawbacks to using the approaches as they are presented in the aforementioned studies to construct the low-fidelity model. These studies have presented different approaches to predict time-invariant quantities associated with wind turbines. However, to understand the behavior of these systems, it is essential to understand the key trends in the dynamic quantities, represented as the controls and outputs database in Fig. 2.3, along with the performance metrics. For example, in controller optimization studies, to understand why a set of $\boldsymbol{x}_c$ identified by the optimizer results in minimiz-



(a) Surrogate modeling approach 1.

(b) Surrogate modeling approach 1.

Figure 2.6: Two different surrogate modeling approaches that can be used for wind turbines. The first approach shown in Fig. 2.6a can be used to train a surrogate model that predicts key performance metrics given information about the load cases. The second approach, shown in Fig. 2.6b, has been used extensively to construct surrogate models that can be used for design optimization studies.



(a) Model construction.

(b) Model usage.

Figure 2.7: An alternate approach to construct a surrogate model that can be used for predicting time-series of key outputs, given inputs and controls.

ing/maximizing the key performance metrics, it is necessary to look at the time series signals of key controls and outputs which $\boldsymbol{x}_c$ results, which these surrogate models do not capture [98]. Additionally, in early-stage design studies, engineers often run multiple preliminary studies to explore the effect of optimizing for different sets of design parameters. When the set of design parameters changes, the process outlined above must be repeated to construct a new surrogate model, which can be computationally expensive and inefficient.

Therefore, a key contribution of this dissertation is to explore different approaches to constructing surrogate or low-fidelity models that can be used to predict the time series of the control variables and key outputs. Typically, in the workflow diagram presented in Fig. 2.5, evaluating OpenFAST is often the most computationally expensive operation. As shown in Fig. 2.5, OpenFAST should be characterized as a high-fidelity model that predicts the key outputs, given inputs, and controls. Therefore, constructing a low-fidelity model that can predict the key outputs and be coupled with the controller to perform closed-loop simulations would address the two prob-

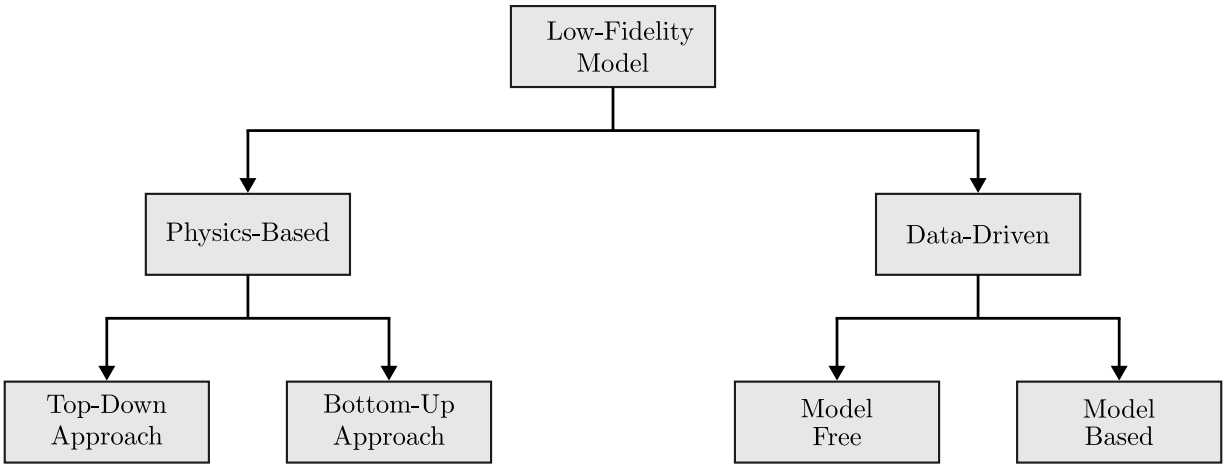


Figure 2.8: Different approaches that can be used to construct low-fidelity models.

lems outlined in Chap. 1. This approach is visualized in Fig. 2.7 using the elements discussed in Fig. 2.5. We limit our focus to testing these low-fidelity models for open-loop and closed-loop controller optimization studies. Primarily, there are two broad approaches that can be used to construct low-fidelity models of wind turbines, namely physics-based and data-driven, which are now summarized.

2.6.1 Physics-Based Models

Physics-based approaches try to approximate the key relationships using prior knowledge of the underlying physics of the system. Typically, this process starts by identifying simple relationships that approximate the behavior of key subsystems [111, 120–122]. Then, parameters in these relationships (e.g., the stiffness constant of the force-displacement relationship) can be found using analytic expressions or data and system response outputs predicted for different inputs. These simplified relationships can also be augmented to obtain more accurate ones that better represent the behavior of the system of interest. This additional model accuracy can lead to detailed mathematical representations closely approximating reality but may be computationally expensive to evaluate [72, 123]. These types of models are often referred to as white-box models, where the relationship between the quantities can be obtained by solving the relevant governing equations. This is a bottom-up way to construct a physics-based low-fidelity model.

A top-down approach could also be used to construct physics-based low-fidelity models. These approaches start from high-fidelity models and obtain different types of reduced-order or low-fidelity models. There are several ways to obtain these reduced-order models, depending on the high-fidelity model. For high-fidelity models such as OpenFAST, using Taylor-series approximations to obtain linearized model(s) is a widely used approach to obtain low-fidelity models [70]. Another approach is to use simplifying assumptions of the system behavior/property that enable simple models [118]. For example, in OpenFAST, assuming that the rotor blades and tower can be represented as Euler-Bernoulli beams with limited deflections allows the users to use a simplified solver to obtain the corresponding states associated with these subsystems [120]. Furthermore, developing these physics-based low-fidelity models can be time-consuming and requires technical expertise and background regarding the fundamental physics of the system. Instead, in this dissertation, we focus on using data-driven approaches that can be used to develop multi-input multi-output low-fidelity models.

2.6.2 Data-Driven Models

Data-driven low-fidelity models capture the input-output relations using different approaches like curve-fitting, machine learning, probabilistic models, etc. [117]. Unlike physics-based low-fidelity models, which can be complex to develop and construct, data-driven models can be straightforward to build. Because they map input-output relations, data-driven approaches can be flexible and are used in a wide variety of scenarios and applications [124]. This facet of data-driven models allows them to capture highly nonlinear and layered relationships between quantities, which may be difficult using physics-based models. These data-driven approaches have traditionally been used to capture the input-output relationship between time-invariant quantities. However, with the advent of deep learning techniques like recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, data-driven approaches have also been used to capture the relationship and trends between dynamic quantities [125].

In addition to machine learning approaches like the ones mentioned above, other approaches can be used to build low-fidelity/surrogate models of dynamic systems. Systems identification approaches have been widely utilized to construct low-fidelity models for various theoretical and real-world dynamic systems, and these approaches are built on rigorous theory [126–129]. At this point, we would like to categorize data-driven approaches into ones that assign a model structure to the input-output relations and ones that do not. Such a distinction has also been made in Ref. [124]. This distinction is helpful as it allows us to discuss specific modeling approaches from the field of systems identification as data-driven modeling approaches. For example, approaches like LSTMs do not assume a model/relationship structure between the inputs and outputs. In contrast, there are many approaches used in systems identification that often assume that the inputs and outputs can be related through a mathematical model. By specifying a model structure, the systems identification approaches can also guarantee certain properties about the identified model as well [130, 131]. For example, it is possible to ensure that the system identified is stable, which is a key requirement when using these models for closed-loop control simulations.

Model-based system identification approaches can further be broadly classified into two categories depending on how the model is constructed, namely parameter estimation methods and subspace identification methods. Parameter estimation methods assume that the inputs and outputs are related through a system model ($\boldsymbol{M}$) that has unknown parameters $\boldsymbol{M}(\boldsymbol{\theta})$ [130]. Parameter estimation methods then find the optimal $\boldsymbol{\theta}^*$ that minimizes the error between the actual model outputs and the outputs predicted by $\boldsymbol{M}(\boldsymbol{\theta}^*)$. On the other hand, subspace identification approaches identify a discrete linear time-invariant (LTI) system that relates how the outputs evolve with the inputs [127]. The linear system states and the corresponding matrices are identified by applying singular value decomposition to the Hankel matrix of the time series data. Unlike parameter estimation methods, subspace identification methods are not iterative.

Other model-based data-driven approaches have also been successfully used to construct surrogate models for wind turbines. These approaches can be generally termed as derivative function surrogate modeling (DFSM) approaches investigated in Refs. [86, 87, 91, 92]. DFSM captures the

relationship between the inputs ($\boldsymbol{u}$) and the system states ($\boldsymbol{\xi}$) and assumes they are related through derivative function or dynamic model $\boldsymbol{f}(\cdot)$ that describes the first-time derivatives of the states ($\dot{\boldsymbol{\xi}}$) such that $\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\xi})$. The DFSM is a surrogate of $\boldsymbol{f}_{\text{DFSM}}(\cdot) \approx \boldsymbol{f}(\cdot)$ that predicts how the state derivatives evolve with respect to the controls and states. This approach was developed to be used for continuous-time open-loop optimal control studies, as popular solution methods for these problems require information about how the state derivatives evolve over time for different controls and states [100, 132].

The assumption that a state-space model can effectively approximate the system behavior is common in DFSM and system-identification approaches. However, because of how these models have been constructed and utilized for different applications, previous studies using the DFSM approach have not tried to understand it from a systems identification perspective. Instead, DFSM has been primarily understood from a surrogate modeling perspective. System-identification approaches typically build models using time series data of the key inputs and outputs of the system. Once constructed, these models have been used for applications such as model prediction, control development, etc. In contrast, the previous instances of the DFSM approach assume that the state-derivative function is available in a form that can be sampled for different values of the controls and states to get the corresponding state-derivative values [86, 87, 91]. Subsequently, a data-fitting method, like radial-basis functions (RBF) or GPR, is used to construct $\boldsymbol{f}_{\text{DFSM}}$. Once constructed, these models have been used primarily for open-loop optimal control studies. Methods from surrogate-assisted optimization have been used to refine the DFSM closer to the identified optima. In this dissertation, we present an approach to constructing a DFSM using principles from systems identification. We construct this model using the input-output time series data.

A key decision that will affect the model quality identified for model-based approaches is the model type. A variety of model types have been explored, such as time-domain vs. frequency-domain, continuous-time vs. discrete-time, linear vs. nonlinear, etc. Here, the DFSM is desired to be a continuous-time model. Once this has been fixed, the next step is to investigate whether a linear or nonlinear system provides more accurate results. A good starting point to investigate the

model type is an LTI system. It is also straightforward to ensure that the identified linear model is stable. We explore the LTI structure for a simple system.

Chapter 3

Development of an Optimal Variable Pitch Controller for Floating Axial-Flow Marine Hydrokinetic Turbines

This chapter discusses the development of an optimal variable-pitch controller for floating, axial-flow marine turbines[3]. In Sec. 3.1, we summarize the traditional controller tuning and design process for marine turbines. Then, in Sec. 3.2, we summarize the different control frameworks that could be used for marine turbines. In Sec. 3.3 we describe the controller optimization problem, and in Sec. 3.4 we discuss the results of this study.

3.1 Controller Tuning and Design

The choice of the control scheme has a significant impact on the design and performance of the marine turbine. Regardless of the control scheme, traditional wind turbine operating regions, control goals, variables, and tuning processes have been adopted for marine turbines as well [133]. The two main control variables in both applications are the blade pitch (β) and the generator torque (τ_g). Based on these variables, the two control schemes that are considered for marine turbines are fixed-pitch variable-speed (FP-VS) control and variable-pitch variable-speed control (VP-VS). Predominantly, an FP-VS scheme has been utilized for marine turbines to keep the design simple and reduce the risk of failure associated with a blade pitch mechanism. However, despite similar concerns, the benefits of using a blade pitch mechanism in minimizing the loads and increasing the power production have led to their widespread adoption for wind turbines [44].

Similar to the control of wind turbines, the generator speed (ω_g) is the main feedback variable for marine turbines. The system is linearized around set operating points for both VP-VS and

[3]This chapter is based on the following publication [98].

FP-VS schemes, and controllers are designed through Bode-shaping [134]. Additional parameters also characterize the controller's performance. The values of the parameters are selected to give a "reasonable" performance. Such an approach has a number of drawbacks that make it unsuitable for early-stage design exploration studies. First, if the turbine's physical design changes, the tuning process must be repeated to design a controller for that specific turbine design realization. Deriving linearized models of the system can be computationally expensive, and the expertise of a control engineer is required to select good values for the control gains [135].

Additionally, to test the performance of the design and the controller, the marine turbine must be simulated over a wide range of design load cases (DLCs) specified by the International Electrotechnical Commission (IEC) [43, 136]. For the design to have a lower levelized cost of energy (LCOE), the controller must be able to generate the rated power for the turbine and minimize loads over the entire range of DLCs. A controller tuned to have a "reasonable" performance cannot meet such requirements. The design space for the controller parameters can be nonlinear. Therefore, finding the optimal parameters is often nonintuitive. Recently, optimization studies have been used to design controllers that can satisfy these challenging requirements [88, 136]. To explore the design space more efficiently, a controller that can be modified easily and used in conjunction with an optimizer is necessary.

Reference Open-Source Controller (ROSCO), introduced in Ref. [33], is an open-source controller for wind turbines that has an automated tuning process while providing industry-standard functionalities. ROSCO uses a generic model of a given turbine to schedule gains for both the torque and blade pitch controllers. The controller and the gains are parameterized by additional variables that can be specified by the user or identified using an optimizer. These features make ROSCO an ideal tool to be used in early-stage design studies.

3.2 Control Frameworks for Marine Turbines

For marine turbines, similar to wind turbines, the inflow speed is divided into below-rated and above-rated regions, and different control goals are utilized in each region [133]. In the below-

rated region, generator torque (τ_g) is the primary control variable [134, 137]. In the above-rated region, two different control strategies are possible [138].

3.2.1 Fixed-Pitch Control

In FP-VS turbines, the blades are fixed to the nacelle at an optimal angle, and τ_g is varied to track the rated power. The lack of a pitching mechanism reduces the capital and maintenance costs and keeps the overall design simple [46]. There are, again, two different ways fixed-pitch controllers can track the rated power, namely, overspeed control or underspeed control [26]. In the overspeed control strategy, generator speed (ω_g) is increased by reducing the generator torque [139, 140]. In underspeed control, τ_g is increased to lower ω_g and track the rated power [141].

Despite allowing a simpler design, FP-VS turbines have several drawbacks. Since larger values of τ_g are required in underspeed control, the size of the generator must be increased [26]. As the target ω_g varies with the current input, there is a larger fluctuation in the power generated [133]. Furthermore, the blades for underspeed turbines are specifically designed to induce stall at higher flow speeds [26, 59, 133, 141]. Because of these issues, the efficiency of fixed-pitch turbines is comparatively lower than their variable-pitch counterparts, and the blades must be designed to withstand higher loads.

3.2.2 Variable-Pitch Control

For VP-VS turbines, in the above-rated region, the blade pitch is varied to keep the turbine rotating at its rated speed [134, 138]. The addition of the blade pitch mechanism increases the capital cost and modes of failure, which subsequently increases the risk of downtime and maintenance costs [26]. However, using blade pitch control results in lower thrust and fatigue loads and better power tracking compared to fixed-pitch controllers [137]. Additionally, as the turbine size increases, the ability of the blade pitch mechanism to reduce the increased loads on the system becomes critical [44]. Therefore, the focus of this study will be on VP-VS turbines.

Equations (2.9a) and (2.9b) can be used to describe certain aspects of the RM1 turbine given in Sec. 1.6.1. Additionally, the RM1 turbine is designed to have a variable-pitch controller in the

above-rated region. Therefore, ROSCO can be used to develop the necessary controllers for the RM1 turbine.

3.3 Controller Optimization

We focus here on optimizing the performance of the system by tuning the blade pitch controller. The performance of the blade pitch controller is characterized by the variables $\boldsymbol{x}_c = [\boldsymbol{\omega}_{\text{pc}}, \boldsymbol{\zeta}_{\text{pc}}, k_{\beta,\text{float}}, \omega_{\text{ptfm}}]$. Therefore, $\boldsymbol{x}_c$ is selected as the set of optimization variables used in this study. The gains $\boldsymbol{\omega}_{\text{pc}}$ and $\boldsymbol{\zeta}_{\text{pc}}$ for two different values of v, one in the near-rated region and one in the above-rated region at $\boldsymbol{v} = [2.0, 2.7]$ m/s are chosen.

For the purposes of this study, the turbine will be simulated using environmental data from Cook Inlet off the coast of Alaska. The metocean (meteorological-oceanographic) conditions are obtained from the study carried out in Ref. [142]. We simulate the model in OpenFAST using 11 different current profiles characterized by the average current speed (v_{avg}) with $v_{\text{avg}} \in [0.5, 3]$ m/s and use the results from the simulations to calculate the fatigue loads, power production, and other quantities. In the rest of the study, one function evaluation implies that the turbine is simulated for these 11 load cases. The turbulence intensity TI [%], which is the ratio of the standard deviation of v to the mean for all 11 load cases, is shown in Fig. 3.1a. The probability distribution for v at this location is also shown in Fig. 3.1b. In addition to the current, the turbine is subjected to irregular waves generated from the JONSWAP spectrum for different significant wave heights H_s [m] and significant wave periods T_p [s] as shown in Fig. 3.1a. Different H_s and T_p are used for each of the 11 load cases.

The optimal controller would help in balancing the power production and minimizing the ultimate and fatigue loads on the turbine. The IEC TS 62600-2 specifies that both the ultimate and fatigue loads must be calculated to assess the performance of the turbine [43]. But in this study, we limit our focus to fatigue loads. Studies on floating wind turbines have reported increased tower-base fatigue damage caused by the additional wave loading [143]. To understand this effect for

marine turbines, the tower-base fatigue loading will be the key load investigated in the subsequent control design study.

3.3.1 Damage Equivalent Load

In this study, we use the damage equivalent load (DEL) metric to characterize the fatigue load on the turbine. DEL approximates the cumulative fatigue damage incurred over a given time span to a single equivalent load that would cause the same damage. This aspect allows the use of DEL as an objective or constraint in the optimization study. We use the rainflow cycle counting method and the Palmgren-Miner rule to estimate the DEL from the time series signals of the tower-base moments. More details regarding the rainflow cycle counting method can be found in Ref. [144]. OpenFAST calculates the tower-base forces and moments in three directions: side-to-side, fore-aft, and yaw. The corresponding moments in each direction are represented as $\mathrm{M}_{t,x}$, $\mathrm{M}_{t,y}$, and $\mathrm{M}_{t,z}$, respectively. The total tower-base moment (M_t) from each load case is calculated from these signals as:

$$\mathrm{M}_t = \sqrt{\mathrm{M}_{t,x}^2 + \mathrm{M}_{t,y}^2 + \mathrm{M}_{t,z}^2} \tag{3.1}$$

Once the time series signal for M_t is available, the rainflow counting method is used to identify the loading cycles in the time series. Next, the amplitude and mean moment for every loading cycle are identified. This step provides a range of values for the amplitude and moment corresponding to different loading cycles. This range of amplitudes for the different cycles is then separated into various bins, and the count for each bin is then calculated to obtain the corresponding $\mathrm{M}_{t,i}$ and n_i pair, where $\mathrm{M}_{t,i}$ refers to the amplitude of the "i-th" bin, and n_i refers to its count. For a more detailed explanation of how DEL_t is calculated, please refer to Refs. [143, 145, 146]. Here, 100 bins are used in the calculation of the DEL_t. The DEL_t for a specific load case is then calculated

as:

$$\text{DEL} = \left(\sum_{i=1}^{100} \frac{M_{t,i}^{m} n_i}{t_{\text{span}}} \right)^{1/m} \tag{3.2}$$

where t_{span} refers to the time span of the time series, and $m = 4$ refers to the slope of the M-N curve for the material. The DEL for all 11 load cases is calculated using the aforementioned steps, and the results are combined to calculate the total tower-base damage equivalent load (DEL_t) as:

$$\text{DEL}_t = \int_v \text{DEL}(v_{\text{avg}}) f_p(v_{\text{avg}}) dv \tag{3.3}$$

where $\text{DEL}(v_{\text{avg}})$ is DEL for a load case, and $f_p(v_{\text{avg}})$ is the probability of v_{avg} obtained from the probability distribution shown in Fig. 3.1b.

Minimizing DEL_t is chosen as the objective for the control design problem investigated in this study. Other objectives that could be used are discussed briefly in Ref. [88]. A constraint is added to ensure ω_g does not exceed 20% of its rated value. The resulting optimization problem formulation is:

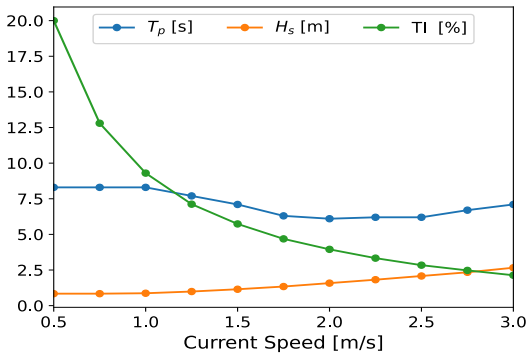
$$\begin{aligned}
\text{changing:} \quad & \boldsymbol{x}_c = [\boldsymbol{\omega}_{\text{pc}}, \boldsymbol{\zeta}_{\text{pc}}, k_{\beta,\text{float}}, \omega_{\text{ptfm}}] && \text{(3.4a)} \\
\text{minimize:} \quad & \text{DEL}_t && \text{(3.4b)} \\
\text{subject to:} \quad & \omega_g \leq 1.2 \cdot \omega_{g,\text{rated}} && \text{(3.4c)} \\
& \boldsymbol{x}_{c,\min} \leq \boldsymbol{x}_c \leq \boldsymbol{x}_{c,\max} && \text{(3.4d)}
\end{aligned}$$

where $\boldsymbol{x}_{c,\min} = [0.1, 0.1, -2, 10^{-5}]$ and $\boldsymbol{x}_{c,\max} = [1.5, 3.0, 0.0, 1.0]$, respectively.

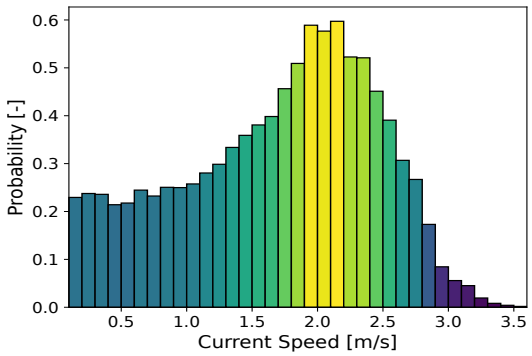
In theory, maximizing AEP could also be considered in the objective. Similar to the DEL_t, the AEP is calculated as:

$$\text{AEP} = 8760 \int_v \bar{P}(v_{\text{avg}}) f_p(v_{\text{avg}}) dv \tag{3.5}$$

where $\bar{P}$ refers to the average power generated for a load case with a mean of v_o. However, for the choice of $\boldsymbol{x}_c$ and its range considered in this study, the AEP does not vary by much from its nominal value. The generator power P is calculated as $P = \tau_g \omega_g$, and the goal of the controller in the above-rated region is to track $\omega_{g,\text{rated}}$ as τ_g is held constant. From Eq. (3.5), it is clear that to increase AEP, $\bar{P}$ must be increased. To increase $\bar{P}$, $\omega_{g,\text{rated}}$ must be increased. Since $\omega_{g,\text{rated}}$ is not changed in this study, $\bar{P}$ and, subsequently, the AEP are not affected in a significant manner.



(a) Wave period (T_p) [s], wave height (H_s) [m], and turbulence intensity (TI) [%].



(b) Probability density of the current speed at Cook Inlet.

Figure 3.1: The metocean conditions and the probability density of the current speed at Cook Inlet, used to simulate the system in the optimization problem in Eq. (3.4).

3.3.2 Notes on Computational Time

All the simulations in this study were run on a high-performance computing cluster with 2 x Intel Xeon Gold 6148 CPU with 40 cores per node and 192 GB of RAM. The IEC TS 62600-2 recommends using simulations that are 600 s long to evaluate the performance of the turbine [43]. The standard also recommends that key metrics be evaluated using the turbine's performance for multiple DLCs (1.1, 1.2, 1.3, etc.). Running all 11 load cases considered in this study (i.e., one function evaluation, in parallel) for a time span of 600 s takes approximately 4.5 hours. Depending on the choice of the optimization algorithm, several hundred function evaluations might be required. The computational expense of simulating the system for 600 s for multiple DLCs per the IEC recommendation can be computationally intractable. Therefore, to reduce the computational

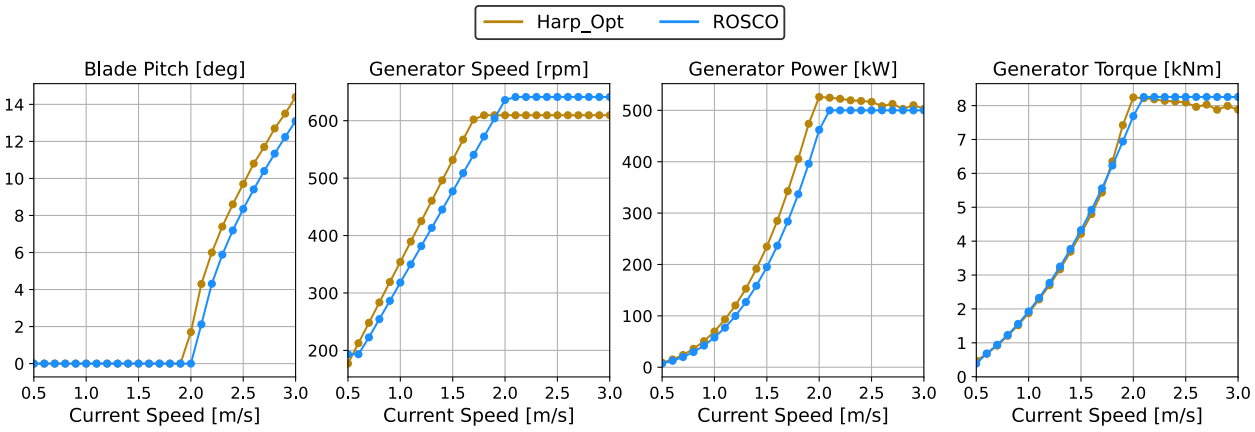


Figure 3.2: Comparison of steady-state values of key signals obtained using `Harp_Opt` as reported in Ref. [17] and ROSCO with $\boldsymbol{x}_c = \boldsymbol{x}_{c,\text{init}}$.

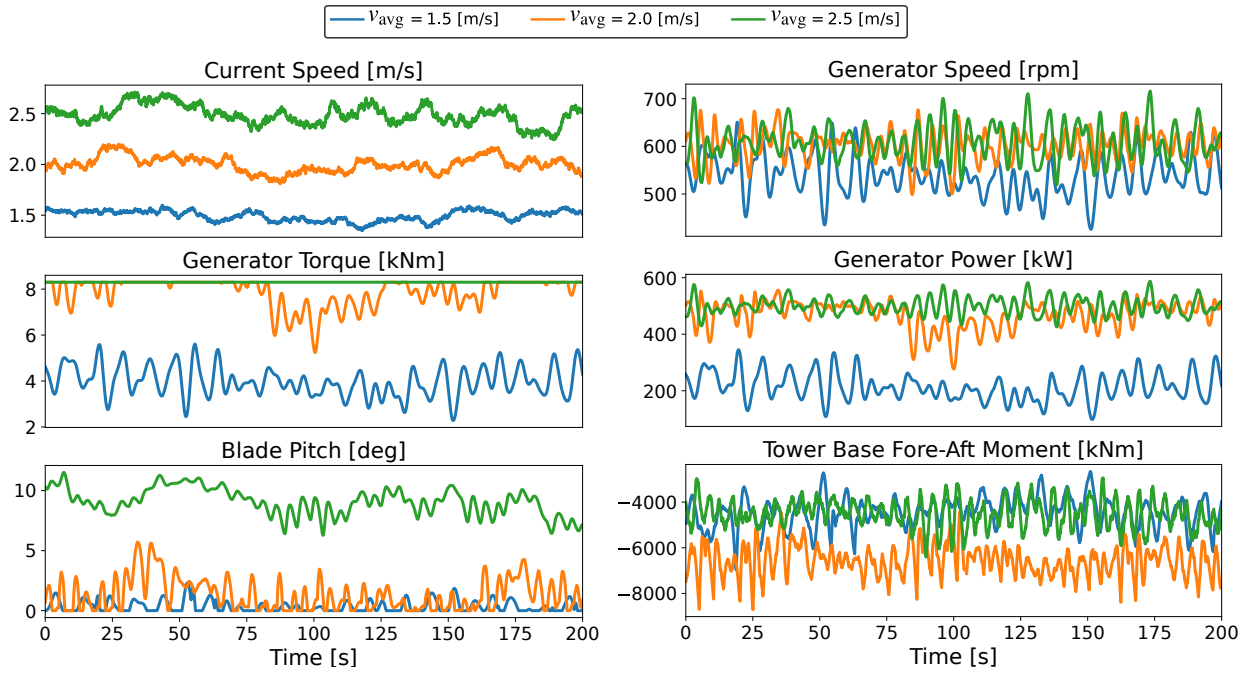
expense, we simulate the system for a time span of 200 s, which takes approximately 1.3 hours per function evaluation. Additionally, we only use load cases from DLC 1.1 to test the turbine's performance at this time, but with more resources, additional DLCs and longer time spans can be selected.

Based on the recommendation provided in Ref. [88] on the choice of the optimizer, a derivative-free optimizer `COBYLA` introduced in Ref. [54] is used. For every iteration of the optimization problem, the model is simulated for the entire range of DLCs, and the relevant quantities are then calculated from these simulations. The WEIS toolbox, available at Ref. [107], is used to run the studies.

3.4 Results

3.4.1 Baseline Performance

Before discussing the results of the optimization study, it is necessary to validate the control response obtained using ROSCO for the F-RM1-Q turbine. The results reported in Ref. [17] are the only other control-related results for the RM1 turbine available in the literature. Comparing the performance obtained using ROSCO to the results presented in Ref. [17] would be a good approach to validate the control scheme obtained using ROSCO. However, there are certain caveats that must be established before such a comparison is carried out. The results presented in Ref. [17] are for



(a) Current speed and control variables.

(b) Key generator outputs and tower-base fore-aft moment.

Figure 3.3: Time series plot of key signals in the below-rated (1.5 m/s), transition (2.0 m/s), and above-rated (2.5 m/s) regions obtained using $\boldsymbol{x}_{c,\text{init}}$.

a fixed configuration of the RM1 turbine. The steady-state values of β, τ_g, and the resulting ω_g and P reported in Ref. [17] were identified using a turbine design tool `Harp_Opt`. Therefore, for a fair comparison, the floating degrees of freedom (surge, sway, heave, roll, pitch, yaw) for the F-RM1-Q model are disabled in OpenFAST for this validation study only. The F-RM1-Q turbine is then simulated using constant current speeds for $v \in [0.5, 3]$ m/s, and the steady-state values of β, τ_g, ω_g and P obtained using ROSCO are plotted against the values reported using `Harp_Opt` in Fig. 3.2. It is clear from Fig. 3.2 that the trends and values of β, τ_g, ω_g, and P obtained using ROSCO are similar to the values reported in Ref. [17]. The difference in the steady-state values can be attributed to the different control goals and methodologies used by these two approaches.

With the control trends validated, we simulated the F-RM1-Q model using $\boldsymbol{x}_{c,\text{init}}$ for the load cases described in Sec. 3.3, with the floating degrees of freedom enabled. The time series plots of some key signals for three different current speeds in the below-rated, transition, and above-rated regions are shown in Fig. 3.3. Figure 3.3a shows how the control variables τ_g and β change in different regions. In the below-rated region, τ_g is the main control variable. Similarly, β is the

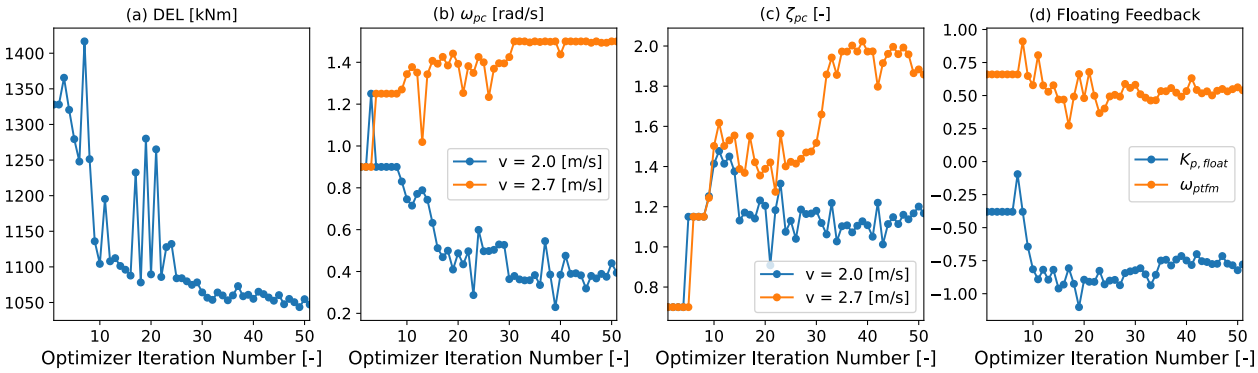


Figure 3.4: Convergence of the objective and variables for the optimization problem in Eq. (3.4) with (a) the objective tower-base damage equivalent loading (DEL_t); (b) natural frequency ($\boldsymbol{\omega}_{\text{pc}}$); (c) damping ratio ($\boldsymbol{\zeta}_{\text{pc}}$); and (d) the proportional gain ($K_{p,\text{float}}$) and the bandwidth of the low-pass filter (ω_{ptfm}) used in the floating feedback controller.

main control variable in the above-rated region, and τ_g is held constant. In the transition region, a combination of these two variables is used. These control trajectories obtained by using the goals outlined in Sec. 2.4 are in line with the trends seen for wind turbines. The DEL_t calculated over the range of DLCs when using $\boldsymbol{x}_{c,\text{init}}$ is 1327.89 kNm as shown in Table. 3.1. The rotor overspeed value is 35.40 %. The performance of the controller can be improved by using an optimizer to find values of $\boldsymbol{x}_c$.

3.4.2 Optimization Results

The optimization problem is solved using $\boldsymbol{x}_{c,\text{init}}$ as the initial value with an optimality tolerance of $o = 10^{-2}$. The optimizer converges to a point with a lower objective value that satisfies the

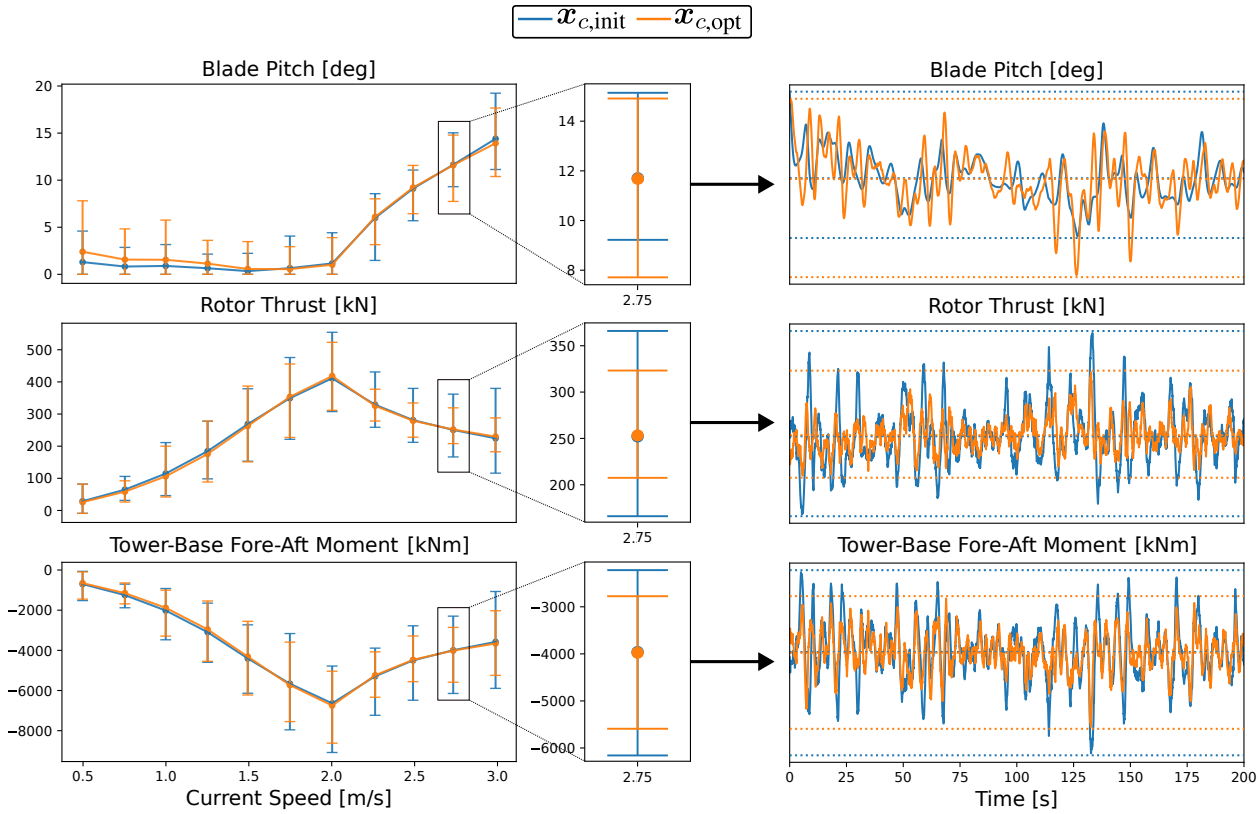
Table 3.1: Optimization results.

Variable	Initial Value ($\boldsymbol{x}_{c,\text{init}}$)	Optimal Value ($\boldsymbol{x}_{c,\text{opt}}$)
$\boldsymbol{\omega}_{\text{pc}}$ [rad/s]	$[0.90, 0.90]$	$[0.40, 1.50]$
$\boldsymbol{\zeta}_{\text{pc}}$ [-]	$[0.70, 0.70]$	$[1.09, 1.79]$
$k_{\beta,\text{float}}$ [-]	-0.38	-0.76
ω_{ptfm} [rad/s]	0.66	0.53
DEL_t [kNm]	1327.89	1052.01
ω_g overspeed [%]	35.40	19.55

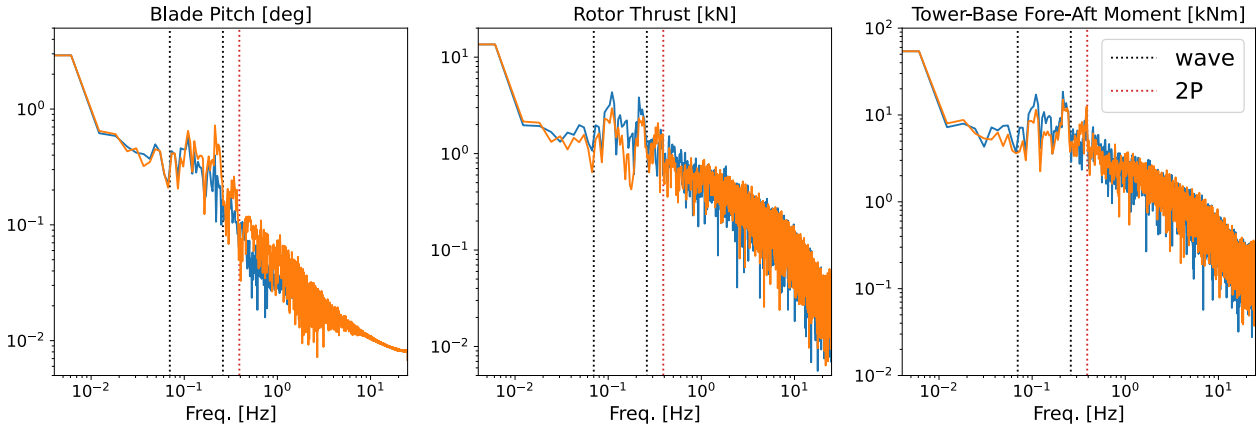
constraints in 51 iterations. It takes a total of 74 hours to solve the optimization on the high-performance computing cluster mentioned in Sec. 3.3.2. The optimal values of $\boldsymbol{x}_c$ denoted by $\boldsymbol{x}_{c,\text{opt}}$ and the objective function DEL_t are presented in Table 3.1. The final value of DEL_t is 1052.01 kNm, which represents a 20.77% reduction from the unoptimized value. The optimal point identified by the optimizer also reduces the rotor overspeed to 19.5%, which satisfies the constraint placed in Eq. (3.4). The convergence of DEL_t and $\boldsymbol{x}_c$ are shown in Fig. 3.4.

Of the signals used to calculate M_t in Eq. (3.1), the tower-base fore-aft moment ($\text{M}_{t,y}$) is the key driver for this system. By understanding how $\text{M}_{t,y}$ changes for $\boldsymbol{x}_{c,\text{init}}$ and $\boldsymbol{x}_{c,\text{opt}}$, we can understand the results of this optimization study. To minimize DEL_t, it is necessary to minimize the amplitude of the loading cycles in $\text{M}_{t,y}$, which subsequently minimizes M_i in Eq. (3.2). The rotor thrust is a key driver for $\text{M}_{t,y}$. The rotor thrust is the force caused by the current flow on the rotor. This force causes a moment at the base of the tower, which is a key driver for $\text{M}_{t,y}$. By minimizing the amplitude of fluctuations in the rotor thrust signal, it is possible to minimize the amplitude of loading cycles in $\text{M}_{t,y}$.

The mean value and range of key quantities like β, $\text{M}_{t,y}$, and the rotor thrust are plotted in Fig. 3.5 with $\boldsymbol{x}_{c,\text{init}}$ and $\boldsymbol{x}_{c,\text{opt}}$ for all 11 load cases. It is clear from Fig. 3.5a, that $\boldsymbol{x}_{c,\text{opt}}$ results in β that minimizes the fluctuations in the rotor thrust and subsequently $\text{M}_{t,y}$. The oscillations in these three quantities are caused by the wave inputs and the 2P or the "twice per revolution" frequency, which corresponds to the frequency at which the rotor blades cross the tower, as shown by the power spectral density plots in Fig. 3.5b. Figure 3.6a shows the time series plot for M_t, and Fig. 3.6b shows the histogram of the amplitudes for different cycles obtained using the rainflow counting method for a load case with $v_{\text{avg}} = 2.75$ m/s for both $\boldsymbol{x}_{c,\text{init}}$ and $\boldsymbol{x}_{c,\text{opt}}$. In Fig. 3.6b, we use $n = 10$ bins to visualize the reduction in amplitude better, but $n = 100$ bins are used to calculate the DEL_t for the optimization problem. These figures clearly illustrate how the reduction of the amplitude in M_t results in the reduction of the DEL_t. The optimal solution $\boldsymbol{x}_{c,\text{opt}}$ reduces the fluctuations in ω_g as well, as shown in Fig. 3.7. Since τ_g is held constant in the above-rated region, this also leads to better tracking of P.

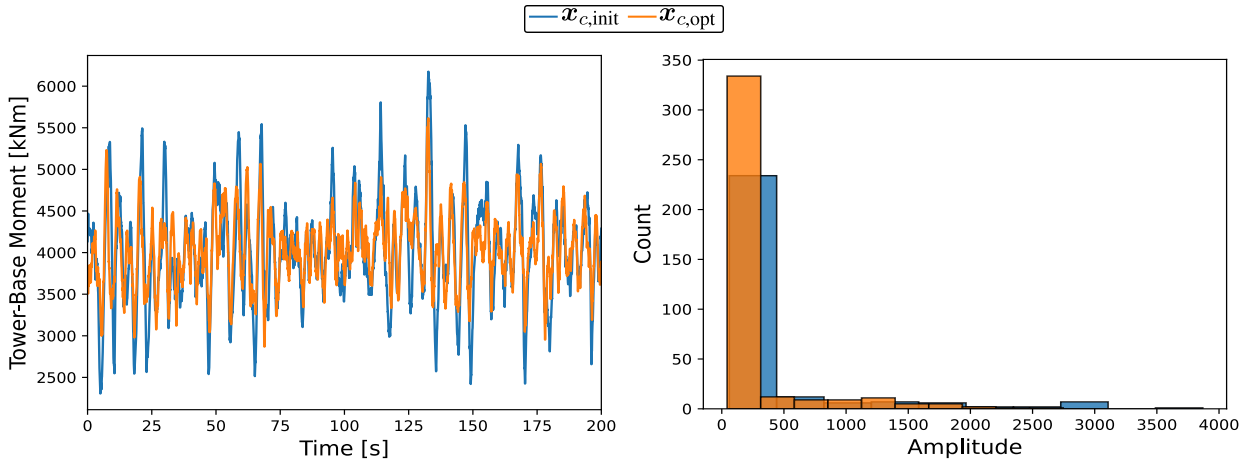


(a) Comparison of the mean values and range of key variables and time series signals for $v_{\text{avg}} = 2.75$ m/s.



(b) Power spectral density plots of the key signals for $v_{\text{avg}} = 2.75$ m/s.

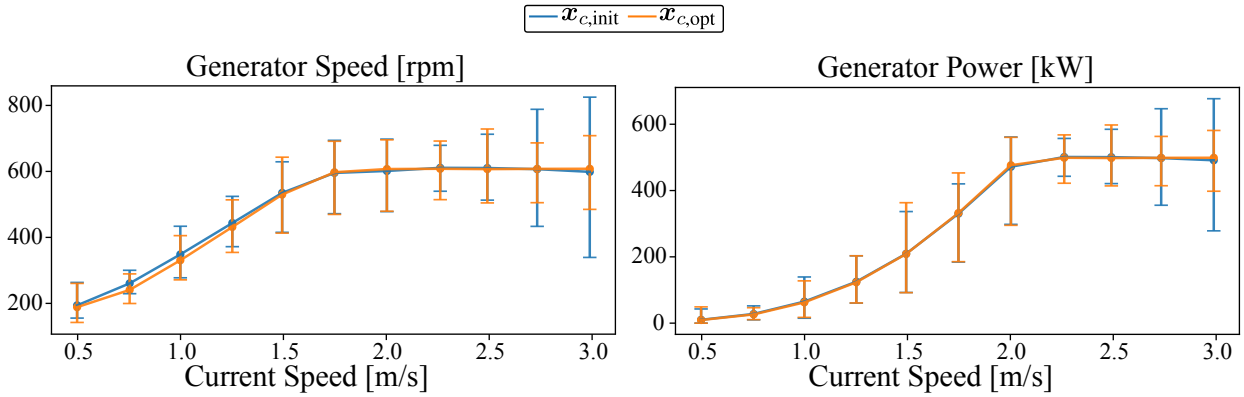
Figure 3.5: Comparison of the mean values of key signals like the blade pitch (β), rotor thrust, and the tower-base fore-aft moment ($M_{t,y}$) between $\boldsymbol{x}_{c,\text{init}}$ and $\boldsymbol{x}_{c,\text{opt}}$ for all 11 load cases, and the time series of the same quantities for $v_{\text{avg}} = 2.75$ m/s.



(a) Tower-base moment.

(b) Histogram plot of the amplitude for different cycles.

Figure 3.6: The time series plot for the tower-base moment (M_t), and the histogram plot of the amplitudes for the different cycles obtained using the rainflow counting method for $n = 10$ bins for $v_{\text{avg}} = 2.75$ m/s for both $\boldsymbol{x}_{c,\text{init}}$ and $\boldsymbol{x}_{c,\text{opt}}$.



(a) Generator speed.

(b) Generator power.

Figure 3.7: The mean value and range of the generator speed (ω_g) and generator power (P) for all 11 load cases for both $\boldsymbol{x}_{c,\text{init}}$ and $\boldsymbol{x}_{c,\text{opt}}$.

The key takeaway from this chapter is that ROSCO can be successfully extended for closed-loop control of marine turbines, and using an optimizer to identify the optimal control parameters can result in a 28% reduction in the damage equivalent loads.

Chapter 4

Low-Fidelity Modeling

This chapter explores the construction and validation of the different data-driven low-fidelity modeling approaches[4]. First, in Sec. 4.1, we present the DFSM approach for a simple 2-d robotic system. Then, in Sec. 4.2 we extend the methods to construct DFSM for wind and marine turbines. Once constructed, we compare and validate the DFSM against other data-driven modeling approaches in Sec. 4.3.

4.1 Illustrative Example

This section explores the proposed DFSM approach for an illustrative example. Consider the derivative function of the two-link robot system described in Example 2.10 of Ref. [147]. The system is characterized by the angle of the two links ($[\theta_1, \theta_2]$), their inertial velocities ($[\dot{\theta}_1, \dot{\theta}_2]$), the torques applied at the links ($[u_1, u_2]$), and the physical characteristics of the system like the length, mass, and moment of inertia of the two links represented as $\boldsymbol{p}$. Assuming the states to be $\boldsymbol{\xi} = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]^T$, and controls to be $\boldsymbol{u} = [u_1, u_2]^T$, the nonlinear state-space model be represented as:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{f} = \begin{bmatrix} \xi_2 \\ f_2(\boldsymbol{\xi}, \boldsymbol{u}, \boldsymbol{p}) \\ \xi_4 \\ f_4(\boldsymbol{\xi}, \boldsymbol{u}, \boldsymbol{p}) \end{bmatrix} \tag{4.1a}$$

$$\boldsymbol{y} = \boldsymbol{\xi} \tag{4.1b}$$

[4]This chapter has been adapted from the publications [90, 99].

The expressions for f_2 and f_4 are highly nonlinear and derived from the Lagrangian of the robotic system. For example, using fixed $\boldsymbol{p}$, f_2 is:

$$\begin{aligned} f_2(\boldsymbol{\xi}, \boldsymbol{u}) = -\Big(&5500(u_1 - u_2) - 100062\cos(\xi_1) + \cdots \\ &+ 29430\cos(\xi_1 + 2\xi_3) + 1800\xi_2^2\sin(2\xi_3) + \cdots \\ &- 6000u_2\cos(\xi_3) + 3300\xi_2^2\sin(\xi_3) + 3300\xi_4^2\sin(\xi_3) + \cdots \\ &+ 6600\xi_2\xi_4\sin(\xi_3)\Big)/(1800\cos(2\xi_3) - 5295) \end{aligned} \tag{4.2}$$

Please refer to Ref. [147] for the detailed derivation of $\boldsymbol{f}$ and values of $\boldsymbol{p}$. This analytic system will be used to demonstrate the different steps associated with the multi-fidelity DFSM approach.

Briefly, the different steps associated with the DFSM approach are:

1. Generate simulations.
2. Extract state derivative information.
3. Construct a low-fidelity model.

4.1.1 Generating Simulations

For the system considered in Eq. (4.1a), we generate total of n_{sim} starting points $\boldsymbol{\xi}(0)$ and control inputs $\boldsymbol{u}$. In this example, we use uniform random signals for $\boldsymbol{u}$, but a more strategic selection of controls might be considered that explores the state space. By simulating the system using these inputs, we can obtain n_{sim} sets of output trajectories $\boldsymbol{y}(t)$ and time mesh $\boldsymbol{t}$. From $\boldsymbol{y}$, the state trajectories $\boldsymbol{\xi}$ can be extracted and organized as:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}^{(1)} & \boldsymbol{t}^{(2)} & \cdots & \boldsymbol{t}^{(n_{\text{sim}})} \end{bmatrix} \tag{4.3a}$$

$$\boldsymbol{I} = \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}^{(1)} & \boldsymbol{u}^{(2)} & \cdots & \boldsymbol{u}^{(n_{\text{sim}})} \\ \boldsymbol{\xi}^{(1)} & \boldsymbol{\xi}^{(2)} & \cdots & \boldsymbol{\xi}^{(n_{\text{sim}})} \end{bmatrix} \tag{4.3b}$$

4.1.2 Extracting State Derivative Information

When the direct evaluation of $\boldsymbol{f}$ is not possible, the state derivative information can be indirectly obtained from the simulated state trajectories. A continuous polynomial approximation of sampled signals is available in many tools. With at least a C^1 approximation, approximate first-order derivatives can be obtained (and higher-order derivatives as well, depending on the method used). Here, a cubic spline polynomial approximation is used to construct continuous $\boldsymbol{\xi}(t)$ and then the exact polynomial derivatives are found for $\dot{\boldsymbol{\xi}}(t)$. Once $\boldsymbol{\xi}$ and $\boldsymbol{t}$ are available, the functions `spline` and `fnder`, available in the curve fitting toolbox in `MATLAB`, and the class `CubicSpline` from `SciPy` can be used to construct the approximation and evaluate the derivatives.

For the illustrative example, Fig. 4.1 compares the actual state derivatives for Eq. (4.1a) and the ones obtained from evaluating the polynomial approximation. The mean error between the actual derivatives and the polynomial derivative approximation evaluated for a hundred random simulations is of the order 10^{-7}, which shows that the polynomial approximation can provide accurate derivative values on a simple analytic example.

Now, the input-output pairs $\langle \boldsymbol{I}, \dot{\boldsymbol{X}} \rangle$ are available to construct $\hat{\boldsymbol{f}}(\cdot)$ where:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\boldsymbol{\xi}}^{(1)} & \dot{\boldsymbol{\xi}}^{(2)} & \dots & \dot{\boldsymbol{\xi}}^{(n_{\text{sim}})} \end{bmatrix} \tag{4.4}$$

4.1.3 Low-Fidelity Model

The low-fidelity portion is found by constructing a least-squares approximation between the inputs $\boldsymbol{I}$ and the state derivatives $\dot{\boldsymbol{X}}$:

$$\hat{\boldsymbol{f}}_{\text{low}}(\boldsymbol{I}) = \hat{\boldsymbol{f}}_{\text{L}}(\boldsymbol{I}) = \boldsymbol{L}\boldsymbol{I} \tag{4.5a}$$

$$\boldsymbol{L} = \boldsymbol{I}^T(\boldsymbol{I}\boldsymbol{I}^T)^{-1}\dot{\boldsymbol{X}} \tag{4.5b}$$

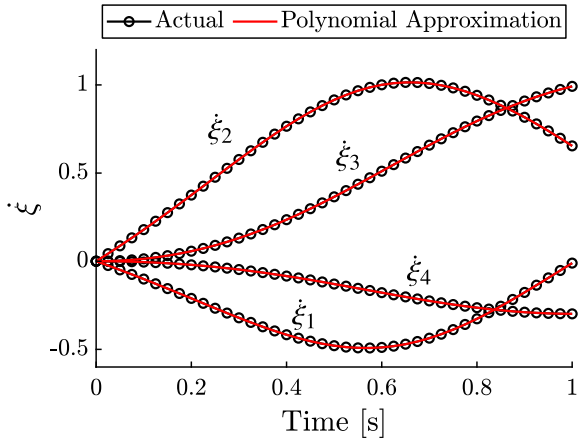


Figure 4.1: Comparison of actual derivative value to the polynomial approximation for the two-link robot system.

The corresponding state $\boldsymbol{A}_{\mathrm{L}}$ and input $\boldsymbol{B}_{\mathrm{L}}$ matrices can be extracted from $\boldsymbol{L}$. This model will be referred to as a 'linear-fit' model as opposed to a 'linear' or 'linearized' to avoid ambiguity.

In addition to the inputs $\boldsymbol{I}$, some systems are characterized by additional parameters $\boldsymbol{w}$ that affect the dynamics. Wind turbine dynamics, for example, depend heavily on the incoming wind speed. A single linear-fit model might not be adequate to capture the dynamics of wind turbines over the entire range of wind speed values. In such cases, multiple linear-fit models can be derived for a range of parameter values and aggregated to provide a more accurate prediction, such that the $\hat{\boldsymbol{f}}_{\mathrm{L}}$ is a function of $\boldsymbol{w}$. This is similar to the idea of linear-parameter varying (LPV) models explored in detail in Ref. [101], where the authors have shown that utilizing multiple linear models can provide a better approximation as opposed to using a single linear model with:

$$\hat{\boldsymbol{f}}_{\mathrm{L}} = \boldsymbol{L}(\boldsymbol{w})\boldsymbol{I} = \begin{bmatrix} \boldsymbol{B}_{\mathrm{L}}(\boldsymbol{w}) & \boldsymbol{A}_{\mathrm{L}}(\boldsymbol{w}) \end{bmatrix} \boldsymbol{I} \tag{4.6}$$

4.1.4 DFSM Validation

Previous DFSM studies made limited attempts to validate the DFSM model before using it within optimal control studies. This decision is because adaptive refinement was used, where the DFSM is successively updated to be more accurate around the trajectories identified by the optimal

control study by adding more points. A drawback of this strategy is that the DFSM is sensitive to the initial sampling and could be overfitted to the identified optimal trajectory.

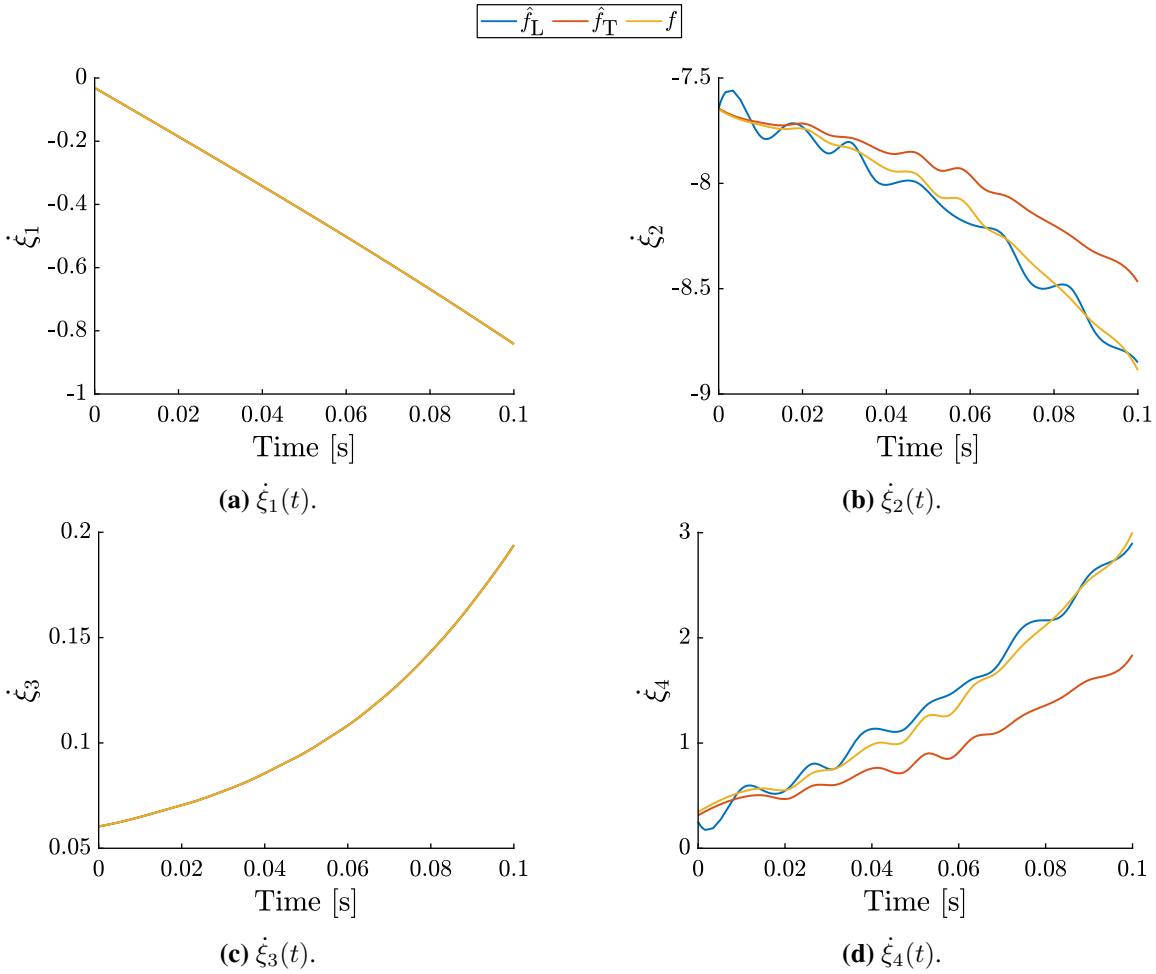
Here, we propose using time-domain simulations to validate the predictive capability of DFSM by comparing the simulation results with the actual model response over a wide range of inputs. To illustrate the validation approach, we again use the two-link robot example described by Eq. (4.1a) to validate the DFSM.

Validating the linear-fit model.

We start by comparing the linear-fit model to the first-order Taylor series approximation of Eq. (4.1a) around the stationary point $\boldsymbol{\xi}_{\text{stat}} = [0.7854, 0, -0.5236, 0]^T$ and $\boldsymbol{u}_{\text{stat}} = [26.1239, 9.4757]^T$, which is the initial point considered in Ref. [148]. This approximation is denoted as $\hat{\boldsymbol{f}}_{\text{T}}$. We obtain $n_{\text{sim}} = 100$ different simulations for the system for perturbations ($\delta = 0.1$) around the expansion point to construct $\hat{\boldsymbol{f}}_{\text{L}}$. Since constructing a linear-fit model is computationally inexpensive, no sub-sampling is performed, and all the training data is used. For example, it takes 0.0068 seconds to construct $\hat{\boldsymbol{f}}_{\text{L}}$ for nearly one million $\langle \boldsymbol{I}, \dot{\boldsymbol{X}} \rangle$ samples.

By comparing the derivatives predicted by these two linear approximations against $\boldsymbol{f}$ and with each other, we can validate the efficacy of using the linear-fit model. The derivatives as predicted by $\hat{\boldsymbol{f}}_{\text{T}}$ and $\hat{\boldsymbol{f}}_{\text{L}}$ for one of the test simulations are shown in Fig. 4.2. From Eq. (4.1a), we can see $\dot{\xi}_1$ and $\dot{\xi}_3$ are linear with respect to the states. From Figs. 4.2a and 4.2c, we can see that $\hat{\boldsymbol{f}}_{\text{T}}$ and $\hat{\boldsymbol{f}}_{\text{L}}$ can capture this relation accurately. If $\hat{\boldsymbol{f}}_{\text{L}}$ can accurately predict some state derivatives, then we do not construct $\boldsymbol{e}$. This insight can lower the time required to construct the DFSM model. Correspondingly, there is some error with the predictions for $\dot{\xi}_2$ and $\dot{\xi}_4$ as a linear model cannot accurately capture the nonlinearities. This error keeps increasing as we move away from the expansion point (as is expected for linear models), and $\boldsymbol{e}$ would be necessary if a more accurate response is desired.

Looking at the eigenvalues of $\boldsymbol{A}_L$ (or the closely-related poles of the transfer function) is another way of comparing $\hat{\boldsymbol{f}}_{\text{L}}$ to $\hat{\boldsymbol{f}}_{\text{T}}$. The poles for $\hat{\boldsymbol{f}}_{\text{T}}$ are at $[\pm 5.2160, \pm 2.6533]$. The eigenvalues from $\hat{\boldsymbol{f}}_{\text{L}}$ are plotted in Fig. 4.3. By sampling around the expansion point, a good approximation



(a) $\dot{\xi}_1(t)$. **(b)** $\dot{\xi}_2(t)$.

(c) $\dot{\xi}_3(t)$. **(d)** $\dot{\xi}_4(t)$.

Figure 4.2: Comparison of the state derivatives predicted by $\hat{\boldsymbol{f}}_L$ and $\hat{\boldsymbol{f}}_T$ to $\boldsymbol{f}$ for a test simulation.

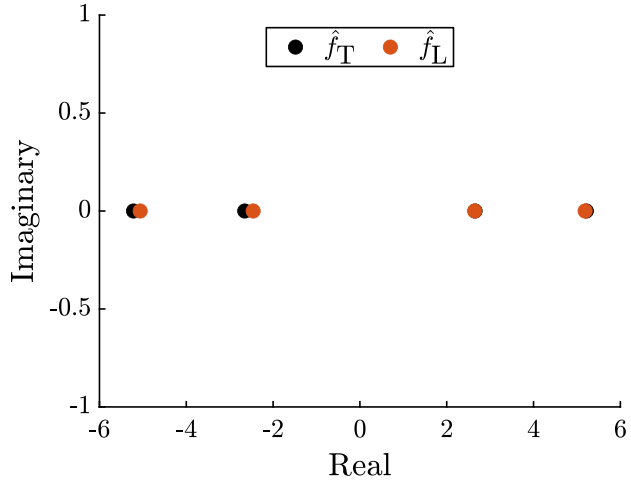


Figure 4.3: Comparisons of the eigenvalues (poles) of $\hat{\boldsymbol{f}}_L$ and $\hat{\boldsymbol{f}}_T$.

of $\hat{\boldsymbol{f}}_{\mathrm{T}}$ is constructed. Considering the ease of constructing the $\hat{\boldsymbol{f}}_{\mathrm{L}}$ model, this insight can be used to approximate the first-order Taylor series expansion. However, care must be taken when approximating certain systems this way. For example, a similar two-link-robot system, discussed in Section 12.4.2 of Ref. [148], has poles at $\boldsymbol{s} = 0$. For such a system, trying to match the poles of $\hat{\boldsymbol{f}}_{\mathrm{L}}$ to $\hat{\boldsymbol{f}}_{\mathrm{T}}$ can be numerically challenging.

4.2 DFSM Construction for Wind and Marine Turbines

4.2.1 Model Structure

The previous study used a relatively simple system to illustrate the steps associated with constructing the DFSM. However, the approach needs to be changed when constructing DFSM for wind and marine turbines. The system response of wind turbines changes over their operating regions from the cut-in to cut-out flow speeds. Therefore, a single LTI model would be insufficient to capture the response of wind turbines over their entire operating region, motivating the need for nonlinear models. However, ensuring the stability of nonlinear models identified through system identification approaches can be difficult. The dynamics and response of wind turbines change over their operating region chiefly as a function of the wind speed. This characteristic of wind turbines has allowed researchers to use linear parameter-varying (LPV) models to construct surrogates [70, 149]. LPV systems are a type of linear time-varying system where the system matrices are continuous functions of certain parameters. Typically, the LPV models for wind systems are constructed using a first-order Taylor series approximation of the underlying system model, with the wind speed being the main parameter. Many articles have explored the construction of LPV models using systems identification principles [150–152]. The approaches used to develop LPV models in these studies rely on limiting assumptions of the system and cannot be generalized to other systems. In this dissertation, a novel yet simple approach is presented for constructing LPV models of wind and marine turbine systems using time series data. This process is shown and validated for a wind turbine system. But the same process can be used to construct a DFSM for a marine turbine as well.

4.2.2 Modeling Quantities

As shown in Figs. 2.3 and 2.7, the three main sets of variables used to construct the low-fidelity models are the primary inputs, controls, and outputs. The system under consideration is the IEA-15 MW turbine with a semisubmersible platform introduced in Refs. [68, 153]. The primary input variables for this system are the wind speed (w) and the wave elevation (η). As mentioned in Chap. 2, the generator torque (τ_g) and the blade pitch (β) are the main control variables for both these turbines. These two sets of variables are combined and represented as:

$$\boldsymbol{u} = [w, \tau_g, \beta, \eta]^{\mathrm{T}} \tag{4.7}$$

where this combined vector contains both kinds of inputs described in Sec. 2.4.4 for simplicity of a single input notation. For both wind and marine turbines, the quantities shown in Eq. (4.7) are used as inputs and controls.

Some key outputs that are considered for both these systems are, the generator speed (ω_g), platform pitch (θ_p), tower base moment in the fore-aft direction ($M_{t,y}$), generator power (P), tower-top rotational acceleration in the fore-aft direction ($\ddot{x}_t$). Of the output quantities listed above, ω_g, P, and $\ddot{x}_t$ are feedback variables that must be provided to the controller. Therefore, to get accurate control responses, the DFSM must capture these variables accurately. Additionally, quantities such as $M_{t,y}$ and P are used to calculate key performance metrics for these systems, such as the damage equivalent load (DEL) and the annual energy production (AEP), respectively. For wind turbines, the platform surge motion (χ_{su}) and the tower-top displacement (δ_{tt}) are also considered. For marine turbines, instead of these quantities, the platform heave (χ_{h}) is considered.

$$\boldsymbol{y}_{\mathrm{WT}} = [\chi_{\mathrm{su}}, \Theta_p, \delta_{tt}, \omega_g, P, M_{t,y}, \ddot{x}_t]^{\mathrm{T}} \tag{4.8a}$$

$$\boldsymbol{y}_{\mathrm{MT}} = [\chi_{\mathrm{h}}, \Theta_p, \omega_g, P, M_{t,y}, \ddot{x}_t]^{\mathrm{T}} \tag{4.8b}$$

All these quantities are time-varying, but this is not shown explicitly for conciseness.

4.2.3 Data Generation and Organization

Once these quantities have been defined, the FOWT model is simulated using the WEIS toolbox available in Ref. [107], using load cases from DLC 1.6. These load cases are primarily categorized by their average wind speed ($\bar{w}$). Since the environmental conditions under which a wind turbine must operate are stochastic, evaluating a single set of load cases is often not sufficient to rigorously characterize the performance of the turbine. Therefore, multiple 'seeds' of load cases are generated to simulate the turbine. Suppose we consider a set of n_w different load cases, each with n_s number of different seeds, the corresponding $\bar{\boldsymbol{w}}$ matrix can be represented as follows to obtain a total of $n_s \times n_w$ simulations:

$$\bar{\boldsymbol{w}} = \begin{bmatrix} \bar{w}_{1,1} & \bar{w}_{1,2} & \dots & \bar{w}_{1,n_w} \\ \bar{w}_{2,1} & \bar{w}_{2,2} & \dots & \bar{w}_{2,n_w} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{w}_{n_s,1} & \bar{w}_{n_s,2} & \dots & \bar{w}_{n_s,n_w} \end{bmatrix} \tag{4.9}$$

The wind turbine is then simulated using OpenFAST for these different load cases to obtain the corresponding combined inputs ($\boldsymbol{U}$) database and outputs ($\boldsymbol{Y}$) database. The sizes of these databases are $(n_u \times n_t)$ and $(n_y \times n_t)$ respectively, where n_u is the number of combined input/control variables in Eq. (4.7), n_y is the number of output variables in Eq. (4.8), and n_t is the number of time points in the simulation. These quantities are then organized as follows:

$$\boldsymbol{\mathcal{U}} = \begin{bmatrix} \boldsymbol{U}_{1,1} & \boldsymbol{U}_{1,2} & \dots & \boldsymbol{U}_{1,n_w} \\ \boldsymbol{U}_{2,1} & \boldsymbol{U}_{2,2} & \dots & \boldsymbol{U}_{2,n_w} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{U}_{n_s,1} & \boldsymbol{U}_{n_s,2} & \dots & \boldsymbol{U}_{n_s,n_w} \end{bmatrix} \quad \boldsymbol{\mathcal{Y}} = \begin{bmatrix} \boldsymbol{Y}_{1,1} & \boldsymbol{Y}_{1,2} & \dots & \boldsymbol{Y}_{1,n_w} \\ \boldsymbol{Y}_{2,1} & \boldsymbol{Y}_{2,2} & \dots & \boldsymbol{Y}_{2,n_w} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{n_s,1} & \boldsymbol{Y}_{n_s,2} & \dots & \boldsymbol{Y}_{n_s,n_w} \end{bmatrix} \tag{4.10}$$

where these matrices correspond to the inputs, controls, and outputs databases shown in Figs. 2.5, 2.6, and 2.7.

4.2.4 Extracting State Derivative Information

In this work, we assume that the selected states are available from the outputs $\boldsymbol{y} \in \mathbb{R}^{n_y}$, and the model does not have any other internal states, such that $\boldsymbol{\xi} \subset \boldsymbol{y}$. The states considered for wind turbines in this study are $[\chi_{su}, \Theta_p, \delta_{tt}, \omega_g]^T$, and their first time derivatives. Similarly, the states considered for marine turbines are $[\chi_{su}, \Theta_p, \delta_{tt}, \omega_g]^T$, and their first time derivatives. The corresponding states and state derivatives are:

$$\boldsymbol{\xi}_{WT} = [\chi_{\text{su}}, \Theta_p, \delta_{tt}, \omega_g, \dot{\chi}_{\text{su}}, \dot{\Theta}_p, \dot{\delta}_{tt}, \dot{\omega}_g]^{\text{T}} \tag{4.11}$$

$$\dot{\boldsymbol{\xi}}_{WT} = [\dot{\chi}_{\text{su}}, \dot{\Theta}_p, \dot{\delta}_{tt}, \dot{\omega}_g, \ddot{\chi}_{\text{su}}, \ddot{\Theta}_p, \ddot{\delta}_{tt}, \ddot{\omega}_g]^{\text{T}} \tag{4.12}$$

$$\boldsymbol{\xi}_{MT} = [\chi_{\text{h}}, \Theta_p, \omega_g, \dot{\chi}_{\text{h}}, \dot{\Theta}_p, \dot{\omega}_g]^{\text{T}} \tag{4.13}$$

$$\dot{\boldsymbol{\xi}}_{MT} = [\dot{\chi}_{\text{h}}, \dot{\Theta}_p, \dot{\omega}_g, \ddot{\chi}_{\text{h}}, \ddot{\Theta}_p, \ddot{\omega}_g]^{\text{T}} \tag{4.14}$$

When the direct evaluation of $\boldsymbol{f}$ is not possible, the state derivative information can be indirectly obtained from the simulated state trajectories, as shown previously. The states and state derivatives are extracted, then organized as follows:

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_{1,1} & \boldsymbol{\xi}_{1,2} & \cdots & \boldsymbol{\xi}_{1,n_w} \\ \boldsymbol{\xi}_{2,1} & \boldsymbol{\xi}_{2,2} & \cdots & \boldsymbol{\xi}_{2,n_w} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\xi}_{n_s,1} & \boldsymbol{\xi}_{n_s,2} & \cdots & \boldsymbol{\xi}_{n_s,n_w} \end{bmatrix} \qquad \dot{\boldsymbol{\Xi}} = \begin{bmatrix} \dot{\boldsymbol{\xi}}_{1,1} & \dot{\boldsymbol{\xi}}_{1,2}, & \cdots & \dot{\boldsymbol{\xi}}_{1,n_w} \\ \dot{\boldsymbol{\xi}}_{2,1} & \dot{\boldsymbol{\xi}}_{2,2} & \cdots & \dot{\boldsymbol{\xi}}_{1,n_w} \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\boldsymbol{\xi}}_{n_s,1} & \dot{\boldsymbol{\xi}}_{n_s,2} & \cdots & \dot{\boldsymbol{\xi}}_{n_s,n_w} \end{bmatrix} \tag{4.15}$$

4.2.5 Model Construction

The goal of this study is to construct a low-fidelity model that can approximate the key states and outputs of wind turbine systems. We first assume a nonlinear state-space model with the

following model structure:

$$\frac{d\boldsymbol{\xi}}{dt} = \dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u}) \approx \boldsymbol{f}_{\text{LF}}(\boldsymbol{\xi}, \boldsymbol{u}) \tag{4.16a}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{\xi}, \boldsymbol{u}) \approx \boldsymbol{g}_{\text{LF}}(\boldsymbol{\xi}, \boldsymbol{u}) \tag{4.16b}$$

We then further assume $\boldsymbol{f}_{\text{LF}}(\cdot)$ and $\boldsymbol{g}_{\text{LF}}(\cdot)$ as a LPV state-space models with the following structure:

$$\dot{\boldsymbol{\xi}} \approx \boldsymbol{f}_{\text{LF}}(\boldsymbol{\xi}, \boldsymbol{u}) = \boldsymbol{A}(w)\boldsymbol{\xi} + \boldsymbol{B}(w)\boldsymbol{u} \tag{4.17a}$$

$$\boldsymbol{y} \approx \boldsymbol{g}_{\text{LF}}(\boldsymbol{\xi}, \boldsymbol{u}) = \boldsymbol{C}(w)\boldsymbol{\xi} + \boldsymbol{D}(w)\boldsymbol{u} \tag{4.17b}$$

A commonly used approach to construct LPV models is to identify the individual state-space matrices $\boldsymbol{\Sigma}_i = (\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{C}_i, \boldsymbol{D}_i)$ corresponding to different values of the system parameter $\bar{w}_i$ where $i \in [1, \ldots, n_w]$, and interpolate over them to obtain a continuous representation as shown in Eq. (4.17). We formulate and solve a set of optimization problems to identify the model parameters associated with the system matrices. Unlike the example shown in the previous section, a least squares approximation is not sufficient to ensure that the system is stable. Therefore, advanced optimization tools must be used for wind and marine turbines. For each value of the parameter $\bar{w}_i$, two different optimization studies are formulated and solved to get the corresponding $\boldsymbol{\Sigma}_i$, first to get the system matrices $(\boldsymbol{A}_i, \boldsymbol{B}_i)$ corresponding to $\boldsymbol{f}_{\text{LF}}$, and the second to get the system matrices $(\boldsymbol{C}_i, \boldsymbol{D}_i)$ corresponding to $\boldsymbol{g}_{\text{LF}}$.

The primary use case for the model constructed in Eq. (4.17) is for time-domain simulations. For the linear system $\boldsymbol{\Sigma}_i$ to be stable when simulated, the real parts of the eigenvalues of the state matrix $\boldsymbol{A}_{\bar{w}_i}$ must be negative. The objective for optimization problem solved to obtain the model parameters corresponding to $(\boldsymbol{A}_i, \boldsymbol{B}_i)$ is to minimize the error between the actual state derivatives $(\dot{\boldsymbol{\xi}}_i)$ and the predicted state derivatives $(\hat{\dot{\boldsymbol{\xi}}}_i)$, and has a constraint to ensure that real parts of the eigenvalues are negative. The optimization problem to identify the model parameters

corresponding to $\boldsymbol{f}_{\text{LF}}$ can be formulated as:

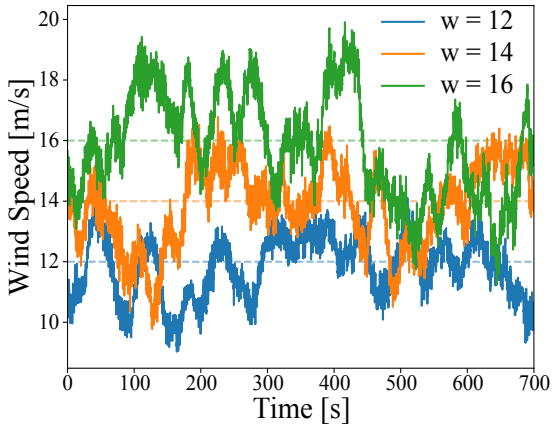
$$\begin{aligned}
\text{changing:} \quad & \boldsymbol{\theta}_{AB,i} = [\boldsymbol{\theta}_{A,i}, \boldsymbol{\theta}_{B,i}] && \text{(4.18a)} \\
\text{minimize:} \quad & ||\dot{\boldsymbol{\xi}}_i - \hat{\dot{\boldsymbol{\xi}}}_i||_2 && \text{(4.18b)} \\
\text{subject to:} \quad & \max(\text{real}(\boldsymbol{\lambda}(\boldsymbol{\theta}_{A,i}))) < 0 && \text{(4.18c)} \\
\text{where:} \quad & \boldsymbol{A}_i = \text{mat}(\boldsymbol{\theta}_{A,i}) && \text{(4.18d)} \\
& \boldsymbol{B}_i = \text{mat}(\boldsymbol{\theta}_{B,i}) && \text{(4.18e)} \\
& \hat{\dot{\boldsymbol{\xi}}}_i = \boldsymbol{A}_i \boldsymbol{\xi}_i + \boldsymbol{B}_i \boldsymbol{u}_i && \text{(4.18f)}
\end{aligned}$$

where $\boldsymbol{\lambda}$ are the eigenvalues of $\boldsymbol{A}_i$, the quantities $(\dot{\boldsymbol{\xi}}_i, \boldsymbol{\xi}_i, \boldsymbol{u}_i)$ are available from the respective columns of the data bases $(\dot{\Xi}, \Xi, \mathcal{U})$ shown in Eqs. (4.10) and (4.15) respectively, and 'mat' in Eqs. (4.18d) and (4.18e) refers to the conversion of a vector into a matrix. A hybrid optimization strategy is used to solve this problem, where a genetic algorithm (GA) is used to obtain a good estimate for $\boldsymbol{\theta}_{AB,i}$. A gradient-based solver is used to find the optimal point $\boldsymbol{\theta}^*_{AB,i}$ starting from the estimate obtained by the GA.

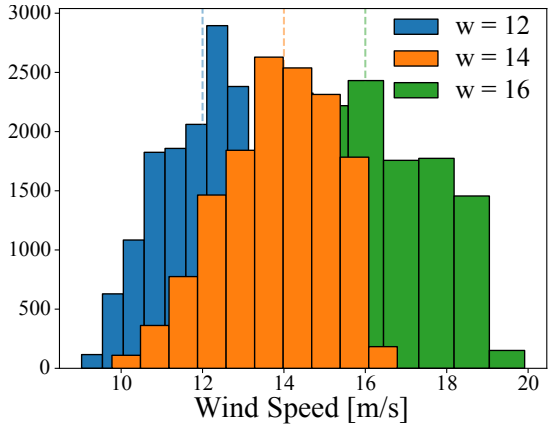
Similarly, the objective for the optimization problem solved to obtain $(\boldsymbol{C}_i, \boldsymbol{D}_i)$ is to minimize the error between actual outputs $(\boldsymbol{y}_i)$ and the predicted outputs $(\hat{\boldsymbol{y}}_i)$. The optimization problem to identify the model parameters corresponding to $\boldsymbol{g}_{\text{LF}}$ can be formulated as:

$$\begin{aligned}
\text{changing:} \quad & \boldsymbol{\theta}_{CD,i} = [\boldsymbol{\theta}_{C,i}, \boldsymbol{\theta}_{D,i}] && \text{(4.19a)} \\
\text{minimize:} \quad & ||\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i||_2^2 && \text{(4.19b)} \\
\text{where:} \quad & \hat{\boldsymbol{y}}_i = \boldsymbol{C}_i \boldsymbol{\xi}_i + \boldsymbol{D}_i \boldsymbol{u}_i && \text{(4.19c)}
\end{aligned}$$

which is a standard linear least-squares estimation problem. These optimization problems are formulated and solved to obtain the complete set of state-space matrices $\boldsymbol{\Sigma}_i$ for different values of w_i.



(a) Wind speed trajectory.



(b) Histogram plot of wind speeds.

Figure 4.4: Wind speed trajectory and histogram plots for three different load cases with $\bar{\boldsymbol{w}} = [12, 14, 16]$ m/s.

Some useful properties of FOWT dynamics can be leveraged to efficiently solve the optimization problem formulated to solve Eq. (4.18). In Fig. 4.4a, three wind speed trajectories for $\bar{\boldsymbol{w}} = [12, 14, 16]$ [m/s] are plotted along with their histograms in Fig. 4.4b. The wind speed trajectory for $\bar{w} = 14$ [m/s] isn't limited to this value, but also contains simulation results where the wind speed is 12 [m/s] and 16 [m/s]. The dynamic response of the FOWT depends heavily on the wind speed. Suppose a state-space model $\boldsymbol{\Sigma}$ is constructed to predict $\dot{\boldsymbol{\xi}}$ and $\boldsymbol{y}$ for a simulation with $\bar{w} = 14$ [m/s]. The model is also trained to be accurate for the parts of the simulation where the wind speed is 12 [m/s] and 16 [m/s]. Therefore, instead of using a GA to identify good starting points for $\boldsymbol{\theta}_{AB,12}$ and $\boldsymbol{\theta}_{AB,16}$, the optimal point $\boldsymbol{\theta}^*_{AB,14}$ can be provided as a feasible starting point to the gradient-based solver to obtain $\boldsymbol{\theta}^*_{AB,12}$ and $\boldsymbol{\theta}^*_{AB,16}$. Subsequently, $\boldsymbol{\theta}^*_{AB,16}$ is used as the starting point for $\boldsymbol{\theta}^*_{AB,18}$, and $\boldsymbol{\theta}^*_{AB,12}$ is used as the starting point for $\boldsymbol{\theta}^*_{AB,10}$, and so on. An interior-point algorithm is used as the gradient-based solver to leverage this property further. The DFSM approach used in this chapter is available at the following Ref. [154].

4.3 Data-Driven Model Results for Wind Turbines

The approach outlined in the previous section is used to construct the DFSM for the FOWT system. All the OpenFAST simulations and low-fidelity models used in this work are constructed

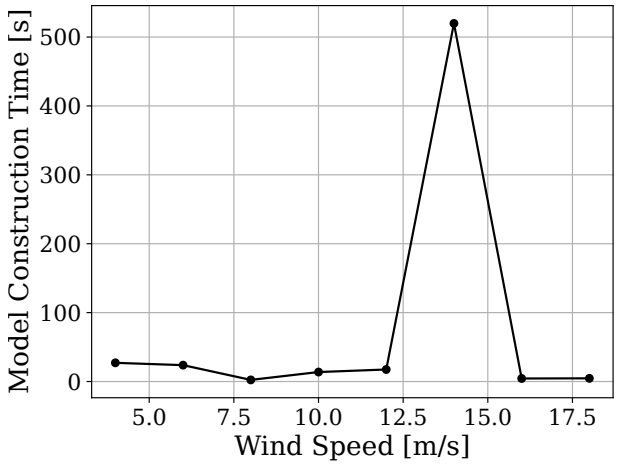


Figure 4.5: Model construction time.

and simulated using a desktop workstation with an AMD 3970X CPU with 32 cores and 128 GB of RAM, with an NVIDIA GeForce GTX 1650 GPU with 4 GB of VRAM.

A total of $n_s = 10$ different simulations are obtained for each wind speed. Of these, $n_{\text{train}} = 5$ seeds are used to train the model, and the rest are used for validation. The model construction time, along with the set of wind speeds considered for these studies, are shown in Fig. 4.5. The model construction process starts from $\bar{w} = 14$ [m/s]. A hybrid optimization scheme is used to identify the model parameters for the $\mathbf{\Sigma}_i$ corresponding to this $\bar{w}_i$. From Fig. 4.5, it can be seen that the time taken to solve Eq. (4.17) for this wind speed is 519 seconds. However, using the approach outlined previously, the solution time for the succeeding wind speeds is significantly reduced, as the parameters identified for $\bar{w} = 14$ [m/s] are used as the starting point for $\bar{w} = 12$ and $\bar{w} = 16$ [m/s]. For example, the construction time for these two succeeding wind speeds is 7 seconds and 3.5 seconds, respectively.

We would also like to compare the performance of the low-fidelity modeling approach proposed here to low-fidelity models developed using other approaches in the literature, like subspace state-space system identification (n4sid) and models built using LSTM networks. Using both these alternative approaches results in a model with discrete dynamics.

4.3.1 Subspace State-Space System Identification

We utilize the n4sid approach to estimate a discrete-time linear shift-invariant system model using input/output time series data with the following structure:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_{\mathrm{d}}\boldsymbol{x}_k + \boldsymbol{B}_{\mathrm{d}}\boldsymbol{u}_k \tag{4.20a}$$

$$\boldsymbol{y}_k = \boldsymbol{C}_{\mathrm{d}}\boldsymbol{x}_k \tag{4.20b}$$

where the subscript 'd' denotes a discrete system model. We use the systems identification toolbox in `MATLAB` to identify this model. In addition to the input/output time series data, the model order n_x must be provided by the user. This selection is identified through a sensitivity study, where we vary n_x for different values and evaluate the response of the identified model. We test the model response for stability and accuracy. Through this process, a model with $n_x = 6$ was identified to predict the system response accurately and provide a stable response. For this system, it is important to use a subspace identification algorithm that can use input/output data obtained using closed-loop simulations [131, 155]. For this reason, we use the 'SSARX' algorithm, introduced in Ref. [131], to identify the model in Eq. (4.20).

4.3.2 Long Short-Term Memory Networks

In parallel, we train an LSTM network to predict the outputs given the inputs at a timestep:

$$\boldsymbol{y}_k = \boldsymbol{f}_{\mathrm{LSTM}}(\boldsymbol{u}_k) \tag{4.21}$$

The network model contains an LSTM layer with 10 LSTM cells and a fully connected layer, and the number of epochs used to train the model is 10. We use the implementation in the `tensorflow` [156] package to construct and train the model.

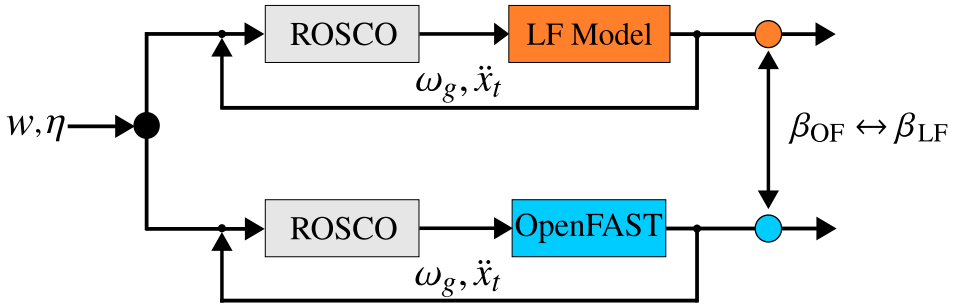


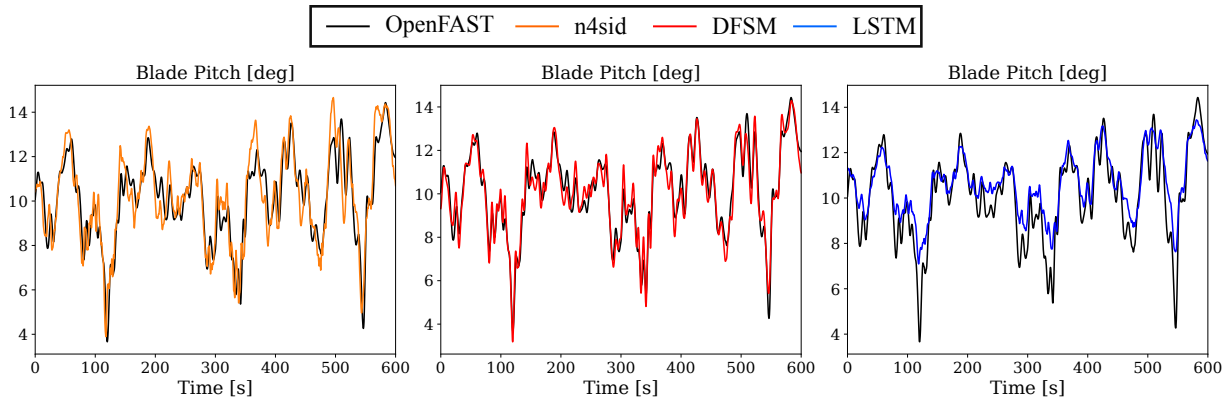
Figure 4.6: Closed-loop validation test. The low-fidelity models and OpenFAST are used as 'plants' in a closed-loop system to provide the main feedback variables of generator speed (ω_g) and tower-top acceleration ($\ddot{x}_t$). For the same wind speed (w) and wave elevation (η) trajectories, the resulting blade pitch (β) signals are compared.

4.4 Closed-Loop Simulation Validation Results

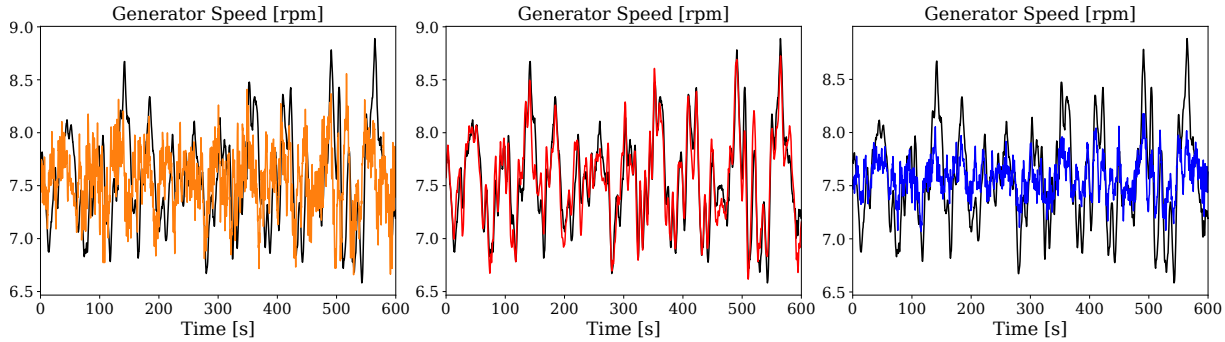
To validate the models constructed in this section, we carry out the closed-loop validation test as shown in Fig. 4.6. The low-fidelity models are used as the 'plant' in the closed-loop system and provide the key feedback variables, namely $\boldsymbol{\omega}_g$ and $\ddot{\boldsymbol{x}}_t$, to the ROSCO controller, which generates the corresponding values for τ_g and β. The resulting control signals obtained using the low-fidelity models are then compared to OpenFAST, which is the high-fidelity model.

We use the process outlined previously to construct the DFSM. We use the same dataset to train the n4sid and LSTM models. Five different seeds are used for training these models. We train the n4sid and LSTM models on simulations with $\bar{w} = 14$ [m/s]. To obtain accurate predictions with the LSTM model, simulations with $\bar{\boldsymbol{w}} = [12, 16]$ [m/s] were included as part of the training data. We utilize the GPU in the desktop workstation for training and evaluating the LSTM model. The time taken to obtain the training data, in terms of CPU hours, is $5\ (n_s) \times 7\ (n_w) \times 0.33$ hours $= 11.6$ hours. Once the data is available, it takes 613 seconds to train the DFSM, 52 seconds to obtain the n4sid model, and 519 seconds to train the LSTM model. Once trained, we simulate all three models for ten different test cases with $\bar{w} = 14$ [m/s].

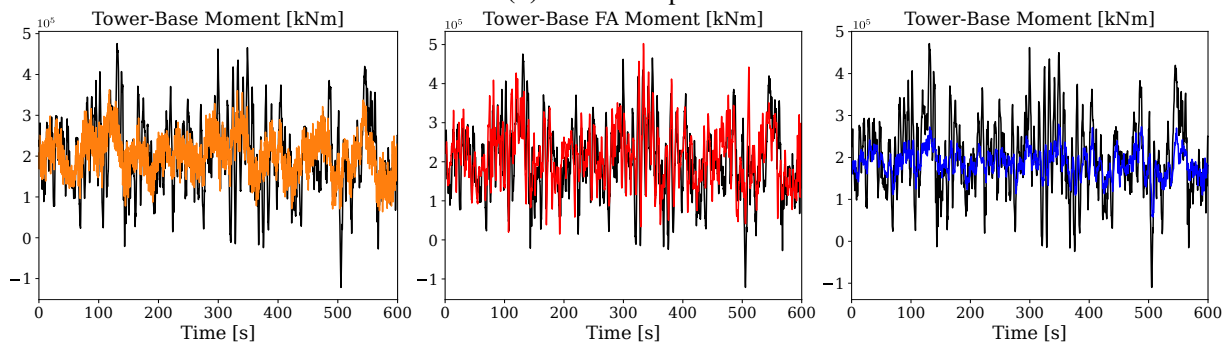
The blade pitch, generator speed, tower-base moment, and platform pitch signals obtained using all three models for one of the test load cases are presented in Fig. 4.7. It is clear from Fig. 4.7 that the accuracy varies between all three low-fidelity models for the three quantities shown. Additionally, in Figs. 4.8, the box plot for the MSE between the OpenFAST and DFSM



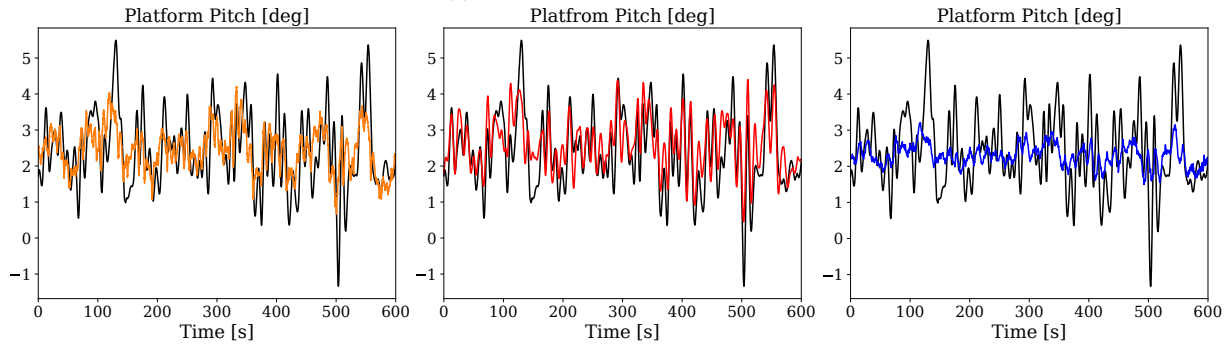
(a) Blade pitch.



(b) Generator speed.



(c) Tower-base fore-aft moment.



(d) Platform pitch.

Figure 4.7: Comparison of the blade pitch (β), generator speed (ω_g), tower-base moment ($M_{y,t}$), platform pitch (Θ_p) response obtained using the three different low-fidelity models and OpenFAST as 'plants' in a closed-loop simulation.

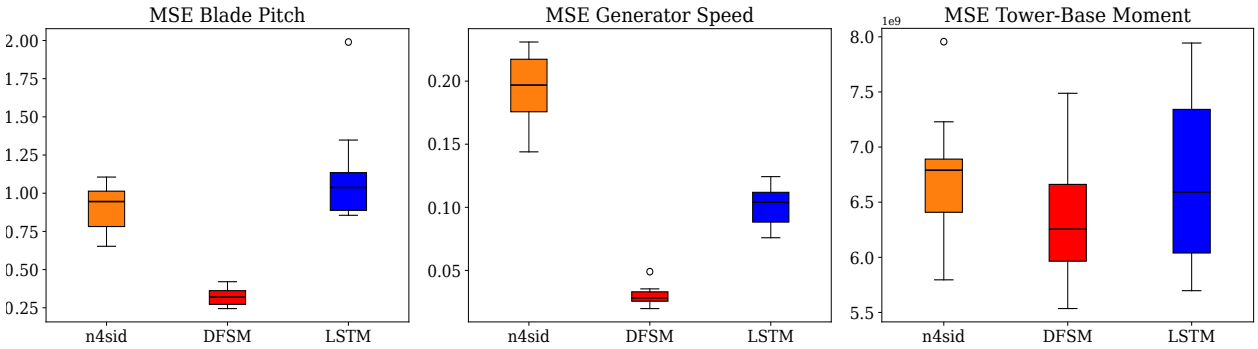


Figure 4.8: Box plots of the MSE for the three quantities based on the low-fidelity model used.

simulations is shown for the three quantities discussed previously, using the three low-fidelity models. The average simulation time using the n4sid model is around 3 seconds, the average simulation time using the DFSM is 35 seconds, and the average simulation time using the LSTM model is around 65 seconds. The average simulation time using OpenFAST is nearly 15 minutes. All three low-fidelity modeling approaches are significantly faster to simulate than OpenFAST. The n4sid model is the least expensive in terms of simulation time. Conversely, the LSTM model has the highest average simulation time, but the variance in the response is high. In contrast, the DFSM achieves a good tradeoff between simulation time, model accuracy, and response variance compared to the n4sid and LSTM models. This can be seen in Fig. 4.7.

In theory, it should be possible to construct an LPV model using the n4sid approach, similar to the DFSM approach. However, the choice of n_x that results in an accurate model response differs for different wind speeds. This makes it hard to obtain an LPV model using the n4sid approaches. Some additional properties of subspace identification methods make them unsuitable for the task at hand. The system states ($\boldsymbol{x}$) shown in Eq. (4.20) and identified by the subspace identification methods are mathematical quantities and do not have any physical meaning. However, engineers know a priori that certain quantities like ω_g and θ_p are system states in physics-based models based on first principles. However, using the n4sid approach, it is not straightforward to model these quantities as the system states, and it might be preferred in the model. Unlike n4sid models, training a single LSTM network to predict the response for multiple wind speeds is straightforward. However, using an LSTM network for iterative predictions can be inefficient. In the study outlined

previously, the n4sid model and DFSM are simulated with a timestep of dt $= 0.01$ [s], while the timestep used for the LSTM model is dt $= 0.5$ [s]. Even with a larger dt, simulating the system using the LSTM model takes significantly longer. Additionally, it can be hard to guarantee the stability of the LSTM model.

The key takeaway from this chapter is that the DFSM approach can be used to model the dynamic response of wind turbines. We present an approach to construct the DFSM using a hybrid optimization approach that can guarantee that the system response is stable. Once constructed, the DFSM provides more accurate simulations when compared to LSTM networks and system-identification models.

Chapter 5

Case Studies

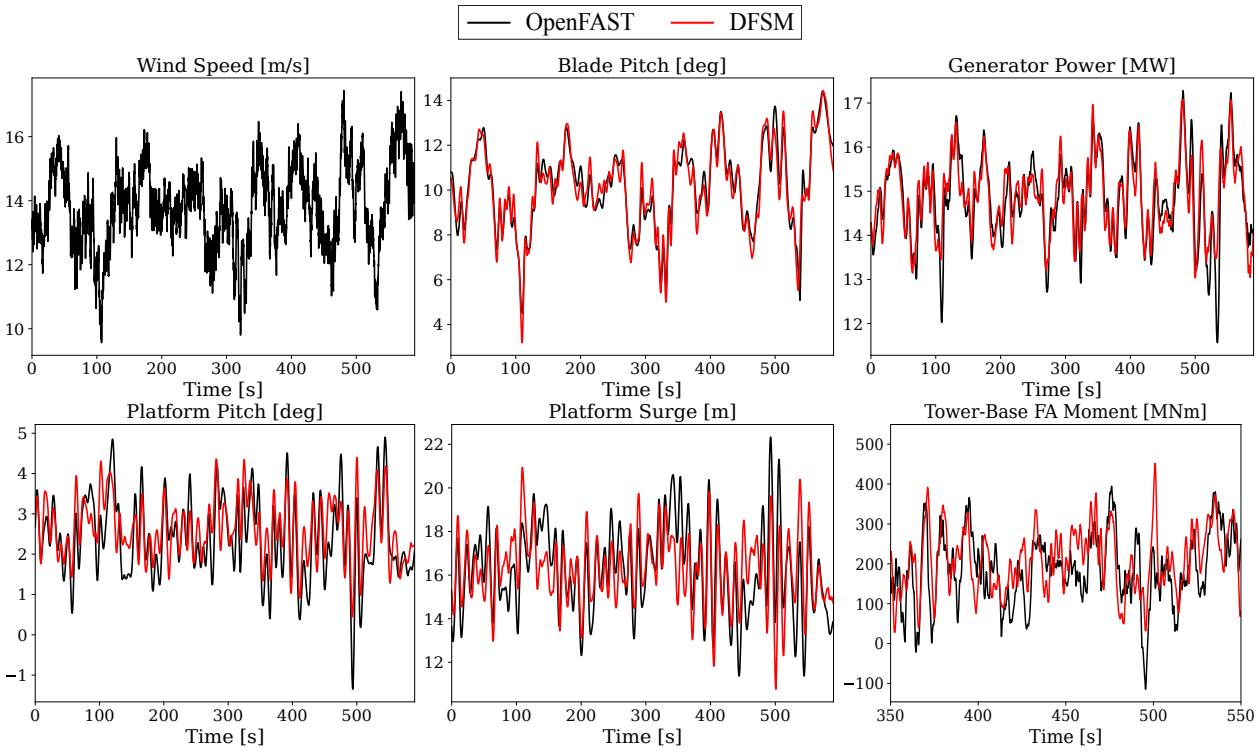
The DFSM approach presented in the previous chapters is used to carry out different case studies in this chapter.[5] Five different case studies are presented.

1. In the first case study, presented in Sec. 5.1, the DFSM approach is used to construct low-fidelity models for five different systems and to estimate their performance for different load cases.
2. The second case study, discussed in Sec. 5.2, involves using the DFSM to perform open-loop optimal control studies for both wind and marine turbines.
3. The third case study, discussed in Sec. 5.3, involves utilizing the DFSM for a multi-fidelity optimization study of the closed-loop control parameters.
4. In the fourth study, presented in Sec. 5.4, the DFSM is used to carry out a multi-objective closed-loop control design study.
5. Finally, in Sec. 5.5, the approach presented in Sec. 5.4 is extended to investigate the multi-objective optimization problem from a CCD perspective.

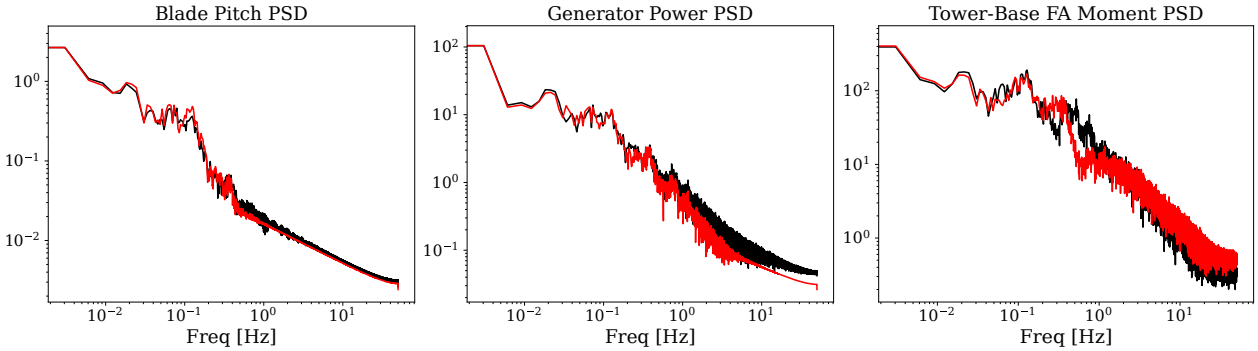
5.1 Predicting System Performance

A primary use case for the DFSM approach presented in the previous chapters is to carry out closed-loop control simulations to estimate the performance of different systems. In addition to design optimization studies, these simulations are required for a number of other applications, primarily to estimate system performance. As seen previously in Chap. 3, simulation times for certain systems can be several hours. Utilizing the DFSM for these systems could help reduce the computational cost associated with them. It is also crucial to show that the DFSM approach can be utilized for different types of systems across multiple wind/current speeds.

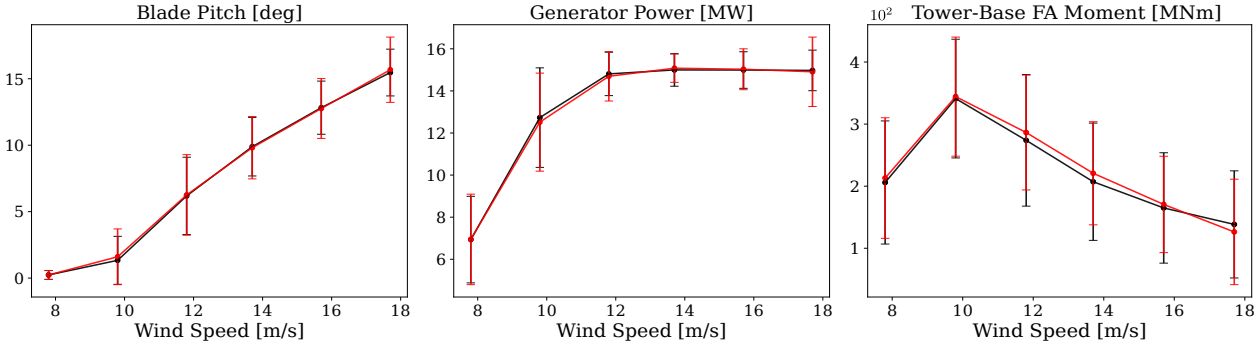
[5] Parts of this chapter are based on Refs. [90, 99].



(a) Timeseries comparison.

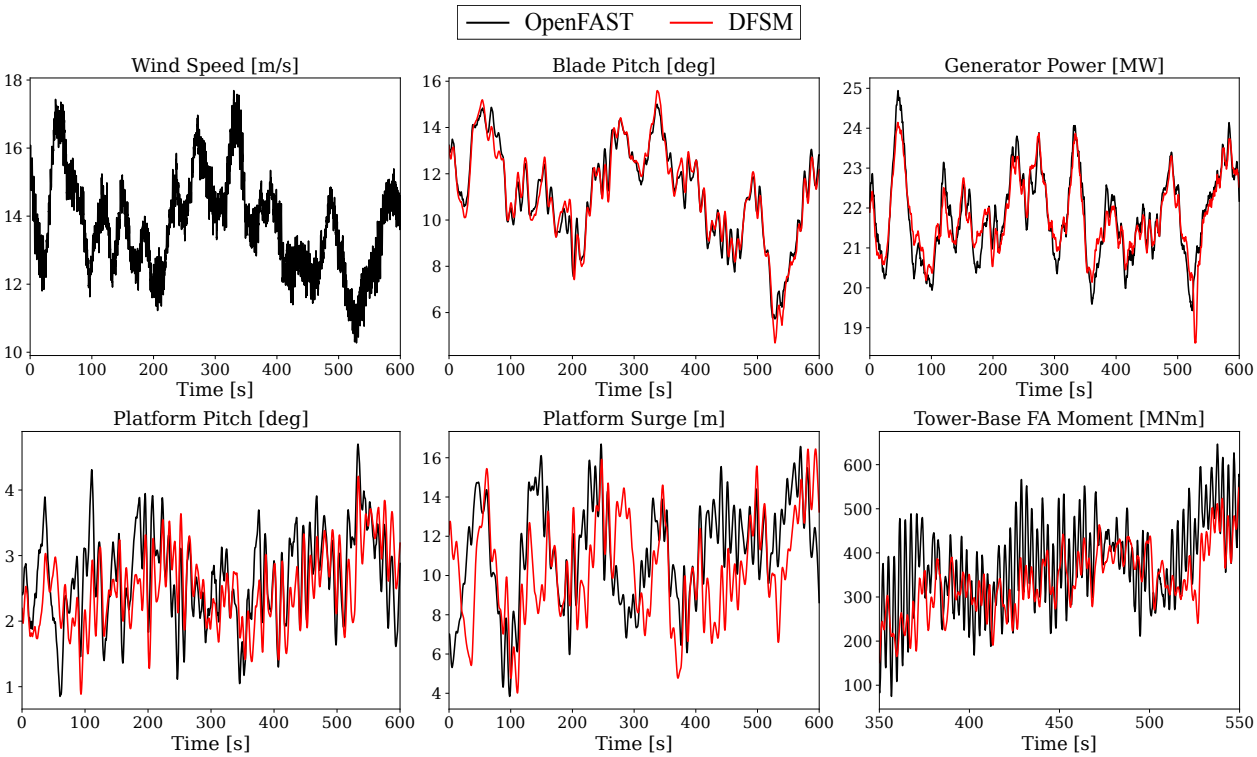


(b) Power spectral density (PSD) for the signals presented in Fig. 5.1a.

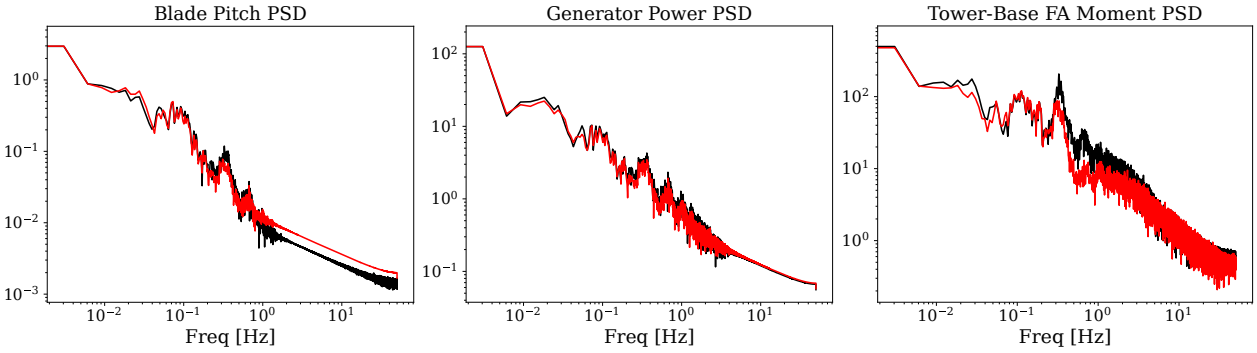


(c) Average mean and standard deviation between DFSM and OpenFAST calculated for all the different load cases.

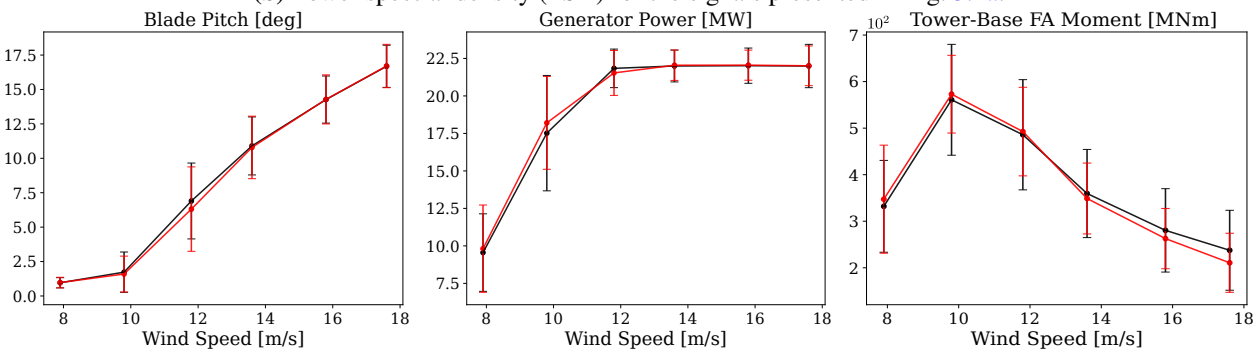
Figure 5.1: Comparison of the blade pitch (β), generator power (ω_g), and tower-base moment ($M_{y,t}$) response obtained using the DFSM and OpenFAST for the IEA-15 MW turbine.



(a) Timeseries comparison.

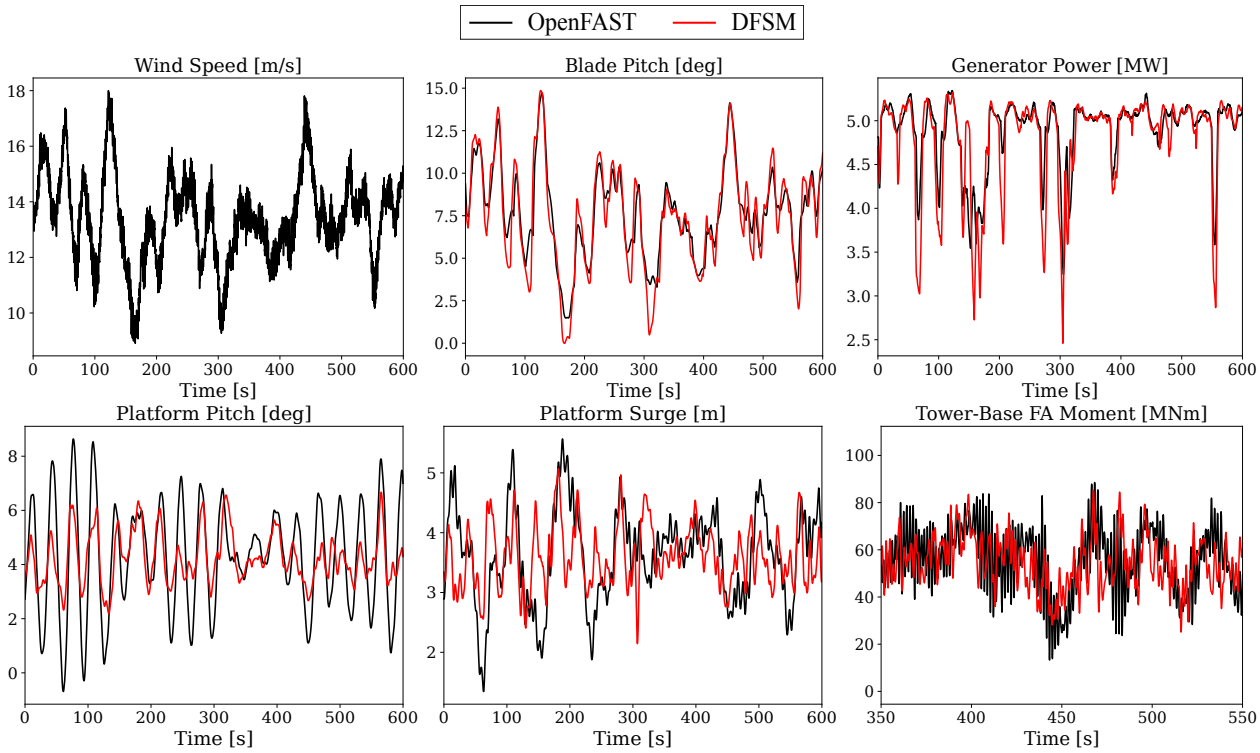


(b) Power spectral density (PSD) for the signals presented in Fig. 5.2a.

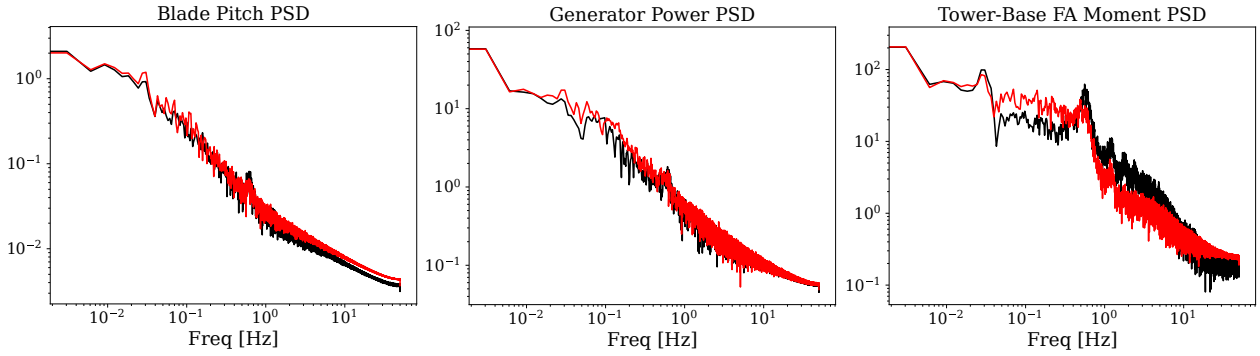


(c) Average mean and standard deviation between DFSM and OpenFAST calculated for all the different load cases.

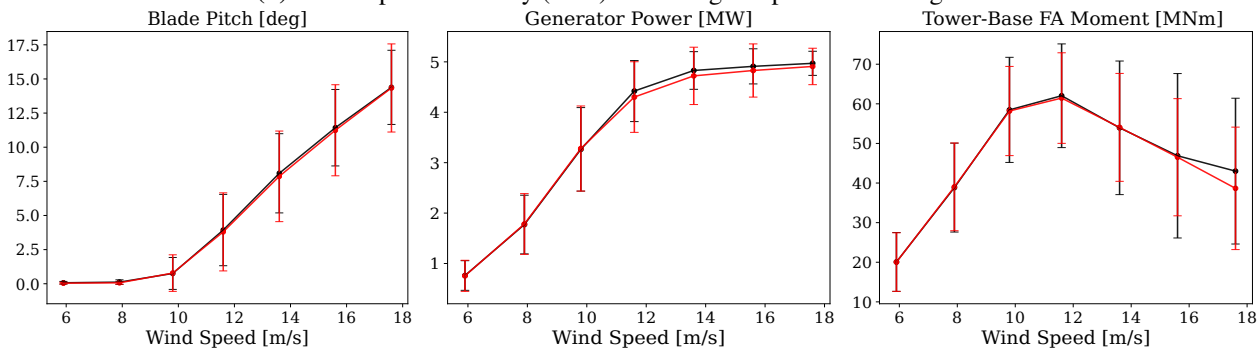
Figure 5.2: Comparison of the blade pitch (β), generator power (ω_g), and tower-base moment ($M_{y,t}$) response obtained using the DFSM and OpenFAST for the IEA-22 MW turbine.



(a) Timeseries comparison.

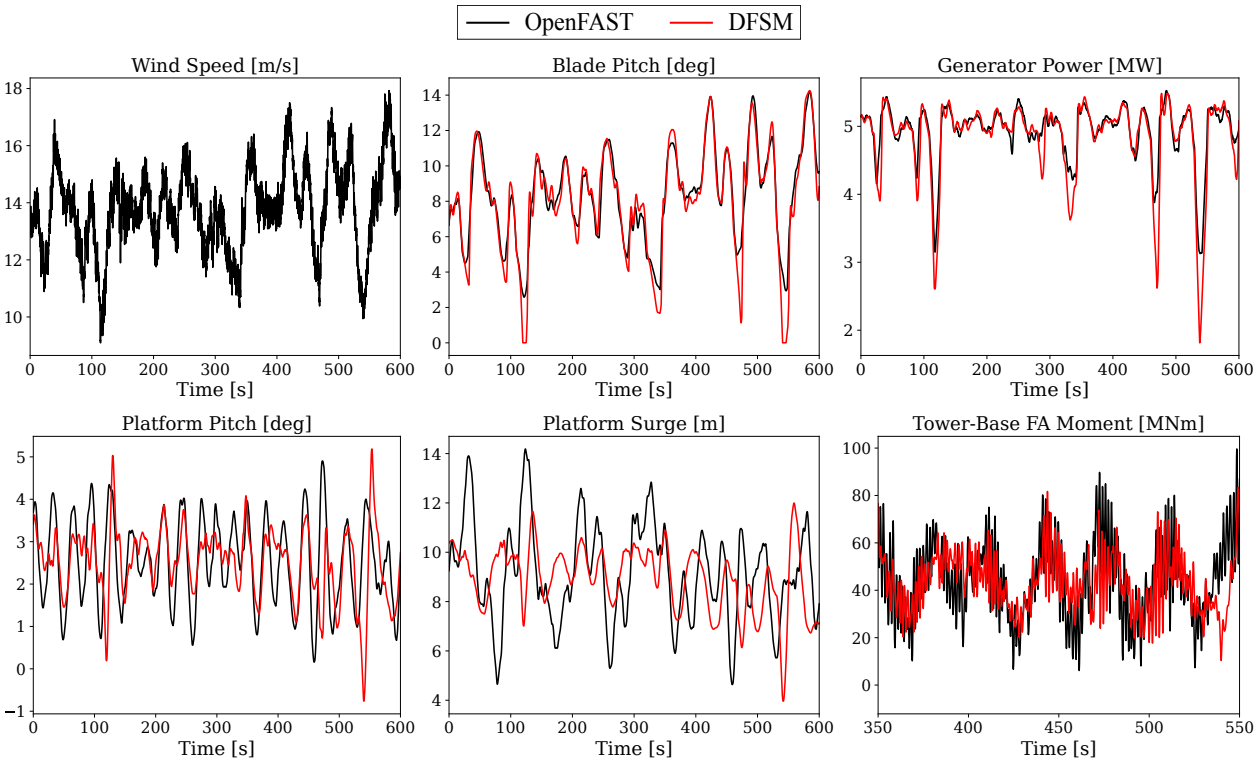


(b) Power spectral density (PSD) for the signals presented in Fig. 5.3a.

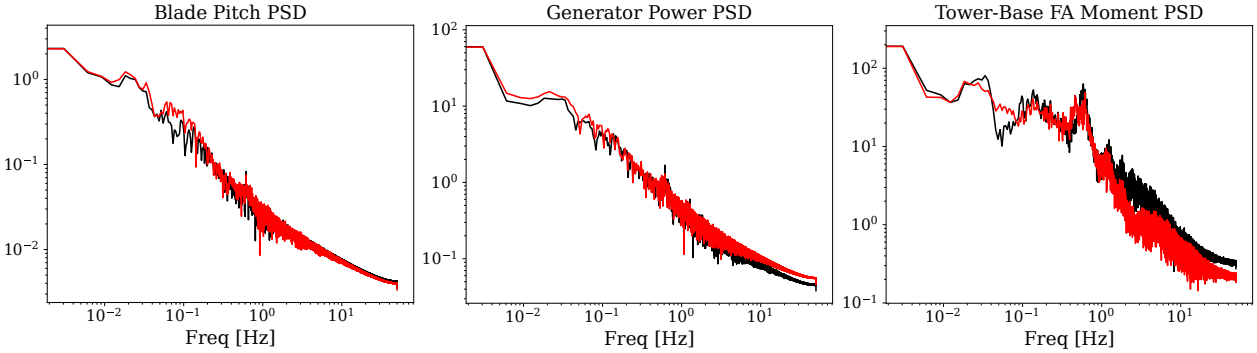


(c) Average mean and standard deviation between DFSM and OpenFAST calculated for all the different load cases.

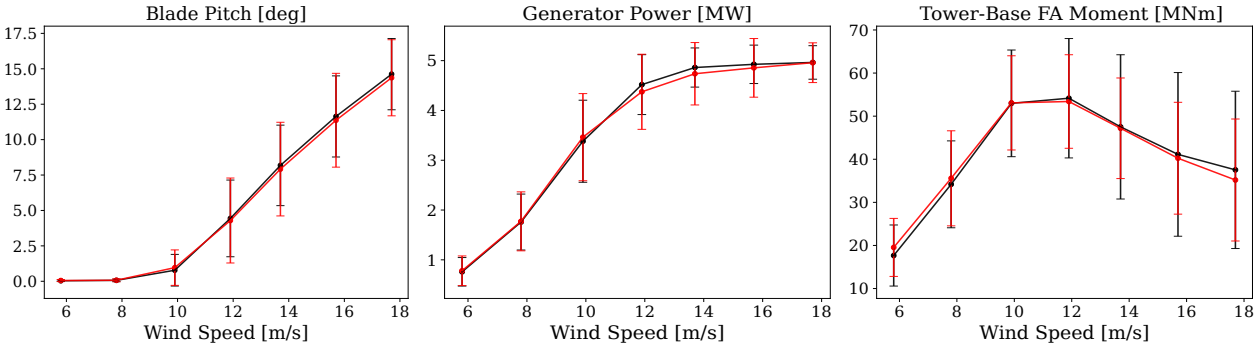
Figure 5.3: Comparison of the blade pitch (β), generator power (ω_g), and tower-base moment ($M_{y,t}$) response obtained using the DFSM and OpenFAST for the NREL 5 MW turbine with a semisubmersible platform.



(a) Timeseries comparison.

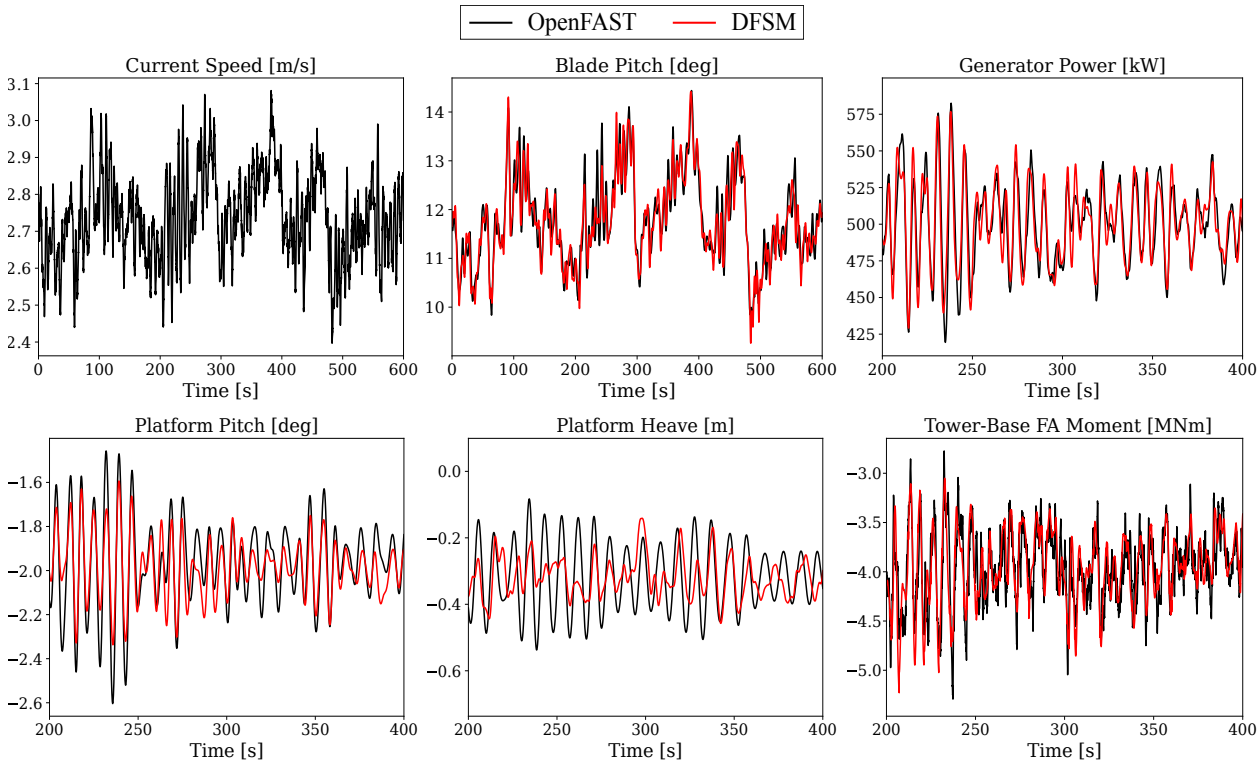


(b) Power spectral density (PSD) for the signals presented in Fig. 5.4a.

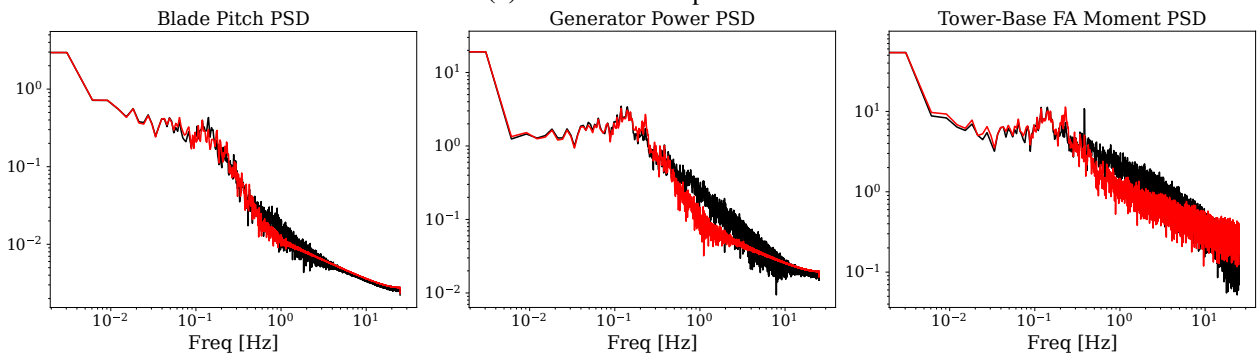


(c) Average mean and standard deviation between DFSM and OpenFAST calculated for all the different load cases.

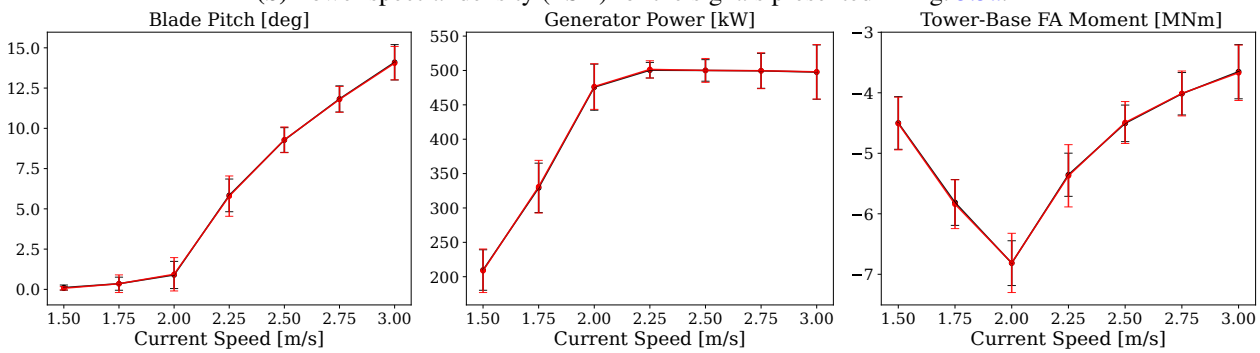
Figure 5.4: Comparison of the blade pitch (β), generator power (ω_g), and tower-base moment ($M_{y,t}$) response obtained using the DFSM and OpenFAST as for the NREL 5 MW turbine with a spar platform.



(a) Timeseries comparison.



(b) Power spectral density (PSD) for the signals presented in Fig. 5.5a.



(c) Average mean and standard deviation between DFSM and OpenFAST calculated for all the different load cases.

Figure 5.5: Comparison of the blade pitch (β), generator power (ω_g), and tower-base moment ($M_{y,t}$) response obtained using the DFSM and OpenFAST for the RM1-Quad marine turbine.

This use case is similar to the validation study discussed in the previous chapter, but instead of simulating the system for a single wind speed, we simulate five different systems for multiple wind speeds between their cut-in and cut-out speeds. These are the IEA-15 MW turbine with a semisubmersible platform, the IEA-22 MW turbine with a semisubmersible platform [69, 74], the NREL 5 MW turbine [66, 157] with a semisubmersible platform, the NREL 5 MW turbine with a spar platform [158], and the RM1-Quad marine turbine. For the sake of conciseness, a set of time series plots of key signals such as $\beta, P, M_{t,y}, \Theta_p, \chi_{su}$ and their corresponding power spectral density (PSD) plots are presented, in addition to the mean and standard deviation of the OpenFAST and DFSM simulations for different wind/current speeds.

5.1.1 IEA-15 MW Turbine with a Semisubmersible Platform

The first system of interest is the IEA-15 MW semisubmersible system. This system is simulated using load cases from DLC 1.6 with $\boldsymbol{w}_{\text{avg}} = [8, \cdots, 16, 18]$. Ten different seeds are used for each value of $\boldsymbol{w}_{\text{avg}}$. The system states, inputs, and outputs shown in Eq. (4.8a) are used. For the same set of wind speeds and wave elevations, both the DFSM and OpenFAST models are simulated, and the key controls and outputs are compared. It takes nearly 15 minutes to simulate this system using OpenFAST, whereas it takes just 35 seconds to simulate each load case using the DFSM.

Figure 5.1 shows the different results obtained using the DFSM and OpenFAST. Figure 5.1a shows the time series plots for these key quantities. Similar to the results seen previously, the DFSM can accurately predict the responses for β and P. From the results of these two quantities, we can infer that ω_g and τ_g are also accurately captured. In addition to these quantities, we see that Θ_p and χ_{su} are also well captured, though not to the same degree as β and P. However, we see that the LPV-based DFSM cannot capture the response for the nonlinear quantity $M_{t,y}$ with the same degree of accuracy. Further evidence for this trend can be seen in the corresponding plots for these quantities in Fig. 5.1b. We see that in the PSD plots for β and P, the key frequencies appear; however, for $M_{t,y}$, the DFSM is unable to capture all key frequencies.

Figure 5.1c shows the mean and standard deviation calculated for all ten seeds for each wind speed used in this study. For quantities like β and P, the mean and standard deviation are accurately captured by the DFSM for all considered load cases. Meanwhile, there is a discrepancy for $M_{t,y}$. From these plots, some general trends can be inferred about the performance metrics calculated using these quantities. For example, the annual energy production (AEP) calculated using the DFSM will be closer to the value obtained using OpenFAST, while the DEL_t will differ. As seen previously, the amplitude of the individual load cycles affects the DEL_t calculation. If these load cycles are not captured accurately, then the resulting DEL_t value will differ.

5.1.2 IEA-22 MW Turbine with a Semisubmersible Platform

The second system of interest is the IEA-22 MW semisubmersible system. This system is a scaled-up version of the IEA-15 MW system discussed in Sec. 5.1.1. The system is simulated using load cases from DLC 1.6 with $\boldsymbol{w}_{\text{avg}} = [8, 10, \cdots, 16, 18]$. Ten different seeds are used for each value of $\boldsymbol{w}_{\text{avg}}$. The same system states, inputs, and outputs shown in Eqs. (4.8a) are used. For the same set of wind speeds and wave elevations, both the DFSM and OpenFAST models are simulated, and the key inputs and outputs are compared. It takes nearly 20 minutes to simulate this system using OpenFAST, whereas it takes just 35 seconds to simulate each load case using the DFSM.

Figure 5.2 shows the different results obtained using the DFSM and OpenFAST. We see similar trends and results to those observed for the IEA-15 MW turbine. Quantities like β and P are captured accurately, but $M_{t,y}$ is not captured to the same degree of accuracy. In addition to these quantities, there are also larger errors in χ_{su} as predicted by the DFSM. Similar trends are seen in the PSD plots and in the plots for the mean and standard deviation of these quantities, as observed for the IEA-15 MW system.

5.1.3 NREL 5MW Turbine with a Semisubmersible Platform

The third system of interest is the NREL 5 MW turbine with a semisubmersible platform. The system is simulated using load cases from DLC 1.6 with $\boldsymbol{w}_{\text{avg}} = [6, 8, \cdots, 16, 18]$. Ten different

seeds are used for each value of $\boldsymbol{w}_{\text{avg}}$. The same system states, inputs, and outputs shown in Eq. (4.8a) are used. For the same set of wind speeds and wave elevations, both the DFSM and OpenFAST models are simulated, and the key inputs and outputs are compared. It takes nearly 7 minutes to simulate this system using OpenFAST, whereas it takes just 35 seconds to simulate each load case using the DFSM.

Figure 5.3 shows the different results obtained using the DFSM and OpenFAST. We see similar trends and results to those observed for the IEA-15 MW and IEA-22 MW turbines for quantities like β, P, and $M_{t,y}$. While β and P are captured accurately, the DFSM cannot capture $M_{t,y}$ with the same degree of accuracy, although it reproduces the overall trends. Unlike the previous two systems, the floating quantities such as Θ_p and χ_{su} are not captured accurately by the DFSM for this system.

5.1.4 NREL 5MW Turbine with a Spar Platform

The fourth system of interest is the NREL 5 MW turbine with a spar platform. This system has a different platform configuration and therefore exhibits different dynamics compared to the systems discussed previously in Secs. 5.1.1, 5.1.2, and 5.1.3, due to the shape of its platform. The system is simulated using load cases from DLC 1.6 with $\boldsymbol{w}_{\text{avg}} = [6, 8, \cdots, 16, 18]$. Ten different seeds are used for each value of $\boldsymbol{w}_{\text{avg}}$. The same system states, inputs, and outputs shown in Eq. (4.8a) are used. For the same set of wind speeds and wave elevations, both the DFSM and OpenFAST models are simulated, and the key inputs and outputs are compared. It takes nearly 7 minutes to simulate this system using OpenFAST, whereas it takes just 35 seconds to simulate each load case using the DFSM.

Figure 5.4 shows the different results obtained using the DFSM and OpenFAST. We observe similar trends and results to those seen for the IEA-15 MW and IEA-22 MW turbines. Quantities such as β and P are captured accurately, while $M_{t,y}$ is not captured to the same degree of accuracy, although the overall trends are preserved. For χ_{su}, however, neither the trend nor the signal is captured accurately by the DFSM.

5.1.5 RM1-Quad Marine Turbine with a Semisubmersible Platform

The fifth system of interest is the RM1-Quad marine turbine discussed in Chap. 3. This system exhibits different dynamics compared to the systems discussed previously in Secs. 5.1.1 and 5.1.2. The system is simulated using load cases from DLC 1.1 with $\boldsymbol{w}_{\text{avg}} = [1.5, 1.75, \cdots, 2.75, 3]$. Six different seeds are used for each value of $\boldsymbol{w}_{\text{avg}}$. The same system inputs shown in Eq. (4.7) are used; however, the states and outputs shown in Eq. (4.8b) are used.

For the same set of current speeds and wave elevations, both the DFSM and OpenFAST models are simulated, and the key inputs and outputs are compared. Figure 5.5 shows the different results obtained using the DFSM and OpenFAST. Similar to the results for the wind turbines, β and P are well captured for this system. Unlike the wind turbines, the trends and the signal for $M_{t,y}$ are also well captured. Additionally, the computational speedup achieved for this system is significantly greater than for the wind turbines discussed previously. It takes nearly 7 hours to simulate this system using OpenFAST, whereas it takes just 35 seconds to simulate each load case using the DFSM.

These results show that the DFSM can be used to predict the response of different wind and marine turbine systems. Additionally, it can be seen that quantities such as β and P are well captured, whereas more nonlinear quantities like $M_{t,y}$ are not captured to the same degree of accuracy. Nevertheless, the DFSM can still be used to predict the mean trends in these signals effectively.

5.1.6 Corrective Functions

The main goal behind the DFSM approach is to obtain a model that balances accuracy and computational cost compared to aero-servo-elastic simulations, while being applicable to a wide variety of use cases. As seen in the preceding section, the DFSM approach can be used to construct low-fidelity models that accurately capture important performance metrics across different systems. However, some use cases that utilize OpenFAST simulations require a higher degree of accuracy in specific signals [159, 160]. To enable the use of the DFSM approach for these applications, it is necessary to improve the accuracy of certain signals predicted by the DFSM.

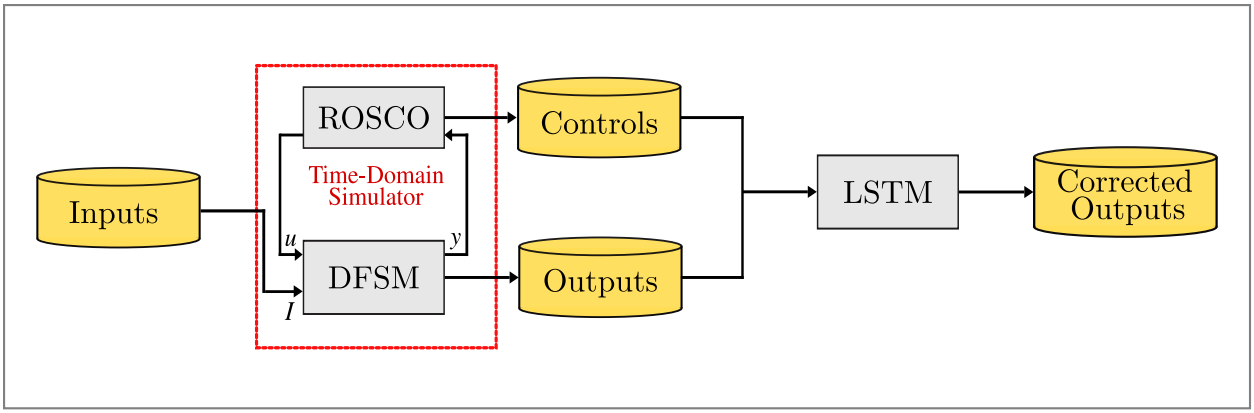


Figure 5.6: Corrective function used to improve the accuracy of certain signals predicted by the DFSM.

To address this, we propose augmenting the DFSM response with a nonlinear corrective function. Based on the discussion in the previous chapter, we employ an LSTM network to construct this corrective function. In the previous study, the LSTM-based low-fidelity model was used in an iterative prediction scheme. From the results of that study, it is evident that this was an inefficient use case for LSTMs. In this study, however, we utilize the LSTM for batch prediction. The DFSM is first simulated as described previously, and the results of this simulation are then corrected using the LSTM. This process is illustrated in Fig. 5.6.

Model quantities and model constructions.

We investigate the efficacy of this approach to improve the accuracy of the low-fidelity model predictions for two systems. The first system is the IEA-22 MW turbine with a semisubmersible platform, and the second system is the RM1-Quad marine turbine. For the IEA-22 MW system, we focus on improving χ_{su}, whereas for the marine turbine system, we focus on improving the $M_{t,y}$ signal. We use the same set of training and testing data that was used to construct and validate the DFSM. We use the simulated data obtained using the closed-loop simulations for both these test sets. Using the training dataset, we train an LSTM network to predict the high-fidelity χ_{su} given $(w, \eta, \chi_{\text{su,DFSM}})$. Similarly, for the second system, we train a network to predict the high-fidelity $M_{t,y}$ given $(w, \eta, M_{t,y})$ predicted by the DFSM. From Fig. 5.5c, it can be seen clearly that the DFSM prediction is not accurate for this signal at current speeds near the rated region. For both

systems, a network consisting of a single LSTM layer with 64 cells and a dense layer is used. The network is trained for 20 epochs for the IEA-22 MW system and for 10 epochs for the marine turbine. For both systems, it takes an average of 300 seconds per epoch to train the networks. Once trained, it takes under a second to obtain the corrected outputs.

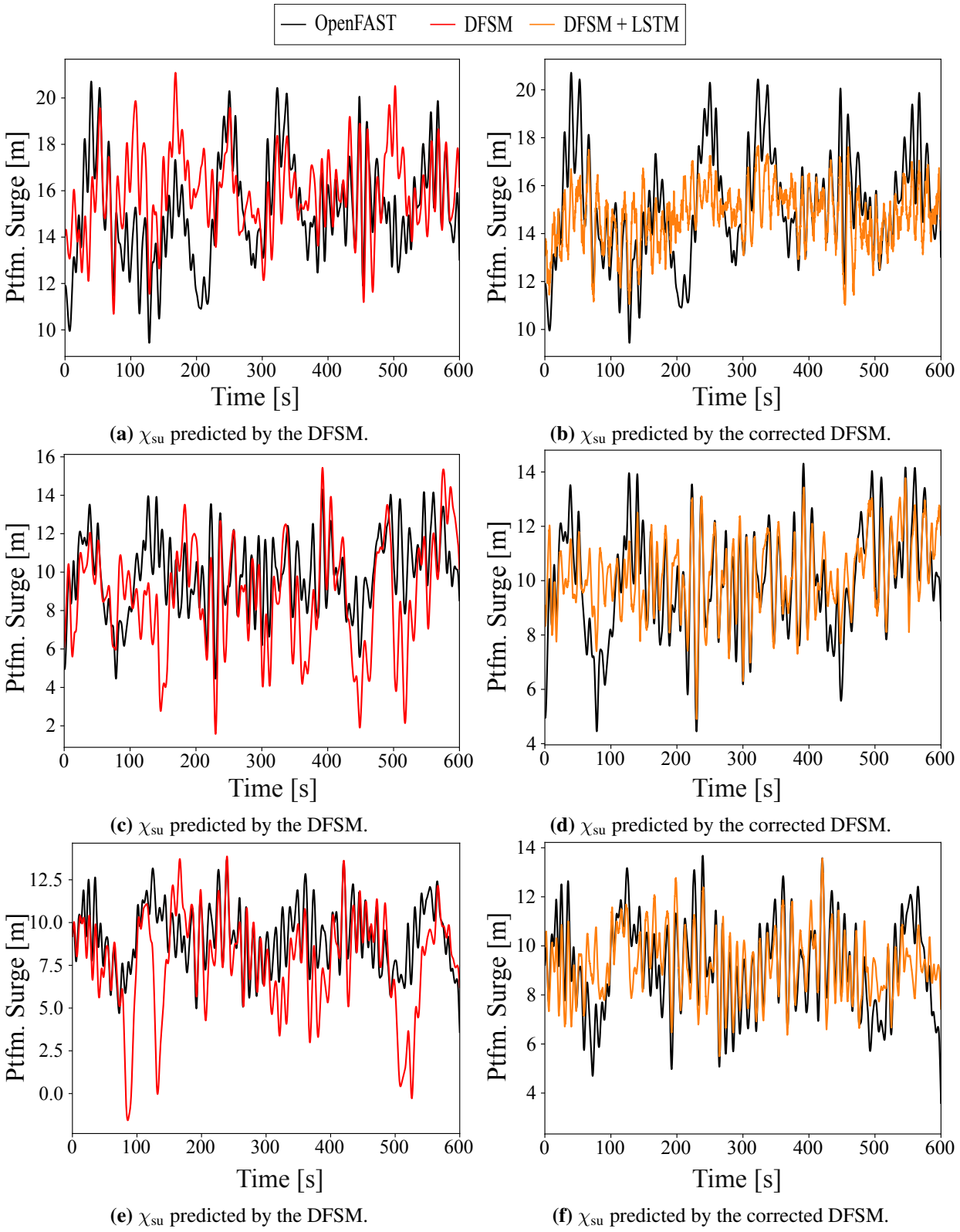
The χ_{su} predicted by this network for different test load cases is shown in Fig. 5.7, and the $M_{t,y}$ predicted by the corrective function for the marine turbine system is shown in Fig. 5.8. As shown in the figures, the corrective function improves the prediction of the DFSM for both systems, particularly in regions of the simulations with higher errors. However, even though the corrective function reduces the error in these signals, it does not entirely eliminate it.

Although this strategy could be used to improve the prediction of the DFSM for all use cases, we do not utilize it. For the remaining case studies discussed in this chapter, both the controller parameters and the plant design are changed. As these variables change, the OpenFAST response also changes. Therefore, to utilize this approach when these variables are modified, the LSTM would need to be continually updated, which can be computationally inefficient. Hence, we recommend that this approach be used in cases where neither the controller nor the plant variables change.

5.2 Open-Loop Optimal Control

5.2.1 Motivation

As mentioned in the previous chapters, the DFSM approach was first introduced in Ref. [86] to carry out open-loop optimal control studies for wind turbines. The main motivation behind this study is to demonstrate that the DFSM approach introduced in this dissertation can also be used to perform similar open-loop optimal control studies. Because an LPV structure is used for these models, the optimal control problems can be formulated and solved efficiently for the reasons outlined in Refs. [104, 132]. We utilize the DFSMs constructed for the IEA-15 MW turbine with a semisubmersible platform and the RM1-Quad marine turbine in this use case. The focus of this section is to use these DFSMs to perform open-loop optimal control studies and explore different



(a) χ_{su} predicted by the DFSM.

(b) χ_{su} predicted by the corrected DFSM.

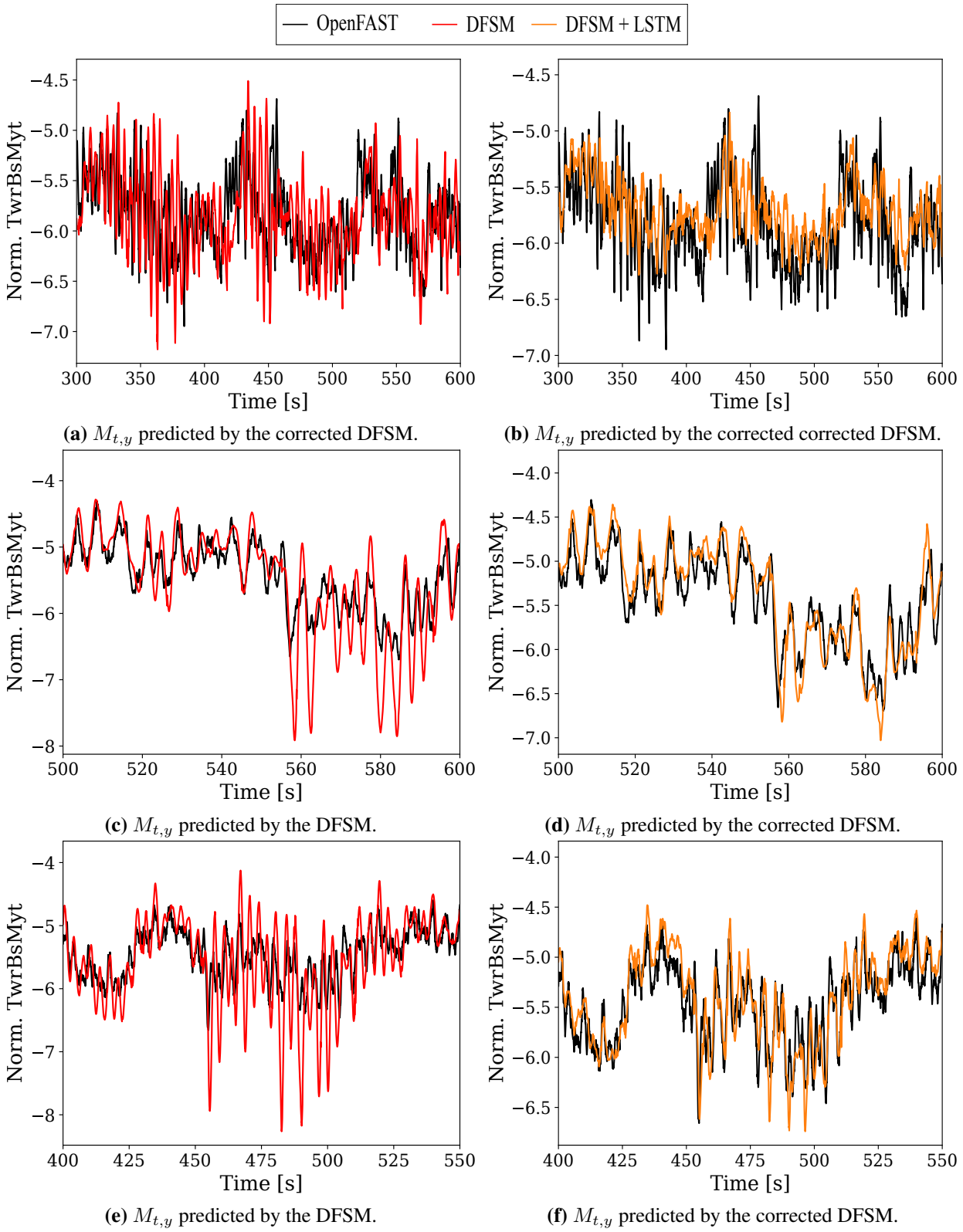
(c) χ_{su} predicted by the DFSM.

(d) χ_{su} predicted by the corrected DFSM.

(e) χ_{su} predicted by the DFSM.

(f) χ_{su} predicted by the corrected DFSM.

Figure 5.7: Comparison of the platform surge (χ_{su}) signal for different load cases as predicted by the DFSM, corrected DFSM (DFSM+LSTM) and OpenFAST.



(a) $M_{t,y}$ predicted by the corrected DFSM.

(b) $M_{t,y}$ predicted by the corrected corrected DFSM.

(c) $M_{t,y}$ predicted by the DFSM.

(d) $M_{t,y}$ predicted by the corrected DFSM.

(e) $M_{t,y}$ predicted by the DFSM.

(f) $M_{t,y}$ predicted by the corrected DFSM.

Figure 5.8: Comparison of the normalized tower-base fore-aft moment $(M_{t,y})$ signal for different load cases as predicted by the DFSM, corrected DFSM (DFSM+LSTM) and OpenFAST.

trade-offs for these systems. Different trade-offs are considered for the wind and marine turbine systems, respectively.

5.2.2 Problem Formulation

Wind turbine case study.

For the wind turbine system, we use the following problem formulation:

$$\text{changing:} \quad [\boldsymbol{u}, \boldsymbol{\xi}] \tag{5.1a}$$

$$\text{minimize:} \quad \int_{t_0}^{t_f} [(-\tau_g \omega_g) + w_1 \tau_g^2 + w_2 \beta^2] d\text{t} \tag{5.1b}$$

$$\text{subject to:} \quad \dot{\boldsymbol{\xi}} = \boldsymbol{A}(w)\boldsymbol{\xi} + \boldsymbol{B}(w)\boldsymbol{u} \tag{5.1c}$$

$$\boldsymbol{y} = \boldsymbol{C}(w)\boldsymbol{\xi} + \boldsymbol{D}(w)\boldsymbol{u} \tag{5.1d}$$

$$\Theta_p \leq \Theta_{p,\text{max}} \tag{5.1e}$$

$$\omega_g \leq \omega_{g,\text{rated}} \tag{5.1f}$$

$$0 \leq \beta \leq 20 \tag{5.1g}$$

$$0 \leq \tau_g \leq 19786.7 \tag{5.1h}$$

Two different values of $\Theta_{p,\text{max}}$ are explored, namely $\Theta_{p,\text{max}} = [4, 7]$. As explained in Refs. [70, 102,105], since the control variable τ_g appears linearly in the objective function, a quadratic penalty must be added to prevent bang-bang behavior. In addition, we also add a penalty on the blade pitch to account for its limited actuation in the transition region. The values of $[w_1, w_2]$ are $[10^{-7}, 10^2]$ for this system. The value of w_2 was chosen to ensure that realistic trajectories for β are identified. The system is simulated for a series of load cases in the transition region, and the results of one of these cases are presented.

Marine turbine case study.

For the marine turbine system, the following problem formulation is used:

$$\text{changing:} \quad [\boldsymbol{u}, \boldsymbol{\xi}] \tag{5.2a}$$

$$\text{minimize:} \quad \int_{t_0}^{t_f} [(-\tau_g \omega_g) + w_1 \tau_g^2 + w_2 \beta^2] d\mathrm{t} \tag{5.2b}$$

$$\text{subject to:} \quad \dot{\boldsymbol{\xi}} = \boldsymbol{A}(w)\boldsymbol{\xi} + \boldsymbol{B}(w)\boldsymbol{u} \tag{5.2c}$$

$$\boldsymbol{y} = \boldsymbol{C}(w)\boldsymbol{\xi} + \boldsymbol{D}(w)\boldsymbol{u} \tag{5.2d}$$

$$\omega_g \leq \omega_{g,\text{max}} \tag{5.2e}$$

$$0 \leq \beta \leq 20 \tag{5.2f}$$

$$0 \leq \tau_g \leq 8.3 \tag{5.2g}$$

Since the platform was explicitly designed to minimize platform pitching motion, no constraints were placed on Θ_p. Instead, two different values of $\omega_{g,\text{max}}$ are explored, namely $\omega_{g,\text{max}} = [609.35, 731.22]$. These values correspond to $\omega_{g,\text{rated}}$ and $1.2 \times \omega_{g,\text{rated}}$, respectively. The values of $[w_1, w_2]$ used in this study are $[10^{-4}, 10^{-2}]$. Load cases from the rated region are used in this study.

5.2.3 Solution Approach

Because both optimal control problems are formulated using quadratic objectives subject to linear constraints, they result in linear quadratic dynamic optimization (LQDO) problems. Such problems can be formulated and solved efficiently. The open-source toolbox `DTQP` is used to formulate and solve these problems. This toolbox utilizes the direct transcription (DT) approach discussed in Chap. 2 to solve the problem. For both problems, $t_0 = 0$ and $t_f = 800$. The horizon between t_0 and t_f is discretized into $n_t = 5000$ points. The trapezoidal rule is used to construct the quadrature and defect constraints. Utilizing the DT approach for this LQDO problem results in a quadratic programming problem, which can be solved using the `quadprog` solver in `MATLAB`. Using this approach, it takes around 15 seconds to solve Prob. (5.1) and 4 seconds to solve Prob. (5.2).

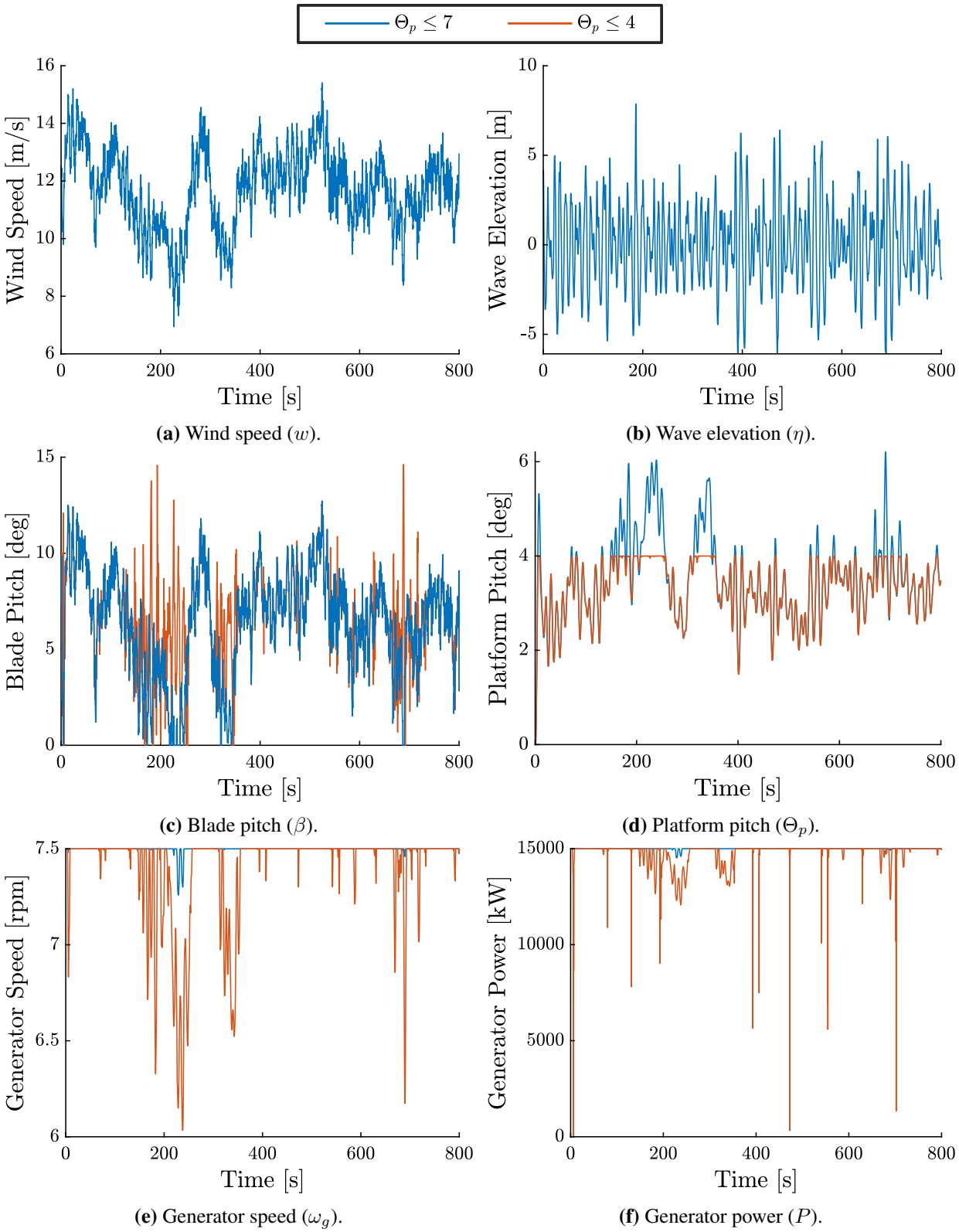
5.2.4 FOWT Results

The optimal control results obtained for one of the load cases for the FOWT system are discussed in this subsection. The optimal trajectories for different quantities are shown in Fig. 5.9. Figures 5.9a and 5.9b show the wind speed and wave elevation profiles for this load case. As can be seen from Fig. 5.9a, this load case is in the transition region. The optimal trajectories for β are shown in Fig. 5.9c. Figures 5.9d to 5.9f show the optimal trajectories obtained for Θ_p, χ, ω_g, and P, respectively.

As noted previously, the trends in β should follow the trends in the wind speed. Figures 5.9a and 5.9c show that the optimizer is able to identify a trajectory for β that satisfies this condition. Since the goal of the problem is to maximize the power produced, the optimizer attempts to maintain the values of ω_g and τ_g at their maximum allowable values. However, to satisfy the constraint on $\Theta_{p,\max}$, β becomes active, which lowers the values of both Θ_p and ω_g, as shown in Figs. 5.9d and 5.9e. This is the reason that β is more active for the constrained case $\Theta_{p,\max} \leq 4$ in Fig. 5.9c. Since β is active to satisfy the constraint in this case, ω_g is also affected, which in turn influences P, as can be seen in Fig. 5.9f. These results follow similar trends to those shown in Ref. [70].

5.2.5 Marine Turbine Results

The optimal trajectories obtained for different quantities are shown in Fig. 5.10. The current speed and wave elevation time series for this load case are shown in Figs. 5.10a and 5.10b, respectively. The optimal β trajectory obtained for both constraint values is shown in Fig. 5.10c. Similar to the FOWT case, the trends in β closely follow the trends in the current speed. Since the goal is to maximize power production, the optimizer seeks to maintain the values of ω_g and τ_g at their maximum allowable limits. Increasing the upper bound on ω_g increases the power production, as can be seen from Figs. 5.10d and 5.10f. To enable this, the value of β must be lower. However, this results in higher values of the tower-base moment, as seen in Fig. 5.10e, which subsequently leads to higher DEL_t. These results illustrate the inherent trade-off between load minimization and power maximization. This study was repeated for five different constraint values corresponding



(a) Wind speed (w).

(b) Wave elevation (η).

(c) Blade pitch (β).

(d) Platform pitch (Θ_p).

(e) Generator speed (ω_g).

(f) Generator power (P).

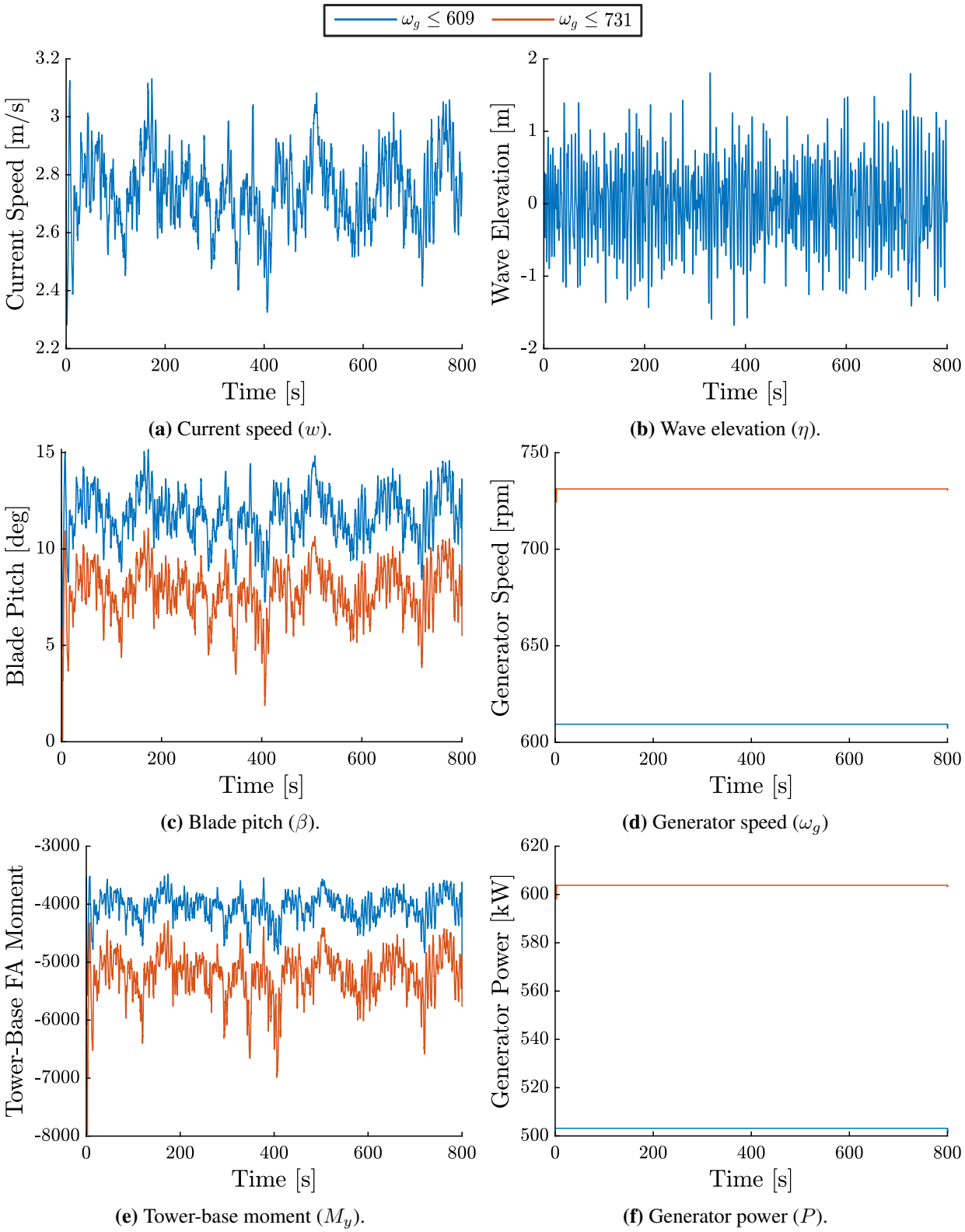
Figure 5.9: Open-loop optimal control results for the FOWT case study using the DFSM for a test load case with $w_{\text{avg}} = 12$ [m/s].

to 0–20% of $\omega_{g,\text{rated}}$ for this load case. The average generator power versus the DEL_t is plotted in Fig. 5.11. For a 5% increase in the average power, there is a 15% increase in the corresponding DEL_t.

In open-loop optimal control studies, it is assumed that all the necessary information about the operating conditions is available a priori. Although this assumption is not realistic, these studies can still be used to identify the maximum achievable performance for the system. Additionally, using LQDO formulations when possible allows the user to quickly explore different trade-offs for key performance metrics. Subsequently, based on the results and insights of these open-loop optimal control studies, closed-loop controllers can be designed.

5.3 Closed-Loop Optimal Control

The third use case for the DFSM modeling approach presented here is to explore the use of these low-fidelity models in design optimization studies for the controllers. We formulate and solve a series of design optimization studies with the goal of finding the optimal controller parameters that minimize the DEL_t for different systems. The blade pitch controller is particularly crucial for minimizing fatigue loads and maximizing power capture. For this reason, many controller optimization studies for wind and marine turbines focus on the blade pitch controller [88, 98]. Before the DFSM can be used for these studies, a validation test must be carried out. Previous DFSM studies assume that the $\boldsymbol{x}_c$ value on which the DFSM is trained is the same as the value it is tested on. However, to use the DFSM for controller optimization studies, it is necessary to verify that the DFSM can predict key performance metrics for different values of $\boldsymbol{x}_c$. To this end, we carry out a series of design of experiments (DOE) studies using a full-factorial sampling scheme. Different samples for $\boldsymbol{x}_c$ are generated using this approach, and the system is simulated using both the DFSM and OpenFAST models at these sample points. At each point, both the low- and high-fidelity models are simulated for different load cases with various values of $\boldsymbol{w}_{\text{avg}}$. The simulation results are then postprocessed to obtain the relevant performance metrics for these systems.



(a) Current speed (w).

(b) Wave elevation (η).

(c) Blade pitch (β).

(d) Generator speed (ω_g)

(e) Tower-base moment (M_y).

(f) Generator power (P).

Figure 5.10: Open-loop optimal control results for the marine turbine case study using the DFSM for a test load case with $v_{\text{avg}} = 2.75$ [m/s].

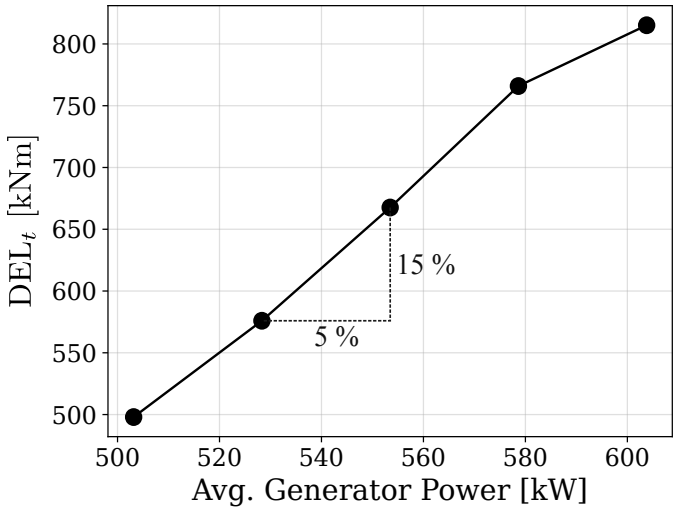


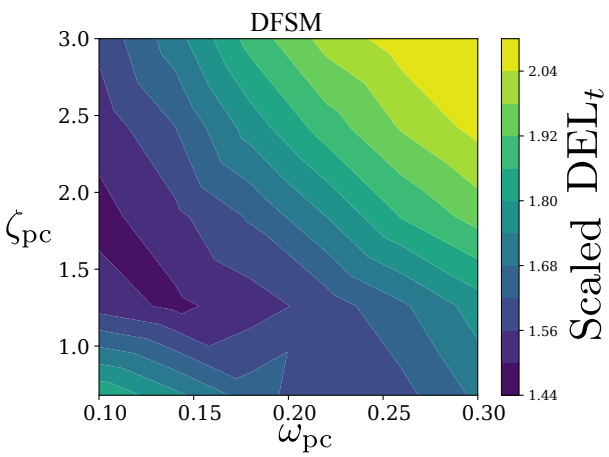
Figure 5.11: Average generator power vs. DEL_t calculated using the open-loop optimal control results for different constraint values on the generator speed for the marine turbine for the load case shown in Fig. 5.10a.

We carry out this study for the IEA-15 MW turbine with a semi-submersible platform. The design variables considered here are the natural frequency and damping ratio of the blade pitch controller:

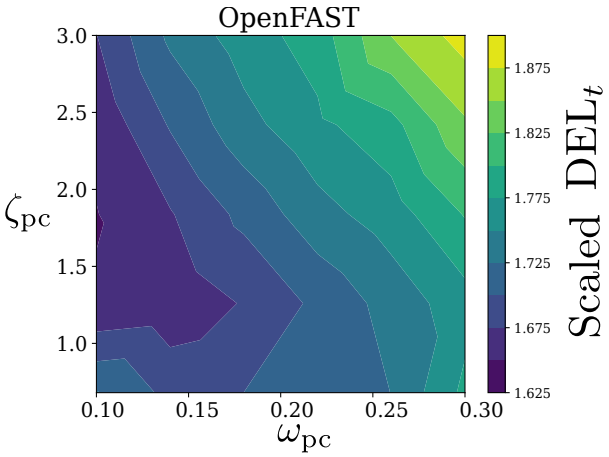
$$\boldsymbol{x}_c = [\omega_{\mathrm{pc}}, \zeta_{\mathrm{pc}}]^{\mathrm{T}} \tag{5.3}$$

where the subscript 'pc' stands for pitch controller. We use a full-factorial scheme with $n = 6$ samples per variable to generate the design space. The system is simulated using both the DFSM and OpenFAST for five key wind speeds, $\bar{\boldsymbol{w}} = [10, 12, 14, 16, 18]$, with $n_s = 5$ seeds per wind speed. These wind speeds are chosen because they are highly weighted in the Weibull PDF used for this system. Consequently, DEL_t values calculated for these load cases carry a higher weight than other wind speeds. All OpenFAST and DFSM simulations in this study are carried out on a high-performance computing cluster equipped with 2× Intel Xeon Gold 6148 CPUs, 40 cores per node, and 192 GB of RAM. On this system, the average simulation time for OpenFAST is approximately 19 minutes, whereas the DFSM takes around 42 seconds per simulation.

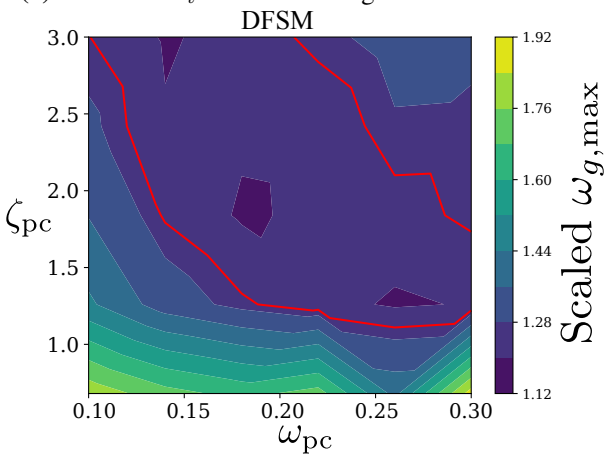
The contour plot of DEL_t is shown in Fig. 5.12, with Fig. 5.12a showing the design space obtained using the DFSM, and Fig. 5.12b showing the design space obtained from OpenFAST.



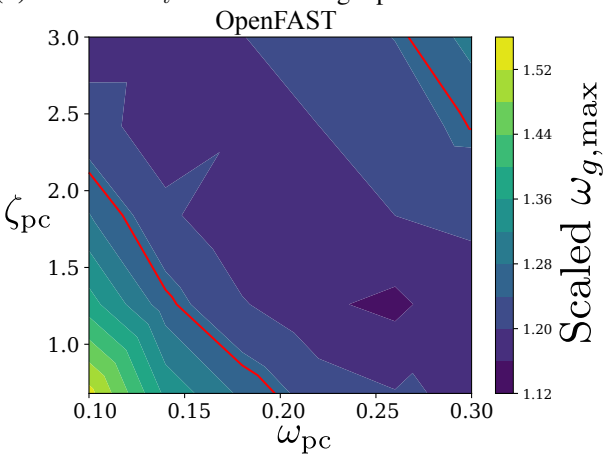
(a) Scaled DEL_t calculated using DFSM simulations.



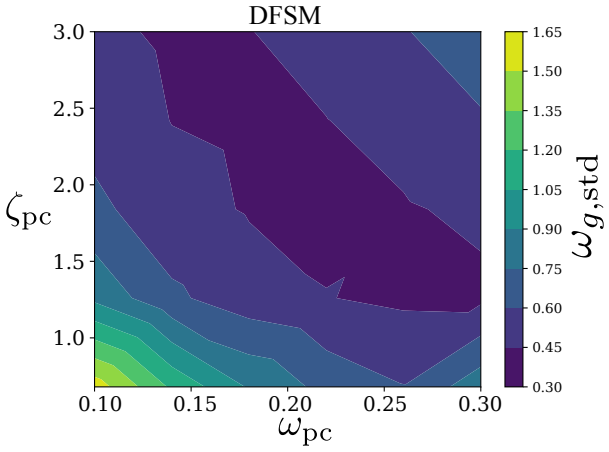
(b) Scaled DEL_t calculated using OpenFAST simulations.



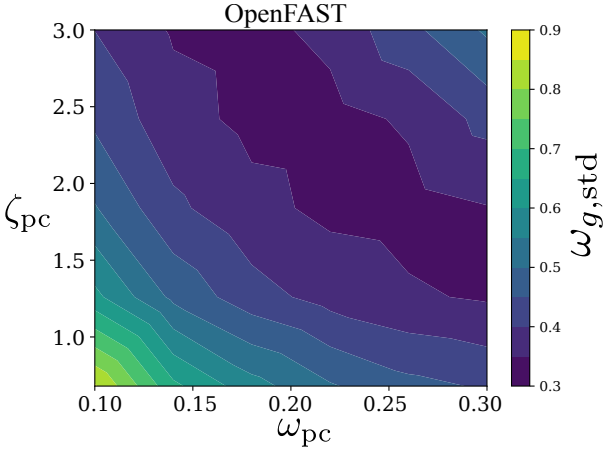
(c) Scaled $\omega_{g,\max}$ calculated using DFSM simulations.



(d) Scaled $\omega_{g,\max}$ calculated using OpenFAST simulations.



(e) $\omega_{g,\mathrm{std}}$ calculated using OpenFAST simulations.



(f) $\omega_{g,\mathrm{std}}$ calculated using OpenFAST simulations.

Figure 5.12: Comparison of the scaled tower base damage equivalent load (DEL_t), scaled maximum generator speed ($\omega_{g,\max}$), and the mean generator speed standard deviation ($\omega_{g,\mathrm{std}}$) vs. $\boldsymbol{x}_c$ calculated using DFSM and OpenFAST simulations over all the load cases.

It is clear from Figs. 5.12a and 5.12b that, although the absolute DEL_t values predicted by the DFSM differ from those predicted by OpenFAST, the overall shape and trends in the design space are similar. Typically, DEL_t is used as the objective in many controller design studies, with the goal of minimizing it. Thus, using the DFSM to formulate and solve the optimization problem would lead the optimizer to identify $\boldsymbol{x}_c$ values near the optima in the high-fidelity design space as well.

In addition to DEL_t, Figs. 5.12c and 5.12d show the scaled maximum generator speed obtained across all simulations using DFSM and OpenFAST. This metric is often used as a constraint in controller optimization studies. In both figures, the region corresponding to scaled $\omega_{g,\max} \leq 1.25$ is highlighted, further confirming that DFSM-based optimization would yield solutions close to those for the high-fidelity model. Figures 5.12f and 5.12e show the comparison of $\omega_{g,\text{std}}$, which serves as a proxy for power quality and is also used as an objective.

Based on these observations, one way to leverage the low computational cost of the DFSM is through a hybrid-optimization scheme. In such a scheme, a GA can be used for a few iterations to explore the design space and identify a promising region, followed by a gradient-based solver to converge to the optimum. Since GAs often require hundreds of function evaluations, using the DFSM in this step significantly reduces computational cost. Other multi-fidelity approaches that utilize both low- and high-fidelity models simultaneously can also be employed to identify high-fidelity optima. In this work, we use one such approach for controller-optimization studies.

5.3.1 Trust-Region-Based Multi-Fidelity Controller Optimization

We use a trust-region-based multi-fidelity optimization scheme introduced in Ref. [119]. This approach has been recently adapted and applied to several wind turbine design optimization studies in Ref. [161]. One of these studies specifically focuses on controller optimization, where OpenFAST serves as the high-fidelity model and linearized OpenFAST models are used as the low-fidelity models. Furthermore, the studies in Ref. [161] leverage the WEIS framework to formulate

and solve these multi-fidelity problems. Given the synergy between the code infrastructure and the application, we adopt this approach and the multi-fidelity implementation it presents.

The trust-region-based multi-fidelity scheme consists of two main components. The first component is a surrogate model used to approximate the high-fidelity response. This surrogate model is constructed as the sum of the low-fidelity model response and a corrective function that captures the discrepancy between the low- and high-fidelity function values. Mathematically, this surrogate model can be expressed as:

$$f_{\text{hf}}(\boldsymbol{x}) \approx f_{\text{corr}}(\boldsymbol{x}) = f_{\text{lf}}(\boldsymbol{x}) + f_{\text{surr}}(\boldsymbol{x}) \tag{5.4a}$$

$$f_{\text{surr}}(\boldsymbol{x}) = f_{\text{hf}}(\boldsymbol{x}) - f_{\text{lf}}(\boldsymbol{x}) \tag{5.4b}$$

where $f_{\text{hf}}(\boldsymbol{x})$, $f_{\text{lf}}(\boldsymbol{x})$, and $\boldsymbol{x}$ represent the high-fidelity function response, the low-fidelity function response, and the design variables, respectively.

The second component of the multi-fidelity scheme is the trust-region approach, which is used to iteratively solve the optimization problem. This approach begins by defining a trust region, which is a subset of the full design space where the corrected low-fidelity model $f_{\text{corr}}(\boldsymbol{x})$ is assumed to provide an accurate approximation of the high-fidelity response. An optimization problem using $f_{\text{corr}}(\boldsymbol{x})$ is then solved within this region to identify the local optimum. The high-fidelity model is subsequently evaluated at this optimum, and the result is compared with the corrected low-fidelity prediction. Based on the level of agreement, the trust-region radius is either expanded or contracted. The corrective function is then updated to incorporate the newly evaluated high-fidelity point. This process is repeated iteratively until a preset number of iterations is reached or the trust-region radius falls below a specified tolerance. Algorithm 1 summarizes this procedure. For additional details, readers are referred to Refs. [119, 161].

5.3.2 Multi-Fidelity Controller Optimization Studies

We utilize the multi-fidelity optimization approach to formulate and solve controller optimization problems for the IEA-15 MW turbine. We utilize the DFSM as the low-fidelity model and

Algorithm 1 Multi-fidelity trust region algorithm from Ref. [161].

TOL $\leftarrow 10^{-3}$
$\gamma_0 \leftarrow 0.25, \gamma_1 \leftarrow 2, \eta \leftarrow 0.25$
$\Delta_{\max} \leftarrow 10^3$
$k \leftarrow 0$
while $\Delta_k >$ TOL **do**
  $\boldsymbol{s}_k \leftarrow \text{argmin} f_{\text{corr}}(\boldsymbol{x}_k + \boldsymbol{s}_k) \quad ||\boldsymbol{s}_k|| \leq \Delta_k$
  `Evaluate and store` $f_{\text{hf}}(\boldsymbol{x}_k + \boldsymbol{s}_k)$
  $\rho_k \leftarrow \frac{f_{\text{hf}}(\boldsymbol{x}_k) - f_{\text{hf}}(\boldsymbol{x}_k + \boldsymbol{s}_k)}{f_{\text{lf}}(\boldsymbol{x}_k) - f_{\text{lf}}(\boldsymbol{x}_k + \boldsymbol{s}_k)}$
  $\boldsymbol{x}_{k+1} \leftarrow \begin{cases} \boldsymbol{x}_k + \boldsymbol{s}_k, \text{ if } \quad \rho_k > 0 \\ \boldsymbol{x}_k, \text{ otherwise} \end{cases}$
  $\Delta_{k+1} \leftarrow \begin{cases} \min\{\gamma_1 \Delta_k, \Delta_{\max}\}, & \text{if } \quad \rho_k \geq \eta \\ \gamma_0 \Delta_k, & \text{if } \quad \rho_k \geq \eta \end{cases}$
  `Update` $f_{\text{surr}}(\cdot)$ `to include` $\boldsymbol{x}_{k+1}$
end while

OpenFAST as the high-fidelity model in this approach. We formulate and solve the following design optimization problem:

$$\text{changing:} \quad \boldsymbol{x}_c = [\omega_{\text{pc}}, \zeta_{\text{pc}}, k_{\beta,\text{float}}, \omega_{\text{ptfm}}]^T \tag{5.5a}$$

$$\text{minimize:} \quad o = \text{DEL}_t \tag{5.5b}$$

$$\text{subject to:} \quad \omega_g \leq 1.25 \times \omega_{g,\text{rated}} \tag{5.5c}$$

$$\boldsymbol{x}_{c,\min} \leq \boldsymbol{x}_c \leq \boldsymbol{x}_{c,\max} \tag{5.5d}$$

$$\text{where:} \quad \boldsymbol{x}_{c,\min} = [0.1, 0.6, -40, 0.0]^T \tag{5.5e}$$

$$\boldsymbol{x}_{c,\max} = [0.3, 3.0, 0.0, 0.4]^T \tag{5.5f}$$

This study is formulated and solved using the same set of load cases described in the previous section. As seen in Figs. 5.12, the DFSM can be effectively used to identify the optima for the 2-dimensional design space. To further test the efficacy of the DFSM in a more challenging controller optimization problem, we increase the number of design variables and solve the optimization problem using the low-fidelity, high-fidelity, and multi-fidelity models. The COBYLA algorithm is used to solve the optimization problems at all three levels.

Table 5.1: Multi-fidelity controller optimization results for Prob. 5.5.

Fidelity	LF calls	HF calls	$\boldsymbol{x}_c$	$\text{DEL}_{t,\text{HF}}$ at $\boldsymbol{x}_c$	$\omega_g/\omega_{g,rated}$ at $\boldsymbol{x}_c$
Low	24	–	$[0.14, 1.65, -07.69, 0.36]^T$	1.62×10^5	1.23
Multi	468	11	$[0.15, 2.12, -15.37, 0.38]^T$	1.56×10^5	1.24
High	–	38	$[0.10, 2.92, -14.42, 0.36]^T$	1.55×10^5	1.23

The results obtained from the three approaches are presented in Table 5.1. All three optimization studies are initialized at the same starting point, $\boldsymbol{x}_{c,\text{init}} = [0.18, 2.2, -4.9, 0.2]^T$, for which the DEL_t value for the high-fidelity model is 1.79×10^5 [kNm]. It is evident from Table 5.1 that both the DFSM and the multi-fidelity approach converge to points with lower DEL_t values. Moreover, the multi-fidelity approach identifies an optimum that is closer to the high-fidelity (OpenFAST) response in terms of the objective function value. The remaining discrepancy between the optima identified by the multi-fidelity and high-fidelity models can be attributed to the trust-region methodology and the optimizer used to solve the trust-region subproblems.

5.4 Multi-Objective Controller Optimization

In this section, we formulate and solve a multi-objective controller optimization problem. For this case study, we use a slightly modified version of the IEA-15 MW turbine with a semisubmersible platform. This model has a different tower and platform design compared to the systems discussed previously; consequently, the DEL_t values reported for this system differ slightly from those presented earlier. We consider load cases corresponding to five different mean wind speeds, $\boldsymbol{w}_{\text{avg}} = [10, 12, 14, 16, 18]$, using six seeds for each wind speed. As described previously, the weighted-objective method is employed to solve this multi-objective problem.

5.4.1 Performance Metrics

As seen previously, maximizing performance metrics such as power production often conflicts with the goal of minimizing fatigue damage, as discussed in Sec. 5.2.5. To visualize this trade-off, consider Fig. 5.12. Figure 5.12b shows the design space for $\boldsymbol{x}_c = [\omega_{pc}, \zeta_{pc}]$ versus DEL_t,

while Fig. 5.12f shows $\boldsymbol{x}_c$ versus $\omega_{g,\text{std}}$, both obtained using OpenFAST simulations. For both performance metrics, the goal is to minimize the values. It is evident from these figures that minimizing one metric generally leads to an increase in the other. Similarly, Figs. 5.12a and 5.12e show the corresponding quantities obtained using DFSM simulations, demonstrating that the DFSM can accurately predict the design spaces for both metrics. Based on this, we formulate and solve a multi-objective optimization problem using the DFSM to investigate the trade-off between these two quantities. In this study, we include variables associated with the floating feedback controller in addition to the variables of the PI controller.

5.4.2 Problem Formulation and Solution Strategy

We formulate the following multi-objective optimization problem:

$$\text{changing:} \quad \boldsymbol{x}_c = [\omega_{\text{pc}}, \zeta_{\text{pc}}, k_{\beta,\text{float}}, \omega_{\text{ptfm}}]^T \tag{5.6a}$$

$$\text{minimize:} \quad \boldsymbol{o} = [\text{DEL}_t, \omega_{g,\text{std}}] \tag{5.6b}$$

$$\text{subject to:} \quad \boldsymbol{x}_{c,\min} \leq \boldsymbol{x}_c \leq \boldsymbol{x}_{c,\max} \tag{5.6c}$$

$$\boldsymbol{x}_{c,\min} \leq \boldsymbol{x}_c \leq \boldsymbol{x}_{c,\max} \tag{5.6d}$$

$$\text{where:} \quad \boldsymbol{x}_{c,\min} = [0.1, 0.6, -40, 0.0]^T \tag{5.6e}$$

$$\boldsymbol{x}_{c,\max} = [0.3, 3.0, 0.0, 0.4]^T \tag{5.6f}$$

We use the weighted-objective method to solve this multiobjective problem. Using this approach, we formulate and solve a series of subproblems for each value $w_1 = [1, 0.9, \ldots, 0]$:

$$\text{changing:} \quad \boldsymbol{x}_c \tag{5.7a}$$

$$\text{minimize:} \quad o = w_1 \text{DEL}_t(\boldsymbol{x}_c) + w_2 \omega_{g,\text{std}}(\boldsymbol{x}_c) \tag{5.7b}$$

$$\text{subject to:} \quad \boldsymbol{x}_{c,\min} \leq \boldsymbol{x}_c \leq \boldsymbol{x}_{c,\max} \tag{5.7c}$$

$$\text{where:} \quad w_1 + w_2 = 1 \tag{5.7d}$$

We solve the subproblems using the multi-fidelity optimization strategy outlined previously, and store the solutions of each subproblem. These results are then post-processed to identify the non-dominated solutions that form the Pareto front.

Using the DFSM alone to formulate and solve these problems would be computationally the most efficient approach. However, as seen previously, the optimal points identified using the DFSM do not necessarily correspond exactly to the optima for OpenFAST, even though the overall design spaces are similar. Therefore, it is necessary to employ the multi-fidelity approach.

We utilize three different strategies to solve the multi-objective, multi-fidelity optimization problem:

S1 Start all subproblems from the same initial point and solve them independently.

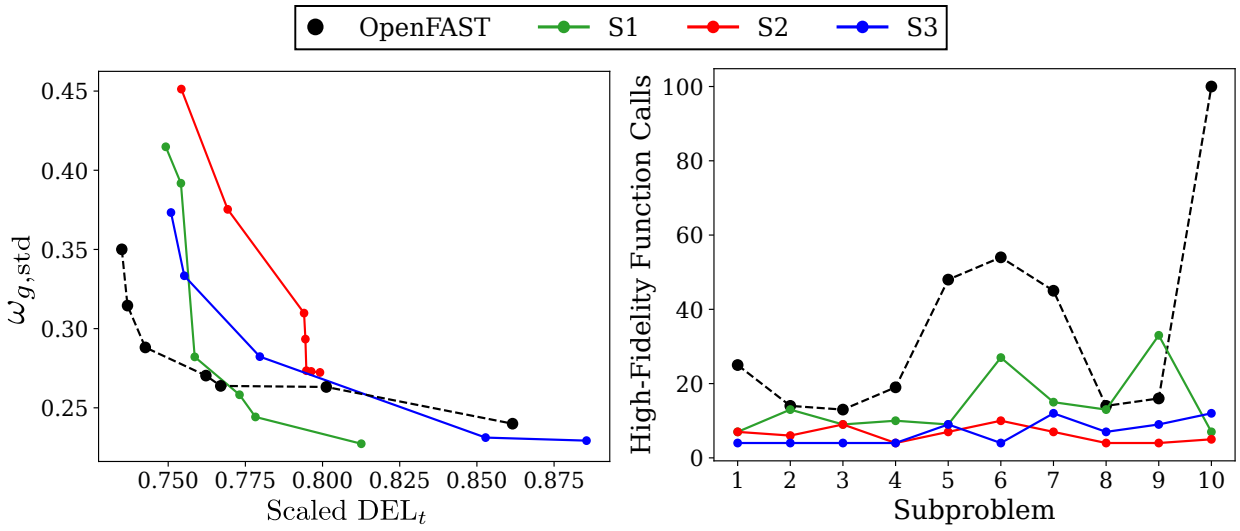
S2 Solve the first subproblem from a random initial point, and then use the optimal point obtained from the previous subproblem as the starting point for the next subproblem, iteratively.

S3 Solve all subproblems using the low-fidelity model first, and then initialize the multi-fidelity approach from the optimal point identified for the low-fidelity model.

We solve the multi-objective problem using all three strategies and compare the results. The trust-region constrained subproblems are solved using the SLSQP solver, which is a gradient-based optimizer. The comparison is based on both the optimal points identified and the computational expense, expressed in terms of the number of high-fidelity function calls.

5.4.3 Results

The Pareto points obtained by solving Prob. 5.6 using all three strategies outlined previously, as well as by using OpenFAST directly, are shown in Fig. 5.13, along with the computational cost measured in terms of the number of high-fidelity function calls. The Pareto points correspond to the high-fidelity function values for both objectives. For all three strategies, the total number of non-dominated points is always less than the total number of subproblems solved, indicating that not all subproblems yield non-dominated solutions. This is one of the inherent drawbacks of the weighted-objective approach for solving multi-objective problems.



(a) Pareto fronts

(b) Number of high-fidelity function calls

Figure 5.13: Comparison of the Pareto fronts and the associated computational expense obtained by solving the multi-objective optimization problem shown in Eqs. (5.6) using three different strategies outlined in S1 – S3 and OpenFAST. In S1, all the subproblems are solved starting from the same point. In S2, we start the first subproblem from a random point, but utilize the optimal point identified for the first subproblem as the starting point for the second subproblem, and so on. In S3, all the subproblems are solved using the low-fidelity model first, and the multi-fidelity approach is started from the optimal point identified for the low-fidelity model.

Comparing the results across strategies, it can be seen that Strategy S1 produces the largest number of non-dominated points, but consequently has the highest computational cost. Strategy S2 has the lowest computational expense but generally results in higher objective function values. Strategy S3 lies between the other two in terms of both computational cost and objective function values. All three strategies require fewer high-fidelity function calls compared to solving the problem directly using OpenFAST.

The initial and final values of the weighted objective function for the non-dominated points obtained using all three strategies are shown in Fig. 5.14. These results demonstrate that the multi-fidelity, multi-objective optimization approach is effective for solving such problems, while also highlighting its limitations. The approach has two primary drawbacks:

1. For non-convex problems, such as those considered in this study, the method tends to converge to a local minimum.

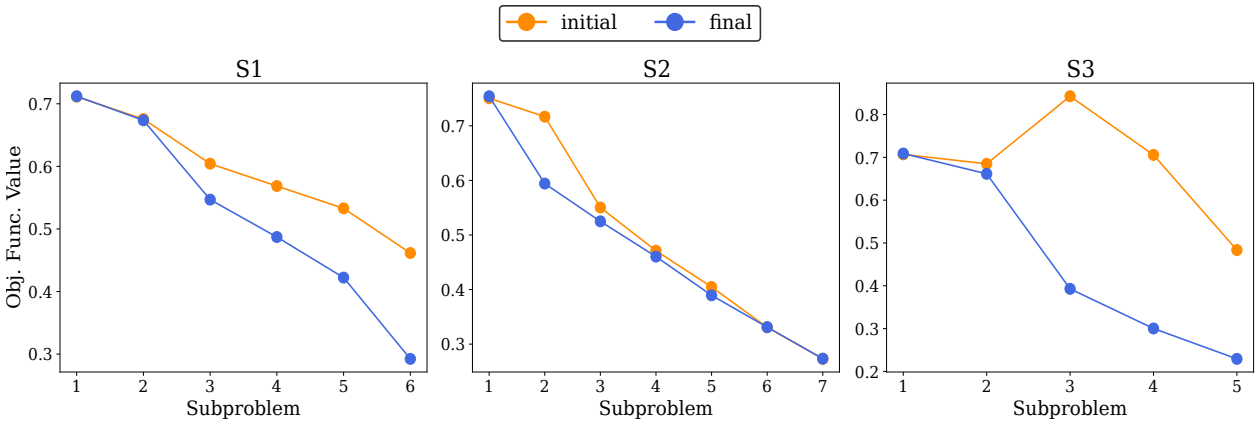


Figure 5.14: Comparison of the initial and final weighted objective function values for the different subproblems that result in a non-dominated point for the three different strategies outlined in S1 - S3.

2. The results are sensitive to the starting point; different initial points can lead to convergence to different solutions.

An interesting trend can be observed for Strategy S3. As the weights change, more high-fidelity function evaluations are required for this approach. This occurs because the low-fidelity model predicts the location of the optimum for DEL_t more accurately than it does for $\omega_{g,\text{std}}$. Consequently, when the weight associated with $\omega_{g,\text{std}}$ is higher, the multi-fidelity approach requires more iterations to converge. This behavior is further illustrated in Fig. 5.14. Based on these observations, we utilize the approach outlined in Strategy S1 for solving multi-objective optimization studies in this dissertation.

5.5 Multi-Objective Controller Optimization from a Control Co-Design Perspective

The previous study considered the controller as the only means to address the trade-off between DEL_t and $\omega_{g,\text{std}}$. An alternative approach is to also modify design variables associated with the supporting platform. Specifically, changing the platform size influences both performance metrics. Larger platforms tend to minimize platform motion and the resulting loads, but this comes at the expense of increased cost. Understanding these trade-offs as the platform design varies constitutes a control co-design (CCD) study. Similar to conventional controller optimization studies, most

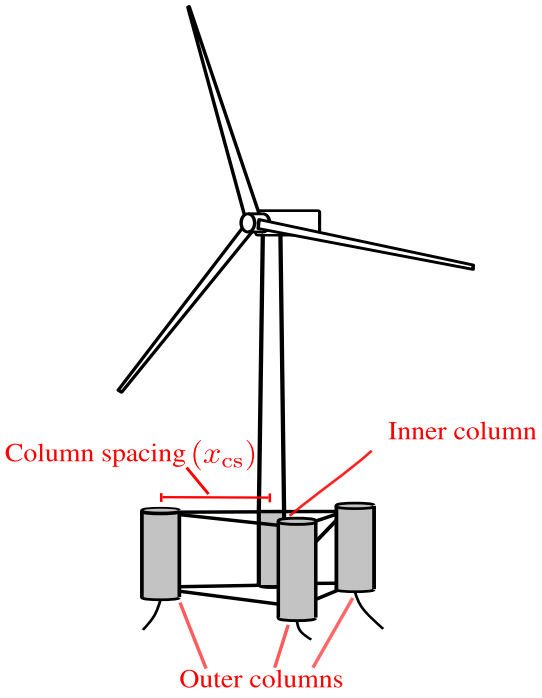


Figure 5.15: Column spacing visualized on the semisubmersible platform for the IEA-15 MW turbine.

previous CCD studies for wind turbines have focused on minimizing a single performance metric. In this study, we extend this approach to investigate CCD from a multi-objective perspective, aiming to understand the trade-off between DEL_t and $\omega_{g,\text{std}}$ as platform and controller parameters are varied simultaneously.

In this study, we focus on the column spacing, denoted as $\boldsymbol{x}_{\text{cs}}$, as the primary plant design variable. This parameter represents the distance between the inner and outer columns of the semisubmersible platform, as illustrated in Fig. 5.15. In addition to $\boldsymbol{x}_{\text{cs}}$, we vary the controller parameters $\boldsymbol{x}_c$ associated with the blade-pitch controller introduced previously.

We adopt a nested control co-design (CCD) formulation to carry out this study. As discussed earlier, the nested formulation consists of an outer loop and an inner loop. In the outer loop, the column spacing is varied, while in the inner loop, we solve the multi-objective controller optimization problem for the corresponding platform configuration. Since the outer loop considers only a single design variable, we perform a sensitivity study by varying the column spacing between its

upper and lower bounds. Specifically, the column spacing is evaluated at five discrete values:

$$\boldsymbol{x}_{\text{cs}} = [35, 42, 51, 57, 65] \tag{5.8}$$

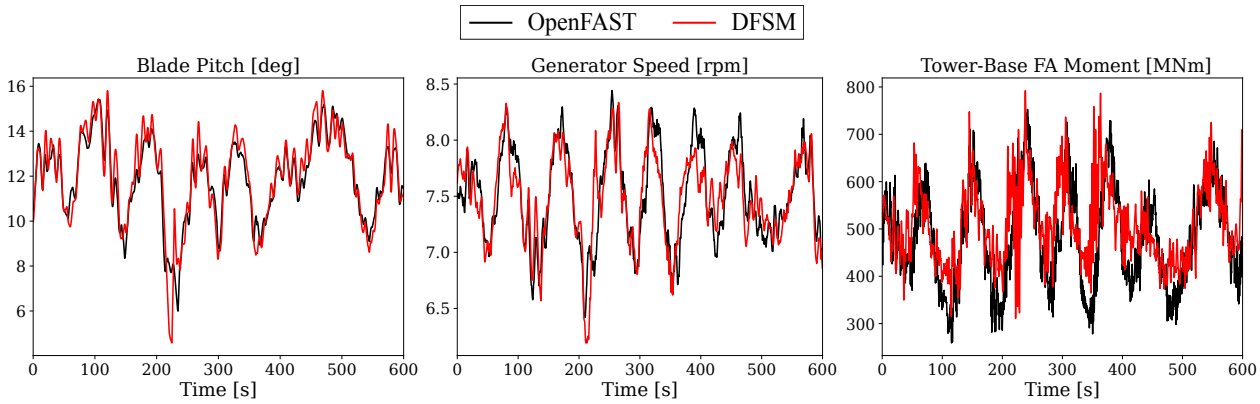
5.5.1 Updating the DFSM

Changing the column spacing affects the dynamic response of the turbine, particularly quantities that capture floating motion and loads, such as Θ_p, χ, and $M_{t,y}$. For the same load case, systems with different values of x_{cs} exhibit distinct responses for these quantities. Therefore, as the column spacing varies, the DFSM must be updated accordingly.

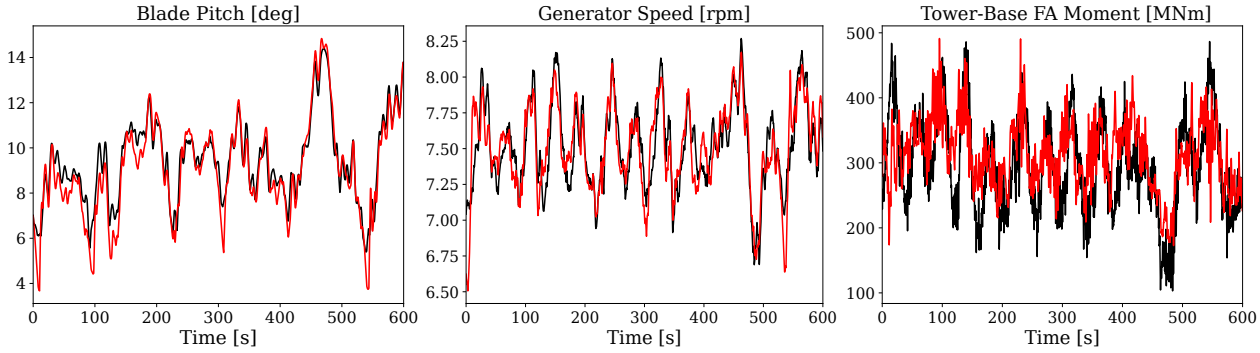
The approach used previously to construct the DFSM—running multiple 600-second load cases for each wind speed—could be applied here as well. However, performing multiple OpenFAST simulations for each iteration of the outer loop can be computationally expensive. To mitigate this, we modify the DFSM construction process for the nested CCD approach.

Previously, model identification began with a hybrid-optimization approach to identify an LTI model for a specific wind speed, which then served as the starting point for successive wind speeds. We adopt the same principle when updating the DFSM for a new value of x_{cs}. We start with the DFSM identified for the nominal column spacing $x_{\text{cs}} = 51$ [m], whose construction began at $w = 14$ m/s. For a platform with $x_{\text{cs}} = 42$ [m], we also start at $w = 14$ m/s, but instead of the hybrid-optimization approach, we use the LTI model identified for $x_{\text{cs}} = 51$ [m] as the initial guess for a gradient-based optimizer. Additionally, we reduce the simulation length from 600 seconds to 300 seconds during model identification. Once the LTI model is identified for $w = 14$ m/s at $x_{\text{cs}} = 42$ [m], it serves as the starting point for $w = 12$ m/s, and so on. Similarly, the LTI model for $x_{\text{cs}} = 42$ [m] is used as the starting point when constructing the DFSM for $x_{\text{cs}} = 35$ [m]. This strategy significantly reduces the computational cost of constructing the DFSM in a nested CCD study.

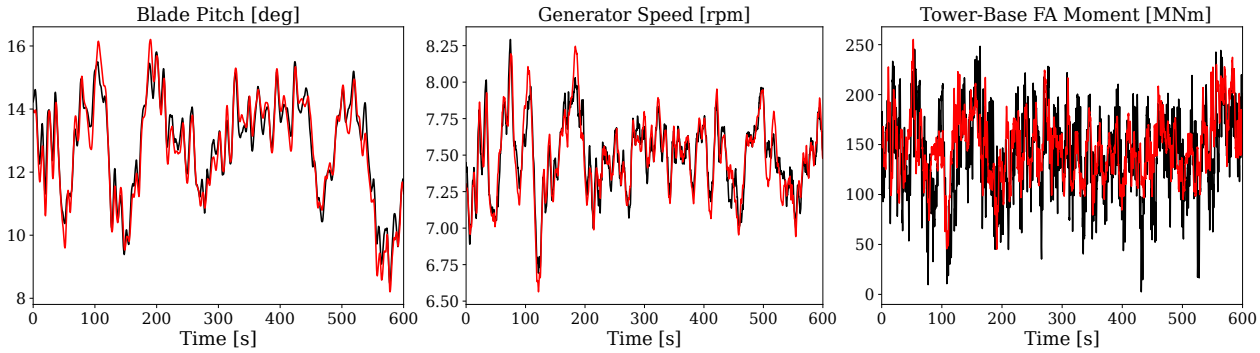
Using this procedure, we construct the DFSM for different values of x_{cs}. Once trained, the models are validated using 600-second closed-loop simulations. Figure 5.16 shows the validation



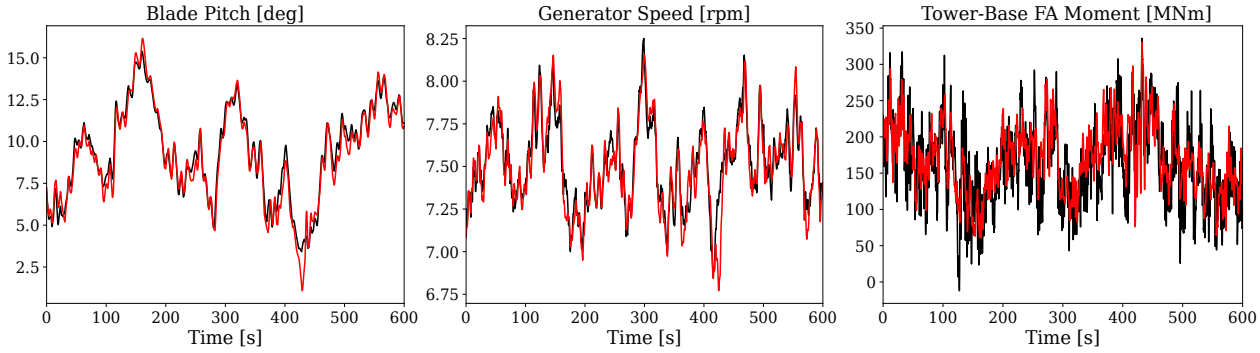
(a) Timeseries comparison for $x_{\text{cs}} = 35$ [m].



(b) Timeseries comparison for $x_{\text{cs}} = 42$ [m].



(c) Timeseries comparison for $x_{\text{cs}} = 57$ [m].



(d) Timeseries comparison for $x_{\text{cs}} = 65$ [m].

Figure 5.16: Comparison of the blade pitch (β), generator speed (ω_g), and tower-base moment ($M_{y,t}$) response obtained using the DFSM and OpenFAST for different values of the column spacing $\boldsymbol{x}_{\text{cs}}$.

results for β, ω_g, and $M_{t,y}$ for one load case at each value of $\boldsymbol{x}_{\text{cs}}$. Despite being trained on 300-second simulations, the models remain accurate for the longer 600-second validation simulations. These results also indicate that changes in x_{cs} have minimal effect on β and ω_g, while the $M_{t,y}$ response varies significantly, highlighting the need to update the DFSM for each column spacing.

5.5.2 Multiobjective Optimization Results

With the DFSM constructed and validated for different values of $\boldsymbol{x}_{\text{cs}}$, we carry out the multi-objective controller optimization study using the multi-fidelity approach outlined previously. The optimization problem is solved for platforms with $\boldsymbol{x}_{\text{cs}} = [42, 51, 57, 65]$ [m], twice for each, starting from different initial points. For $x_{\text{cs}} = 35$ [m], this is not performed, as preliminary studies indicated that both objectives are significantly higher compared to the other platforms. The resulting Pareto fronts for these multi-objective optimization problems are shown in Fig. 5.17. Across all Pareto fronts, it is evident that DEL_t and $\omega_{g\text{std}}$ remain conflicting objectives, although the level of conflict varies between different platforms.

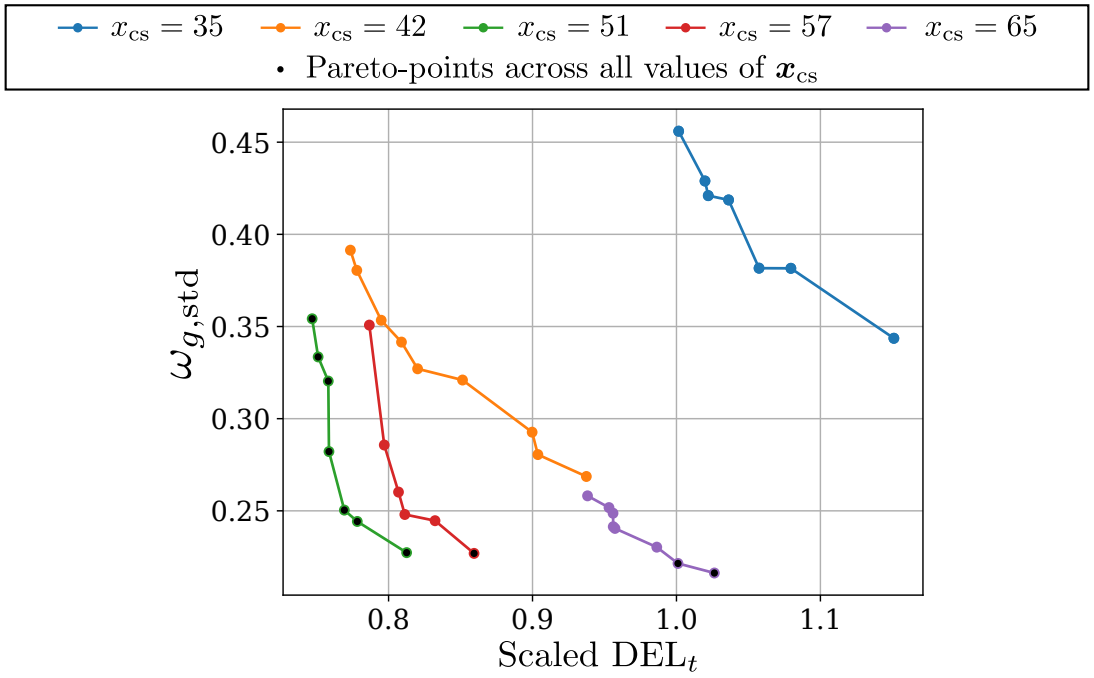
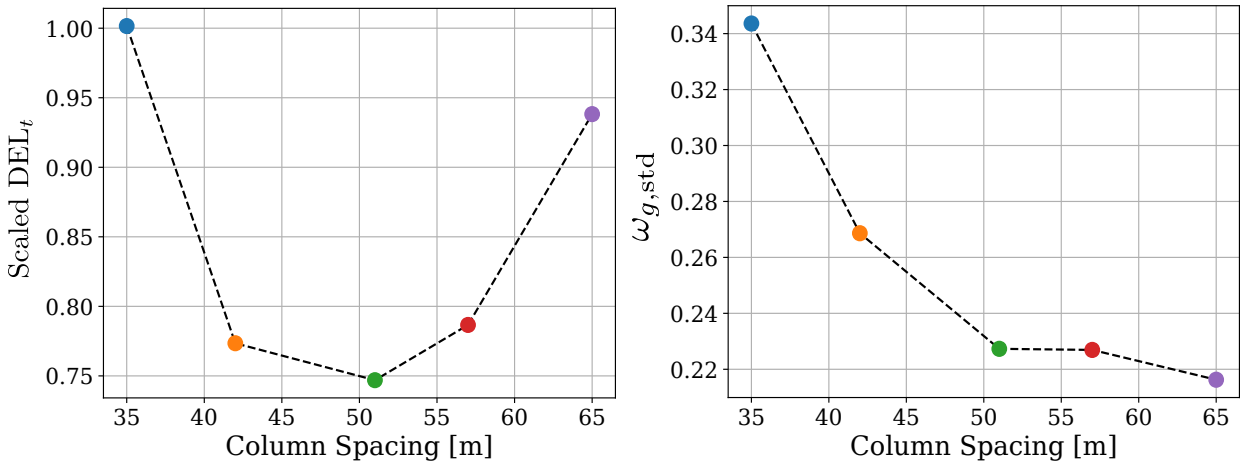


Figure 5.17: Pareto front between DEL_t vs. $\omega_{g,\text{std}}$ obtained for different values of column spacing for the IEA-15 MW turbine with a semisubmersible platform.



(a) Optimal DEL$_t$ vs. column spacing.

(b) Optimal $\omega_{g,\mathrm{std}}$ vs. column spacing.

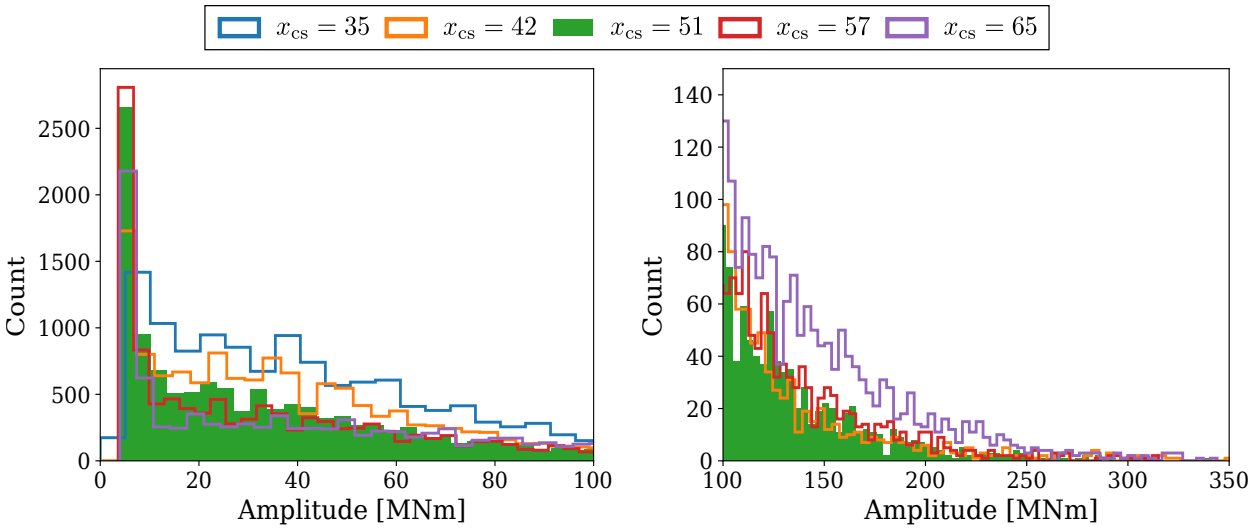
Figure 5.18: Comparison of the optimal DEL$_t$ and $\omega_{g,\mathrm{std}}$ vs. column spacing obtained as part of the multi-objective optimization results shown in Fig. 5.17.

The behavior of the optimal values of DEL$_t$ and $\omega_{g\mathrm{std}}$ differs as x_{cs} is varied. DEL$_t$ exhibits a nonlinear trend with respect to column spacing, whereas $\omega_{g\mathrm{std}}$ monotonically decreases as the column spacing increases. The platform with $x_{\mathrm{cs}} = 35$ [m] has the highest optimal $\omega_{g\mathrm{std}}$, and the platform with $x_{\mathrm{cs}} = 65$ [m] has the lowest, as shown in Figs. 5.18a and 5.18b, respectively.

The trends in Fig. 5.18a can be explained physically. For $x_{\mathrm{cs}} = 35$ [m], the platform is smaller and more susceptible to large motions, which dominate DEL$_t$. For $x_{\mathrm{cs}} = 65$ [m], the platform is heavier and stiffer, resulting in smaller motion amplitudes. However, reduced compliance means more aerodynamic thrust is transmitted directly into the tower, increasing base loads despite smaller platform motions.

These trends can be further understood by examining the DEL$_t$ calculation and corresponding histogram plots. Figure 5.19 shows histograms of load cycle amplitudes from 30 simulations for all values of $\boldsymbol{x}_{\mathrm{cs}}$, using the controller parameters that yield the lowest DEL$_t$. Figure 5.19a presents amplitudes below 100 [MNm], while Fig. 5.19b shows amplitudes above 100 [MNm] for $\boldsymbol{x}_{\mathrm{cs}} = [42, 51, 57, 65]$ [m], with the histogram for $x_{\mathrm{cs}} = 51$ highlighted.

From Fig. 5.19a, it is evident that the platform with $x_{\mathrm{cs}} = 35$ [m] has more high-amplitude load cycles, whereas the platform with $x_{\mathrm{cs}} = 65$ [m] has a higher count of cycles with mean amplitudes between 100–250 [MNm]. Consequently, these systems experience higher cumulative damage and



(a) Amplitude and count in the range of 0–100 [MNm].

(b) Amplitude and count in the range of 100–350 [MNm].

Figure 5.19: The amplitude and count for the different load cycles from all thirty simulations used to calculate the DEL_t for the different values of $\boldsymbol{x}_{cs}$.

DEL_t. Platforms with $x_{cs} = [51, 57]$ [m] have more load cycles with lower amplitudes, while $x_{cs} = 42$ [m] has a higher count of cycles in the 0–100 [MNm] range. The key factor influencing DEL_t is the number of cycles in the 100–250 [MNm] range. The platform with $x_{cs} = 51$ [m] satisfies both criteria—high count of low-amplitude cycles and low count of high-amplitude cycles—resulting in the lowest DEL_t.

The combined Pareto front, containing the non-dominated points across all values of x_{cs}, is shown in Fig. 5.20. All points from the individual Pareto fronts are part of this combined front. This further validates selecting $x_{cs} = 51$ [m] as the optimal platform design, as it contributes the highest number of points to the combined Pareto front.

For this platform, the optimal controller parameters must be sub-selected. Consider points labeled 1–5 in Fig. 5.20: point 1 has the lowest DEL_t and highest $\omega_{g,\text{std}}$, while point 5 has the lowest $\omega_{g,\text{std}}$ and highest DEL_t, illustrating the trade-off between these two objectives.

Comparing point 2 to point 1, we observe a 1.56% increase in DEL_t but a 30.46% increase in $\omega_{g,\text{std}}$ relative to point 5. For point 23, the DEL_t increases by 1.99% and $\omega_{g,\text{std}}$ increases by 15.78%. Similarly, comparing point 4 to points 1 and 5, there is a 4.19% increase in DEL_t and a 12.93%

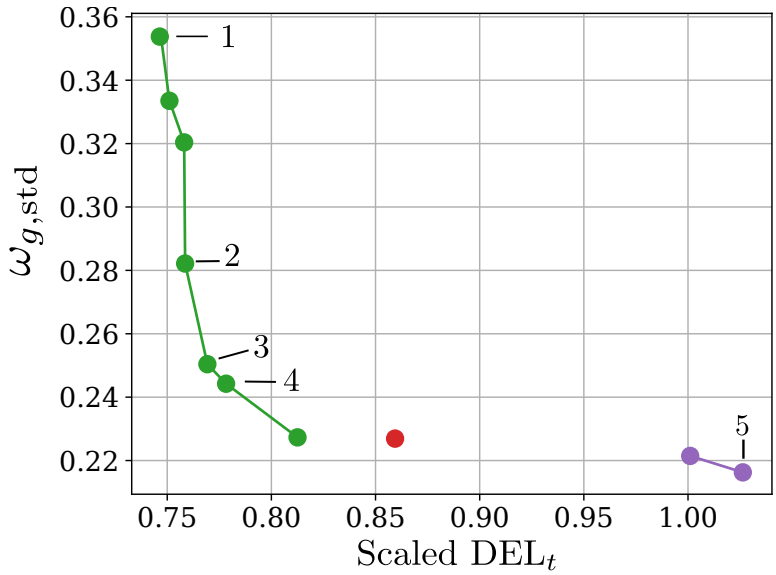


Figure 5.20: Pareto front obtained using the CCD study between DEL_t vs. $\omega_{g,\mathrm{std}}$ from Fig. 5.17.

Table 5.2: DEL_t and $\omega_{g,\mathrm{std}}$ values for the five points labeled in Fig. 5.20.

Point	DEL_t	$\omega_{g,\mathrm{std}}$	$100 \times \frac{(\mathrm{DEL}_t - \mathrm{DEL}_{t,\min})}{\mathrm{DEL}_{t,\min}}$ [%]	$100 \times \frac{(\omega_{g,\mathrm{std}} - \omega_{g,\mathrm{std},\min})}{\omega_{g,\mathrm{std},\min}}$ [%]
1	0.7469	0.3541	0.00	63.77
2	0.7585	0.2821	1.56	30.46
3	0.7692	0.2504	1.99	15.78
4	0.7782	0.2442	4.19	12.93
5	1.0263	0.2162	37.40	0.00

increase in $\omega_{g,\mathrm{std}}$. Considering all these points, point 3 achieves a favorable trade-off between the two objectives.

The corresponding optimal design values are:

$$x_{\mathrm{cs,opt}} = 51 \text{ [m]} \tag{5.9}$$

$$\boldsymbol{x}_{c,\mathrm{opt}} = [0.1931, 2.885, -14.464, 0.2185]^T \tag{5.10}$$

This study highlights the inherent trade-offs between DEL_t and $\omega_{g,\mathrm{std}}$ and demonstrates the value of the control co-design (CCD) approach in systematically exploring these trade-offs.

The key takeaway from this chapter is that the DFSM approach can be applied across different use cases. It enables the construction of low-fidelity models for a variety of wind and marine turbine systems while maintaining accuracy in key performance signals. Moreover, when used within a multi-fidelity framework, the DFSM facilitates the identification of optimal solutions closer to high-fidelity results with significantly fewer function evaluations. This capability allows for efficient multi-objective optimization studies, both for controller tuning alone and from a control co-design perspective, providing valuable insights into performance trade-offs.

Chapter 6

Conclusion

This chapter summarizes the research carried out in this dissertation and provides directions for future research. The first section, Sec. 6.1, provides a summary and key takeaways, and the second section, Sec. 6.2, presents directions for future research.

6.1 Summary

This dissertation outlines the main issues researchers and engineers face when designing wind and marine turbines using design optimization studies and proposes solutions to them. There is a wide variety of models that can be used for the design of wind and marine turbines. These models can be classified from low-fidelity to high-fidelity based on the complexity of the encoded physics. For the process to be effective, models that balance accuracy with computational expense are required. Typically, models that are detailed in terms of physics are preferred as they can capture the behavior and response of the individual subsystems and the interactions with other subsystems, but are cheaper compared to high-fidelity CFD/FEA studies. These models are called aero-servo-hydro-elastic models.

Recently, these models have been linked to optimizers to more efficiently explore the design space associated with key parameters. The use of optimizers allows researchers to identify designs that satisfy different performance constraints. Researchers have also identified the importance of the controller on the system performance. Since these systems must function efficiently in highly stochastic offshore environments, a controller is necessary to ensure that the loads on the turbine are minimized and that it produces sufficient energy to remain economically viable. These optimization studies have allowed engineers to identify optimal designs that minimize the levelized cost of energy (LCOE) associated with these systems, which would help in the wide-scale adoption of these technologies.

However, extending these design optimization studies to include more design variables requires overcoming certain drawbacks. Primarily, it can take from several minutes to hours to evaluate these mid-fidelity models. In design optimization studies, the models might need to be evaluated several hundred times. Therefore, over the course of these evaluations, the computational cost adds up, making them expensive. Additionally, these mid-fidelity models have been designed to evaluate system performance and not necessarily for design optimization studies. For example, gradient information for key design variables, which many optimizers require, is not available using these mid-fidelity models. Additionally, with these turbine systems, there are multiple conflicting performance metrics of interest. Designers are also interested in understanding the trade-offs associated with these metrics, instead of identifying optimal parameters that result in a single optimal design. These studies require multi-objective optimization studies, which further involve solving a set of optimization problems. Additionally, for marine turbines, only recently has an open-source modeling tool been made available for system simulation. In order to effectively simulate these systems, a controller needs to be developed for them.

6.1.1 Development of an Optimal Controller for Marine Turbines

ROSCO, an open-source tool developed by NREL, is used to derive the necessary controllers for a floating configuration of the Reference Model 1 (RM1) turbine. An optimization problem was formulated to identify the tuning parameters associated with the blade pitch controller to minimize the tower-base damage equivalent loading (DEL). The parameters identified by the controller result in a 20% reduction in the DEL.

6.1.2 Different Approaches for Constructing Low-fidelity Models

There are different types of surrogate modeling approaches that have been used to offset the computational expense associated with these aero-servo-hydro-elastic models. For use in design optimization studies for wind and marine turbines, in place of these aero-servo-hydro-elastic models, low-fidelity modeling approaches that can be trained on time-series simulations of these models and be used to predict the key system response for a variety of load cases would be ideal.

Several such modeling approaches have been presented, ranging from traditional subspace system identification approaches to recurrent neural networks such as long short-term memory networks or gated recurrent units. In addition to these, a set of approaches titled derivative function surrogate model (DFSM) has also been utilized to approximate the dynamics of wind turbines.

6.1.3 Derivative Function Surrogate Models

DFSM approaches assume that the model response can be approximated as a continuous-time state-derivative function. Typically, previous studies that have utilized this approach assume that the high-fidelity model is available in a form where it can be evaluated directly. A space-filling sampling approach is used to evaluate the high-fidelity function, and approaches such as radial basis functions (RBF) or Gaussian process models (GPR) are used to construct a surrogate between the inputs and outputs. However, in the case studies discussed in this dissertation, such an assumption does not hold. Instead, the model will have to be constructed using input-output time-series data. In order to do that, we introduce a novel method to construct a DFSM using time-series data. This approach is inspired by system identification approaches. The DFSM is assumed to have a linear parameter varying (LPV) state-space model, with wind speed being the main parameter. This LPV model is constructed by identifying individual linear time-invariant (LTI) models corresponding to different wind speeds, and interpolating over them to get a continuous model. The individual LTI models are, in turn, identified using an optimization problem formulated to identify the model parameters that minimize the predicted state derivatives and outputs. A hybrid optimization scheme is used to solve the problem, where a genetic algorithm (GA) is used to identify a good starting point, and a gradient-based solver is used to identify the local minima starting from the point identified by the GA. A constraint is added to this problem to ensure that the identified LTI model is stable. Specifically, the constraint specifies that the maximum value of the real parts of the eigenvalues of the state matrix should be below zero. Certain properties of wind/marine turbine dynamics are leveraged to solve this problem more efficiently. Instead of solving a hybrid optimization problem for each individual LTI model, we use the model identified for one wind speed

as the starting point for the gradient-based optimizer for the next wind speed. Since this model satisfies the constraint, the gradient-based solver efficiently finds the associated model parameters.

6.1.4 Comparison With Other Data-Driven Modeling Approaches

We construct low-fidelity models for a wind turbine system using all three of these different approaches. Once constructed, these models are validated using closed-loop control simulations by comparing the key control and output signals identified using these low-fidelity models to the high-fidelity model for ten different test simulations. Different signals are compared, such as the blade pitch, generator power, and the tower-base moment. This study showed that all three approaches could be used to build a low-fidelity model of the systems. However, there is a wide difference in accuracy and consistency between these three approaches. The model constructed using the system identification approach assumes that the system response can be approximated using a discrete-time linear time-invariant model (LTI). Therefore, this model is the fastest to simulate, taking around 3 seconds on average. It is also possible to ensure that this model is stable. However, there is some error in the predicted controls and outputs. Additionally, it is not possible to extend this approach to construct an LPV model, as the model order that provides the most accurate simulations for different wind speeds is different. Next, the LSTM model, which was trained to predict the key outputs given the controls, is tested. This model takes the longest to simulate compared to the other approaches, but the variance in the output is less compared to the system identification model. This approach is also less accurate in terms of the mean squared error (MSE). Additionally, it is possible to extend this approach to predict the response for other wind speeds, but it is not possible to ensure that the system response is stable. For the DFSM approach, it takes an average of around 35 seconds to simulate the system. DFSM outperforms the other two approaches in terms of accuracy and variance in prediction. Unlike the previous two approaches, the DFSM can be used to predict the response for multiple wind speeds and ensure that the system response is stable. Because of this, the DFSM is used as the low-fidelity model for design optimization studies. Using the approach outlined, the DFSM is constructed, and its

accuracy in predicting the key system outputs is demonstrated for a marine turbine system and other wind turbine systems.

6.1.5 Case Studies

Open-loop optimal control.

The efficacy of using the DFSM in optimal control studies is then demonstrated. First, the DFSM is used to formulate and solve different open-loop optimal control problems for wind and marine turbine systems, respectively. For the wind turbine system, a problem is formulated to identify the optimal control trajectories that maximize power production while satisfying constraints on the platform pitching motion. For the marine turbine system, a problem is formulated to identify the optimal control trajectories that maximize power for different limits on the generator speed. The results from both these studies show that the DFSM can be used to identify optimal solutions for these problems that are in line with optimal control results obtained using linearized OpenFAST models. Because the DFSM used for both these studies has an LPV structure, the optimal control problems can be formulated and solved efficiently.

Closed-loop optimal control.

The efficacy of using the DFSM for closed-loop optimal control studies, with the goal of identifying the controller parameters that minimize the fatigue loads on the system, is then demonstrated. The validation studies carried out for the DFSM assume that the underlying controller parameters do not change. To verify if the DFSM can still accurately predict the controls and outputs for closed-loop simulations when the controller parameters change, a design of experiments study is carried out. A full factorial sampling scheme is used to sample the design space associated with the controller parameters, and at each sample, the low- and high-fidelity models are simulated for different load cases, and a contour plot of the key fatigue loads is plotted against the design space. The results show that the DFSM can capture the key trends in the design space. However, there is a difference in the predicted values, which would prohibit the direct use of the DFSM for closed-loop controller optimization studies. This motivates the use of multi-fidelity approaches to perform the

optimization studies. A trust-region-based multi-fidelity approach is used. Controller optimization studies are carried out using this approach, and the results show that a multi-fidelity approach utilizing the DFSM as the low-fidelity model can find a solution that is closer to the high-fidelity model values while balancing the computational cost.

Multi-objective optimization studies.

Finally, multi-objective optimization studies are carried out to identify the trade-offs between different conflicting objectives. First, a multi-objective problem is solved to identify the Pareto front utilizing the controller parameters as the only set of design variables. A weighted-objective method is used to carry out these multi-objective optimization studies. Each sub-problem is solved using the multi-fidelity approach outlined previously. The multi-objective optimization study seeks to identify the trade-offs between minimizing fatigue damage on the system and improving power quality by minimizing generator speed standard deviation. The results show a clear trade-off between these two objectives. Additionally, utilizing the multi-fidelity approach allows the user to identify points that are closer to the high-fidelity optima with fewer high-fidelity function calls.

This study is then extended to include plant variables as well. The column spacing associated with the semisubmersible platform is varied using a sensitivity study, and the multi-objective optimization study is carried out for different values of this variable. The results show that the two objectives also vary significantly with the plant variables. This study also extends the Pareto front, and the optimal point on this solution set that balances both objectives is identified.

All these studies show that the DFSM approach can balance accuracy and computational cost and be utilized for a variety of use cases.

6.2 Future Work

There are several possible directions that could be explored to expand the studies discussed in this dissertation. These directions can be grouped under three broad headings, which are discussed next.

6.2.1 Exploring Other Data-Driven Modeling Approaches

The first set of studies that would be valuable is to explore other data-driven modeling approaches. Different approaches from system identification or deep learning could be used. For example, several studies have explored how to construct nonlinear system identification models [162]. However, these approaches have been shown to be effective for simple systems and theoretical examples only. Identifying and exploring how these approaches can be used for modeling wind and marine turbines would be helpful.

In addition to system identification approaches, deep learning approaches such as continuous-time echo-state networks and more sophisticated architectures of LSTM networks could also be explored. As seen earlier, there are two main drawbacks to using deep neural networks. Deep learning approaches that are efficient for use cases involving iterative predictions are important. In addition, the ability to guarantee the open-loop and closed-loop stability of the system would also be necessary. Approaches such as physics-informed neural networks (PINNs) could be useful in training networks that can guarantee stability [163, 164]. Additionally, deep state-space models, which combine aspects of system identification and deep learning methods, could also be explored to approximate the response of these systems [165–167]. These approaches, similar to system identification approaches, have been explored for simple systems. Identifying their efficacy in modeling wind and marine turbines would improve the state-of-the-art for both these models and wind/marine turbine applications. Additionally, networks such as continuous-time echo-state networks (CTESN) or neural ordinary differential equations (nODE) also offer an approach to model continuous-time systems [168–170].

6.2.2 Construction of the DFSM

Several improvements could be made to how the DFSM is constructed in this dissertation. Currently, a semi-automated approach is used to identify the DFSM. The model parameters are identified using an optimization study, but there are some hyperparameters associated with this process as well. These hyperparameters, for example, include the wind speed at which the iden-

tification process is started, the number of seeds used per wind speed to construct the model, the maximum value of the real parts of the eigenvalues, etc. A hyperparameter optimization approach could be used to automate the process. In addition, providing derivative information regarding the objective and the constraints could also be useful. However, since the constraint involves the eigenvalues of the state matrix and is highly nonlinear, advanced approaches are required to calculate the derivative accurately [171].

The current approach tests for open-loop stability of the system. However, for some values of the hyperparameters, the system can be unstable for some values of the control parameters. Therefore, in addition to open-loop stability, closed-loop stability must also be tested. The identification procedure used for the DFSM has been tested empirically by constructing a DFSM for multiple systems and simulating them. However, the process does not have any theoretical guarantees. Such guarantees are often customary with other system identification approaches. Understanding the model identification process and developing a unified theory behind it would allow users to apply this process to other systems as well. This could also help in the construction of continuous-time system identification approaches.

Synthetic simulations are used to construct the DFSM in this dissertation. Being able to utilize real-world data to construct these models would be helpful. Integrating real-world data in the construction of these models would enable the use of the DFSM approach for digital-twin applications of these turbines as well [172]. However, real-world data can be noisy. Obtaining real-world simulations and understanding how to process them appropriately to construct the DFSM would be key. Additionally, multiple simulations are used to construct the model. It would be interesting to explore how the DFSM can be constructed using a single persistently exciting signal that explores the associated state-space [173]. The current approach could also be used to identify the requirements of such a signal. This would make the process of constructing the DFSM less computationally expensive.

6.2.3 Case Studies

Finally, the DFSM can be explored in more use cases than the ones discussed in this dissertation and in the wind turbine design optimization literature. Its flexibility, in terms of accuracy and computational speedup, could be leveraged to carry out different studies that would be inefficient to carry out using only a high-fidelity model. One such study is using the open-loop optimal control results to train closed-loop controllers. Since the DFSM can be used for both open-loop and closed-loop optimal control studies, it would be useful in identifying different strategies that could be used to train the closed-loop controller and the trade-offs between them [174]. Because of the LPV structure of the DFSM, it could also be used for robust-control studies, which would not be feasible using the high-fidelity model [175].

The DFSM could also be utilized in investigating novel FOWT concepts. For example, multi-loop control systems, actively controlled mooring lines, and active ballast control are different concepts that have been explored to make FOWTs viable [30, 31, 176–178]. The DFSM could be used to aid these investigations.

The DFSM can further be used for different CCD studies. Some of these possible CCD studies are listed below:

1. For marine turbines, when utilizing fixed-pitch control schemes, the blade shape is specifically designed to induce stall at higher flow speeds. This results in higher loads on the turbine blades. To ensure that there is no system failure, the blades of these turbines need to be designed to withstand these loads. A CCD study could be used to obtain a system that balances the trade-offs between rotor design and controller. Because of the computational time involved in simulating the system, a DFSM approach could be applied for CCD studies related to this system.
2. Marine turbines, because they operate in a heavier fluid, are subject to larger loads. In addition to these loads, floating marine turbines are also subject to additional loads because of the motion of the system. The optimal design of the rotor of a fixed-bottom turbine and

a floating turbine would be different to account for these different loads. Designing both systems using a CCD approach and comparing their designs would be useful.

3. Recently, researchers have realized the effect that the controller has on the design of the mooring system of offshore turbines. Formulating and solving different CCD problems that explore the trade-offs between these two systems would be useful.
4. Different supporting platform types have different dynamics and different requirements for the tower and the mooring system. Additionally, the tower and mooring system design would also depend on the controller. Therefore, formulating and solving a CCD that explores the trade-off between the different platform types and the tower and mooring system design would be useful.

Utilizing an efficient global-optimization (EGO) based approach as part of the multi-fidelity optimization studies would help overcome the issues with local minima encountered when using the trust-region-based multi-fidelity approach [179–181]. Additionally, investigations can be carried out to utilize the DFSM as a surrogate to obtain derivative information for the high-fidelity model. As mentioned previously, it can be difficult to obtain derivative information for high-fidelity models. The DFSM could be utilized to construct approximate derivative information. This derivative information could be used to carry out optimization studies using gradient-based solvers.

Bibliography

[1] R. McKenna *et al.*, "System impacts of wind energy developments: Key research challenges and opportunities," *Joule*, vol. 9, no. 1, p. 101799, Jan. 2025, doi: 10.1016/j.joule.2024.11.016

[2] "Yearly Electricity Data," [Online], url: https://ember-energy.org/data/yearly-electricity-data/

[3] "Global Wind Atlas," [Online], url: https://globalwindatlas.info/en/

[4] R. Joshi, D. von Terzi, and R. Schmehl, "System design and scaling trends in airborne wind energy demonstrated for a ground-generation concept," *Wind Energy Science*, vol. 10, no. 4, pp. 695–718, Apr. 2025, doi: 10.5194/wes-10-695-2025

[5] Q. Hassan, S. Algburi, A. Z. Sameen, H. M. Salman, and M. Jaszczur, "A review of hybrid renewable energy systems: Solar and wind-powered solutions: Challenges, opportunities, and policy implications," *Results in Engineering*, vol. 20, p. 101621, Dec. 2023, doi: 10.1016/j.rineng.2023.101621

[6] J. J. Thomas *et al.*, "Wind turbine design optimization for hydrogen production," *IET Conference Proceedings*, vol. 2024, no. 2, pp. 177–184, Jun. 2024, doi: 10.1049/icp.2024.1835

[7] J. M. Jonkman, "Dynamics modeling and loads analysis of an offshore floating wind turbine," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-500-41958, Nov. 2007, doi: 10.2172/921803

[8] "Wind Turbines: the Bigger, the Better," [Online], url: https://www.energy.gov/eere/articles/wind-turbines-bigger-better

[9] J. Jonkman, A. Wright, G. Barter, M. Hall, J. Allison, and D. R. Herber, "Functional requirements for the WEIS toolset to enable controls co-design of floating offshore wind turbines,"

in *ASME International Offshore Wind Technical Conference*, no. IOWTC2021-3533, Feb. 2021, doi: 10.1115/iowtc2021-3533

[10] P. Veers *et al.*, "Grand challenges in the design, manufacture, and operation of future wind turbine systems," *Wind Energy Science*, vol. 8, no. 7, pp. 1071–1131, Jul. 2023, doi: 10.5194/wes-8-1071-2023

[11] M. Garcia-Sanz, "Control co-design: An engineering game changer," *Advanced Control for Applications*, vol. 1, no. 1, Oct. 2019, doi: 10.1002/adc2.18

[12] ——, "A metric space with LCOE isolines for research guidance in wind and hydrokinetic energy systems," *Wind Energy*, vol. 23, no. 2, pp. 291–311, Nov. 2019, doi: 10.1002/we.2429

[13] V. S. Neary *et al.*, "Methodology for design and economic analysis of marine energy conversion (MEC) technologies," Sandia National Laboratories, Tech. Rep. SAND2014-3561C; 517036, Mar. 2014, url: https://www.osti.gov/biblio/1143279

[14] "Offshore Wind Prototypes," [Online], url: https://www.dnv.com/publications/prototype-testing-of-floating-offshore-wind-turbines-96456/.

[15] "Marine Energy Prototype," [Online], url: https://tethys.pnnl.gov/project-sites/kvalsund-tidal-turbine-prototype2

[16] "Project-Blue-Economy," [Online], url: https://www.energy.gov/sites/default/files/2022-07/wpto-pbe-brochure-july2022.pdf

[17] G. S. Bir, M. J. Lawson, and Y. Li, "Structural design of a horizontal-axis tidal current turbine composite blade," in *ASME International Conference on Ocean, Offshore and Arctic Engineering*, no. OMAE2011-50063, Jun. 2011, doi: 10.1115/omae2011-50063

[18] L. Kilcher, M. Fogarty, and M. Lawson, "Marine energy in the United States: An overview of opportunities," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5700-78773, Feb. 2021, doi: 10.2172/1766861

[19] A. Salmon, B. Meadows, L. Kilcher, and B. Hirsch, "Igiugig's journey towards sustainability," National Renewable Energy Laboratory, Tech. Rep. NREL/BR-5700-83816, Dec. 2022, url: https://www.osti.gov/biblio/1903772

[20] L. Kilcher, K. Haas, and A. Muscalus, "Tidal resource gaps analysis technical report," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5700-86692, Sep. 2023, doi: 10.2172/2007002

[21] R. Meadows *et al.*, "Feasibility study for renewable energy technologies in alaska offshore waters," U.S. Department of the Interior, Bureau of Ocean Energy Management, Tech. Rep. OCS Study BOEM 2023-076, Dec. 2023, doi: 10.2172/2322404

[22] J. R. McVey, L. Kilcher, and Z. Yang, "Tidal energy resource characterization measurements at cook inlet's east foreland: Velocity and turbulence," *Renewable Energy*, vol. 256, p. 123990, Jan. 2026, doi: 10.1016/j.renene.2025.123990

[23] L. Kilcher and R. Thresher, "Marine hydrokinetic energy site identification and ranking methodology part I: Wave energy," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5000-66038, Oct. 2016, doi: 10.2172/1330617

[24] M. Deb, Z. Yang, T. Wang, and L. Kilcher, "Turbulence modeling to aid tidal energy resource characterization in the Western Passage, Maine, USA," *Renewable Energy*, vol. 219, p. 118694, Dec. 2023, doi: 10.1016/j.renene.2023.04.100

[25] T. Lake *et al.*, "Strain gauge measurements on a full scale tidal turbine blade," *Renewable Energy*, vol. 170, pp. 985–996, Jun. 2021, doi: 10.1016/j.renene.2021.01.137

[26] K. Van Ness, C. Hill, J. Burnett, A. Aliseda, and B. Polagye, "Experimental comparison of blade pitch and speed control strategies for horizontal-axis current turbines," *Journal of*

Ocean Engineering and Marine Energy, vol. 7, no. 1, pp. 83–96, Feb. 2021, doi: 10.1007/s40722-021-00188-w

[27] W. J. Shaw *et al.*, "Scientific challenges to characterizing the wind resource in the marine atmospheric boundary layer," *Wind Energy Science*, vol. 7, no. 6, pp. 2307–2334, Nov. 2022, doi: 10.5194/wes-7-2307-2022

[28] A. Clifton *et al.*, "Grand challenges in the digitalisation of wind energy," *Wind Energy Science*, vol. 8, no. 6, pp. 947–974, Jun. 2023, doi: 10.5194/wes-8-947-2023

[29] M. O'Malley *et al.*, "Grand challenges of wind energy science – meeting the needs and services of the power system," *Wind Energy Science*, vol. 9, no. 11, pp. 2087–2112, Nov. 2024, doi: 10.5194/wes-9-2087-2024

[30] D. Stockhouse, M. Phadnis, E. Grant, K. Johnson, R. Damiani, and L. Pao, "Control of a floating wind turbine on a novel actuated platform," in *American Control Conference*, Jun. 2022, doi: 10.23919/acc53348.2022.9867498

[31] J. D. Dinius, R. Damiani, K. Johnson, E. Grant, L. Y. Pao, and M. Phadnis, "Control actuation options for the SpiderFLOAT floating offshore wind substructure," in *AIAA SCITECH 2022 Forum*, no. AIAA 2022-2295, Jan. 2022, doi: 10.2514/6.2022-2295

[32] L. Y. Pao *et al.*, "Control co-design of 13 MW downwind two-bladed rotors to achieve 25% reduction in levelized cost of wind energy," *Annual Reviews in Control*, vol. 51, pp. 331–343, 2021, doi: 10.1016/j.arcontrol.2021.02.001

[33] N. Abbas, D. Zalkind, L. Pao, and A. Wright, "A reference open-source controller for fixed and floating offshore wind turbines," *Wind Energy Science*, vol. 7, no. 1, pp. 53–73, Apr. 2021, doi: 10.5194/wes-2021-19

[34] Q. Pan, W. Yu, and P. W. Cheng, "Simplified modeling of floating offshore wind farms with shared mooring line configurations," *Ocean Engineering*, vol. 327, p. 121018, May 2025, doi: 10.1016/j.oceaneng.2025.121018

[35] J. Liang, B. Kato, and Y. Wang, "Constructing simplified models for dynamic analysis of monopile-supported offshore wind turbines," *Ocean Engineering*, vol. 271, p. 113785, Mar. 2023, doi: 10.1016/j.oceaneng.2023.113785

[36] J. López-Queija, E. Robles, J. I. Llorente, I. Touzon, and J. López-Mendia, "A simplified modeling approach of floating offshore wind turbines for dynamic simulations," *Energies*, vol. 15, no. 6, p. 2228, Mar. 2022, doi: 10.3390/en15062228

[37] T. T. Tran, H. Ross, W. Wiley, L. Wang, and S. Sirnivas, "High- and mid-fidelity modeling comparison for a floating marine turbine system," in *ASME International Conference on Ocean, Offshore and Arctic Engineering*, no. OMAE2024-128251, Jun. 2024, doi: 10.1115/omae2024-128251

[38] H. Darling *et al.*, "OC6 phase IV: Validation of CFD models for Stiesdal TetraSpar floating offshore wind platform," *Wind Energy*, vol. 28, no. 1, Dec. 2024, doi: 10.1002/we.2966

[39] R. P. Tavares, V. Bouwman, and W. Van Paepegem, "Finite element analysis of wind turbine blades subjected to torsional loads: Shell vs solid elements," *Composite Structures*, vol. 280, p. 114905, Jan. 2022, doi: 10.1016/j.compstruct.2021.114905

[40] P. M. O. Gebraad *et al.*, "Wind plant power optimization through yaw control using a parametric model for wake effects-a CFD simulation study: Wind plant optimization by yaw control using a parametric wake model," *Wind Energy*, vol. 19, no. 1, pp. 95–114, Dec. 2014, doi: 10.1002/we.1822

[41] B. G. Knight *et al.*, "Multifidelity CFD analysis for ducted hydrokinetic turbine design," *Renewable Energy*, vol. 252, p. 123307, Oct. 2025, doi: 10.1016/j.renene.2025.123307

[42] L. Wang, R. Quant, and A. Kolios, "Fluid structure interaction modelling of horizontal-axis wind turbine blades based on CFD and FEA," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 158, pp. 11–25, Nov. 2016, doi: 10.1016/j.jweia.2016.09.006

[43] IEC, "Wind energy generation systems – part 3-2: Design requirements for floating offshore wind turbines," International Electrotechnical Commission, Technical Specification 61400-3-2:2025, Jan. 2025.

[44] L. Y. Pao and K. E. Johnson, "A tutorial on the dynamics and control of wind turbines and wind farms," in *American Control Conference*, 2009, doi: 10.1109/acc.2009.5160195

[45] L. Y. Pao, M. Pusch, and D. S. Zalkind, "Control co-design of wind turbines," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, no. 1, pp. 201–226, Jul. 2024, doi: 10.1146/annurev-control-061423-101708

[46] E. García, A. Correcher, E. Quiles, F. Tamarit, and F. Morant, "Control and supervision requirements for floating hybrid generator systems," *International Journal of Environmental Research and Public Health*, vol. 19, no. 19, p. 12781, Oct. 2022, doi: 10.3390/ijerph191912781

[47] H. Ross *et al.*, "Development of a control co-design modeling tool for marine hydrokinetic turbines," in *ASME International Mechanical Engineering Congress and Exposition*, no. IMECE2022-94483, Oct. 2022, doi: 10.1115/imece2022-94483

[48] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*. Cambridge University Press, Nov. 2021, doi: 10.1017/9781108980647

[49] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer New York, 2006, doi: 10.1007/978-0-387-40065-5

[50] S. Forrest, "Genetic algorithms: Principles of natural selection applied to computation," *Science*, vol. 261, no. 5123, pp. 872–878, Aug. 1993, doi: 10.1126/science.8346439

[51] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017

[52] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948, doi: 10.1109/icnn.1995.488968

[53] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671

[54] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*. Springer Netherlands, 1994, pp. 51–67, doi: 10.1007/978-94-015-8330-5_4

[55] N. Abbas, "Enabling automated control co-design for wind turbines," Ph.D. Dissertation, University of Colorado at Boulder, Boulder, CO, USA, 2022.

[56] S. Watson *et al.*, "Future emerging technologies in the wind power sector: A European perspective," *Renewable and Sustainable Energy Reviews*, vol. 113, p. 109270, Oct. 2019, doi: 10.1016/j.rser.2019.109270

[57] X. Guo, J. Yang, W. Lu, and X. Li, "Dynamic responses of a floating tidal turbine with 6-DOF prescribed floater motions," *Ocean Engineering*, vol. 165, pp. 426–437, Oct. 2018, doi: 10.1016/j.oceaneng.2018.07.017

[58] J. C. Mankins, "Technology readiness levels," White Paper, Apr. 1995.

[59] J. M. Jonkman, "Influence of control on the pitch damping of a floating wind turbine," in *AIAA Aerospace Sciences Meeting and Exhibit*, no. AIAA 2008-1306, Jan. 2008, doi: 10.2514/6.2008-1306

[60] S. Butterfield, W. Musial, J. Jonkman, and P. Sclavounos, "Engineering challenges for floating offshore wind turbines," in *Copenhagen Offshore Wind Conference*, Oct. 2007, url: https://www.osti.gov/biblio/917212

[61] J. M. Jonkman and D. Matha, "Dynamics of offshore floating wind turbines—analysis of three concepts," *Wind Energy*, vol. 14, no. 4, pp. 557–569, Jan. 2011, doi: 10.1002/we.442

[62] K. P. Thiagarajan and H. J. Dagher, "A review of floating platform concepts for offshore wind energy generation," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 136, no. 2, Mar. 2014, doi: 10.1115/1.4026607

[63] "OpenFAST," [Online], url: https://github.com/OpenFAST/openfast

[64] "Bladed," [Online], url: https://store.veracity.com/bladed

[65] "HAWC2," [Online], url: https://www.hawc2.dk/about-hawc2

[66] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW reference wind turbine for offshore system development," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-500-38060, Feb. 2009, doi: 10.2172/947422

[67] C. Bak *et al.*, "The DTU 10-MW reference wind turbine," in *Danish Wind Power Research*, 2013, url: https://orbit.dtu.dk/en/publications/the-dtu-10-mw-reference-wind-turbine

[68] E. Gaertner *et al.*, "IEA wind TCP task 37: definition of the IEA 15-megawatt offshore reference wind turbine," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5000-75698, Mar. 2020, doi: 10.2172/1603478

[69] F. Zahle *et al.*, "Definition of the IEA wind 22-megawatt offshore reference wind turbine," Technical University of Denmark, Tech. Rep., 2024, doi: 10.11581/DTU.00000317

[70] A. K. Sundarrajan, Y. Hoon Lee, J. T. Allison, D. S. Zalkind, and D. R. Herber, "Open-loop control co-design of semisubmersible floating offshore wind turbines using linear parameter-varying models," *Journal of Mechanical Design*, vol. 146, no. 4, Nov. 2023, doi: 10.1115/1.4063969

[71] J. M. Hegseth, E. E. Bachynski, and J. R. R. A. Martins, "Design optimization of spar floating wind turbines considering different control strategies," *Journal of Physics: Conference Series*, vol. 1669, no. 1, p. 012010, Oct. 2020, doi: 10.1088/1742-6596/1669/1/012010

[72] S. Bayat, Y. H. Lee, and J. T. Allison, "Nested control co-design of a spar buoy horizontal-axis floating offshore wind turbine," *Ocean Engineering*, vol. 328, p. 121037, Jun. 2025, doi: 10.1016/j.oceaneng.2025.121037

[73] N. J. Abbas, J. Jasa, D. S. Zalkind, A. Wright, and L. Pao, "Control co-design of a floating offshore wind turbine," *Applied Energy*, vol. 353, p. 122036, Jan. 2024, doi: 10.1016/j.apenergy.2023.122036

[74] D. Zalkind and P. Bortolotti, "Control co-design studies for a 22 MW semisubmersible floating wind turbine platform," *Journal of Physics: Conference Series*, vol. 2767, no. 8, p. 082020, Jun. 2024, doi: 10.1088/1742-6596/2767/8/082020

[75] D. Stockhouse *et al.*, "A tutorial on the control of floating offshore wind turbines: Stability challenges and opportunities for power capture," *IEEE Control Systems*, vol. 44, no. 5, pp. 28–57, Oct. 2024, doi: 10.1109/mcs.2024.3433208

[76] L. Brandetti, S. P. Mulders, R. Merino-Martinez, S. Watson, and J.-W. van Wingerden, "Multi-objective calibration of vertical-axis wind turbine controllers: balancing aero-servo-elastic performance and noise," *Wind Energy Science*, vol. 9, no. 2, pp. 471–493, Mar. 2024, doi: 10.5194/wes-9-471-2024

[77] L. Brandetti, S. P. Mulders, Y. Liu, S. Watson, and J.-W. van Wingerden, "Analysis and multi-objective optimisation of wind turbine torque control strategies," *Wind Energy Science*, vol. 8, no. 10, pp. 1553–1573, Oct. 2023, doi: 10.5194/wes-8-1553-2023

[78] M. Lara, F. Vázquez, J.-W. van Wingerden, S. P. Mulders, and J. Garrido, "Multi-objective optimization of individual pitch control for blade fatigue load reductions for a 15 MW wind turbine," in *European Control Conference*, Jun. 2024, pp. 669–674, doi: 10.23919/ecc64448.2024.10590830

[79] L. Zhang, S.-q. Wang, Q.-h. Sheng, F.-m. Jing, and Y. Ma, "The effects of surge motion of the floating platform on hydrodynamics performance of horizontal-axis tidal current tur-

bine," *Renewable Energy*, vol. 74, pp. 796–802, Feb. 2015, doi: 10.1016/j.renene.2014.09.002

[80] T. Dewhurst, M. R. Swift, M. Wosnik, K. Baldwin, J. DeCew, and M. Rowell, "Dynamics of a floating platform mounting a hydrokinetic turbine," *Marine Technology Society Journal*, vol. 47, no. 4, pp. 45–56, Jul. 2013, doi: 10.4031/mtsj.47.4.13

[81] T. de Jesus Henriques *et al.*, "The effects of wave–current interaction on the performance of a model horizontal axis tidal turbine," *International Journal of Marine Energy*, vol. 8, pp. 17–35, Dec. 2014, doi: 10.1016/j.ijome.2014.10.002

[82] R. Martinez *et al.*, "Analysis of the effects of control strategies and wave climates on the loading and performance of a laboratory scale horizontal axis tidal turbine," *Ocean Engineering*, vol. 212, p. 107713, Sep. 2020, doi: 10.1016/j.oceaneng.2020.107713

[83] S. Tatum *et al.*, "Wave–current interaction effects on tidal stream turbine performance and loading characteristics," *International Journal of Marine Energy*, vol. 14, pp. 161–179, Jun. 2016, doi: 10.1016/j.ijome.2015.09.002

[84] S. Ordonez-Sanchez *et al.*, "Analysis of a horizontal-axis tidal turbine performance in the presence of regular and irregular waves using two control strategies," *Energies*, vol. 12, no. 3, p. 367, Jan. 2019, doi: 10.3390/en12030367

[85] W. Wiley, H. Ross, A. K. Sundarrajan, , and T. T. Tran, "Design and modeling of an open-source baseline floating marine turbine," in *University Marine Energy Research Community Conference*, Oct. 2023, url: https://www.osti.gov/biblio/2202791

[86] A. P. Deshmukh and J. T. Allison, "Design of dynamic systems using surrogate models of derivative functions," *Journal of Mechanical Design*, vol. 139, no. 10, Aug. 2017, doi: 10.1115/1.4037407

[87] T. Lefebvre, F. De Belie, and G. Crevecoeur, "A trajectory-based sampling strategy for sequentially refined metamodel management of metamodel-based dynamic optimization in

mechatronics," *Optimal Control Applications and Methods*, vol. 39, no. 5, pp. 1786–1801, May 2018, doi: 10.1002/oca.2442

[88] D. Zalkind, N. J. Abbas, J. Jasa, A. Wright, and P. Fleming, "Floating wind turbine control optimization," *Journal of Physics: Conference Series*, vol. 2265, no. 4, p. 042021, May 2022, doi: 10.1088/1742-6596/2265/4/042021

[89] S. Azad and D. R. Herber, "Concurrent probabilistic control co-design and layout optimization of wave energy converter farms using surrogate modeling," in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, no. DETC2023-116896, Aug. 2023, doi: 10.1115/detc2023-116896

[90] A. K. Sundarrajan and D. R. Herber, "Using high-fidelity time-domain simulation data to construct multi-fidelity state derivative function surrogate models for use in control and optimization," in *ASME International Mechanical Engineering Congress & Exposition*, no. IMECE2023-112316, Nov. 2023, doi: 10.1115/IMECE2023-112316

[91] Q. Zhang, Y. Wu, and L. Lu, "A novel surrogate model-based solving framework for the black-box dynamic co-design and optimization problem in the dynamic system," *Mathematics*, vol. 10, no. 18, p. 3239, Sep. 2022, doi: 10.3390/math10183239

[92] P. Qiao, Y. Wu, J. Ding, and Q. Zhang, "A new sequential sampling method of surrogate models for design and optimization of dynamic systems," *Mechanism and Machine Theory*, vol. 158, p. 104248, Apr. 2021, doi: 10.1016/j.mechmachtheory.2021.104248

[93] S. Azad, D. R. Herber, S. Khanal, and G. Jia, "Concurrent geometry, control, and layout optimization of wave energy converter farms in probabilistic irregular waves using surrogate modeling," *Ocean Engineering*, vol. 320, p. 120183, Mar. 2025, doi: 10.1016/j.oceaneng.2024.120183

[94] R. Reeves and D. T. Griffith, "Intracycle RPM control for vertical axis wind turbines: Impacts on system design and costs," in *AIAA SCITECH 2025 Forum*, no. AIAA 2025-0648, Jan. 2025, doi: 10.2514/6.2025-0648

[95] M. Sadman Sakib, D. Todd Griffith, S. Hossain, S. Bayat, and J. T. Allison, "Intracycle RPM control for vertical axis wind turbines," *Wind Energy*, vol. 27, no. 3, pp. 202–224, Dec. 2023, doi: 10.1002/we.2885

[96] P. Klanatsky, F. Veynandt, and C. Heschl, "Grey-box model for model predictive control of buildings," *Energy and Buildings*, vol. 300, p. 113624, Dec. 2023, doi: 10.1016/j.enbuild.2023.113624

[97] S. S. Tohidi, D. Cali, M. Tamm, J. Ortiz, J. Salom, and H. Madsen, "From white-box to grey-box modelling of the heat dynamics of buildings," *E3S Web of Conferences*, vol. 362, p. 12002, 2022, doi: 10.1051/e3sconf/202236212002

[98] A. K. Sundarrajan, T. T. Tran, W. Wiley, H. Ross, D. Zalkind, and D. R. Herber, "Development of an optimal variable-pitch controller for floating axial-flow marine hydrokinetic turbines," in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, no. DETC2024-141271, Aug. 2024, doi: 10.1115/detc2024-141271

[99] A. K. Sundarrajan and D. R. Herber, "Comparison of data-driven modeling approaches for control optimization of floating offshore wind turbines," in *ASME International Design Engineering Technical Conferences*, no. DETC2025-167963, Aug. 2025, doi: 10.48550/arXiv.2505.14515

[100] J. T. Allison, T. Guo, and Z. Han, "Co-design of an active suspension using simultaneous dynamic optimization," *Journal of Mechanical Design*, vol. 136, no. 8, Jun. 2014, doi: 10.1115/1.4027335

[101] A. K. Sundarrajan and D. R. Herber, "Towards a fair comparison between the nested and simultaneous control co-design methods using an active suspension case study," in *American Control Conference*, May 2021, doi: 10.23919/ACC50511.2021.9482687

[102] D. R. Herber, "Dynamic system design optimization of wave energy converters utilizing direct transcription," M.S. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, May 2014, url: http://hdl.handle.net/2142/49463.

[103] L. T. Biegler, "An overview of simultaneous strategies for dynamic optimization," *Chemical Engineering and Processing: Process Intensification*, vol. 46, no. 11, pp. 1043–1053, Nov. 2007, doi: 10.1016/j.cep.2006.06.021

[104] D. R. Herber, "Advances in combined architecture, plant, and control design," Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Dec. 2017.

[105] J. T. Betts, *Practical Methods for Optimal Control and Estimation using Nonlinear Programming*. SIAM, Jan. 2010, doi: 10.1137/1.9780898718577

[106] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009, url: https://www.researchgate.net/profile/Anil-Rao-9/publication/268042868_A_Survey_of_Numerical_Methods_for_Optimal_Control/links/5557b5fc08ae6fd2d824f27d/A-Survey-of-Numerical-Methods-for-Optimal-Control.pdf.

[107] "WEIS," [Online, version gd3ea7969], url: https://github.com/WISDEM/WEIS/tree/develop

[108] "The DTQP project," [Online, version ad43dfb], url: https://github.com/danielrherber/dt-qp-project

[109] M. Hall, S. Housner, D. Zalkind, P. Bortolotti, D. Ogden, and G. Barter, "An open-source frequency-domain model for floating wind turbine design optimization," *Journal*

of Physics: Conference Series, vol. 2265, no. 4, p. 042020, May 2022, doi: 10.1088/1742-6596/2265/4/042020

[110] J. M. Jonkman and B. J. Jonkman, "FAST modularization framework for wind turbine simulation: full-system linearization," *Journal of Physics: Conference Series*, vol. 753, p. 082010, Sep. 2016, doi: 10.1088/1742-6596/753/8/082010

[111] J. M. Jonkman, E. S. P. Branlard, and J. P. Jasa, "Influence of wind turbine design parameters on linearized physics-based models in OpenFAST," *Wind Energy Science*, vol. 7, no. 2, pp. 559–571, Mar. 2022, doi: 10.5194/wes-7-559-2022

[112] "ROSCO," [Online, 989ce5b], url: https://github.com/NREL/ROSCO/tree/develop

[113] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Pearson London, 2015, vol. 33.

[114] G. J. van der Veen, I. J. Couchman, and R. O. Bowyer, "Control of floating wind turbines," in *American Control Conference*, Jun. 2012, pp. 3148–3153, doi: 10.1109/acc.2012.6315120

[115] T. J. Larsen and T. D. Hanson, "A method to avoid negative damped low frequent tower vibrations for a floating, pitch controlled wind turbine," *Journal of Physics: Conference Series*, vol. 75, p. 012073, Jul. 2007, doi: 10.1088/1742-6596/75/1/012073

[116] Y. Collette and P. Siarry, *Multiobjective Optimization*. Springer Berlin Heidelberg, 2004, doi: 10.1007/978-3-662-08883-8

[117] R. Haghi and C. Crawford, "Data-driven surrogate model for wind turbine damage equivalent load," *Wind Energy Science*, vol. 9, no. 11, pp. 2039–2062, Nov. 2024, doi: 10.5194/wes-9-2039-2024

[118] B. Peherstorfer, K. Willcox, and M. Gunzburger, "Survey of multifidelity methods in uncertainty propagation, inference, and optimization," *SIAM Review*, vol. 60, no. 3, pp. 550–591, Jan. 2018, doi: 10.1137/16m1082469

[119] A. March and K. Willcox, "Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives," *AIAA Journal*, vol. 50, no. 5, pp. 1079–1089, May 2012, doi: 10.2514/1.j051125

[120] Q. Wang, M. A. Sprague, J. Jonkman, N. Johnson, and B. Jonkman, "BeamDyn: a high-fidelity wind turbine blade solver in the FAST modular framework," *Wind Energy*, vol. 20, no. 8, pp. 1439–1462, Mar. 2017, doi: 10.1002/we.2101

[121] R. Vercellino *et al.*, "Control co-design optimization of natural gas power plants with carbon capture and thermal storage," in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, no. DETC2022-90021, Aug. 2022, doi: 10.1115/detc2022-90021

[122] S. Azad, Z. Gulumjanli, and D. R. Herber, "A general framework for supporting economic feasibility of generator and storage energy systems through capacity and dispatch optimization," in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, no. DETC2024-142667, Aug. 2024, doi: 10.1115/detc2024-142667

[123] Y. H. Lee, S. Bayat, J. T. Allison, M. S. Hossain, and D. T. Griffith, "Multidisciplinary modeling and control co-design of a floating offshore vertical-axis wind turbine system," *Journal of Mechanical Design*, vol. 147, no. 6, Mar. 2025, doi: 10.1115/1.4068072

[124] Y. Zhao, C. Jiang, M. A. Vega, M. D. Todd, and Z. Hu, "Surrogate modeling of nonlinear dynamic systems: A comparative study," *Journal of Computing and Information Science in Engineering*, vol. 23, no. 1, May 2022, doi: 10.1115/1.4054039

[125] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *IEEE International Conference on Big Data*, Dec. 2019, pp. 3285–3292, doi: 10.1109/bigdata47090.2019.9005997

[126] I. Kollár, R. Pintelon, and J. Schoukens, "Frequency domain system identification toolbox for MATLAB: Improvements and new possibilities," *IFAC Proceedings Volumes*, vol. 30, no. 11, pp. 943–946, Jul. 1997, doi: 10.1016/s1474-6670(17)42968-5

[127] P. V. Overschee and B. D. Moor, *Subspace Identification for Linear Systems*. Springer, 1996, doi: 10.1007/978-1-4613-0465-4

[128] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, "Past, present and future of nonlinear system identification in structural dynamics," *Mechanical Systems and Signal Processing*, vol. 20, no. 3, pp. 505–592, Apr. 2006, doi: 10.1016/j.ymssp.2005.04.008

[129] K. Worden *et al.*, "On evolutionary system identification with applications to nonlinear benchmarks," *Mechanical Systems and Signal Processing*, vol. 112, pp. 194–232, Nov. 2018, doi: 10.1016/j.ymssp.2018.04.001

[130] L. Ljung, *System Identification*. Birkhäuser Boston, 1998, pp. 163–173, doi: 10.1007/978-1-4612-1768-8_11

[131] M. Jansson, "A new subspace identification method for open and closed loop data," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 500–505, 2005, doi: 10.3182/20050703-6-cz-1902.00084

[132] D. R. Herber and A. K. Sundarrajan, "On the uses of linear-quadratic methods in solving nonlinear dynamic optimization problems with direct transcription," in *ASME International Mechanical Engineering Congress and Exposition*, no. IMECE2020-23885, Nov. 2020, doi: 10.1115/IMECE2020-23885

[133] B. Whitby and C. E. Ugalde-Loo, "Performance of pitch and stall regulated tidal stream turbines," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 1, pp. 64–72, Jan. 2014, doi: 10.1109/tste.2013.2272653

[134] T. Gunnink, "Analysis and regulation of the coupled dynamics of a two- turbine floating tidal energy converter," M.S. Thesis, Delft University of Technology, Delft, NL, Aug. 2015, url: http://resolver.tudelft.nl/uuid:4670bb38-ffbb-44c1-86ac-0d6498d9433f

[135] D. S. Zalkind, E. Dall'Anese, and L. Y. Pao, "Automatic controller tuning using a zeroth-order optimization algorithm," *Wind Energy Science*, vol. 5, no. 4, pp. 1579–1600, Nov. 2020, doi: 10.5194/wes-5-1579-2020

[136] D. S. Zalkind *et al.*, "System-level design studies for large rotors," *Wind Energy Science*, vol. 4, no. 4, pp. 595–618, Nov. 2019, doi: 10.5194/wes-4-595-2019

[137] P. Qian, B. Feng, H. Liu, X. Tian, Y. Si, and D. Zhang, "Review on configuration and control methods of tidal current turbines," *Renewable and Sustainable Energy Reviews*, vol. 108, pp. 125–139, Jul. 2019, doi: 10.1016/j.rser.2019.03.051

[138] Y.-j. Gu, Y.-g. Lin, Q.-k. Xu, H.-w. Liu, and W. Li, "Blade-pitch system for tidal current turbines with reduced variation pitch control strategy based on tidal current velocity preview," *Renewable Energy*, vol. 115, pp. 149–158, Jan. 2018, doi: 10.1016/j.renene.2017.07.034

[139] K. Gracie-Orr, T. M. Nevalainen, C. M. Johnstone, R. E. Murray, D. A. Doman, and M. J. Pegg, "Development and initial application of a blade design methodology for overspeed power-regulated tidal turbines," *International Journal of Marine Energy*, vol. 15, pp. 140–155, Sep. 2016, doi: 10.1016/j.ijome.2016.04.006

[140] S. Djebarri, J.-F. Charpentier, F. Scuiller, and M. Benbouzid, "Influence of fixed-pitch tidal turbine hydrodynamic characteristic on the generator design," in *European Wave and Tidal Energy Conference*, Sep. 2015, url: https://api.semanticscholar.org/CorpusID:112395408

[141] M. Arnold, F. Biskup, and P. W. Cheng, "Load reduction potential of variable speed control approaches for fixed pitch tidal current turbines," *International Journal of Marine Energy*, vol. 15, pp. 175–190, Sep. 2016, doi: 10.1016/j.ijome.2016.04.012

[142] X. Wang, X. Song, W. Sun, C. Sun, and Z. Liu, "Surrogate based co-design for combined structure and control design problems," *IEEE Access*, vol. 8, pp. 184 851–184 865, 2020, doi: 10.1109/access.2020.3029390

[143] A. Natarajan, "Damage equivalent load synthesis and stochastic extrapolation for fatigue life validation," *Wind Energy Science*, vol. 7, no. 3, pp. 1171–1181, Jun. 2022, doi: 10.5194/wes-7-1171-2022

[144] C. Amzallag, J. Gerey, J. Robert, and J. Bahuaud, "Standardization of the rainflow counting method for fatigue analysis," *International Journal of Fatigue*, vol. 16, no. 4, pp. 287–293, Jun. 1994, doi: 10.1016/0142-1123(94)90343-3

[145] G. Freebury and W. Musial, "Determining equivalent damage loading for full-scale wind turbine blade fatigue tests," in *ASME Wind Energy Symposium*, Jan. 2000, doi: 10.2514/6.2000-50

[146] Y. Miao, M. N. Soltani, and A. Hajizadeh, "A machine learning method for modeling wind farm fatigue load," *Applied Sciences*, vol. 12, no. 15, p. 7392, Jul. 2022, doi: 10.3390/app12157392

[147] E. Hendricks, O. Jannerup, and P. H. Sørensen, "Linear control system design," in *Linear Systems Control*. Springer, pp. 193–291, doi: 10.1007/978-3-540-78486-9_4

[148] R. Luus, *Iterative Dynamic Programming*. Chapman and Hall/CRC, Sep. 2019, doi: 10.1201/9780429123641

[149] D. P. Martin, K. E. Johnson, D. S. Zalkind, and L. Y. Pao, "LPV-based torque control for an extreme-scale morphing wind turbine rotor," in *American Control Conference*, May 2017, doi: 10.23919/ACC.2017.7963146

[150] Y. Zhao, B. Huang, H. Su, and J. Chu, "Prediction error method for identification of LPV models," *Journal of Process Control*, vol. 22, no. 1, pp. 180–193, Jan. 2012, doi: 10.1016/j.jprocont.2011.09.004

[151] F. Casella and M. Lovera, "LPV/LFT modelling and identification: overview, synergies and a case study," in *International Conference on Computer-Aided Control Systems*, 2008, pp. 852–857, doi: 10.1109/cacsd.2008.4627358

[152] R. Tóth, *Modeling and Identification of Linear Parameter-Varying Systems*. Springer Berlin Heidelberg, 2010, doi: 10.1007/978-3-642-13812-6

[153] C. Allen *et al.*, "Definition of the UMaine VolturnUS-S reference platform developed for the IEA wind 15-megawatt offshore reference wind turbine," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5000-76773, Jul. 2020, doi: 10.2172/1660012

[154] "WEIS-CTOpt," [Online], url: https://github.com/NREL/WEIS-CTOpt

[155] P. Van den Hof, "Closed-loop issues in system identification," *Annual Reviews in Control*, vol. 22, pp. 173–186, Jan. 1998, doi: 10.1016/s1367-5788(98)00016-9

[156] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, url: https://www.tensorflow.org/

[157] A. Robertson *et al.*, "Definition of the semisubmersible floating system for phase II of OC4," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5000-60601, Sep. 2014, https://docs.nrel.gov/docs/fy14osti/60601.pdf.

[158] J. Jonkman and W. Musial, "Offshore code comparison collaboration (OC3) for IEA task 23 offshore wind technology and deployment," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5000-48191, Mar. 2010, https://docs.nrel.gov/docs/fy11osti/48191.pdf.

[159] R. Meklenborg *et al.*, "IEA wind task 42: Lifetime extension assessment," Danish Technical University, Tech. Rep., Dec. 2022, url: https://iea-wind.org/wp-content/uploads/2023/06/IEA-Task-42-Report-D56_DataDrivenLifePredictionAndComparison.pdf.

[160] K. Brown *et al.*, "One-to-one aeroservoelastic validation of operational loads and performance of a 2.8 MW wind turbine model in OpenFAST," *Wind Energy Science*, vol. 9, no. 8, pp. 1791–1810, Aug. 2024, doi: 10.5194/wes-9-1791-2024

[161] J. Jasa, P. Bortolotti, D. Zalkind, and G. Barter, "Effectively using multifidelity optimization for wind turbine design," *Wind Energy Science*, vol. 7, no. 3, pp. 991–1006, May 2022, doi: 10.5194/wes-7-991-2022

[162] T. B. Schön, A. Wills, and B. Ninness, "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, Jan. 2011, doi: 10.1016/j.automatica.2010.10.013

[163] S. Baisthakur and B. Fitzgerald, "Physics-informed neural network surrogate model for bypassing blade element momentum theory in wind turbine aerodynamic load estimation," *Renewable Energy*, vol. 224, p. 120122, Apr. 2024, doi: 10.1016/j.renene.2024.120122

[164] J. D. Osorio, M. D. Florio, R. Hovsapian, C. Chryssostomidis, and G. E. Karniadakis, "Physics-informed machine learning for solar-thermal power systems," *Energy Conversion and Management*, vol. 327, p. 119542, Mar. 2025, doi: 10.1016/j.enconman.2025.119542

[165] D. Gedon, N. Wahlström, T. B. Schön, and L. Ljung, "Deep state space models for nonlinear system identification," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 481–486, 2021, doi: 10.1016/j.ifacol.2021.08.406

[166] G. Pillonetto, A. Aravkin, D. Gedon, L. Ljung, A. H. Ribeiro, and T. B. Schön, "Deep networks for system identification: A survey," *Automatica*, vol. 171, p. 111907, Jan. 2025, doi: 10.1016/j.automatica.2024.111907

[167] M. Hauser, S. Gunn, S. Saab, and A. Ray, "State-space representations of deep neural networks," *Neural Computation*, vol. 31, no. 3, pp. 538–554, Mar. 2019, doi: 10.1162/neco_a_01165

[168] R. Anantharaman *et al.*, "Accelerating simulation of stiff nonlinear systems using continuous-time echo state networks," arXiv, Oct. 2020, doi: 10.48550/arXiv.2010.04004

[169] C. Roberts *et al.*, "Continuous-time echo state networks for predicting power system dynamics," *Electric Power Systems Research*, vol. 212, p. 108562, Nov. 2022, doi: 10.1016/j.epsr.2022.108562

[170] R. T. Q. Chen *et al.*, "Neural ordinary differential equations," arXiv, Jun. 2018, doi: 10.48550/arXiv.1806.07366

[171] S. He, Y. Shi, E. Jonsson, and J. R. Martins, "Eigenvalue problem derivatives computation for a complex matrix using the adjoint method," *Mechanical Systems and Signal Processing*, vol. 185, p. 109717, Feb. 2023, doi: 10.1016/j.ymssp.2022.109717

[172] E. Branlard, J. Jonkman, C. Brown, and J. Zhang, "A digital twin solution for floating offshore wind turbines validated using a full-scale prototype," *Wind Energy Science*, vol. 9, no. 1, pp. 1–24, Jan. 2024, doi: 10.5194/wes-9-1-2024

[173] K. S. Narendra and A. M. Annaswamy, "Persistent excitation in dynamical systems," in *American Control Conference*, Jul. 1984, pp. 336–338, doi: 10.23919/acc.1984.4788399

[174] S. Bayat and J. T. Allison, "Impact of control strategies on the control co-design of spar floating offshore wind turbines," *Ocean Engineering*, vol. 336, p. 121763, Sep. 2025, doi: 10.1016/j.oceaneng.2025.121763

[175] A. Henry, M. Pusch, and L. Pao, "Investigation of $\mathcal{H}_\infty$-tuned individual pitch control for wind turbines," *Wind Energy*, vol. 27, no. 10, pp. 1074–1094, Aug. 2024, doi: 10.1002/we.2945

[176] Y. Li and Z. Wu, "Stabilization of floating offshore wind turbines by artificial muscle based active mooring line force control," in *American Control Conference (ACC)*, Jul. 2016, pp. 2277–2282, doi: 10.1109/acc.2016.7525257

[177] D. Stockhouse, M. Pusch, R. Damiani, S. Sirnivas, and L. Pao, "Robust multi-loop control of a floating wind turbine," *Wind Energy*, vol. 27, no. 11, pp. 1205–1228, Sep. 2023, doi: 10.1002/we.2864

[178] M. Phadnis and L. Pao, "Lyapunov-based framework for platform motion control of floating offshore wind turbines," *Energies*, vol. 18, no. 15, p. 3969, Jul. 2025, doi: 10.3390/en18153969

[179] Z. Liu, H. Huang, X. Xu, M. Xiong, and Q. Li, "An efficient global optimization algorithm combining revised expectation improvement criteria and Kriging," *Engineering Optimization*, vol. 56, no. 4, pp. 608–624, Feb. 2023, doi: 10.1080/0305215x.2023.2170367

[180] Y. Diouane, V. Picheny, R. L. Riche, and A. S. Di Perrotolo, "TREGO: a trust-region framework for efficient global optimization," arXiv, Jan. 2021, doi: 10.48550/arXiv.2101.06808

[181] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec. 1998, doi: 10.1023/a:1008306431147

Appendix A

Additional Validation Results

This appendix contains additional time series results for the five different wind/marine turbine systems discussed in Sec. 5.1.

1. Additional results for the IEA-15 MW turbine:
    - Time series validation results for a load case in the below-rated region are shown in Fig. A.1.
    - Time series validation results for a load case in the transition regions are shown in Fig. A.2.
2. Additional results for the IEA-22 MW turbine:
    - Time series validation results for a load case in the below-rated region are shown in Fig. A.3.
    - Time series validation results for a load case in the transition region are shown in Fig. A.4.
3. Additional results for the NREL 5 MW turbine with a semisubmersible platform:
    - Time series validation results for a load case in the below-rated region are shown in Fig. A.5.
    - Time series validation results for a load case in the transition region are shown in Fig. A.6.
4. Additional results for the NREL 5 MW turbine with a spar platform:
    - Time series validation results for a load case in the below-rated region are shown in Fig. A.7.
    - Time series validation results for a load case in the transition region are shown in Fig. A.8.
5. Additional results for the RM1-quad marine turbine:

- Time series validation results for a load case in the below-rated region are shown in Fig. A.9.
- Time series validation results for a load case in the transition region are shown in Fig. A.10.

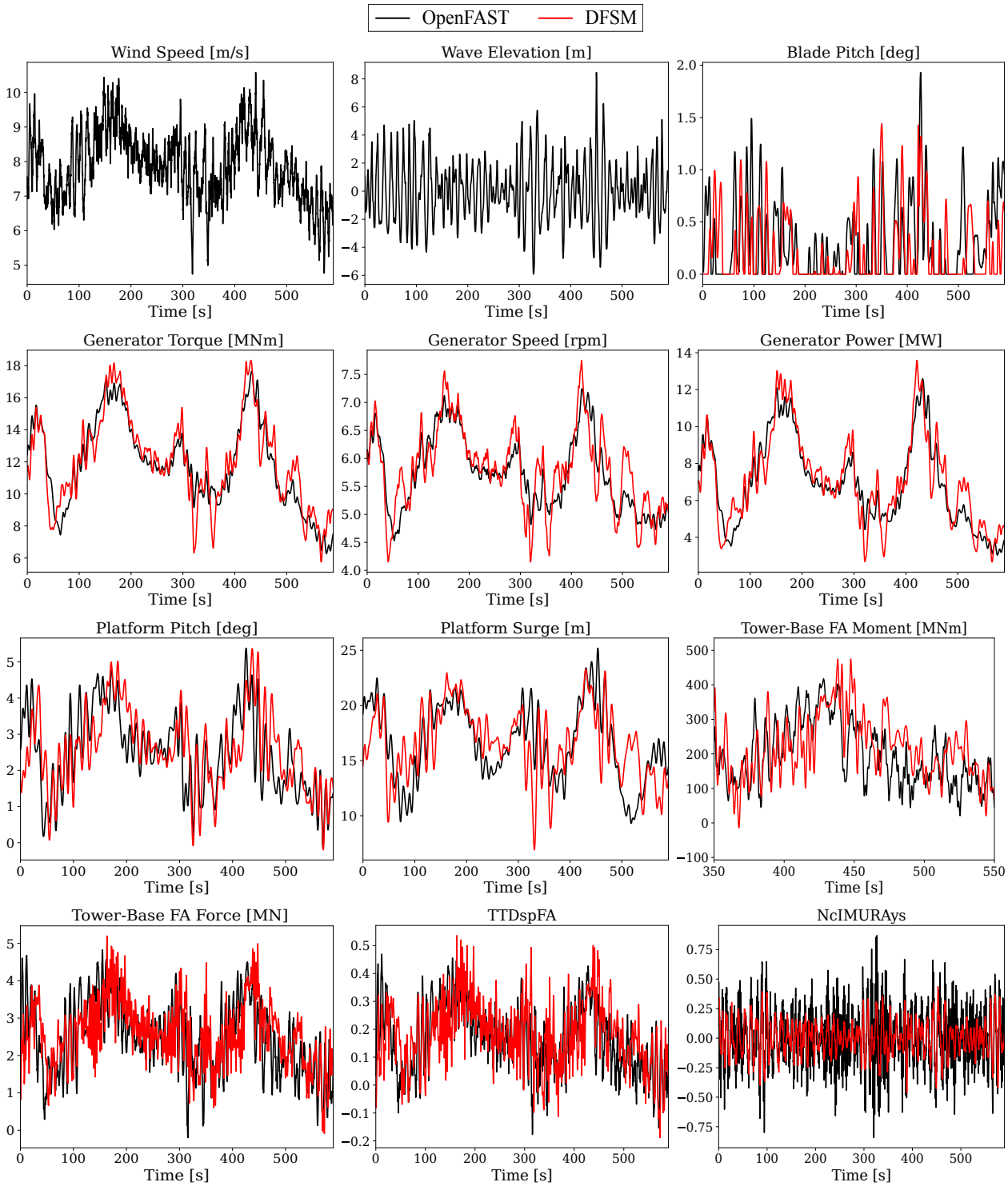


Figure A.1: Time series validation results for the IEA-15 MW turbine for a load case in the below-rated region.

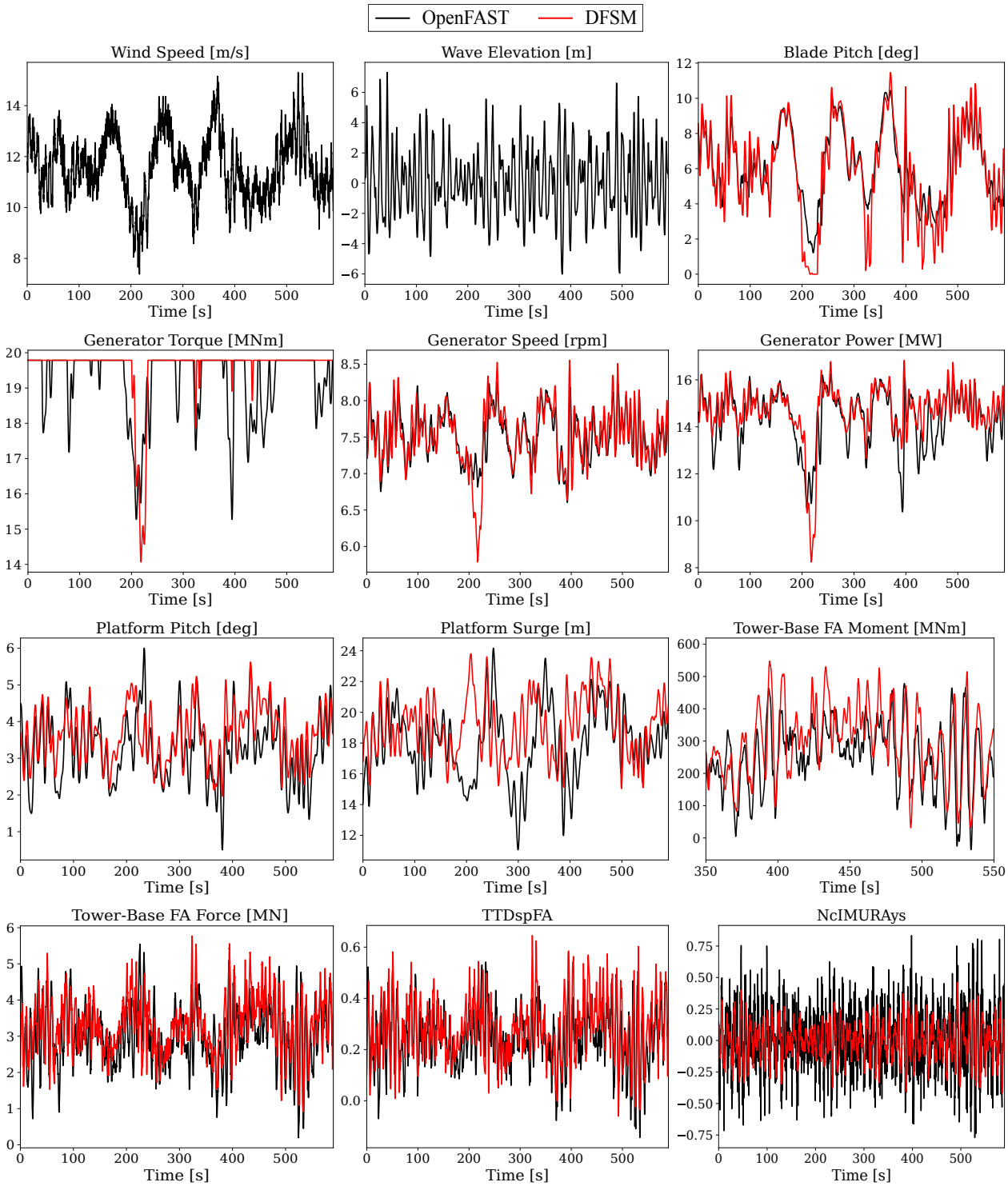


Figure A.2: Time series validation results for the IEA-15 MW turbine for a load case in the transition region.

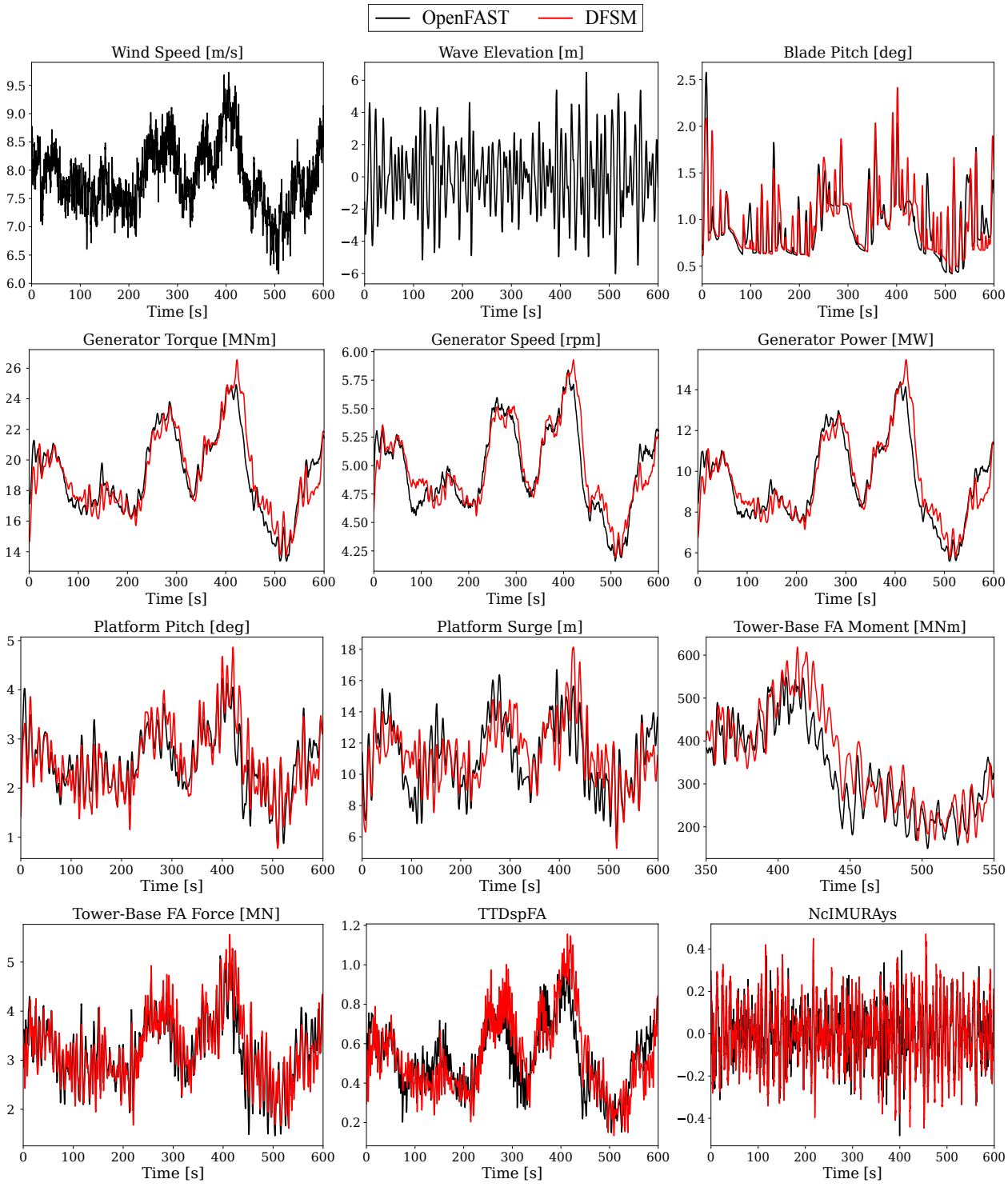


Figure A.3: Time series validation results for the IEA-22 MW turbine for a load case in the below-rated region.

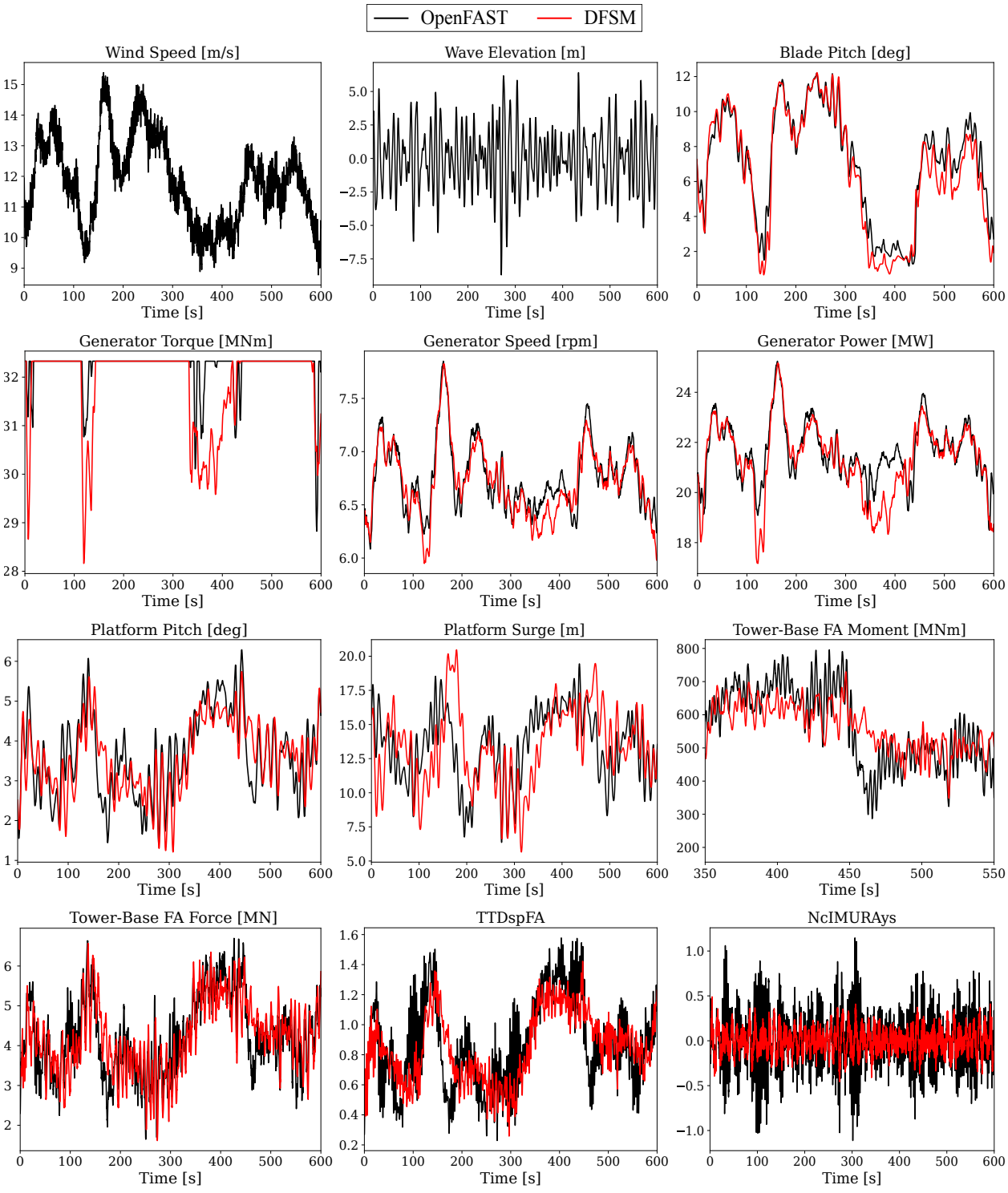


Figure A.4: Time series validation results for the IEA-22 MW turbine for a load case in the transition region.

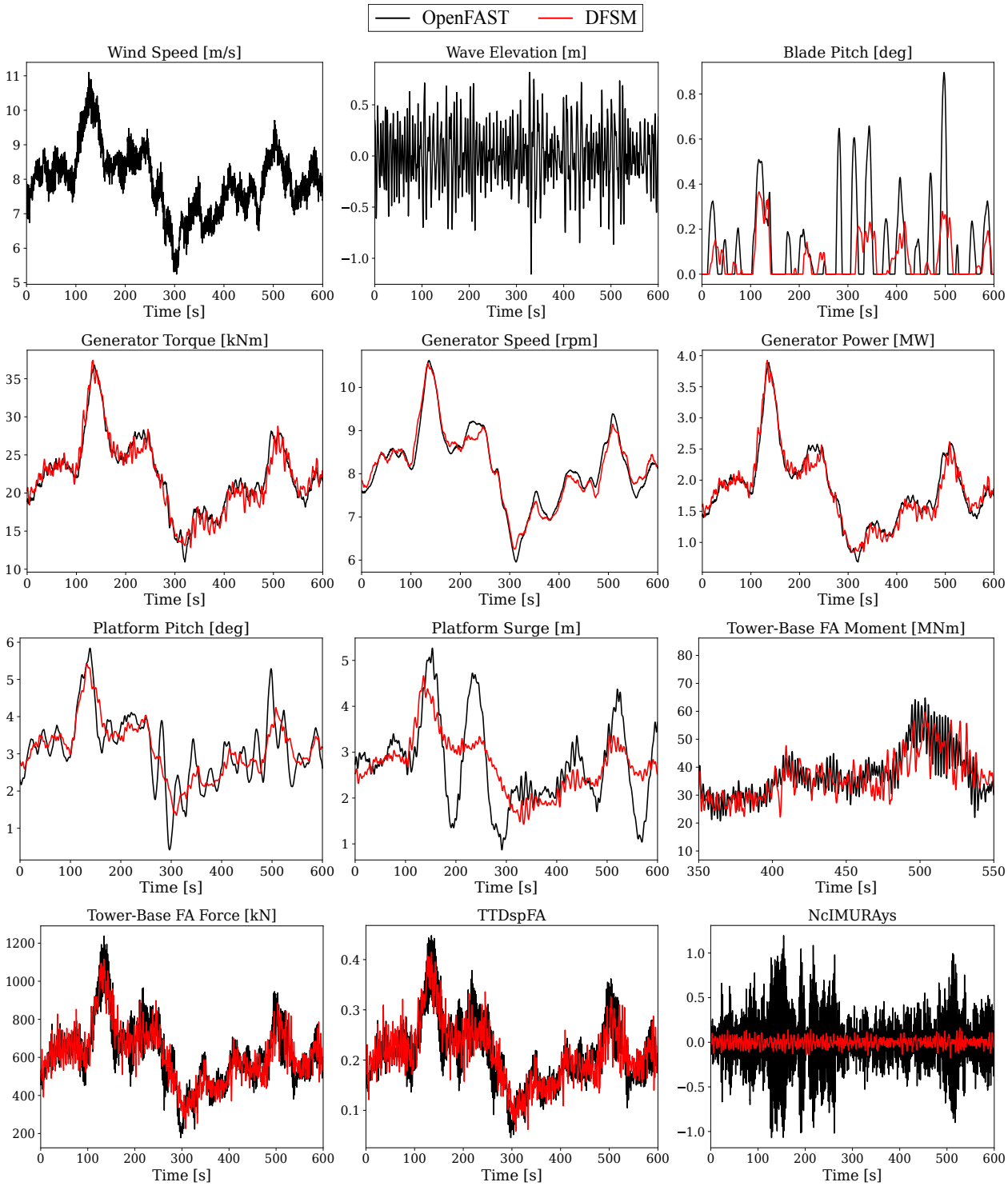


Figure A.5: Time series validation results for the NREL 5 MW turbine with a semisubmersible for a load case in the below-rated region.

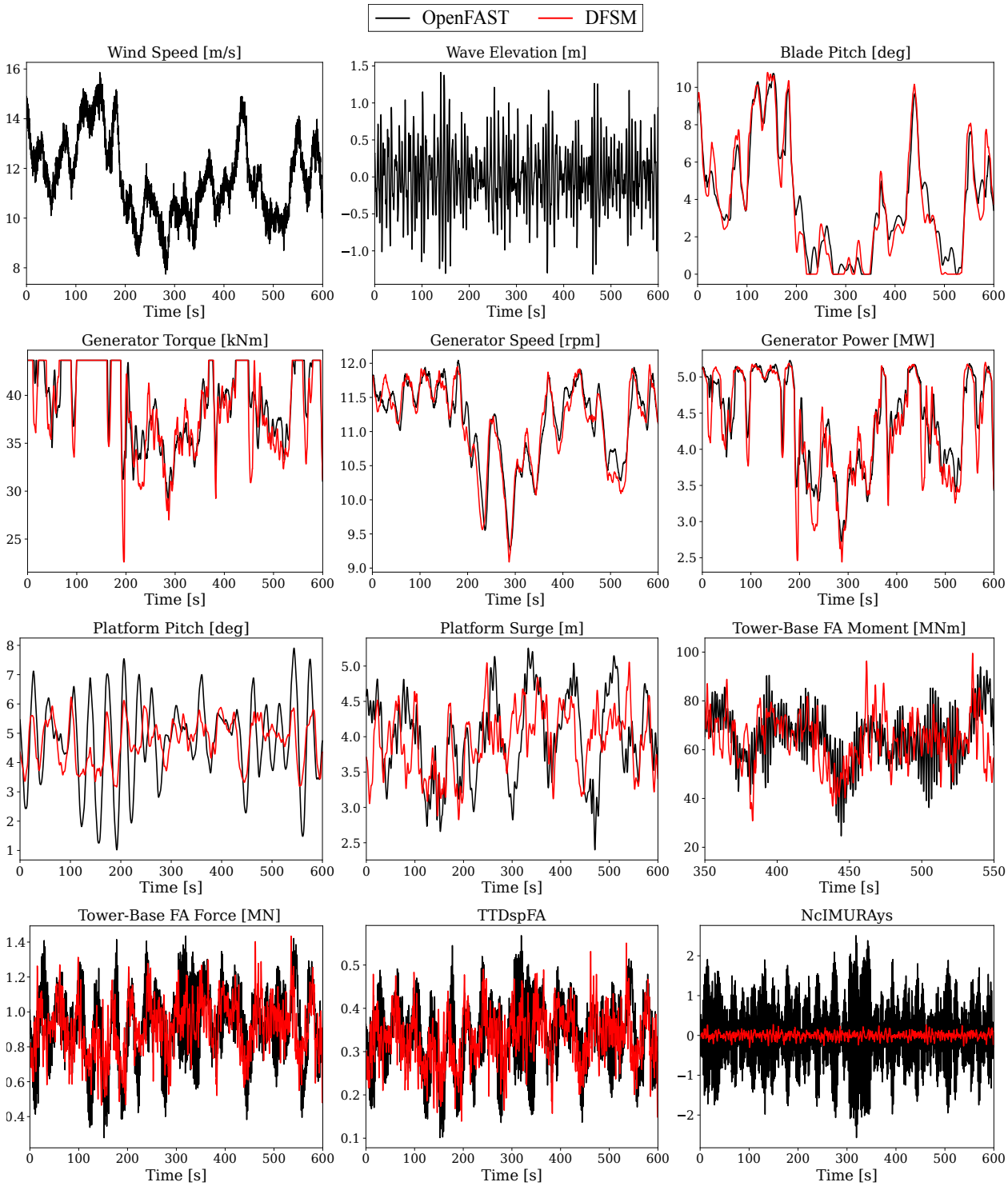


Figure A.6: Time series validation results for the NREL 5 MW turbine with a semisubmersible for a load case in the transition region.

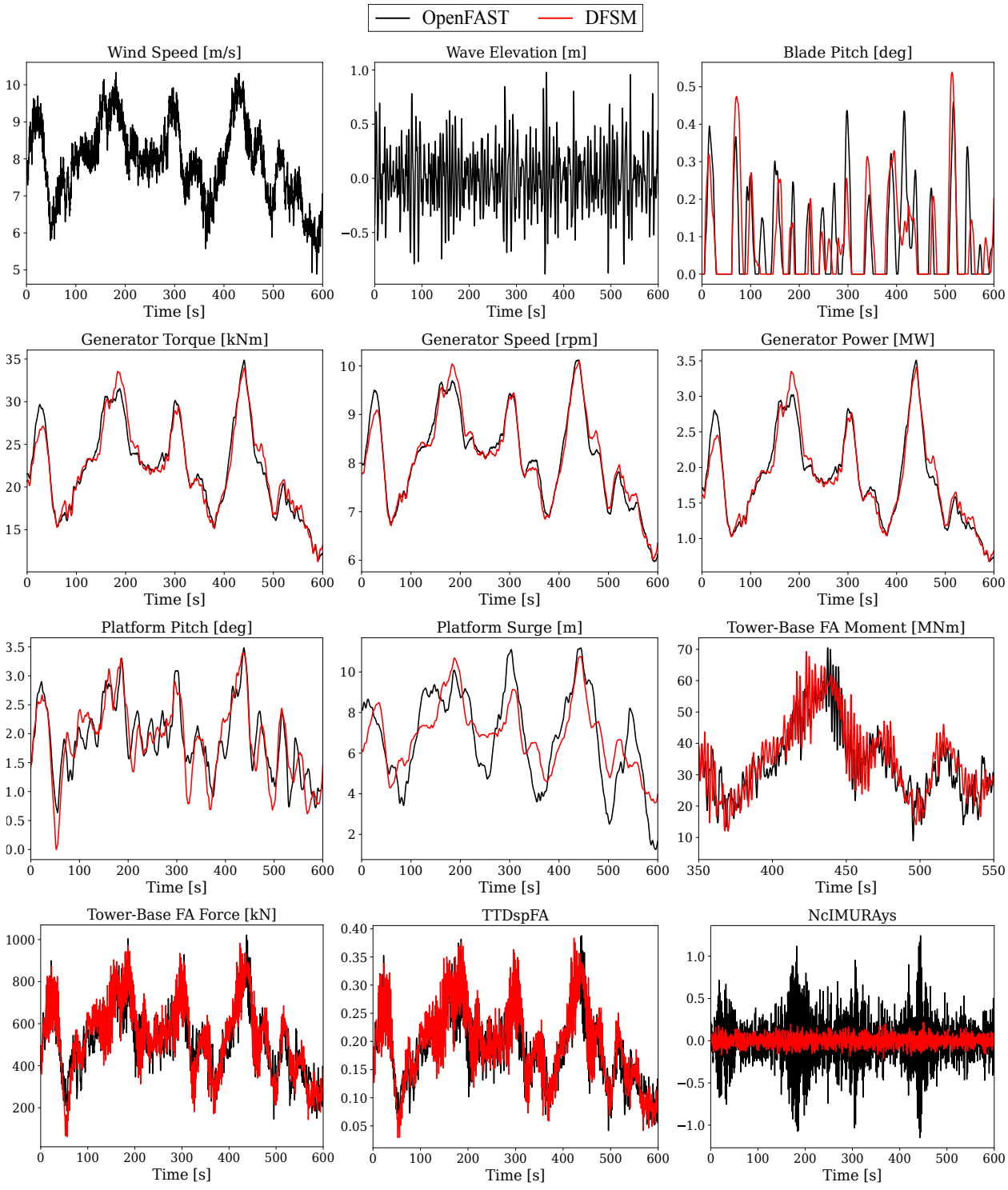


Figure A.7: Time series validation results for the NREL 5 MW turbine with a spar for a load case in the below-rated region.

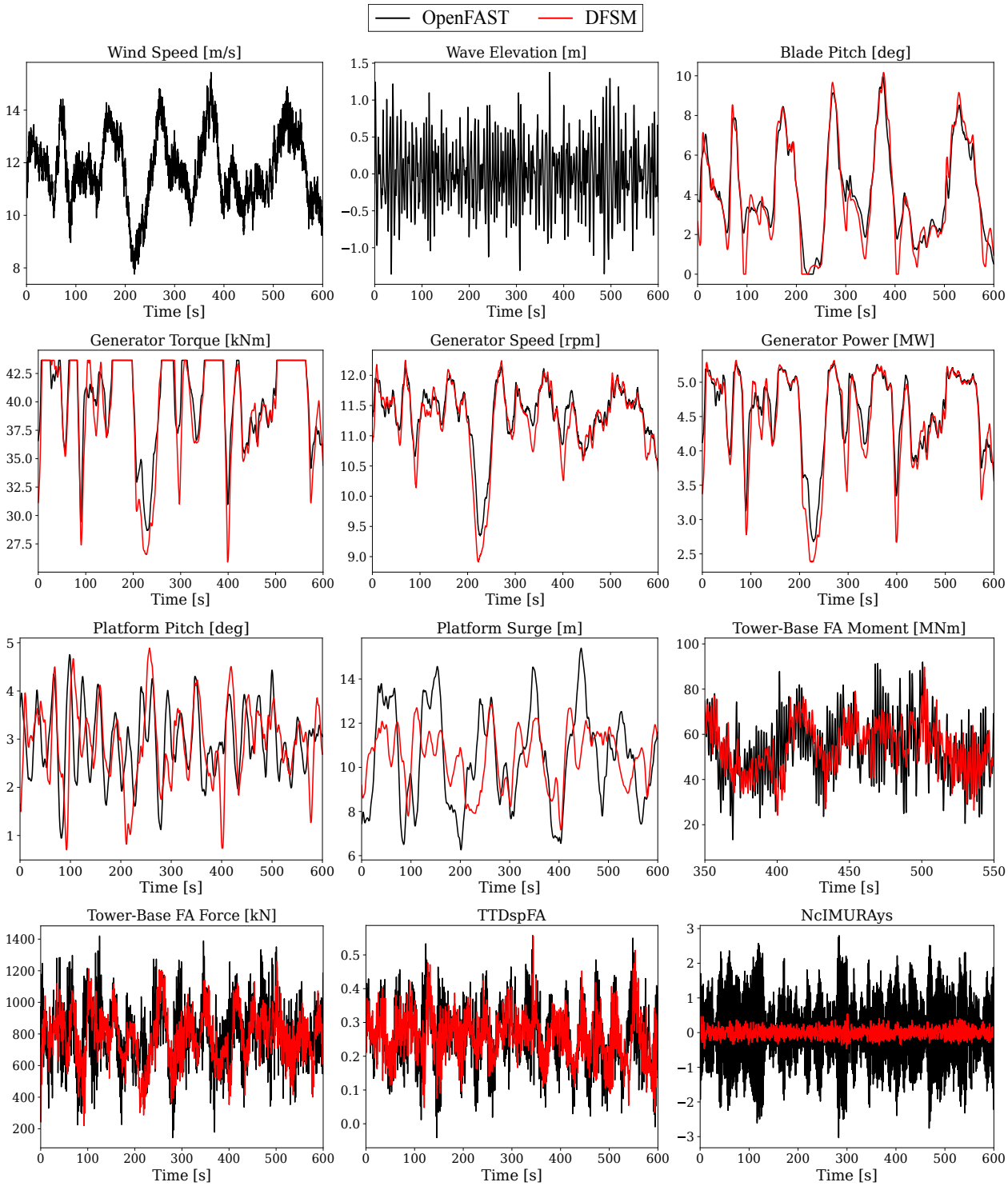


Figure A.8: Time series validation results for the NREL 5 MW turbine with a spar for a load case in the transition region.

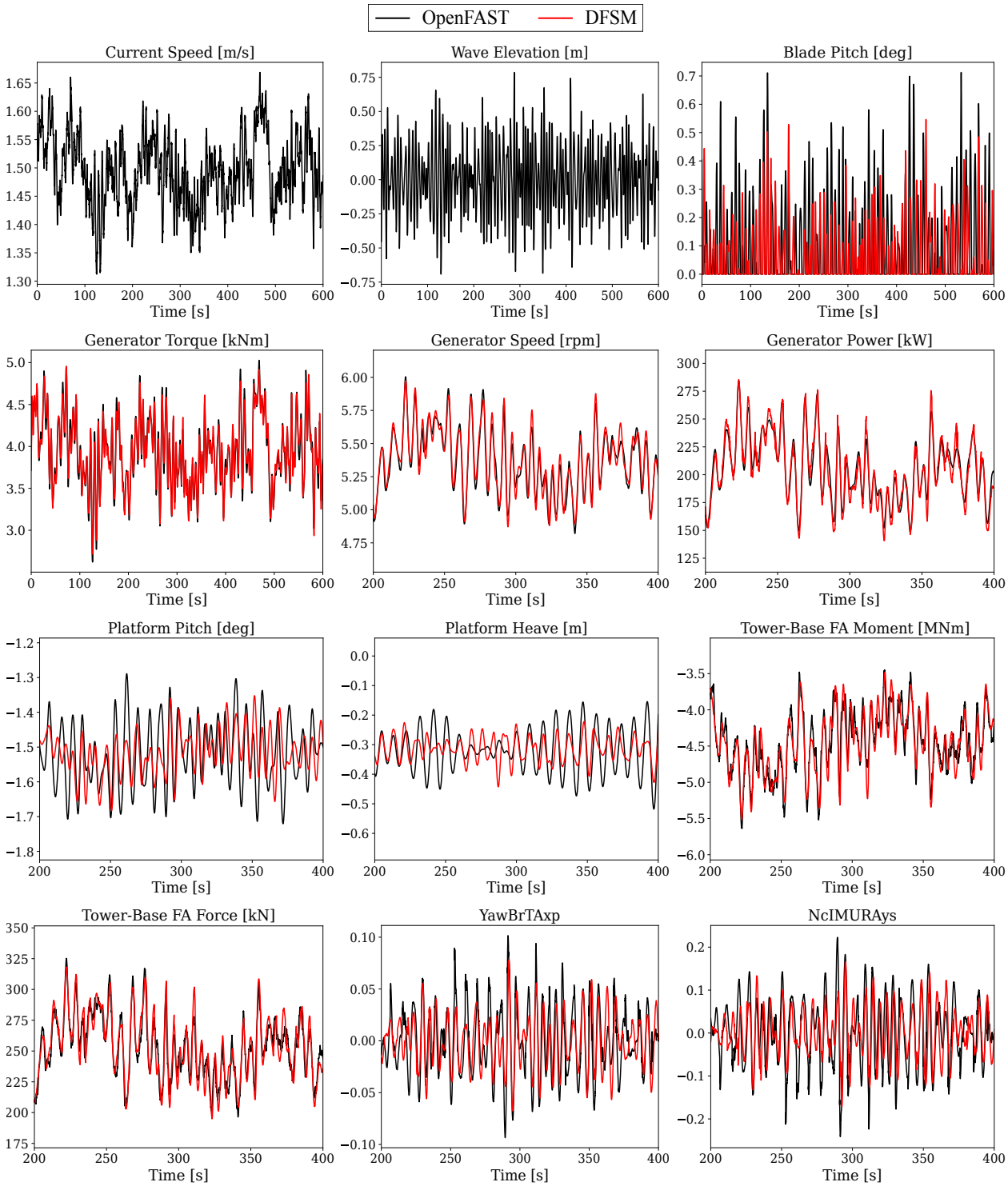


Figure A.9: Time series validation results for the RM1-quad marine turbine for a load case in the below-rated region.

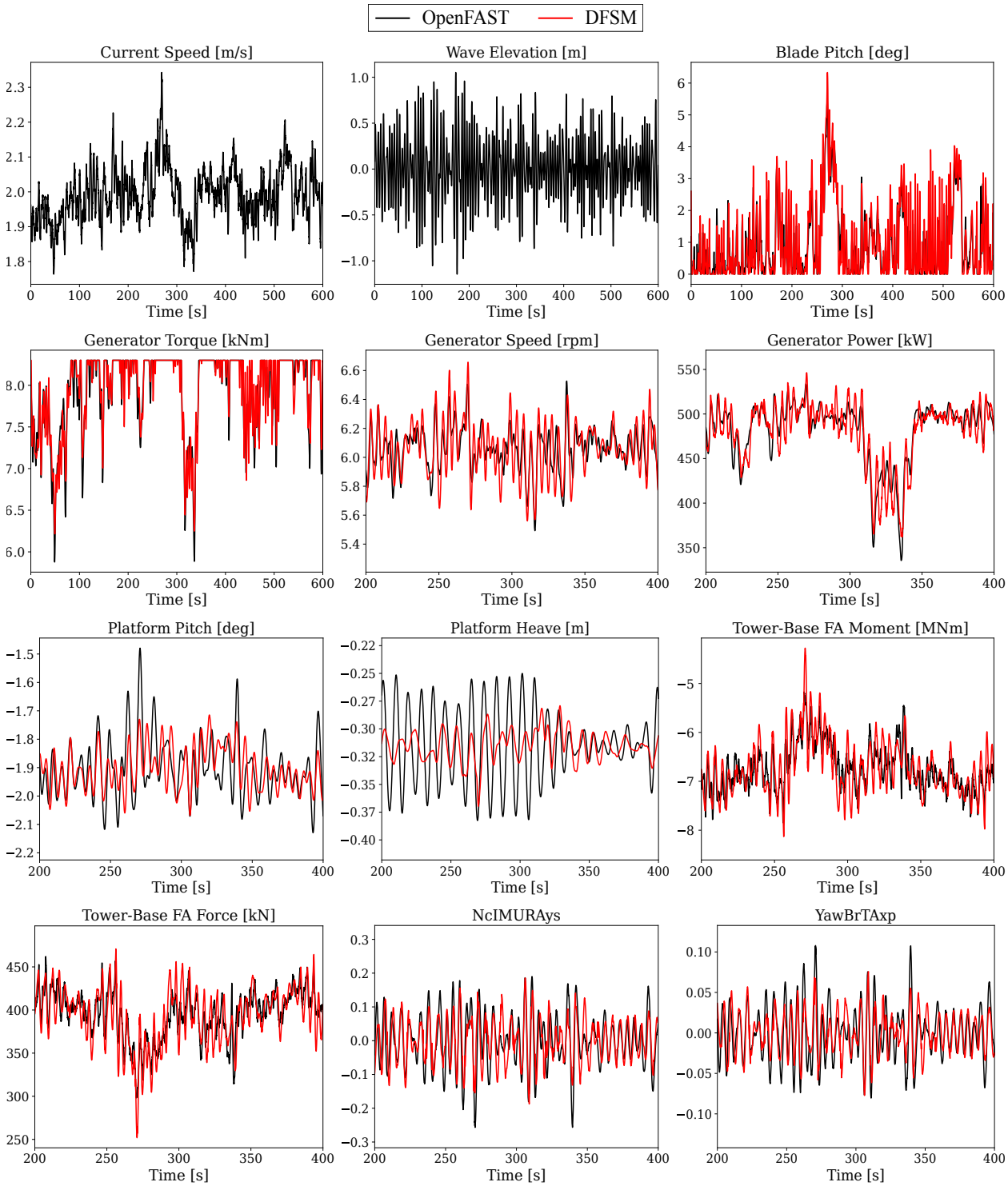


Figure A.10: Time series validation results for the RM1-quad marine turbine for a load case in the transition region.