Graphical Abstract

Neural Network-based Super-Twisting Control for Floating Wind Turbines: Design and Real-Time Validation

Flavie Didier, Hedi Basbas, Djelalli Larioumlil, Salah Laghrouche, Daniel Depernet

Highlights

Neural Network-based Super-Twisting Control for Floating Wind Turbines: Design and Real-Time Validation

Flavie Didier, Hedi Basbas, Djelalli Larioumlil, Salah Laghrouche, Daniel Depernet

- Research highlight 1
- Research highlight 2

Neural Network-based Super-Twisting Control for Floating Wind Turbines: Design and Real-Time Validation

Flavie Didier[a,*], Hedi Basbas[a], Djelalli Larioumlil[a], Salah Laghrouche[a], Daniel Depernet[a]

[a]*Energy Department, Université Marie et Louis Pasteur, UTBM, CNRS, FEMTO-ST Institute (UMR 6174), Belfort, 90010, France*

Abstract

This paper presents a novel model-based control strategy for collective blade pitch regulation in Region III of a 5 MW semi-submersible FOWT. The proposed approach combines a Radial Basis Function Neural Network (RBFNN) with a Super-Twisting Algorithm (STA) to enhance robustness against disturbances and improve dynamic performance. The control design is based on a reformulated nonlinear control-oriented model to support the synthesis of adaptive control laws. In addition to the control strategy, this work introduces a real-time Hardware-in-the-Loop (HIL) platform as a key contribution, specifically designed for the validation of FOWT control strategies. This platform provides a strictly synchronized execution framework for accurate real-time emulation and integrates OpenFAST for high-fidelity dynamic simulations. By enabling real-time validation, the HIL platform plays a crucial role in assessing control performance under realistic operating conditions. Experimental results confirm both the real-time feasibility of the proposed RBFNN-based STA controller and the effectiveness of the HIL platform in evaluating its performance, particularly in regulating generator speed and mitigating platform pitch motion under varying wind conditions.

Keywords: Floating offshore wind turbine, collective blade pitch control, super-twisting, neural networks, adaptive control, real-time validation

Nomenclature

Table 1: Comprehensive list of mathematical symbols and notations used throughout the manuscript.

Symbol	Description
$\mathcal{F}^0$	Inertial (global) coordinate system
$\mathcal{F}^b$	Body (tower base) coordinate system
$\mathcal{F}^n$	Nacelle coordinate system
$\mathcal{F}^s$	Shaft coordinate system
$\mathcal{F}^r$	Rotor coordinate system
$\mathcal{F}^g$	Generator coordinate system
$\mathbf{x}$	Full system state vector
$\mathbf{x}_g$	Translational displacement vector (m)
$\boldsymbol{\theta}$	Rotational angle vector (rad)
$\mathbf{u}$	Control input vector
θ_r	Rotor azimuth angle (rad)

Continued on next page

*Corresponding author.

Email address: flavie.didier@utbm.fr (Flavie Didier)

Table 1 – continued from previous page

Symbol	Description
θ_g	Generator azimuth angle (rad)
γ	Nacelle yaw angle (rad)
θ_{tilt}	Shaft tilt angle (rad)
ω_r	Rotor angular speed (rad/s)
ω_y	Platform pitch angular velocity (rad/s)
β	Collective blade pitch angle (rad)
T_g	Generator torque (Nm)
$\mathbf{R}(\boldsymbol{\theta})$	Rotation matrix from $\mathcal{F}^b$ to $\mathcal{F}^0$
$\mathbf{R}(\theta_{\text{tilt}})$	Rotation matrix from $\mathcal{F}^s$ to $\mathcal{F}^n$
f	Nonlinear dynamics function
$\mathbf{v}$	Wind disturbance vector (m/s²)
$\mathbf{w}$	Wave disturbance vector (m/s²)
m_g	Total mass of the FOWT (kg)
$\mathbf{m}_a$	Added hydrodynamic mass (kg)
F_A, F_B, F_C, F_D	Aerodynamic, buoyancy, catenary, and hydrodynamic forces (N)
T_A, T_B, T_C, T_D	Aerodynamic, buoyancy, catenary, and hydrodynamic torques (Nm)
P_A	Aerodynamic power (W)
$\mathbf{I}_g$	Inertia tensor in $\mathcal{F}^b$ (kg·m²)
J_l	Equivalent low-speed shaft inertia (kg·m²)
λ	Tip-speed ratio
R_r	Rotor radius (m)
ρ_a	Air density (kg/m³)
$\mathbf{r}_{gt}^b$	Vector from turbine CG to aerodynamic thrust center in $\mathcal{F}^b$ (m)
C_p, C_t	Power and thrust coefficients (–)
n_{ij}	Polynomial coefficients for approximating $C_t(\lambda, \beta)$
g_{ct}, f_{ct}	Polynomial functions used in the approximation of C_t
ΔC_t	Approximation error in the thrust coefficient model
o_{ij}	Polynomial coefficients for approximating $C_p(\lambda, \beta)$
g_{cp}, f_{cp}	Polynomial functions used in the approximation of C_p
ΔC_p	Approximation error in the power coefficient model
$\mathbf{v}_{\text{rel}}$	Relative wind velocity vector at the rotor center (m/s)
$\mathbf{v}_n$	Effective wind velocity normal to rotor (m/s)
η_v	Effective wind velocity factor, projected along rotor axis (m/s)
a_1, a_2, a_3	Geometric coupling coefficients from position vector $\mathbf{r}_{gt}^b$ and shaft tilt
D_g	Unmodeled torque disturbances and generator torque effects (Nm)
D_t	Total external disturbance torque (Nm)
D_θ	Aggregate disturbance in rotational dynamics (Nm)
D_{θ_y}	Pitch-axis component of the disturbance vector D_θ (Nm)
d_θ	Scaling factor in pitch rate dynamics coupling aerodynamic and inertial terms
b_1, b_2, b_3	Reciprocal inertia terms: $1/I_{xx}$, $1/I_{yy}$, $1/I_{zz}$
b_4, b_5, b_6	Inertial coupling coefficients based on $\mathbf{I}_g$ components
g_r, g_y	Nonlinear control gain functions for rotor and pitch dynamics
D_r, D_y	Lumped disturbance terms in respective dynamics
g_s	Composite control gain function in sliding variable dynamics
D_s	Lumped disturbance in sliding variable dynamics
ρ_{r1}, ρ_{r2}	Upper bounds on D_r and $\dot{D}_r$
ρ_{s1}, ρ_{s2}	Upper bounds on D_s and $\dot{D}_s$
P_g	Generator electrical power (W)

Continued on next page

Table 1 – continued from previous page

Symbol	Description
η_g	Gearbox ratio (–)
η_e	Generator efficiency (-)
e_r	Rotor speed tracking error (rad/s)
e_y	Platform pitch rate tracking error (rad/s)
s	Sliding variable (rad/s)
$\dot{s}$	Time derivative of the sliding variable (rad/s²)
k_y	Empirical gain for platform pitch rate compensation (s)
k_s	Composite tracking error gain, $k_s = k_y\omega_{rd}$ (rad)
ω_{rd}	Rated rotor speed (rad/s)
ω^*_{rd}	Modified reference rotor speed accounting for platform motion (rad/s)
T_{gd}	Rated generator torque (Nm)
β_{STA}	Super-Twisting control component of β
β_{ANN}	Neural network-based disturbance compensation component of β
β_ϵ	Estimation error compensation term in the control law
k_1, k_2	Gains of the Super-Twisting Algorithm (STA)
$\hat{D}_s$	Estimate of lumped disturbance D_s by the neural network
ε	Neural network estimation error
$\mathbf{W}$	Weight vector of the RBFNN output layer
$\mathbf{W}^*$	Optimal weight vector minimizing estimation error
$\Delta\mathbf{W}$	Weight estimation error: $\Delta\mathbf{W} = \mathbf{W} - \mathbf{W}^*$
ε^*	Minimal estimation error of the neural network
$\Delta\varepsilon$	Error between current and minimal estimation error
x_1, x_2	Inputs to the RBFNN: sliding variable and its derivative
c_{ji}	Center of the j^{th} radial basis function for input i
δ_{ji}	Width of the j^{th} radial basis function for input i
h_j	Output of the j^{th} neuron in the hidden layer
$\mathbf{h}$	Vector of outputs from hidden layer neurons
g_j	Gaussian basis function for neuron j
η_1, η_2	Learning rates for $\mathbf{W}$ and ε adaptation
ϕ	Auxiliary integral term in Lyapunov stability analysis
V	Lyapunov candidate function

Table 2: List of abbreviations used throughout the manuscript.

Abbreviation	Definition
AHOSMC	Adaptive High-Order Sliding Mode Control
ANN	Artificial Neural Network
BEMT	Blade Element Momentum Theory
COM	Control-Oriented Model
cRIO	compactRIO
CTE	Composite Tracking Error
DEL	Damage Equivalent Load
DoF	Degree of Freedom
DRL	Deep Reinforcement Learning
EC	Environmental Condition
EoM	Equation of Motion

Table 2 – continued from previous page

Abbreviation	Definition
FOWT	Floating Offshore Wind Turbine
GSPI	Gain-Scheduled Proportional-Integral
HIL	Hardware-in-the-Loop
HOSMC	High-Order Sliding Mode Control
MIL	Model-in-the-Loop
PHIL	Power-Hardware-in-the-Loop
RBFNN	Radial Basis Function Neural Network
RMSE	Root Mean Square Error
SMC	Sliding Mode Control
STA	Super-Twisting Algorithm
SWL	Still Water Level
UDP	User Datagram Protocol
VPPC	Variable Pitch Power Control

1. Introduction

Despite growing climatic concerns, the global demand for electrical energy continues to rise, with a projected increase of 3.3% in 2024 [1]. To meet this demand while limiting carbon emissions, renewable energy innovations such as Floating Offshore Wind Turbines (FOWTs) present a promising solution. Unlike bottom-fixed turbines, FOWTs rely on floating platforms anchored by mooring lines, enabling power generation in deep waters where winds are both stronger and more stable. These conditions provide multiple benefits for energy production: reduced wind obstacles simplify system control, lower mechanical strain, and improve turbine efficiency [2]. Additionally, higher wind intensities in deep waters allow greater energy extraction compared to shallow waters or onshore locations. By being situated far from coastlines, FOWTs also mitigate visual and noise concerns for coastal communities, making them a viable and sustainable alternative to traditional bottom-fixed wind turbines.

However, the floating nature of FOWTs introduces significant control challenges, notably six additional Degrees of Freedom (DoFs), making them more sensitive to wind, wave, and current disturbances. Although mooring systems provide some stabilization, these disturbances can still impact turbine stability, power generation, and operational costs. Effective control strategies are therefore essential to ensure optimal power capture and structural stability, particularly in Region III operation, where the turbine functions at its rated wind speed. In this region, the combination of strong winds and platform motion can lead to a phenomenon known as negative damping [3]. This occurs when the conventional pitch-to-feather control strategy, designed to reduce power output, unintentionally amplifies platform pitch motions. As a result, collective blade pitch control plays a crucial role in regulating generator speed and minimizing platform pitch oscillations. However, the system's highly nonlinear behavior, combined with unknown and varying disturbances, necessitates the development of advanced and robust control methodologies.

In the field of model-based approaches, the majority of studies rely on linear models for FOWTs [4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 10, 11, 17, 18, 19, 20, 21], which are derived from specific operating points based on wind conditions and rotor speed. Linear control strategies struggle when turbine dynamics shift from the modeled operating point, reducing control accuracy. As a result, controllers must be continuously adjusted for each new operating point, making real-time implementation impractical and labor-intensive.

In contrast, adaptive nonlinear control methods provide a more flexible alternative. They handle system nonlinearities effectively, improving regulation precision, robustness against disturbances, and adaptability to environmental variations. This property is particularly relevant for FOWTs operating in Region III, where wind conditions are highly unpredictable and subject to rapid variations due to external influences such as wind and waves. Specifically, the transition towards nonlinear algorithms is particularly compelling given the robust response capabilities of nonlinear control approaches such as Sliding Mode Control (SMC) to external disturbances, unmodeled dynamics, and modeling uncertainties [24]. When designed using a nonlinear model, SMC-based controllers ensure rigorous control law synthesis, system stability, and optimized performance. The development of robust controllers in this context

inherently requires a deep understanding of the system's coupled aero-hydro-servo-elastic behavior. In recent years, significant efforts have been made toward modeling these interactions comprehensively. For instance, Chen et al. [22] proposed a coupled aero-hydro-servo-elastic method to evaluate FOWT dynamics and control performance under various sea states. More recently, another study in [23] extended this approach to include wake interaction effects using a high-fidelity coupling strategy, emphasizing the growing demand for integrated modeling frameworks that capture the full system behavior.

While these methods provide accurate simulation platforms, they often rely on high-dimensional numerical models that are computationally intensive and not directly suitable for real-time control synthesis. Consequently, a shift towards comprehensive analytical and Control-Oriented Models (COM) for floating wind turbine systems is essential. However, existing COMs, such as those proposed by Betti [26], Lemmer [27], and Homer [28], are scarce in the literature and are not fully exploitable for nonlinear control synthesis. Recent studies [25, 30, 31, 32] have demonstrated the potential of adapting the nonlinear Betti COM for the Tension-Leg Platform (TLP)-based FOWT, ensuring the preservation of essential dynamics while enabling robust control design. In [77], the nonlinear Homer COM for the semi-submersible FOWT was adapted to developed a modified super-twisting sliding-mode algorithm, where functions within the rotor speed and platform pitch rate dynamics were approximated by locally valid constants around an operating point.

A key advantage of using a nonlinear analytical model is that it allows for the systematic synthesis of control laws with formal stability guarantees, unlike purely data-driven approaches. This structured framework provides a solid foundation for designing robust controllers that can adapt to dynamic environmental conditions. To overcome the limitations of existing approaches, this work leverages a nonlinear analytical COM to design an advanced control strategy that ensures both robustness and adaptability to varying operating conditions. Adapting a COM to a fully analytical form is a crucial step in advancing nonlinear control for FOWTs. Once such a model is available, robust and adaptive control laws can be synthesized to address the challenges of Region III. Notably, the Super-Twisting Algorithm (STA) approach, along with other High-Order SMC (HOSMC) methods and their adaptive versions, have demonstrated significant potential in previous studies [25, 30, 31, 32, 33, 34]. Adaptive SMC methods, either first-order or higher-order (AHOSMC) [31, 50, 51, 52, 53, 33, 34, 35, 36, 37], dynamically adjust the controller's parameters in response to parametric variations and disturbances, particularly in changing environmental conditions. However, their main drawback lies in convergence time, which can lead to under-performance and overshoots during rapid disturbances, especially when the control sampling period is relatively long. Additionally, although these methods are robust against uncertainties, they rely on sufficiently accurate dynamic models of the FOWT, which becomes problematic in extreme conditions where floating platform dynamics change abruptly.

To address these challenges, data-driven control strategies have recently emerged as a compelling alternative to traditional model-based approaches [38]. Methods such as fuzzy logic, Machine learning, and Deep Learning with Artificial Neural Networks (ANNs) bypass the need for explicit analytical models by learning the system's behavior directly from input-output data. This makes them well suited for handling the nonlinear and high-dimensional characteristics of FOWTs operating under variable wind and wave conditions. While their application in this field remains limited [39, 40], several recent works have demonstrated promising results in enhancing control robustness and performance. For instance, [41] proposed an ANN-based model predictive control framework for a 5 MW FOWT, where the ANN estimates the nonlinear steady-state behavior, while the dynamic part is identified using an AutoRegressive with eXogenous input (ARX) model. This allowed fast torque-pitch coordination without nonlinear optimization, improving response to frequency disturbances. Similarly, [42] introduced a gated recurrent neural network (GRNN) to predict FOWT dynamics for multi-objective pitch control, which reduced blade root loads and improved power tracking compared to GSPI strategies. In a different context, [44, 45] developed multi-layer perceptron-based surrogate models of hybrid FOWT-wave energy platforms and used fuzzy logic controllers to dampen platform oscillations, showing improved stability under coupled wind-wave conditions.

Beyond supervised learning, Deep Reinforcement Learning (DRL) offers a model-free control paradigm that is particularly suited to highly uncertain and nonlinear environments. DRL agents learn optimal control policies through direct interaction with the environment by maximizing cumulative rewards, thereby avoiding the need for explicit modeling. DRL methods such as Deep Deterministic Policy Gradient (DDPG) [46, 47, 48] and Incremental Dual Heuristic Programming (IDHP) [49] have been applied to FOWT control, demonstrating their capacity to handle multiple objectives and adapt to changing conditions. However, these approaches still face challenges, including training instability, sensitivity to hyperparameters, and computational demands—especially in real-time applications.

Despite these limitations, data-driven control represents a promising research direction for FOWTs, particularly when integrated with robust nonlinear techniques. In this context, the combination of SMC and HOSMC with ANN-based approximators forms a robust hybrid framework capable of managing modeling uncertainties and external disturbances [54, 55, 56, 57, 58, 76]. Unlike adaptive control techniques such AHOSMC, ANN-based controllers do not rely on pre-defined adaptation laws but continuously learn and approximate the system's unknown dynamics, enabling faster and more precise compensation of uncertainties. While AHOSMC dynamically tunes controller gains to mitigate disturbances, it often suffers from slow convergence and may struggle with abrupt variations in system dynamics. In contrast, an ANN can capture and compensate for complex, time-varying dynamics in real time, significantly reducing adaptation delays. The synergy between ANN and the STA further enhances performance: the ANN efficiently estimates and counteracts unmodeled dynamics and external disturbances, while the STA ensures robust tracking and mitigates chattering, which is particularly beneficial for FOWTs operating under highly uncertain and turbulent conditions. This hybrid approach leverages the ANN's ability to adapt to complex, nonlinear interactions, while preserving the stability guarantees of robust model-based control.

Validating control strategies for FOWTs is essential to ensure their reliability in real-world conditions. While full-scale offshore testing offers the most accurate validation, it is costly, time-consuming, and highly dependent on unpredictable environmental factors, making iterative controller development impractical. Scaled wind turbine tests in wave basins [78, 79] provide useful aero-hydro-servo-elastic insights but remain expensive and do not fully replicate full-scale operating conditions, particularly in terms of aerodynamic loads and control responses. To address these limitations, this work introduces a Hardware-in-the-Loop (HIL) platform as an intermediate validation step. The HIL framework enables real-time testing of control strategies in a controlled environment, allowing for rapid iteration and optimization before costly field trials [80]. By integrating the actual controller hardware with a high-fidelity FOWT simulation, HIL provides a cost-effective and flexible alternative for assessing computational performance, system delays, and robustness to uncertainties. This approach bridges the gap between numerical simulations and full-scale deployment, accelerating development while minimizing financial and logistical constraints.

This paper advances FOWT control strategies by proposing an ANN-based STA control law, built upon an adjusted nonlinear model. The proposed approach enables a systematic and constructive synthesis of the controller, ensuring formal stability guarantees. Specifically, the Radial Basis Function Neural Network (RBFNN) observer is integrated with the STA (RBFNN-based STA) to enhance robustness against disturbances and compensate for unmodeled dynamics. While numerous studies on FOWT control explore nonlinear and intelligent control strategies, most remain at the simulation level using software such as FAST, without experimental validation in real-time environments. To address this gap, this work implements and validates the RBFNN-based STA collective blade pitch controller on a HIL platform, ensuring real-time feasibility and performance assessment under conditions that better reflect real-world constraints. To the best of our knowledge, this is the first time an adaptive RBFNN-based STA control strategy for FOWTs has been validated in real-time using an HIL platform.

The main contributions of this article can be summarized as follows:

- Development and reformulation of a nonlinear analytical model for a 5 MW semi-submersible FOWT, ensuring a structured foundation for robust control synthesis.
- Design of an ANN-based STA control strategy built upon an adjusted nonlinear model, where a RBFNN observer estimates and compensates for unmodeled dynamics and external disturbances, while the STA ensures robust control performance across varying environmental conditions.
- Design, construction, and implementation of a real-time HIL platform, featuring a strictly synchronized execution framework to maintain real-time interaction between the control system and the FOWT emulator.
- Experimental validation of the proposed RBFNN-based STA controller through closed-loop HIL testing, demonstrating superior generator speed regulation and platform motion mitigation under real-time operating conditions, outperforming both the baseline GSPI blade pitch controller and the STA alone.

The remainder of this article is organized as follows: Section II introduces the Homer nonlinear COM, detailing its reformulation and validation for model-based control development in Region III. Section III defines the control objectives and presents the design of the RBFNN-based STA controller. Section IV describes the HIL platform,

followed by the implementation and validation of the RBFNN-based STA through comparative co-simulation tests. Finally, Section V outlines conclusions and future research directions.

2. FOWT Modeling

To develop an adaptive nonlinear model-based control law for FOWTs operating in Region III, a sufficiently accurate nonlinear model is essential to facilitate the synthesis of control laws. After introducing the considered Homer COM, this section details the simplifications made to the nonlinear dynamics of the COM to achieve a fully analytical model. This refined model serves as a foundation for accurately capturing the dynamic behavior of the FOWT system and implementing the RBFNN-based STA collective blade pitch controller.

2.1. Homer Control-Oriented Model

A detailed presentation of the Homer's 3-Dimensional (3-D) COM is given in [28]. The Homer COM is designed for a 5 MW FOWT based on a semi-submersible platform, whose specifications are given in [29]. The overall system, composed of the floating platform and the wind turbine, is regarded as a single rigid body, except for the drive shaft, which is treated as a double-mass system with torsional flexibility. The six right-hand 3-D coordinate systems used in the model include a global coordinate system ($\mathcal{F}^0$) and five body-fixed coordinate systems ($\mathcal{F}^b, \mathcal{F}^n, \mathcal{F}^s, \mathcal{F}^r, \mathcal{F}^g$). These are illustrated in Figure 1 and defined as follows:

- **Global Coordinate System** ($\mathcal{F}^0$): A static reference frame with its origin in the Still Water Level (SWL). The z-axis points vertically upward, while the x- and y-axes are horizontal, parallel to the SWL.
- **FOWT System Coordinate System** ($\mathcal{F}^b$): A dynamic frame attached to the FOWT's centroid. The z-axis extends vertically from the platform to the nacelle, parallel to the turbine tower. The x-axis aligns with the nominal wind direction, assuming no nacelle yaw movement.
- **Nacelle Coordinate System** ($\mathcal{F}^n$): Originates at the intersection of the yaw axis and rotor axis. Its z-axis aligns with $\mathcal{F}^b$, while the x- and y-axes rotate around the z-axis when yaw motion occurs.
- **Shaft Coordinate System** ($\mathcal{F}^s$): Shares its origin and y-axis with $\mathcal{F}^n$ but is tilted around the y-axis by the shaft tilt angle θ_{tilt}.
- **Rotor and Generator Coordinate Systems** ($\mathcal{F}^r$ **and** $\mathcal{F}^g$): These are dynamic frames attached along the shaft. Each rotates independently around their respective x-axes, both aligned with $\mathcal{F}^s$.

The states vector $\mathbf{x}$ incorporates sixteen variables, including the platform's six DoFs, the rotor and generator azimuth angles, and all corresponding velocities:

$$\mathbf{x} = \left[x_g, y_g, z_g, \theta_x, \theta_y, \theta_z, \theta_r, \theta_g, \dot{x}_g, \dot{y}_g, \dot{z}_g, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z, \omega_r, \omega_g\right]^\top. \tag{1}$$

Thus, the model defines three position states, $\mathbf{x}_g = \left[x_g, y_g, z_g\right]^\top$, as well as three orientation angles, $\boldsymbol{\theta} = \left[\theta_x, \theta_y, \theta_z\right]^\top$, representing the displacement vector and rotational position from the body frame $\mathcal{F}^b$ relative to the world frame $\mathcal{F}^0$. The orientation angles are part of a rotation matrix, $\mathbf{R}(\boldsymbol{\theta})$, approximated under the small angle assumption to simplify calculations and defined as:

$$\mathbf{R}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix}. \tag{2}$$

The possible control inputs within the COM are the blade pitch angle (β), the generator torque (T_g), and the nacelle's yaw angle (γ):

$$\mathbf{u} = \left[\beta, T_g, \gamma\right]^\top. \tag{3}$$

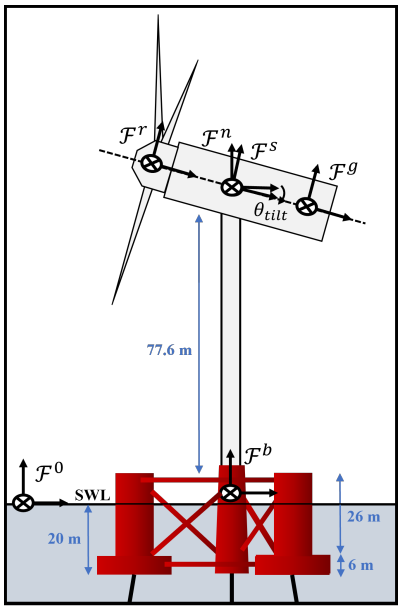

Figure 1: Schema of the semi-submersible FOWT with the reference frames considered for modeling [28].

The EoM for the Homer COM are articulated using the Newton-Euler formalism for translational and rotational dynamics. The nonlinear dynamics function f encapsulates the system's behavior as follows:

$$f(\mathbf{x},\mathbf{u},\mathbf{v},\mathbf{w}) = \begin{bmatrix} \dot{\mathbf{x}}_g \\ \dot{\boldsymbol{\theta}} \\ \omega_r \\ \omega_g \\ \left(m_g\mathbf{I}_3 + \mathbf{m}_a\mathbf{I}_3\right)^{-1} \sum_j F_j(\mathbf{x},\mathbf{u},\mathbf{v},\mathbf{w}) \\ \left(\mathbf{R}(\boldsymbol{\theta})\mathbf{I}_g^{-1}\mathbf{R}(\boldsymbol{\theta})^\top\right) \sum_j T_j(\mathbf{x},\mathbf{u},\mathbf{v},\mathbf{w}) \\ \frac{1}{J_r} \sum_{k_r} Q_{kr}(\mathbf{x},\mathbf{u},\mathbf{v},\mathbf{w}) \\ \frac{1}{J_g} \sum_{k_g} Q_{kg}(\mathbf{x},\mathbf{u},\mathbf{v},\mathbf{w}) \end{bmatrix}, \tag{4}$$

where wind and waves disturbances are represented by vectors $\mathbf{v}$ and $\mathbf{w}$, respectively. m_g is the total mass of the FOWTs, $\mathbf{I}_3$ is a 3×3 identity matrix, and $\mathbf{m}_a$ is the vector of hydrodynamic added mass. $\mathbf{I}_g = diag(I_{xx}, I_{yy}, I_{zz})$ is the inertia tensor matrix of the wind turbine in the body frame $\mathcal{F}^b$. J_r and J_g represent inertia about the rotor-side shaft and generator-side shaft, respectively, and Q_{kr} and Q_{kg} represent the k_r-th and k_g-th torque about each respective shaft. Summations of forces F_j and torques T_j include aerodynamic (F_A and T_A), buoyancy (F_B and T_B), catenary (F_C and T_C), and hydrodynamic (F_D and T_D) components.

Within the model, the aerodynamic force F_A is represented as a single thrust force that captures the combined effects of the wind interacting with the FOWT. This force is assumed to act parallel to the rotor's axis of rotation in frame $\mathcal{F}^s$, and is applied at the blades' center of thrust located at the hub:

$$F_A = \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda,\beta)\|\mathbf{v}_n\|^2, \tag{5}$$

where C_t is the thrust coefficient dependent on the tip-speed ratio (λ) and blade pitch angle (β), ρ_a denotes air density, R_r is the rotor radius, and $\mathbf{v}_n$ is the equivalent velocity vector normal to the face of the rotor blade.

The corresponding torque T_A generated by the aerodynamic force is then calculated as:

$$T_A = \mathbf{R}(\boldsymbol{\theta})\mathbf{r}_{gt}^b F_A + \mathbf{R}(\boldsymbol{\theta})\mathbf{R}(\theta_{tilt})\eta_g T_g \hat{\mathbf{e}}_1^s, \tag{6}$$

where $\mathbf{r}_{gt}^b = \left[(\mathbf{r}_{gt}^b)_x, (\mathbf{r}_{gt}^b)_y, (\mathbf{r}_{gt}^b)_z\right]$ is the position vector from the turbine center of gravity to the force application point in the body frame $\mathcal{F}^b$. η_g is the gearbox ratio, and $\hat{\mathbf{e}}_1^s$ is the first orthogonal unit vector of $\mathcal{F}^s$. $\mathbf{R}(\theta_{tilt})$ defines the orientation of $\mathcal{F}^s$ with respect to $\mathcal{F}^n$ expressed as:

$$\mathbf{R}(\theta_{tilt}) = \begin{bmatrix} \cos(\theta_{tilt}) & 0 & -\sin(\theta_{tilt}) \\ 0 & 1 & 0 \\ \sin(\theta_{tilt}) & 0 & \cos(\theta_{tilt}) \end{bmatrix}. \tag{7}$$

The aerodynamic power P_A is given by:

$$P_A = \frac{1}{2}\rho_a \pi R_r^2 C_p(\lambda, \beta)\|\mathbf{v}_n\|^3, \tag{8}$$

where C_p is the power coefficient.

Typically, the power (C_p) and thrust (C_t) coefficients, which are highly nonlinear functions of λ and β, are expressed using lookup tables generated offline by external software like AeroDyn [59].

2.2. Adaptation of the Homer COM

The formal structure of the presented Homer COM does not support the development of nonlinear control laws. Specifically, its aerodynamic module employs a simplified Blade Element Momentum Theory (BEMT), which uses lookup tables for calculating the thrust coefficient (C_t) and power coefficient (C_p) based on turbine performance data. These tables, generated by the blade pitch control, intertwine analytical equations with data from third-party software, complicating the direct application of nonlinear control laws that require fully analytical models.

To overcome these limitations, the Homer model must be reformulated into a fully analytical form, eliminating dependence on externally sourced tables and enabling the seamless integration of model-based nonlinear control strategies.

2.2.1. Simplified EoM

In this adaptation, the misalignment between wind direction and the x-axis of $\mathcal{F}^r$ is not considered. Consequently, the nacelle yaw motion is omitted. Furthermore, unlike the original two-mass flexible shaft model used by Homer, this study employs a one-mass rigid shaft model to describe the drivetrain dynamics:

$$\dot{\omega}_r = \frac{1}{J_l}\left(\frac{P_A}{\omega_r} - \eta_g T_g\right). \tag{9}$$

Here, $J_l = (J_r + n_g^2 J_g)$ denotes the equivalent inertia of the low-speed shaft, incorporating both the turbine (J_r) and generator (J_g) inertia.

Considering this simplified drivetrain model, the state selection for our study includes the platform's spatial and angular positions and their time derivatives, the rotor azimuth angle θ_r and the rotor speed ω_r:

$$\mathbf{x} = \left[x_g, y_g, z_g, \theta_x, \theta_y, \theta_z, \theta_r, \dot{x}_g, \dot{y}_g, \dot{z}_g, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z, \omega_r\right]^\top. \tag{10}$$

Given the adoption of a pitch-to-feather strategy for the development of a collective blade pitch controller operating in Region III, and given that the generator torque T_g is fixed at its rated value and nacelle yaw motion is excluded, the control input selected consists solely of the blade pitch angle β:

$$\mathbf{u} = \beta. \tag{11}$$

Consequently, the EoM for the semi-submersible FOWT can be rewritten to reflect these initial simplifications made to the model:

$$f(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{w}) = \begin{bmatrix} \dot{\mathbf{x}}_g \\ \dot{\boldsymbol{\theta}} \\ \omega_r \\ \left(m_g \mathbf{I}_3 + m_a \mathbf{I}_3\right)^{-1} \sum_j F_j(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{w}) \\ \left(\mathbf{R}(\boldsymbol{\theta})\, \mathbf{I}_g^{-1} \mathbf{R}(\boldsymbol{\theta})^\top\right) \sum_j T_j(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{w}) \\ \frac{1}{J_l}\left(\frac{P_A}{\omega_r} - \eta_g T_g\right) \end{bmatrix}, \tag{12}$$

2.2.2. Dynamics of the platform pitch rate ω_y

Within the framework of the Homer model, the thrust coefficient (C_t) of the wind turbine is approximated using a polynomial function (13). While this method introduces a relatively high fitting error, this deviation will be effectively managed by the neural network disturbance observer.

$$\begin{aligned} C_t(\lambda,\beta) &= g_{ct}\beta + f_{ct} \\ &= \left(n_{10} + n_{11}\lambda + n_{12}\lambda^2 + n_{13}\lambda^3 + n_{14}\lambda^4\right)\beta \\ &+ \left(n_{00} + n_{01}\lambda + n_{02}\lambda^2 + n_{03}\lambda^3 + n_{04}\lambda^4 + n_{05}\lambda^5\right) \\ &+ \Delta C_t, \end{aligned} \tag{13}$$

where ΔC_t represents the fitting error. Polynomial coefficients given in Table 3 are determined using MATLAB's curve fitting toolbox.

n_{ij}	**Value**	n_{ij}	**Value**	n_{ij}	**Value**
n_{00}	-0.03825	n_{01}	0.2559	n_{02}	-0.06493
n_{03}	0.02576	n_{04}	-0.003477	n_{05}	-0.0005361
n_{10}	-0.4266	n_{11}	0.3144	n_{12}	-0.01908
n_{13}	0.02513	n_{14}	-0.000702		

Table 3: Values of polynomial coefficients for C_t.

By projecting the relative wind velocity vector $\mathbf{v}_{\text{rel}}$ between the wind and the FOWT at the center of thrust along the x-axis of the coordinate system $\mathcal{F}^s$, $\mathbf{v}_n$ can be calculated as:

$$\mathbf{v}_n = \mathbf{R}(\boldsymbol{\theta})\mathbf{R}(\theta_{\text{tilt}})\text{diag}(1,0,0)\mathbf{R}^\top(\theta_{\text{tilt}})\mathbf{R}^\top(\boldsymbol{\theta})\mathbf{v}_{\text{rel}}, \tag{14}$$

with:

$$\mathbf{v}_{\text{rel}} = \mathbf{v} - \dot{x}_g - \dot{\mathbf{R}}(\boldsymbol{\theta})\mathbf{r}^b_{gt}, \tag{15}$$

where $\mathbf{v} = \begin{bmatrix} v_x, v_y, v_z \end{bmatrix}$ is the total wind velocity vector.

Based on (5), (14), and (15), the aerodynamic force F_A is then calculated as:

$$F_A = \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda,\beta)\mathbf{v}_n^2 \mathbf{R}(\boldsymbol{\theta})\mathbf{R}(\theta_{\text{tilt}}) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{16}$$

which based on (2) and (7), simplifies to:

$$F_A = \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda,\beta)\eta_v^2 \begin{bmatrix} \cos(\theta_{\text{tilt}}) + \theta_y \sin(\theta_{\text{tilt}}) \\ \theta_z \cos(\theta_{\text{tilt}}) - \theta_x \sin(\theta_{\text{tilt}}) \\ -\theta_y \cos(\theta_{\text{tilt}}) + \sin(\theta_{\text{tilt}}) \end{bmatrix}, \tag{17}$$

where η_v is a scalar function representing the effective velocity factor, dependent on the relative wind velocity and rotor orientation, such as:

$$\begin{aligned} \eta_v = \|\mathbf{v}_n\| &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{R}^\top(\theta_{\text{tilt}})\mathbf{R}^\top(\boldsymbol{\theta})v_{\text{rel}} \\ &= \left(\cos(\theta_{\text{tilt}}) + \theta_y \sin(\theta_{\text{tilt}})\right)\left(v_x - \dot{x}_g - (\mathbf{r}^b_{gt})_z\dot{\theta}_y + (\mathbf{r}^b_{gt})_y\dot{\theta}_z\right) \\ &+ \left(\theta_z \cos(\theta_{\text{tilt}}) - \theta_x \sin(\theta_{\text{tilt}})\right)\left(v_y - \dot{y}_g - (\mathbf{r}^b_{gt})_z\dot{\theta}_x - (\mathbf{r}^b_{gt})_x\dot{\theta}_z\right) \\ &+ \left(-\theta_y \cos(\theta_{\text{tilt}}) + \sin(\theta_{\text{tilt}})\right)\left(v_z - \dot{z}_g - (\mathbf{r}^b_{gt})_y\dot{\theta}_x + (\mathbf{r}^b_{gt})_x\dot{\theta}_y\right). \end{aligned} \tag{18}$$

Given the expression of the aerodynamic torque T_A in (6), and substituting (7) and (17) into (6), T_A can be rewritten as:

$$\begin{aligned} T_A = & \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda,\beta)\eta_v^2 \\ & \begin{bmatrix} a_1\theta_x^2 + a_3\theta_x\theta_y - a_2\theta_x\theta_z - a_2\theta_y - a_3\theta_z + a_1 \\ a_3\theta_y^2 + a_1\theta_x\theta_y - a_2\theta_y\theta_z + a_2\theta_x + a_1\theta_z + a_3 \\ -a_2\theta_z^2 + a_1\theta_x\theta_z + a_3\theta_y\theta_z + a_3\theta_x - a_1\theta_y - a_2 \end{bmatrix} \\ & + D_g. \end{aligned} \tag{19}$$

where D_g represents disturbances and generator torque influence not captured by the simplified model, and the coefficients are defined as follows:

$$a_1 = (\mathbf{r}_{gt}^b)_y \sin(\theta_{tilt}), \quad a_2 = (\mathbf{r}_{gt}^b)_y \cos(\theta_{tilt}),$$
$$a_3 = (\mathbf{r}_{gt}^b)_z \cos(\theta_{tilt}) - (\mathbf{r}_{gt}^b)_x \sin(\theta_{tilt}).$$

Given that the nacelle yaw motion is disregarded in this study, the y-component of the distance vector from the centroid of the FOWT to the center of thrust ($(\mathbf{r}_{gt}^b)_y$) is zero. Consequently, the coefficients a_1 and a_2, which depend on $(\mathbf{r}_{gt}^b)_y$, are also zero. This simplification allows the aerodynamic torque to be expressed in a more compact form:

$$T_A = \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda,\beta)\eta_v^2 a_3 \begin{bmatrix} \theta_x\theta_y - \theta_z \\ \theta_y^2 + 1 \\ \theta_y\theta_z + \theta_x \end{bmatrix} + D_g. \tag{20}$$

The Euler equations, which govern the rotational dynamics of the FOWT, can be expressed as follows:

$$\ddot{\boldsymbol{\theta}} = \mathbf{R}(\boldsymbol{\theta})\mathbf{I}_g^{-1}\mathbf{R}(\boldsymbol{\theta})^\top \left(T_A + T_B + T_C + T_D\right). \tag{21}$$

Expanding the matrix multiplication, the dynamics can be detailed as:

$$\begin{aligned} \ddot{\boldsymbol{\theta}} = & \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} 1 & \theta_z & -\theta_y \\ -\theta_z & 1 & \theta_x \\ \theta_y & -\theta_x & 1 \end{bmatrix} \\ & (T_A + T_B + T_C + T_D) \\ = & \begin{bmatrix} b_1 + b_3\theta_y^2 + b_2\theta_z^2 & -b_3\theta_x\theta_y + b_4\theta_z & -b_2\theta_x\theta_z + b_6\theta_y \\ -b_3\theta_x\theta_y + b_4\theta_z & b_2 + b_3\theta_x^2 + b_1\theta_z^2 & -b_1\theta_y\theta_z + b_5\theta_x \\ -b_2\theta_x\theta_z + b_6\theta_y & -b_1\theta_y\theta_z + b_5\theta_x & b_3 + b_2\theta_x^2 + b_1\theta_y \end{bmatrix} T_A \\ & + D_t, \end{aligned} \tag{22}$$

where I_{xx}, I_{yy}, and I_{zz} are the moments of inertia about the respective axes. The coefficients are defined as:

$$b_1 = \frac{1}{I_{xx}}, \quad b_2 = \frac{1}{I_{yy}}, \quad b_3 = \frac{1}{I_{zz}},$$
$$b_4 = \frac{I_{yy} - I_{xx}}{I_{xx}I_{yy}}, \quad b_5 = \frac{I_{zz} - I_{yy}}{I_{yy}I_{zz}}, \quad b_6 = \frac{I_{xx} - I_{zz}}{I_{xx}I_{zz}},$$

and $D_t = \mathbf{R}(\boldsymbol{\theta})\mathbf{I}_g^{-1}\mathbf{R}(\boldsymbol{\theta})^\top(T_B + T_C + T_D)$ represents the total disturbance torque including contributions from all external forces.

Substituting (20) into (22), the Euler equations for the FOWT can be rewritten as:

$$
\begin{aligned}
\ddot{\boldsymbol{\theta}} &= \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda,\beta)\eta_v^2 a_3 \\
&\begin{bmatrix} b_1 + b_3\theta_y^2 + b_2\theta_z^2 & -b_3\theta_x\theta_y + b_4\theta_z & -b_2\theta_x\theta_z + b_6\theta_y \\ -b_3\theta_x\theta_y + b_4\theta_z & b_2 + b_3\theta_x^2 + b_1\theta_z^2 & -b_1\theta_y\theta_z + b_5\theta_x \\ -b_2\theta_x\theta_z + b_6\theta_y & -b_1\theta_y\theta_z + b_5\theta_x & b_3 + b_2\theta_x^2 + b_1\theta_y \end{bmatrix} \begin{bmatrix} \theta_x\theta_y - \theta_z \\ \theta_y^2 + 1 \\ \theta_y\theta_z + \theta_x \end{bmatrix} \\
&+ D_\theta \\
&= \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda,\beta)\eta_v^2 a_3 \begin{bmatrix} -b_2\theta_z(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1) \\ b_2(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1) \\ -b_2\theta_x(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1) \end{bmatrix} + D_\theta \\
&= \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda,\beta)\eta_v^2 a_3 b_2 \left(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1\right) \begin{bmatrix} -\theta_z \\ 1 \\ -\theta_x \end{bmatrix} + D_\theta \\
&= d_\theta C_t(\lambda,\beta)\eta_v^2 \left(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1\right) \begin{bmatrix} -\theta_z \\ 1 \\ -\theta_x \end{bmatrix} + D_\theta,
\end{aligned} \tag{23}
$$

where D_θ represents the cumulative disturbance, and is defined as:

$$
D_\theta = \mathbf{R}(\boldsymbol{\theta})\mathbf{I}_g^{-1}\mathbf{R}(\boldsymbol{\theta})^\top D_g + D_t. \tag{24}
$$

while d_θ is a scaling factor, calculated as:

$$
d_\theta = \frac{1}{2}\rho_a \pi R_r^2 a_3 b_2. \tag{25}
$$

Based on (23), the dynamics of the platform pitch rate of the semi-submersible FOWT can be written as:

$$
\dot{\omega}_y = d_\theta C_t(\lambda,\beta)\eta_v^2 \left(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1\right) + D_{\theta_y}, \tag{26}
$$

where D_{θ_y} represents the y-axis component of D_θ in the frame $\mathcal{F}^0$.

By substituting the thrust coefficient C_t with its polynomial approximation (13), the dynamics can be further expanded as:

$$
\dot{\omega}_y = d_\theta(g_{ct}\beta + f_{ct})\eta_v^2 \left(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1\right) + D_{\theta_y}. \tag{27}
$$

This can be simplified and expressed in terms of control influence and disturbances:

$$
\dot{\omega}_y = g_y\beta + D_y, \tag{28}
$$

where g_y is a known function considering the dynamics influenced by the rotor configuration and environmental variables, and D_y encapsulates the polynomial fitting error and cumulative disturbances affecting the pitch rate dynamics, defined as:

$$
g_y = d_\theta g_{ct}\eta_v^2(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1), \tag{29}
$$

$$
D_y = d_\theta f_{ct}\eta_v^2(\theta_x^2 + \theta_y^2 + \theta_z^2 + 1) + D_{\theta_y}. \tag{30}
$$

Assumption 1. *The disturbance D_y and its time derivative $\dot{D}_y$ are bounded such that: $|D_y| \le \rho_{y1}$ and $|\dot{D}_y| \le \rho_{y2}$ where ρ_{y1} and ρ_{y2} are two unknown positive constants.*

2.2.3. Dynamics of the rotational rotor speed ω_r

Similar to the thrust coefficient of the wind turbine, the power coefficient C_p is approximated using a polynomial function, where the coefficients o_{ij} are determined through polynomial regression and are provided in Table 4:

$$
C_p(\lambda,\beta) = g_{cp}\beta + f_{cp}, \tag{31}
$$

where:

$$g_{cp}(\lambda) = o_{10} + o_{11}\lambda + o_{12}\lambda^2 + o_{13}\lambda^3 + o_{14}\lambda^4, \tag{32}$$

and

$$f_{cp}(\lambda) = \left(o_{00} + o_{01}\lambda + o_{02}\lambda^2 + o_{03}\lambda^3 + o_{04}\lambda^4 + o_{05}\lambda^5\right) + \Delta C_p. \tag{33}$$

with the term ΔC_p representing the fitting error inherent in the polynomial approximation.

o_{ij}	**Value**	o_{ij}	**Value**	o_{ij}	**Value**
o_{00}	-0.4835	o_{01}	0.7824	o_{02}	-0.3106
o_{03}	-0.03193	o_{04}	-0.01383	o_{05}	0.001998
o_{10}	-0.8757	o_{11}	-0.7761	o_{12}	-0.3552
o_{13}	-0.05165	o_{14}	-0.002998		

Table 4: Values of polynomial coefficients for C_p.

Substituting the aerodynamic power P_A expression (8) into the one-mass rigid shaft model, the dynamics of the rotor speed of the semi-submersible can be expressed as:

$$\dot{\omega}_r = \frac{1}{J_l}\left(\frac{1}{2}\rho_a \pi R_r^2 \frac{C_p(\lambda,\beta)}{\omega_r}\|\mathbf{v}_n\|^3 - \eta_g T_g\right). \tag{34}$$

Subsequently, substituting C_p by its polynomial expression (31), the dynamic equation for the rotor speed can be rewritten as:

$$\begin{aligned}\dot{\omega}_r &= \frac{1}{J_l}\left(\frac{1}{2}\rho_a \pi R_r^2 \frac{(g_{cp}\beta + f_{cp})}{\omega_r}\|\mathbf{v}_n\|^3 - \eta_g T_g\right) \\ &= \frac{1}{J_l}\left(\frac{\rho_a \pi R_r^2}{2\omega_r}(g_{cp}\beta + f_{cp})\eta_v^3 - \eta_g T_g\right).\end{aligned} \tag{35}$$

Thus, the dynamic of the rotor speed can be expressed as:

$$\dot{\omega}_r = g_r\beta + D_r, \tag{36}$$

with

$$g_r = \frac{\rho_a \pi R_r^2 g_{cp}\eta_v^3}{2J_l\omega_r}, \quad \text{and} \quad D_r = \frac{\rho_a \pi R_r^2 f_{cp}\eta_v^3}{2J_l\omega_r} - \frac{\eta_g}{J_l}T_g, \tag{37}$$

where g_r is a known function, and D_r encapsulates the polynomial fitting of C_p, excluding the pitch component, along with cumulative disturbances.

Assumption 2. *The disturbance D_r and its time derivative $\dot{D}_r$ are bounded such that: $|D_r| \leq \rho_{r1}$ and $|\dot{D}_r| \leq \rho_{r2}$ where ρ_{r1} and ρ_{r2} are two unknown positive constants.*

2.3. Validation of the reformulated dynamics

The original Homer COM was reproduced in the Matlab/Simulink environment under specific wind and wave conditions that align with those used in its published research [28]. Comparative simulations presented in [28, 25] indicate a strong correlation between the primary motion behaviors predicted by this COM and the outputs from the high-fidelity OpenFAST software [60].

Developed by NREL, OpenFAST provides high-fidelity modeling of FOWT dynamics by integrating detailed aerodynamic, hydrodynamic, and structural models. Figure 2 illustrates the key OpenFAST modules used for FOWT emulation, which are defined as follows:

- **InflowWind:** Provides wind field inputs at the rotor, including speed and turbulence intensity.
- **ElastoDyn:** Simulates elastic structural dynamics of the drivetrain, tower, and nacelle.

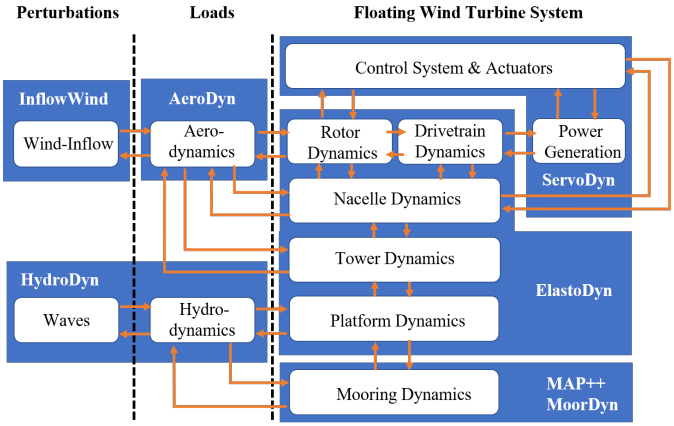

Figure 2: Architecture of the OpenFAST simulation software, depicting key modules used in FOWT system emulation.

Table 5: General, aerodynamic, and generator properties of the Homer model.

Property	**Unit**	**Value**
Rated power	MW	5
ω_{rd}	rpm	12.1
T_{gd}	N·m	43,093.55
m_g	kg	14,072,718
I_{xx}	kg·m^2	1.695×10^{10}
I_{yy}	kg·m^2	1.695×10^{10}
I_{zz}	kg·m^2	1.845×10^{10}
θ_{tilt}	deg	5
R_r	m	62.94
ρ_a	kg/m^3	1.225
$\mathbf{r}_{gt}^b$	m	$[-5, 0, 99.889]^T$
J_r	kg·m^2	3.5444×10^7
J_g	kg·m^2	534.116
η_g	-	0.97

- **BeamDyn:** Models blade flexibility using beam-type finite element methods.
- **AeroDyn:** Calculates aerodynamic loads using BEMT.
- **HydroDyn:** Simulates hydrodynamic forces acting on floating structures.
- **Mooring:** Models mooring system dynamics via MAP++, FEAMooring, or MoorDyn.
- **ServoDyn:** Simulates pitch and torque actuator dynamics for control systems.

OpenFAST includes 26 preconfigured wind turbines with 24 DoFs, among which the DeepCWind OC4 5 MW FOWT model closely represents the semi-submersible wind turbine foundation used in the Homer COM. Table 5 summarizes its key properties.

The strong agreement between the Homer COM and OpenFAST across all considered DoFs, as demonstrated in [28, 25], confirms the reliability of this COM for control design while maintaining computational efficiency. Similarly, to validate the reformulated dynamics for rotor speed ω_r (36) and platform pitch rate ω_y (28), an open-loop response test was conducted under realistic wind and wave conditions characteristic of Region III.

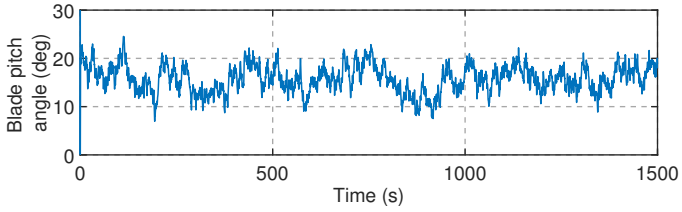

Figure 3: Collective blade pitch control input for open-loop validation.

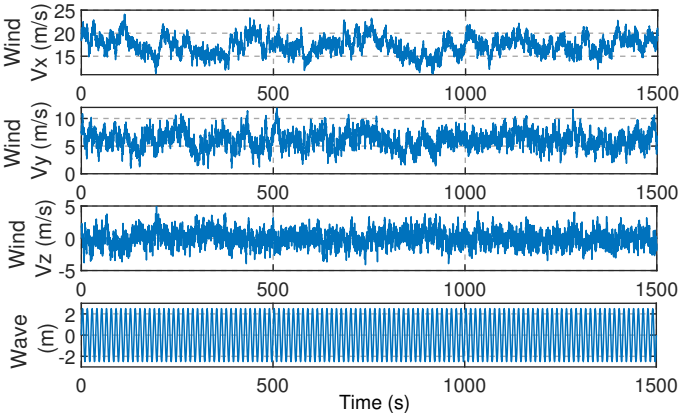

Figure 4: Environmental conditions used for open-loop validation, including 3-D wind speed profiles and wave height variations.

2.3.1. Open-loop configuration

This test aims to compare the time responses of the proposed ω_r- and ω_y-dynamics with those of the OpenFAST model in operating Region III. In accordance with the assumptions made during the derivation of these dynamics, the generator torque control input is fixed at its rated value, $T_{\text{gd}} = 43\,093.55$ N.m, while the nacelle yaw motion is neglected and set to zero. Consequently, only the collective blade pitch control input is considered for open-loop validation. The applied blade pitch control input, as specified in the reference study [28], is shown in Figure 3.

The simulation also employs the same wind and wave conditions as those presented in the reference study. The corresponding 3-D wind speed profiles and wave height variations are illustrated in Figure 4.

2.3.2. Results

The time series results for the simulations conducted in both OpenFAST and the reformulated dynamics are presented in Figure 5. The statistical comparisons, including the mean, maximum, minimum, and standard deviation (STD) values, are summarized in Table 6. Additionally, the Root Mean Square Error (RMSE) is computed with respect to OpenFAST, which serves as the reference model, for both dynamics.

The open-loop validation results demonstrate a strong agreement between the reformulated dynamics and OpenFAST. For rotor speed, the mean values align closely (10.6154 rpm vs. 10.2494 rpm), with a low RMSE of 0.39027 rpm, confirming that the reformulated model effectively captures the main rotor speed variations. For platform pitch rate, the reformulated model accurately predicts the general trend but slightly underestimates extreme values, with a lower standard deviation (0.18605 deg/s vs. 0.29323 deg/s). The RMSE of 0.35759 deg/s remains acceptable, ensuring suitability for control-oriented applications.

3. RBFNN-based STA for collective blade pitch control in Region III

By leveraging the reformulated dynamics of rotor speed (36) and platform pitch rate (28) from the adapted Homer COM, this section outlines the design framework for an STA control law integrating an RBFNN as a disturbance ob-

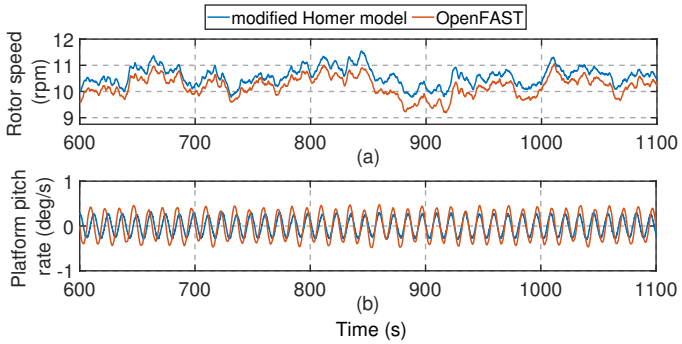

Figure 5: Open-loop validation of the modified Homer model compared to OpenFAST: (a) Rotor speed and (b) Platform pitch rate.

Table 6: Statistical comparison between OpenFAST and the modified Homer model.

	modified COM	**OpenFAST**
Rotor speed ω_r (rpm)		
Mean	10.6154	10.2494
Max	12.05	12.3601
Min	9.2612	9.0421
STD	0.38436	0.39718
RMSE (w.r.t. OpenFAST)	0.39027	
Platform pitch rate ω_y (deg/s)		
Mean	-0.00027	0.00170
Max	0.31722	0.94478
Min	-0.32971	-0.51657
STD	0.18605	0.29323
RMSE (w.r.t. OpenFAST)	0.35759	

server. It first defines the control objectives, followed by the formulation of the sliding variable and its time derivative. Subsequently, the STA controller design is detailed, including the neural network architecture.

3.1. Control objectives

In Region III, the primary control objectives are to regulate the generator's power to its rated value while ensuring stability in the platform's pitching motion. The generator power, P_g, can be expressed as follows:

$$P_g = \eta_e \eta_g T_g \omega_r, \tag{38}$$

where η_e represents the efficiency of the generator.

Given that T_g is fixed at its rated value, the first control objective simplifies to maintaining the rotor speed at its rated value, $\omega_{rd} = 12.1$ rpm, expressed as:

$$e_r = \omega_r - \omega_{rd}. \tag{39}$$

The second control objective focuses on reducing the platform's pitching motion. This is achieved by driving the platform pitch rate, ω_y, towards zero:

$$e_y = \omega_y. \tag{40}$$

3.2. Definition of the sliding variable

In an effort to enhance the stability of the FOWT, [61] introduces the concept known as Variable Pitch Power Control (VPPC). This strategy dynamically adjusts the nominal generator speed, increasing above the constant nominal value when the platform pitches upwind and decreasing when pitching downwind. This adjustment is realized by making the nominal generator speed, ω_{rd}, a linear function of the platform pitch rate, ω_y.

Drawing inspiration from [61], a practical solution for the underactuated control issue, arising from the presence of the same control input β in both the platform pitch rate ω_y and rotor speed ω_r dynamics, is to modify the reference rotor speed from a fixed value to a linear function related to the platform pitch rate:

$$\omega_{rd}^{*} = \omega_{rd}\left(1 - k_y\omega_y\right), \tag{41}$$

where ω_{rd}^{*} is the modified reference rotor speed and k_y is a positive constant defined empirically.

Thus, the primary control objective for the FOWT in Region III is to drive the Composite Tracking Error (CTE) to zero:

$$CTE = \omega_r - \omega_{rd}^{*} = \omega_r - \omega_{rd}\left(1 - k_y\omega_y\right) = e_r + k_s e_y, \tag{42}$$

where $e_r = \omega_r - \omega_{rd}$, $e_y = \omega_y - 0$, and $k_s = k_y\omega_{rd}$ is a positive scalar constant with units of $rpm \cdot s/deg$.

This variable is used as the sliding variable in all sliding mode controllers discussed in this chapter. The time derivative of the sliding variable s can then be expressed as:

$$\dot{s} = \dot{e_r} + k_s\dot{e_y} = \dot{\omega_r} + k_s\dot{\omega_y}. \tag{43}$$

Substituting the two reformulated dynamics expressions (36) and (28), the derivative simplifies to:

$$\dot{s} = (g_r\beta + D_r) + k_s\left(g_y\beta + D_y\right) = g_s\beta + D_s, \tag{44}$$

where $g_s = g_r + k_s g_y$ is a known function and $D_s = D_r + k_s D_y$ encapsulates the unknown lumped uncertainties and external disturbances.

Assumption 3. *Under Assumptions 1 and 2, it is held that $|D_s| \leq \rho_{s1}$ and $|\dot{D}_s| \leq \rho_{s2}$ where $\rho_{s1} = \rho_{r1} + k_s\rho_{y1}$ and $\rho_{s2} = \rho_{r2} + k_s\rho_{y2}$ are two unknown positive constants, respectively representing bounds on disturbances and their rates of change.*

3.3. Structure of RBFNN-based STA controllers

The dynamics of the sliding variable (44) incorporate the term D_s, which accounts for unmodeled dynamics, external disturbances, and modeling uncertainties within the adapted Homer model. To estimate and compensate for these lumped disturbances, a neural network is employed.

The overall structure of the proposed controller, integrating the STA with a neural network-based disturbance observer, is illustrated in Figure 6.

In this configuration, the control law β is composed of three distinct components:

$$\beta = \beta_{STA} - \beta_{ANN} - \beta_{\epsilon}, \tag{45}$$

where:

- β_{STA}: this component represents the standard STA control law for collective blade pitch control, ensuring the convergence of the system state to a vicinity of the origin, and defined as:

$$\beta_{STA} = -\frac{1}{g_s}\left(k_1\sqrt{|s(t)|}\mathrm{sgn}(s(t)) + k_2\int_0^t \mathrm{sgn}(s(\tau))\,d\tau\right), \tag{46}$$

  where $k_1 > 0$ and $k_2 > 0$ are controller gains.

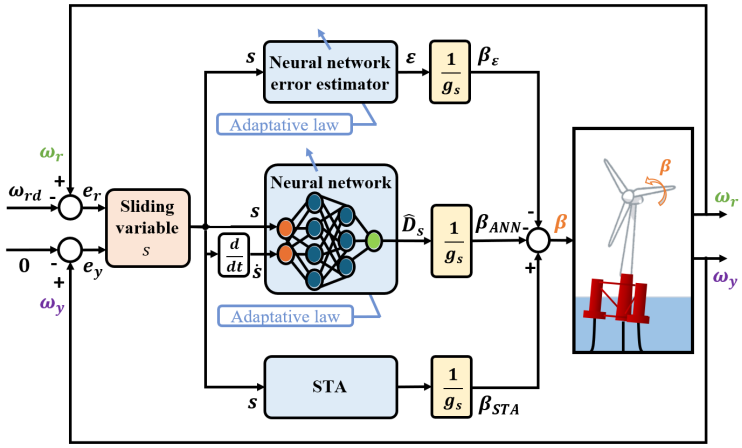

Figure 6: Block diagram of the neural network-based STA controller.

- β_{ANN}: this component aims to capture and compensate for the unknown lumped disturbances D_s. It is expressed as:

$$\beta_{ANN} = \frac{1}{g_s}\hat{D}_s, \tag{47}$$

  where $\hat{D}_s$ is the disturbance estimate output by the neural network observer.

- β_ϵ: this component addresses the approximation error ε inherent in the neural network.

$$\beta_\epsilon = \frac{1}{g_s}\varepsilon. \tag{48}$$

By incorporating a neural network to approximate the lumped uncertainties and disturbances D_s, the proposed controller aims to regulate the rotor speed to its rated value while minimizing the platform's pitching motion in Region III. An error estimator is integrated to offset the theoretical minimum estimation error of the neural network, thereby enhancing the accuracy and robustness of the control strategy.

3.4. RBFNN-based STA controller design

The well-explored class of RBFNN is highly regarded for its efficacy in control systems within dynamic environments, such as those encountered in FOWTs. Known for their capability as universal function approximators, RBFNNs can model any continuous function with remarkable accuracy, provided they are equipped with sufficient neurons and optimally tuned parameters. This feature, combined with their rapid convergence and simple architecture, makes RBFNNs ideally suited for tasks requiring quick and accurate modeling of nonlinear dynamics.

At its core, the RBFNN employs radial basis functions as activation functions, which play a critical role in transforming input data into a space where relationships among the data can be more easily interpreted and used by the neural network. RBFNNs have been effectively combined with various nonlinear control techniques, including SMC [62, 63, 64, 65, 66, 67, 70]. In particular, the integration of RBFNN-based STA controllers has been explored in applications like MEMS gyroscopes [68] or PMSG systems [69], indicating a broad applicability across various fields.

3.4.1. RBFNN architecture

The architecture of an RBFNN typically comprises three layers: an input layer, a hidden layer featuring radial basis functions, and an output layer. This structure is depicted in Figure 7.

Input layer: Directly captures operational parameters pertinent to FOWT dynamics. For this application, the sliding variable s and its derivative $\dot{s}$ are chosen as inputs, denoted x_1 and x_2 respectively, to allow for immediate reaction to dynamic changes:

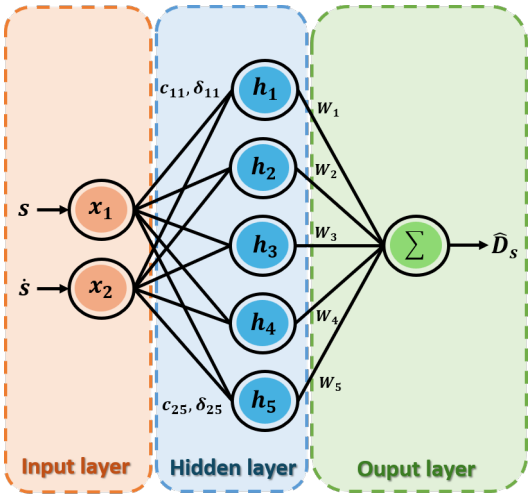

Figure 7: RBFNN architecture within the RBFNN-based STA.

$$x_1 = s, \quad x_2 = \dot{s}. \tag{49}$$

Hidden layer: This layer uses radial basis functions to transform the input space into a higher-dimensional space where linear separability is more achievable. Each of the n neurons in this layer processes the inputs using Gaussian functions with centers c_{ji} and widths δ_{ji}. The output h_j of each neuron j is given by:

$$h_j = e^{-g_j}, \tag{50}$$

with

$$g_j = \frac{\left(x_1 - c_{j1}\right)^2}{2\delta_{j1}^2} + \frac{\left(x_2 - c_{j2}\right)^2}{2\delta_{j2}^2} \tag{51}$$

Output layer: This layer aggregates the outputs from the hidden layer to form the final output of the network, $\hat{D}_s$, which is an estimate of the disturbances D_s:

$$\hat{D}_s = \mathbf{W}^\top \mathbf{h}, \tag{52}$$

where $\mathbf{W} = [W_1, \ldots, W_n]$ is the weight vector from the hidden to the output layer, and $\mathbf{h} = [h_1, \ldots, h_n]$ is the vector of outputs from the hidden layer.

With its universal approximation property, the RBFNN can optimally estimate the disturbance D_s when the output layer weights, $\mathbf{W}$, are ideally tuned. The theoretically minimal estimation error, ε^*, defines the precision limit of the network:

$$D_s = \mathbf{W}^{*\top} \mathbf{h} + \varepsilon^*, \tag{53}$$

where $\mathbf{W}^*$ represents the optimal weight configuration.

3.4.2. Stability analysis

The primary objective of the RBFNN is to estimate and compensate for the unknown lumped disturbance D_s, using inputs $\mathbf{x} = [s, \dot{s}]$, and producing the output $\hat{D}_s$, detailed in equation (52). The adaptation of the weight vector $\mathbf{W}^\top$ is governed by an adaptive law, formulated based on principles of Lyapunov stability.

Substituting the dynamics of the sliding variable (44) leads to the equation:

$$
\begin{aligned}
\dot{s} &= g_s\beta + D_s = g_s\left(\beta_{STA} - \beta_{ANN} - \beta_\epsilon\right) + D_s \\
&= -k_1\sqrt{|s|}\mathrm{sgn}(s) - k_2\int_0^t \mathrm{sgn}(s)\,d\tau - \hat{D}_s - \varepsilon + D_s \\
&= -k_1\sqrt{|s|}\mathrm{sgn}(s) - k_2\int_0^t \mathrm{sgn}(s)\,d\tau - \Delta\mathbf{W}^\top\mathbf{h} - \Delta\varepsilon
\end{aligned}
\tag{54}
$$

where $\Delta\mathbf{W} = \mathbf{W} - \mathbf{W}^*$ and $\Delta\varepsilon = \varepsilon - \varepsilon^*$.

Assumption 4. *It is considered that $\mathbf{W}^*$ and ε^* are constant in the derivation of the adaptive laws.*

The following theorem establishes the criteria for selecting the control gains and learning laws of the RBFNN-based STA controller to ensure the asymptotic convergence of s to the origin.

Theorem 1. *For the system defined, with control gains $k_1 > 0$ and $k_2 > 0$, and properly selected learning laws, s will converge asymptotically to the origin. The learning laws are defined as:*

$$
\dot{\mathbf{W}}^\top = \eta_1 k_2 sgn(s)\mathbf{h}, \tag{55}
$$

$$
\dot{\varepsilon} = \eta_2 k_2 sgn(s), \tag{56}
$$

where η_1 and η_2 are positive learning rates.

Proof. The system dynamics (54) can be converted into:

$$
\begin{cases}
\dot{s} = -k_1\sqrt{|s|}\mathrm{sgn}(s) - \Delta\mathbf{W}^\top\mathbf{H} - \Delta\varepsilon + \phi, \\
\dot{\phi} = -k_2\mathrm{sgn}(s).
\end{cases}
\tag{57}
$$

Based on [71], the following Lyapunov candidate function is selected:

$$
V = k_2|s| + \frac{1}{2}\phi^2 + \frac{1}{2\eta_1}\Delta\mathbf{W}^\top\Delta\mathbf{W} + \frac{1}{2\eta_2}\Delta\varepsilon^2. \tag{58}
$$

The first-time derivative of V is then expressed as:

$$
\begin{aligned}
\dot{V} &= k_2\dot{s}\mathrm{sgn}(s) + \phi\dot{\phi} + \frac{1}{\eta_1}\Delta\mathbf{W}^\top\Delta\dot{\mathbf{W}} + \frac{1}{\eta_2}\Delta\varepsilon\Delta\dot{\varepsilon} \\
&= -k_1k_2\sqrt{|s|} - \frac{1}{\eta_1}\Delta\mathbf{W}^\top\left(\eta_1 k_2\mathbf{h}\mathrm{sgn}(s) - \Delta\dot{\mathbf{W}}\right) \\
&\quad - \frac{1}{\eta_2}\Delta\varepsilon\left(\eta_2 k_2\mathrm{sgn}(s) - \Delta\dot{\varepsilon}\right).
\end{aligned}
\tag{59}
$$

Thus, based on Assumption 4 and the learning laws (55) and (56), the first-time derivative of V simplifies to:

$$
\dot{V} = -k_1k_2\sqrt{|s|}. \tag{60}
$$

Since $\dot{V} \leq 0$, it follows that $\dot{V}$ is negative semi-definite, indicating the asymptotic convergence of s to the origin. This completes the proof.

□

4. HIL platform description and real-time validation

HIL validation enables real-time testing of control laws by interfacing actual controllers with a simulated turbine environment. This approach mimics real-world FOWT behavior while providing a safe and controlled setting for evaluating control strategies before deployment. Real-time simulation ensures accurate performance assessment, allowing developers to analyze system responses, detect potential faults, and refine control algorithms under realistic operating conditions. This step is critical for validating controller robustness and effectiveness before integration into a physical system.

4.1. HIL platform description

The HIL platform (see Figure 8) developed at FEMTO-ST laboratory enables real-time simulation of the OpenFAST model, ensuring accurate emulation of FOWT dynamics while facilitating smooth communication between hardware and software components.

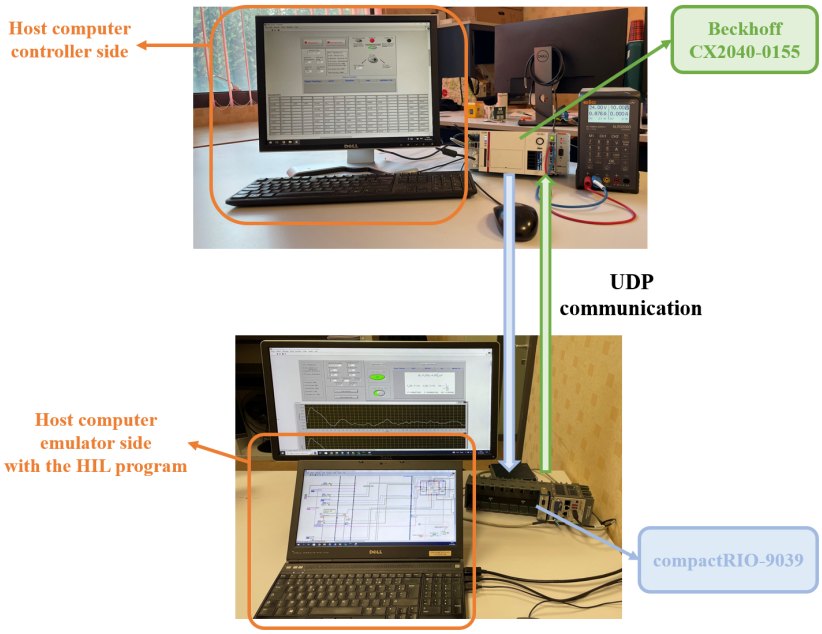

Figure 8: Hardware setup featuring the compactRIO-9039 at the bottom and the remote Beckhoff CX2040-0155 at the top, communicating via UDP.

The platform consists of the following key elements:

- **Software emulator:** The software emulator represents the analytical or numerical model that simulates the FOWT's dynamic behavior. OpenFAST v3.1.0 is used as the high-fidelity model for real-time emulation of the OC4 DeepCWind 5 MW semi-submersible FOWT.
- **Real-time emulator implementation:** To execute OpenFAST in real-time, the compactRIO-9039 (cRIO) is used due to its high processing capabilities and real-time performance characteristics (Table 7). The OpenFAST model is compiled into an executable program optimized for NI Linux RTOS, a UNIX-based real-time operating system that ensures efficient execution.

Table 7: Characteristics of the NI card used in the HIL setup.

Specification	Details
Brand	National Instruments (NI)
Model	compactRIO-9039
OS	NI Linux Real-Time
Processor	Intel Atom Quad-Core 1.91 GHz
RAM	2 GB
Storage	16 GB

- **Remote control unit:** For executing control laws and ensuring robust communication, the Beckhoff CX2040-0155 industrial PC is used (Table 8). Widely employed in industrial automation and control systems, the Beckhoff is responsible for loading and executing real-time control processes while facilitating data exchange with the emulator. Its high-speed Ethernet ports support various communication protocols, ensuring minimal latency.

Table 8: Characteristics of the Beckhoff CX2040-0155 control card in the HIL setup.

Specification	Details
Brand	Beckhoff
Model	CX2040-0155
OS	Windows
Processor	Intel Core™ i7, 2715QE 2.1 GHz, 4 cores
RAM	4 GB
Storage	40 GB

The controller equations are encoded in LabVIEW, compiled into an executable, and deployed on the Beckhoff unit.

- **Communication protocol:** The HIL platform uses the User Datagram Protocol (UDP) for real-time data exchange between the emulator (cRIO) and control unit (Beckhoff). UDP is chosen for its low latency, connectionless operation, and minimal overhead, making it well suited for fast and efficient real-time control applications.

- **Communication interfaces:** LabVIEW interfaces manage communication between OpenFAST (on cRIO) and the controller (Beckhoff), allowing users to monitor and interact with the HIL simulation.

  On the cRIO host computer, a LabVIEW wrapper integrates OpenFAST with LabVIEW, executing key simulation functions via dynamic library calls. The Wrapper includes:

  - `sim_init()`: Initializes the OpenFAST simulation (called once at startup).
  - `sim_equilibrium()`: Computes the equilibrium output at $t = 0s$.
  - `sim_timestep()`: Updates OpenFAST outputs at each control step.
  - `sim_end()`: Terminates the OpenFAST simulation.

  These functions run sequentially within LabVIEW, using *Call Library Node* blocks, which invoke specific functions from the dynamic library. Especially, the `sim_timestep()` function is embedded in a timed loop to maintain real-time execution and synchronization throughout the simulation.

4.2. Integration of OpenFAST into the HIL platform

To achieve this setup, OpenFAST must be compiled on the cRIO platform, integrating dynamic libraries that enable seamless interaction with the LabVIEW Wrapper. As shown in Figure 9, the compilation process creates a Linux executable and shared libraries (.so files), which are linked with the LabVIEW Wrapper. These shared libraries form a unified real-time simulation environment, allowing OpenFAST to function within LabVIEW's control interface.

4.2.1. Interfacing external control inputs with OpenFAST

To integrate external control inputs into OpenFAST within the cRIO, the function `FAST_SetExternalInputs()` propagates variables through OpenFAST modules. The key control variables—generator power (P_g), generator torque (T_g) and collective blade pitch angle (β)—are the only actively managed variables. The data flow follows this operational sequence:

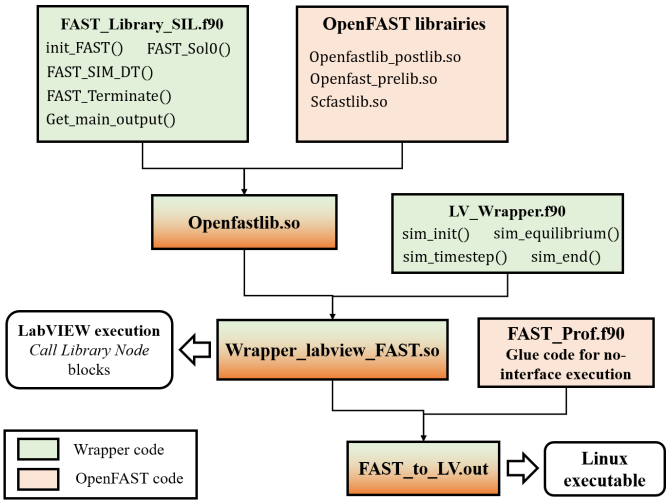

Figure 9: Block diagram of the interactions between OpenFAST library and LabVIEW Wrapper.

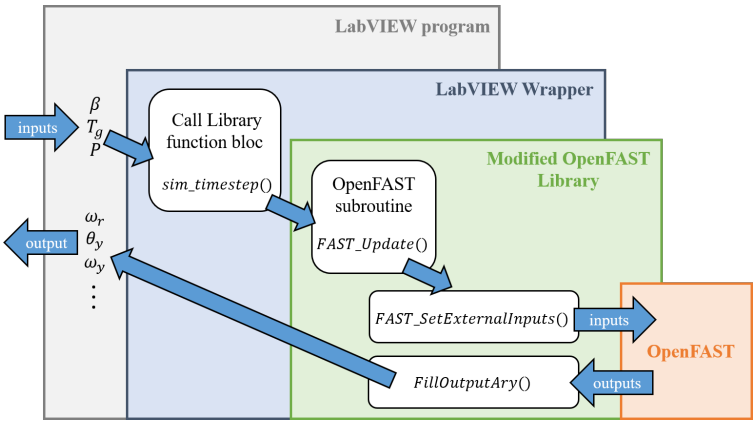

Figure 10: Workflow for integrating an external LabVIEW command within OpenFAST.

1. LabVIEW receives control inputs from the Beckhoff controller, passing them to `sim_timestep()` via a Call Library Node.
2. The `sim_timestep()` function invokes OpenFAST subroutine `FAST_Update`.
3. `FAST_Update` propagates the control data within OpenFAST by calling two crucial functions:
   - `FAST_SetExternalInputs()`: Updates control inputs, particularly in `ServoDyn`, which computes the next system states.
   - `FillOutputAry()`: Returns the computed OpenFAST outputs to LabVIEW.

This workflow, illustrated in Figure 10, ensures that control inputs are accurately integrated into OpenFAST, and the real-time dynamics of the FOWT system are effectively simulated and updated within the HIL framework.

This integration bridges OpenFAST's high-fidelity modeling with LabVIEW's control environment, enabling real-time HIL testing for advanced FOWT control validation.

4.2.2. HIL execution and real-time synchronization

The block diagram in Figure 11 illustrates the key components and interconnections within the real-time system on the cRIO. At its core, the four compiled OpenFAST functions—`sim_init()`, `sim_equilibrium()`, `sim_timestep()`, and `sim_end()`—handle model initialization, dynamic simulation, and termination.

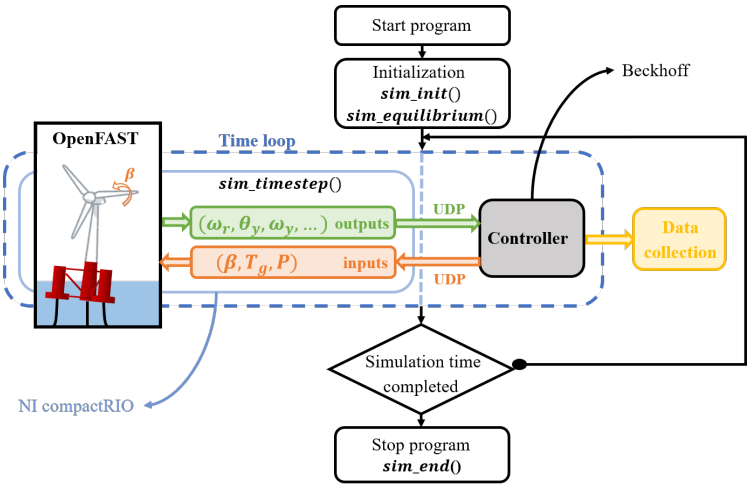

Figure 11: Block diagram of the LabVIEW program within the cRIO host computer.

Two key mechanisms ensure real-time performance. First, timer loop regulation enforces a strictly timed execution loop, ensuring simulation time steps align with the physical system's constraints. A 0.027 s time step is chosen to maintain synchronization between the controller and emulator, enabling real-time control simulations. Second, UDP-based communication facilitates real-time data exchange between the Beckhoff controller and cRIO, allowing the control system to dynamically adapt and accurately replicate FOWT operation. Additionally, a data collection module logs key simulation parameters, enabling detailed post-simulation analysis to compare expected vs. actual control performance.

4.2.3. Data exchange between the controller and emulator

The UDP connection between the emulator (cRIO) and controller (Beckhoff) is structured into two main modules on each side, one for receiving data and one for transmitting data.

Controller side:.

- Data reception (Module 1): The controller receives critical system states from OpenFAST, including rotor speed, platform pitch angle, and pitch rate. This data is handled by the UDP READ VI, where the "ID receive data" input specifies the cRIO's IP address as the sender. A start/stop control signal ensures synchronization with the emulator. If no data is received within 5 ms, an error code is generated, halting the simulation to prevent communication failures.
- Data transmission (Module 2): After processing the received states, the controller computes and sends back control inputs to the emulator. These inputs include generator power, torque, and blade pitch angle.

Emulator side:.

- Data reception (Module 1): The emulator receives control inputs from the controller, verifies the data format, and applies the torque, power, and blade pitch commands to the real-time OpenFAST simulation, updating the turbine's operational states accordingly.
- Data transmission (Module 2): The cRIO transmits updated system states along with the start/stop signal back to the controller in real time.

In total, the emulator transmits 32 states variables from OpenFAST to the controller, including: the rotor and generator speeds, the 6 DoFs platform motions, comprising the 3 translational motions (surge, sway, and heave), and the 3 rotational motions (roll, pitch, and yaw). In return, the controller sends the generator power, generator torque, and collective blade pitch angle.

4.3. HIL real-time validation

The following subsection details the closed-loop validation process of control laws on the HIL platform.

4.3.1. Closed-loop validation process

The LabVIEW HIL program operation is structured as follows and illustrated in Figure 14.

1. **Controller program initialization:**
   The controller program is launched on the host computer connected to the Beckhoff system. In the LabVIEW interface (Figure 12), the user must enter: IP address of the cRIO (HIL host computer) to establish communication, program iteration time, desired controller (selected from those implemented on the Beckhoff). Once settings are configured, the program starts and enters standby mode, indicated by the "Stop Loop" light turning red.

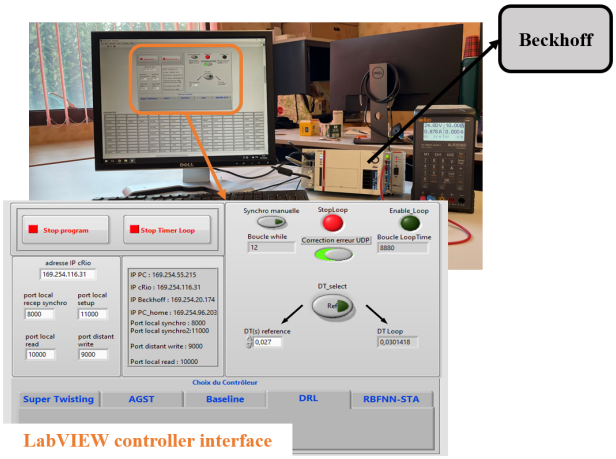

Figure 12: LabVIEW user interface for controller setup on the Beckhoff host computer.

2. **HIL setup on the cRIO host PC:**
   The cRIO host computer (Figure 13) runs the HIL program and requires: controller computer's IP address, simulation duration, program loop iteration time. Once all settings are entered, the HIL program is started.
3. **Initialization phase:**
   The cRIO emulator initializes OpenFAST by calling `sim_init()`, which sets up the initial conditions. The `sim_equilibrium()` function computes the initial system outputs, ensuring a consistent starting point.
4. **Synchronous start:**
   Once initialized, the HIL program synchronizes with the controller and starts the real-time loop. When the simulation starts on the cRIO, the controller executable is automatically triggered, ensuring full synchronization between both systems.
5. **Real-time closed-loop execution:**
   Both the controller (Beckhoff) and emulator (cRIO) run simultaneously during the simulation. At each simulation time step, the `sim_timestep()` function updates the 32 system state variables based on the latest control inputs. The controller receives the new states via UDP, computes the control inputs (torque, power, blade pitch), and transmits them back to the emulator via UDP. All exchanged data are continuously recorded for post-simulation analysis.
6. **Simulation termination:**
   Once the simulation reaches its end time, the HIL program automatically stops by calling `sim_end()`. The controller returns to standby mode until a new simulation starts.
7. **Data collection and analysis:**
   After the simulation, key system parameters such as blade pitch angle, rotor speed, platform motions, can be

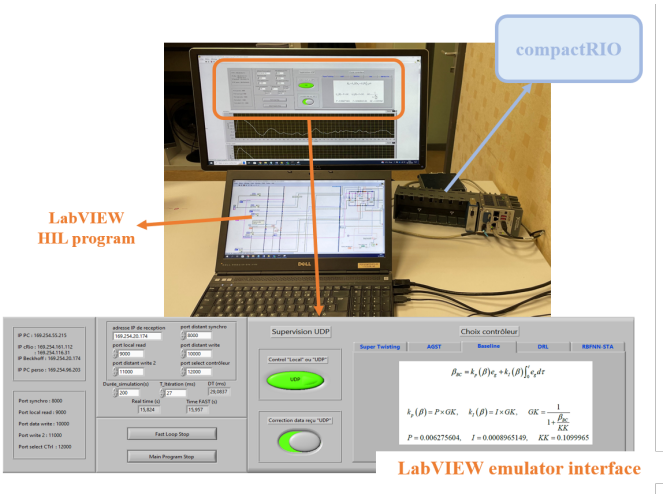

Figure 13: LabVIEW user interface for HIL setup on the cRIO host computer.

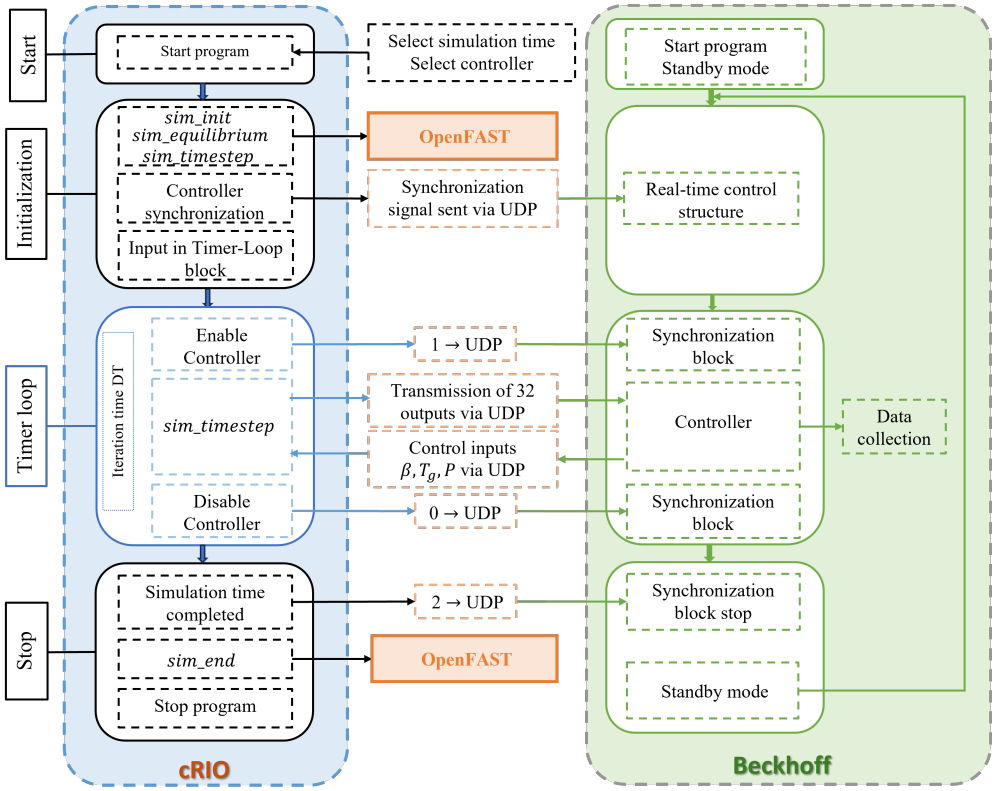

Figure 14: Detailed operating of the LabVIEW program for the HIL testing of a controller.

visualized in real-time graphs on the LabVIEW interface of the cRIO. This ensures that the recorded data is suitable for further analysis and controller performance evaluation.

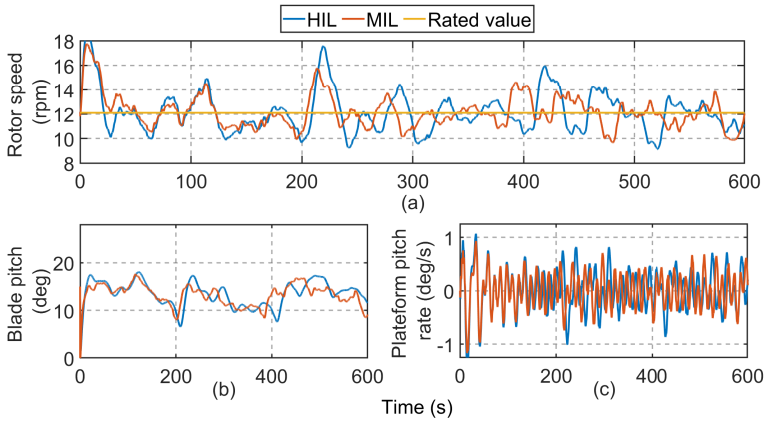

Figure 15: Comparison of the Baseline controller performance between MIL and HIL for: (a) Rotor speed, (b) Platform pitch rate, and (c) Blade pitch angle.

4.3.2. Baseline controller validation

To assess the accuracy and reliability of the HIL platform, a comparative validation was conducted between the GSPI baseline controller [72] running on the HIL platform and its Model-in-the-Loop (MIL) implementation. The objective was to ensure that the real-time execution of OpenFAST on the cRIO emulator, coupled with real-time control execution on the Beckhoff controller, accurately reproduces the system dynamics observed in MIL simulations.

The GSPI controller, widely used as a reference for FOWT rotor and platform control, was implemented on both MIL and HIL under identical wind and wave conditions presented in Figure 4. The MIL simulation was performed using MATLAB-Simulink, where OpenFAST was interfaced with Simulink to simulate the floating wind turbine's dynamics. In this setup, the controller was implemented as a Simulink block running synchronously with OpenFAST, ensuring ideal execution without real-time constraints. In contrast, the HIL setup ran the same baseline controller on the Beckhoff industrial PC, executing in real time while exchanging data with OpenFAST on the cRIO emulator via UDP communication.

The control parameters used in the HIL implementation are detailed in Table 9.

Table 9: Baseline GSPI: control parameters.

Variable	**Value**	**Description**
K_p	0.006275604	Proportional gain for the pitch controller
K_i	0.0008965149	Integral gain for pitch control
K_k	0.1099965	Pitch angle at which power derivative doubles
$\beta_{\max}$	1.570796	Maximum blade pitch setting (rad)
$\beta_{\min}$	0.0	Minimum blade pitch setting (rad)
β_{rate}	0.1396263	Maximum blade pitch rate (rad/s)
ω_{cfreq}	1.570796	Corner frequency of the low-pass filter (rad/s)
ω_{rd}	12.1	Rated rotor speed (rpm)

To assess the accuracy of the HIL execution, key system variables were compared, including rotor speed ω_r, blade pitch angle β, platform pitch rate ω_y, and all 6 DoFs of platform motion. The time series comparison between MIL and HIL simulations, shown in Figures 15 and 16, shows general consistency during the first 200 seconds, with qualitative similarity in dynamics despite some minor deviations that begin to emerge early in the simulation. Beyond 200 seconds, a progressive phase offset appears, particularly evident in the platform motions.

To quantify this discrepancy, RMSE was computed for key system variables using MIL results as reference. The computed RMSE values after 200 seconds are summarized in Table 10.

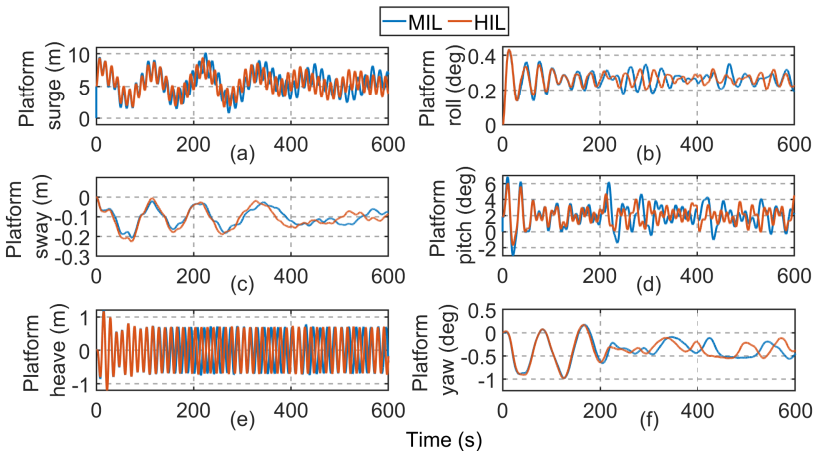

Figure 16: Comparison of the Baseline controller performance between MIL and HIL for the 6 DoFs platform motions: (a) Surge, (b) Roll, (c) Sway, (d) Pitch, (e) Heave, and (f) Yaw motion.

Table 10: RMSE values comparing HIL and MIL outputs after 200s.

Variable	**RMSE (HIL vs MIL)**
Rotor speed (ω_r)	0.61582 rpm
Blade pitch angle (β)	0.94427 deg
Platform pitch rate (ω_y)	0.097485 deg/s
Platform surge (x_g)	0.4137 m
Platform sway (y_g)	0.010716 m
Platform heave (z_g)	0.027934 m
Platform roll angle (θ_x)	0.013133 deg
Platform pitch angle (θ_y)	0.48693 deg
Platform yaw angle (θ_z)	0.028645 deg

The observed divergence after 200 seconds likely results from the accumulation of small-scale effects inherent to HIL execution, including:

- **Real-time execution constraints in HIL:** Unlike MIL, which runs as fast as possible in an ideal environment, HIL enforces strict timing, where even small jitter in the solver or controller can accumulate over time, leading to the observed delay.
- **UDP communication latency:** UDP is used for exchanging data between the cRIO emulator and the Beckhoff controller. While efficient, its connectionless nature can lead to minor but accumulating timing inconsistencies.
- **Loop drift in LabVIEW:** The HIL simulation loop is governed by a LabVIEW timed loop. Any deviation between intended and actual loop execution time can introduce progressive drift, leading to phase shifts after extended operation.

Despite these effects, the RMSE values remain within acceptable bounds for real-time validation. Indeed, the rotor speed tracking error remains low at 0.61582 rpm, which is negligible in practical operation. The blade pitch error (0.94427 deg RMSE) is also small and does not introduce major deviations in aerodynamic performance. The platform pitch rate and pitch angle errors are 0.097485 deg/s and 0.48693 deg, respectively, confirming minimal impact on stability. The 6 DoFs platform motion errors remain within a narrow range, with platform roll, pitch, and yaw angles remaining below 0.5 degrees, ensuring structural stability.

These results confirm the reliability and accuracy of the HIL platform for real-time control validation of FOWTs, providing a realistic and robust environment to assess controller performance.

4.4. RBFNN-based STA real-time validation

To design the RBFNN-STA control law, a nonlinear COM derived from the modified HOMER framework is employed. This model simplifies the coupled dynamics of the FOWT to allow for real-time control synthesis. However, to rigorously assess the controller's robustness, all simulations within the HIL platform are conducted using OpenFAST. As such, the RBFNN-based STA controller is inherently tested under model mismatch conditions, and its performance demonstrates its ability to adapt to real-time unmodeled disturbances and uncertainties. This approach eliminates reliance on perfect model knowledge and supports the robustness of the proposed control strategy in realistic offshore operating scenarios.

The developed RBFNN-based STA controller was successfully integrated into the Beckhoff system and tested on the HIL platform for real-time validation. Both the forward pass, responsible for generating control inputs, and the backward pass, responsible for updating the RBFNN parameters, were fully implemented in LabVIEW. This enabled real-time adaptation during HIL simulations, allowing the network to dynamically learn and adjust its parameters under real-time conditions.

The performance of the RBFNN-based STA was evaluated against the STA without neural networks and the Baseline GSPI, assessing their ability to achieve control objectives across four test scenarios.

4.4.1. Environmental conditions

The five Environmental Conditions (ECs) were selected to comprehensively cover the operational range of Region III (Table 11). They include three different mean wind speeds with varying turbulence intensities, and three distinct wave profiles. The corresponding wind and wave profiles are illustrated in Figure 17 and 18, respectively.

EC	Wind Speed (m/s)	Turbulence (%)	Wave Height (m)	Wave Period (s)
EC1	18	5	2.6	8.36
EC2	20	5	3.5	9.08
EC3	18	15	2.6	8.36
EC4	20	12	3.5	9.08
EC5	30	20	5	11.5

Table 11: Environmental conditions used for simulation scenarios.

Wind conditions were generated using NREL's TurbSim software [73], applying a power law wind profile and the Kaimal turbulence model [74] over 600 seconds. Among the tested scenarios, EC3—with a turbulence intensity of 15%—produces instantaneous wind speeds ranging from 10.72 m/s to 25.32 m/s. This fully spans the Region III operating range (from rated to cut-out wind speed), making it a particularly stringent robustness test. Additionally, EC5, with its high mean wind speed of 30 m/s, approaches the turbine's cut-out threshold and represents a highly demanding operational condition for controller evaluation.

Wave conditions were generated using the HydroDyn module in OpenFAST, based on the Pierson-Moskowitz spectrum [75], to simulate irregular ocean wave patterns that closely replicate real-world conditions.

4.4.2. Validation results

The HIL simulations were conducted for 600 seconds with a sampling period of 0.027 seconds. The sliding mode and super-twisting parameters, which remained constant across all controllers and ECs, are as follows: the sliding variable coefficient is $k_y = 0.05$, and the STA gains are $k_1 = 1.2$ and $k_2 = 1.6$. The neural network configuration of the RBFNN-based STA is defined as:

- Hidden layer with $n = 5$ neurons.
- Gaussian basis function parameters:

$$c = \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix}, \quad b = 2.$$

- Initial weight values: $W = \text{ones}(1, n)$.

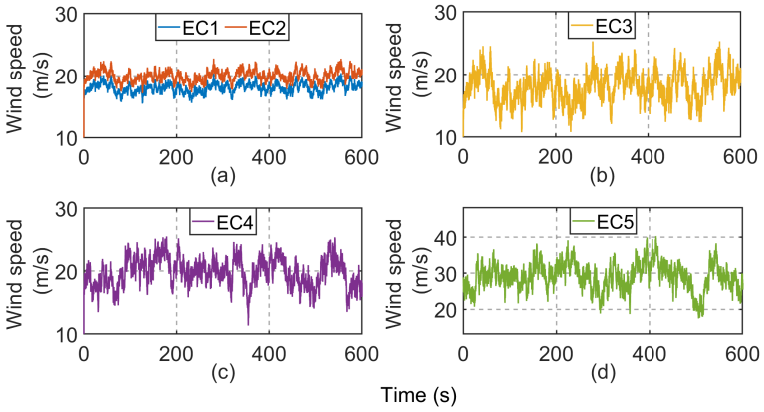

Figure 17: Wind speed profiles for EC1 and EC2 with 5% turbulence intensity (a), wind speed profile for EC3 with 15% turbulence intensity (b), wind speed profile for EC4 with 12% turbulence intensity (c), and wind speed profil for EC5 with 20% intensity (d).

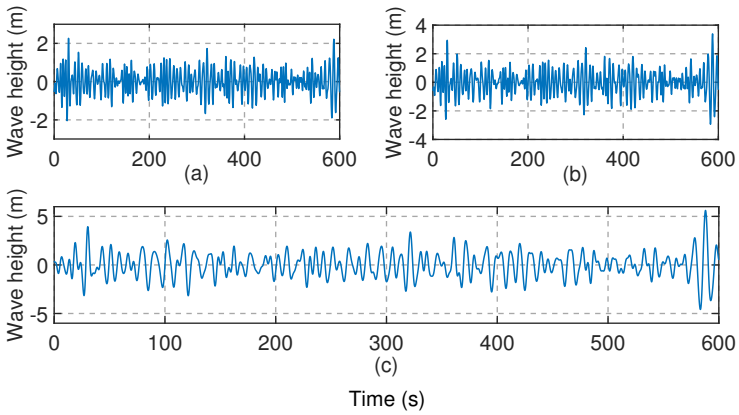

Figure 18: Wave profile for EC1 and EC3 (a), wave profile for EC2 and EC4 (b), and wave profile for EC5 (c).

- Learning rates for adaptive laws: $\eta_1 = 0.1$ and $\eta_2 = 0.5$.

Additionally, in order to maintain consistent output limitations, the same maximum blade pitch angle ($\beta_{\max}$), minimum blade pitch angle ($\beta_{\min}$), and blade pitch rate (β_{rate}) are applied to the tested RBFNN-based STA and the STA alone. This ensures that all controllers operate within the same physical constraints, enabling a fair comparison of their control performance.

Figures 19 to 23 illustrate the controller's performance in regulating rotor speed, mitigating platform pitch rate, and stabilizing the FOWT platform's motion in real-time. The response curves for rotor speed, platform pitch angle, platform pitch rate, and blade pitch angle across EC1 to EC5 are compared between the proposed RBFNN-based STA (RBFNN-STA), the STA alone, and the Baseline GSPI. Additionally, Figure 24 presents the sliding variable and its derivative for the RBFNN-STA, under the turbulent conditions of EC3, providing insight into the controller's dynamic behavior.

To quantitatively evaluate the controllers' performance, mean, standard deviation (STD), and min-max values for rotor speed, platform pitch rate, and blade pitch angle are presented in Figure 25. These metrics provide insight into tracking accuracy and variance under each EC. Additionally, the RMSE for rotor speed and platform pitch rate is detailed in Table 12. Statistical analysis is conducted from 100s to 600s to assess stabilized performance, minimizing the influence of initial transients.

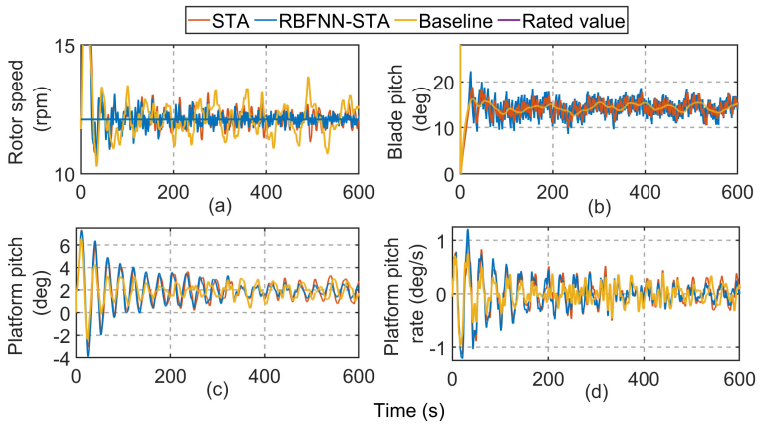

Figure 19: Performance comparison of RBFNN-STA, STA, and GSPI for EC1: (a) Rotor speed, (b) Blade pitch angle, (c) Platform pitch angle, and (d) Platform pitch rate.

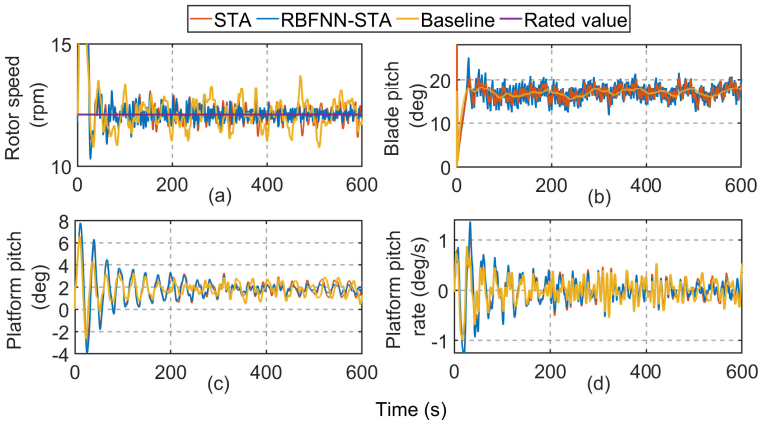

Figure 20: Performance comparison of RBFNN-STA, STA, and GSPI for EC2: (a) Rotor speed, (b) Blade pitch angle, (c) Platform pitch angle, and (d) Platform pitch rate.

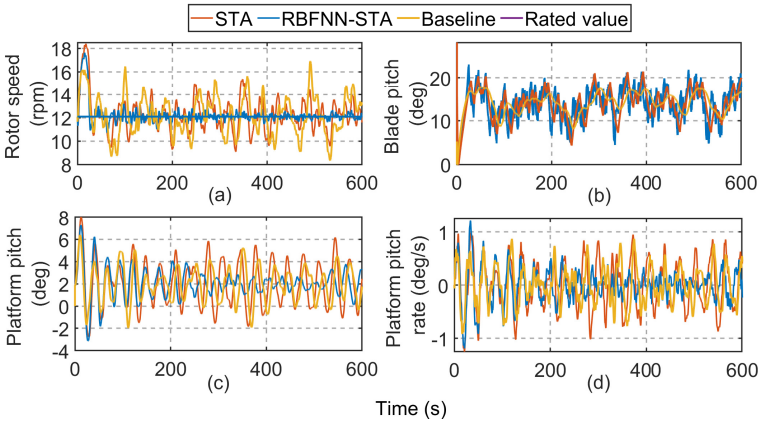

Figure 21: Performance comparison of RBFNN-STA, STA, and GSPI for EC3: (a) Rotor speed, (b) Blade pitch angle, (c) Platform pitch angle, and (d) Platform pitch rate.

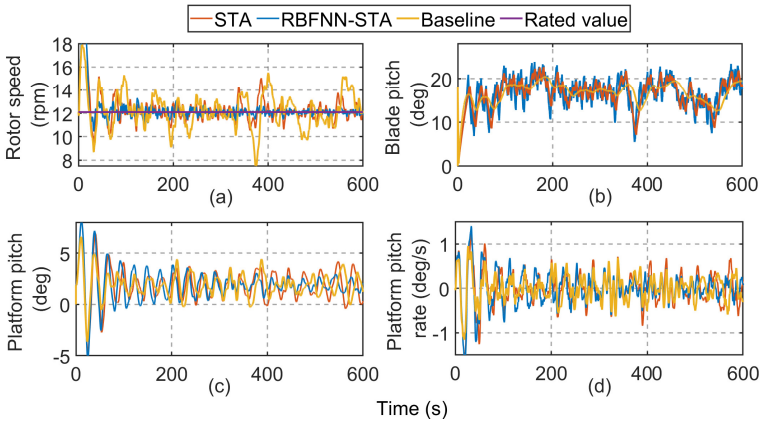

Figure 22: Performance comparison of RBFNN-STA, STA, and GSPI for EC4: (a) Rotor speed, (b) Blade pitch angle, (c) Platform pitch angle, and (d) Platform pitch rate.

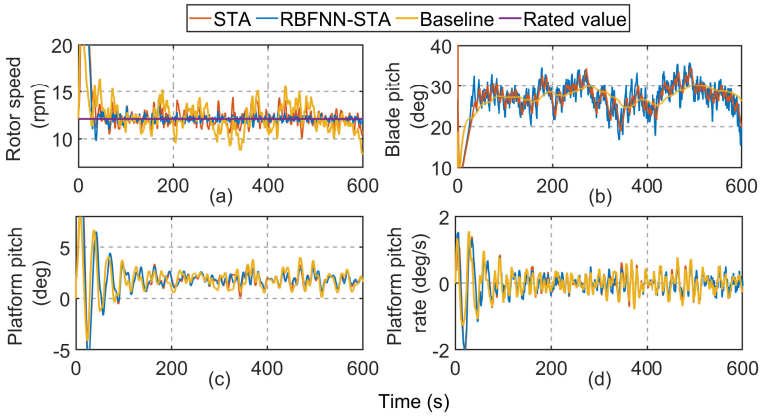

Figure 23: Performance comparison of RBFNN-STA, STA, and GSPI for EC5: (a) Rotor speed, (b) Blade pitch angle, (c) Platform pitch angle, and (d) Platform pitch rate.

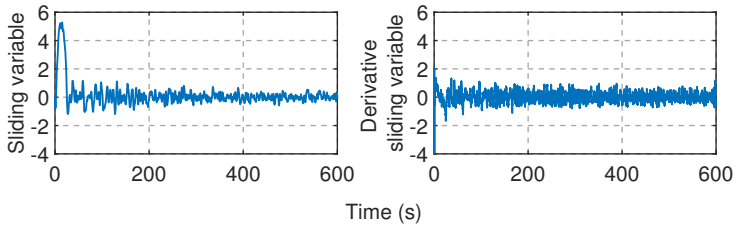

Figure 24: Sliding variable (a) and its derivative (b) for the RBFNN-STA under EC3.

4.4.3. Discussion

The comparative analysis of the proposed RBFNN-based STA controller, the STA alone, and the Baseline GSPI under four ECs provides key insights into their real-time performance on the HIL platform.

Rotor speed regulation: Across all five ECs, the RBFNN-based STA controller demonstrates superior rotor speed tracking performance compared to both the STA and GSPI controllers. The RMSE values for rotor speed (ω_r) consistently show lower errors for the RBFNN-STA, especially under high turbulence (EC3, EC4 and EC5), where

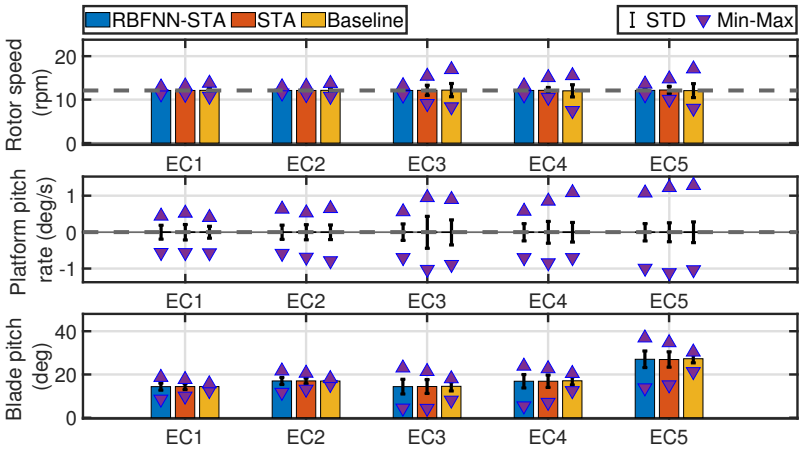

Figure 25: Statistical metrics (mean, STD, min-max) across EC1 to EC5 for: Rotor speed, Platform pitch rate, and Blade pitch angle.

Table 12: RMSE for rotor speed (ω_r) and platform pitch rate (ω_y).

EC	Controller	RMSE ω_r (rpm)	RMSE ω_y (deg/s)
EC1	RBFNN-based STA	0.22169	0.19022
	STA	0.31617	0.21163
	Baseline	0.53238	0.16675
EC2	RBFNN-based STA	0.23352	0.19453
	STA	0.33712	0.20695
	Baseline	0.5218	0.20039
EC3	RBFNN-based STA	0.29165	0.2257
	STA	1.1633	0.4383
	Baseline	1.524	0.34413
EC4	RBFNN-based STA	0.2877	0.23622
	STA	0.66112	0.2991
	Baseline	1.4046	0.26919
EC5	RBFNN-based STA	0.30292	0.24563
	STA	0.84144	0.25994
	Baseline	1.6039	0.28432

the controller maintains a lower deviation from the reference speed.

In EC3, the RBFNN-STA achieves an RMSE of 0.29165 rpm, whereas the STA alone exhibits a significantly higher RMSE of 1.1633 rpm, and the Baseline GSPI reaches 1.524 rpm. A similar trend is observed in EC4, where the RBFNN-STA maintains 0.2877 rpm RMSE, significantly outperforming the STA (0.66112 rpm RMSE) and GSPI (1.4046 rpm RMSE). Notably, under the extreme wind scenario EC5 (30 m/s mean wind, 20% turbulence), the RBFNN-STA maintains robust performance with a rotor speed RMSE of 0.30292 rpm, far lower than STA (0.84144 rpm) and Baseline (1.6039 rpm). These results highlight the controller's adaptability and robustness in challenging operating conditions, with the RBFNN component significantly enhancing disturbance rejection and rotor speed stability.

Platform pitch rate reduction: Platform pitch rate (ω_y) is a key indicator of floating turbine stability, particularly in response to wind and wave disturbances. The results demonstrate that the proposed RBFNN-based STA consistently achieves lower pitch rate fluctuations than the other controllers.

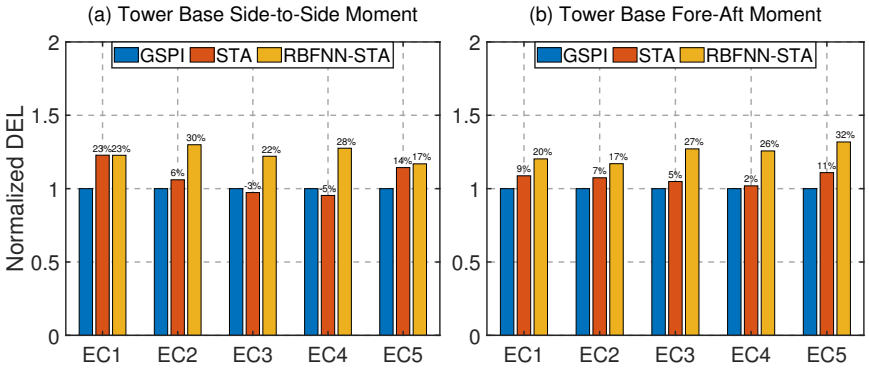

Figure 26: Normalized 1 Hz DELs at the tower base for EC1 to EC5: (a) Side-to-side bending moment and (b) Fore-aft bending moment.

Under moderate conditions (EC1 and EC2), the Baseline GSPI exhibits the lowest RMSE in ω_y. However, under higher turbulence (EC3-EC5), the RBFNN-STA outperforms both the STA and Baseline GSPI, significantly reducing pitch rate oscillations. In EC3, the RBFNN-STA achieves an RMSE of 0.2257 deg/s, while the STA reaches 0.4383 deg/s and the GSPI 0.34413 deg/s. Similarly, in EC4, the RBFNN-STA maintains 0.23622 deg/s RMSE, outperforming both the STA (0.2991 deg/s) and GSPI (0.26919 deg/s). In EC5, despite the extremely harsh wind conditions, the RBFNN-STA continues to slightly outperform the other methods with an RMSE of 0.24563 deg/s, compared to 0.25994 deg/s for the STA and 0.28432 deg/s for the Baseline. This consistency across increasingly difficult scenarios confirms the controller's capacity to mitigate platform pitch rate excursions effectively, contributing to improved overall turbine stability.

Blade pitch angle adaptation: Blade pitch angle (β) adjustments reflect the controller's response to varying wind conditions and its capacity to regulate rotor speed while maintaining stability. As seen in Figures 19 to 23, the Baseline GSPI produces the smoothest β profiles, with the lowest standard deviations, owing to its more conservative control nature.

In contrast, the RBFNN-STA introduces more aggressive and frequent adjustments in β, especially under turbulent and high wind scenarios like EC3 and EC5. These larger fluctuations allow for finer control of rotor speed and platform stability but come at the cost of increased actuator activity, as also seen in the min-max and STD statistics.

Fatigue Load Analysis: Figure 26 presents the normalized 1 Hz Damage Equivalent Loads (DELs) computed at the tower base in both the side-to-side and fore-aft directions for all environmental conditions (EC1-EC5). The DELs are normalized with respect to the Baseline GSPI controller, with percentage values indicating the relative increase in DEL compared to GSPI. As expected, the GSPI controller, designed to ensure smooth pitch activity, generally achieves the lowest DEL values across all conditions, serving as a reference for fatigue performance.

In the side-to-side direction (Figure 26a), the RBFNN-STA controller exhibits increased DELs compared to GSPI, with values ranging from +22% to +30% in EC2–EC5. The STA controller achieves more moderate increases or even slight reductions (e.g., –3% in EC3 and –5% in EC4), highlighting a trade-off between aggressive disturbance rejection and structural fatigue. The increase observed for RBFNN-STA is attributable to the more active pitch control strategy, which, while effective in stabilizing rotor speed and pitch rate, introduces higher-frequency load variations that amplify tower base fatigue. Similarly, in the fore-aft direction (Figure 26b), the RBFNN-STA consistently yields the highest DEL values, particularly under turbulent and high wind conditions such as EC3 and EC5, with increases of +27% and +32%, respectively. In contrast, STA shows smaller increases, ranging from +2% to +11% across scenarios.

This fatigue analysis highlights an important trade-off between performance and structural loading. While the RBFNN-STA demonstrates superior control precision and robustness, especially in challenging conditions (EC3–EC5), its increased actuator activity results in higher DELs that must be considered when assessing long-term reliability

Overall, the proposed RBFNN-based STA controller exhibits significantly improved control performance compared to both the STA and GSPI controllers, particularly in the most demanding environmental conditions. The results obtained in EC3, EC4, and EC5 clearly demonstrate enhanced rotor speed tracking, platform stabilization, and disturbance rejection capabilities.

EC3 and EC5—respectively representing a scenario that spans the full Region III operational range and another characterized by extreme wind and high turbulence—offer stringent tests of controller robustness. Despite the severity of these conditions, the RBFNN-based STA maintained effective rotor speed and platform pitch regulation, consistently achieving the lowest RMSE values among all controllers. These results confirm the scalability and robustness of the proposed controller, even near the turbine's cut-out wind speed. The ability of the RBFNN-STA to sustain performance under such adverse conditions highlights the benefit of integrating adaptive learning (via RBFNN) within a robust nonlinear control framework (STA).

Nonetheless, the increased blade pitch activity observed across all ECs requires further investigation to assess its impact on actuator longevity and long-term operational reliability. Future work should aim to incorporate fatigue-aware control constraints or penalization terms to balance control precision and structural endurance.

Additionally, the validation setup itself serves as an implicit sensitivity analysis of the control design with respect to model accuracy. The proposed controller is developed based on the reformulated analytical Homer COM but is validated in real time using a HIL configuration interfaced with OpenFAST. The inherent model-plant mismatch between the analytical COM and high-fidelity aero-hydro-servo-elastic OpenFAST simulator naturally introduces structural and hydrodynamic discrepancies, along with real-time execution constraints and communication latencies. As such, the HIL results demonstrate the controller's robustness in compensating for deviations between the modeled and actual system dynamics, confirming its reliability for realistic offshore deployment.

These findings validate not only the HIL platform's capability to handle complex, adaptive control algorithms requiring real-time learning but also the superiority of the proposed RBFNN-STA controller over both the STA alone and the Baseline GSPI controller.

5. Conclusion

This study presented the real-time validation of an Radial Basis Function Neural Network-based Super-Twisting Algorithm (RBFNN-based STA) controller for Floating Offshore Wind Turbines (FOWTs) through Hardware-in-the-Loop (HIL) testing. The proposed control strategy was designed to enhance rotor speed regulation and platform stability in the challenging operating conditions of Region III.

The nonlinear control law for the RBFNN-based STA controller was derived from the Homer Control-Oriented Model (COM), representing a 5MW semi-submersible FOWT. To facilitate its use in control law derivation, the Homer COM was reformulated into a fully analytical model, eliminating dependencies on external lookup tables. The proposed controller integrates an RBFNN, acting as a real-time disturbance observer, to compensate for unknown dynamics within the reformulated model, while the STA ensures robustness and guarantees convergence of the sliding variable through a Lyapunov-based stability analysis.

The HIL platform was designed to accurately emulate real-time FOWT dynamics, integrating OpenFAST with LabVIEW on a CompactRIO embedded system for real-time simulation, while a remote Beckhoff unit executed the control algorithms. A UDP-based communication framework facilitated real-time data exchange between the controller and emulator, ensuring accurate replication of FOWT behavior. To establish the reliability of the HIL platform, an initial baseline validation was conducted by comparing the GSPI controller running on the HIL setup against its Model-in-the-Loop (MIL) implementation. During the first 200 seconds, the dynamic responses from both MIL and HIL platforms showed a high degree of similarity across key variables. After this point, noticeable deviations emerged, particularly in time alignment, which are attributed to real-time execution delays and communication latencies inherent to the HIL setup. These discrepancies were quantitatively assessed using RMSE metrics, which, while non-negligible, remained within operationally acceptable thresholds for controller validation. Rather than being ignored, these differences highlight the inherent challenges of maintaining perfect synchronization in real-time systems and underscore the importance of considering such delays when designing and deploying FOWT controllers. Overall, despite the divergence after 200s, the HIL platform remains a reliable tool for evaluating real-time control strategies under realistic conditions.

The performance of the proposed RBFNN-based STA controller was then evaluated against the STA alone and the Baseline GSPI controller under four distinct environmental conditions (ECs), characterized by varying wind turbulence and wave profiles. The experimental results demonstrated that the RBFNN-based STA consistently outperformed both the STA and GSPI, achieving superior rotor speed tracking and platform pitch rate mitigation, particularly under high turbulence scenarios. The integration of the RBFNN as a real-time disturbance observer significantly improved disturbance rejection, leading to lower RMSE values across all key performance variables. However, it was observed that the RBFNN-based STA induced more aggressive blade pitch adjustments, resulting in higher pitch angle fluctuations compared to the Baseline GSPI. While this behavior contributes to better adaptability under varying wind conditions, it may also increase actuator wear over time. This trade-off suggests that further optimization of the control strategy could focus on balancing adaptability and actuator longevity.

Overall, the results confirm that the proposed RBFNN-based STA controller provides a more effective control strategy for FOWTs operating in Region III, enhancing system stability and performance under realistic offshore conditions. The HIL validation underscores the feasibility of real-time implementation of advanced control algorithms and offers a robust testbed for assessing control sensitivity to model-plant mismatch. Notably, while the controller is synthesized using a simplified analytical model, it is deployed in a closed-loop with OpenFAST, whose high-fidelity modeling introduces realistic structural and hydrodynamic deviations. This inherently tests the robustness of the control strategy against modeling inaccuracies, supporting the controller's practical applicability. Future research could focus on extending this evaluation through systematic perturbation-based sensitivity analyses and Power-Hardware-in-the-Loop (PHIL) testing to further bridge the gap toward field deployment.

Beyond the present validation of the RBFNN-based STA controller, future work will focus on benchmarking against modern data-driven control approaches, particularly Deep Reinforcement Learning (DRL) methods. These strategies offer model-free control capabilities and have demonstrated significant potential in managing complex, nonlinear systems such as FOWTs under turbulent environmental conditions. Integrating DRL agents with robust control frameworks may further enhance adaptability and control precision in Region III, providing a valuable direction for advancing FOWT control strategies.

Acknowledgments

This work was supported by the ANR Project (CREATIF, ANR-20-CE05-0039), the EIPHI Graduate School (contract ANR-17-EURE-0002) and the Region Bourgogne Franche-Comté.

References

[1] International Energy Agency (IEA), *Electricity Market Report Update: Outlook for 2023 and 2024*, Tech. Rep., IEA, Paris, France, 2024.

[2] J. M. Jonkman, *Dynamics Modeling and Loads Analysis of an Offshore Floating Wind Turbine*, Technical Report NREL/TP-500-41958, National Renewable Energy Laboratory (NREL), November 2007.

[3] Skaare, B. and Hanson, T. D. and Nielsen, F. G., *Importance of Control Strategies on Fatigue Life of Floating Wind Turbines*, Volume 5: Ocean Space Utilization, Polar and Arctic Sciences and Technology, The Robert Dean Symposium on Coastal and Ocean Engineering, Special Symposium on Offshore Renewable Energy, International Conference on Offshore Mechanics and Arctic Engineering, pp. 493–500, 2007. DOI: 10.1115/OMAE2007-29277.

[4] H. Namik and K. Stol, *Individual blade pitch control of floating offshore wind turbines*, Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology, vol. 13, no. 1, pp. 74–85, 2010. Wiley Online Library.

[5] H. Namik and K. Stol, *Performance analysis of individual blade pitch control of offshore wind turbines on two floating platforms*, Mechatronics, vol. 21, no. 4, pp. 691–703, 2011. Elsevier.

[6] H. Namik and K. Stol, *Individual blade pitch control of a spar-buoy floating wind turbine*, IEEE Transactions on Control Systems Technology, vol. 22, no. 1, pp. 214–223, 2013. IEEE.

[7] S. Christiansen, T. Knudsen, and T. Bak, *Optimal control of a ballast-stabilized floating wind turbine*, in Proceedings of the 2011 IEEE International Symposium on Computer-Aided Control System Design (CACSD), pp. 1214–1219, 2011. IEEE.

[8] S. Christiansen, T. Knudsen, and T. Bak, *Extended onshore control of a floating wind turbine with wave disturbance reduction*, Journal of Physics: Conference Series, vol. 555, no. 1, pp. 012018, 2014. IOP Publishing.

[9] F. Lemmer, D. Schlipf, and P.W. Cheng, *Control design methods for floating wind turbines for optimal disturbance rejection*, Journal of Physics: Conference Series, vol. 753, no. 9, pp. 092006, 2016. IOP Publishing.

[10] O. Bagherieh and R. Nagamune, *Gain-scheduling control of a floating offshore wind turbine above rated wind speed*, Control Theory and Technology, vol. 13, no. 2, pp. 160–172, 2015. Springer.

[11] P. Zhao and R. Nagamune, *Switching LPV control of a floating offshore wind turbine on a semi-submersible platform*, in Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), pp. 664–669, 2019. IEEE.

[12] T. Bakka and H.R. Karimi, *Robust output feedback H-infinity control synthesis with pole placement for offshore wind turbine system: An LMI approach*, in Proceedings of the 2012 IEEE International Conference on Control Applications, pp. 1467–1472, 2012. IEEE.

[13] X. Li and H. Gao, *Load mitigation for a floating wind turbine via generalized H_∞ structural control*, IEEE Transactions on Industrial Electronics, vol. 63, no. 1, pp. 332–342, 2015. IEEE.

[14] C.J. Cortes Sanchez, *Wind and wave disturbance rejection control of floating offshore wind turbines*, Master's Thesis, University of British Columbia, 2018.

[15] T. Bakka, H.R. Karimi, and N.A. Duffie, *Gain Scheduling for Output H8 Control of Offshore Wind Turbine*, in Proceedings of the ISOPE International Ocean and Polar Engineering Conference, pp. ISOPE–I, 2012. ISOPE.

[16] N. Hara, Y. Nihei, K. Iijima, and K. Konishi, *Blade pitch control for floating wind turbines: Design and experiments using a scale model*, in Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA), pp. 481–486, 2017. IEEE.

[17] M.S. Mahmoud and M.O. Oyedeji, *Adaptive and predictive control strategies for wind turbine systems: A survey*, IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 2, pp. 364–378, 2019, doi: 10.1109/JAS.2019.1911375.

[18] F. Lemmer, S. Raach, D. Schlipf, and P.W. Cheng, *Prospects of linear model predictive control on a 10 MW floating wind turbine*, in International Conference on Offshore Mechanics and Arctic Engineering, vol. 56574, pp. V009T09A071, 2015, American Society of Mechanical Engineers.

[19] A. Cunha, E. Caetano, P. Ribeiro, and G. Müller, *Reducing blade fatigue and damping platform motions of floating wind turbines using model predictive control*, in International Conference on Structural Dynamics, 2014.

[20] Y. Okada, K. Haneda, T. Chujo, and T. Ohtsuka, *Parameter-varying Modeling and Nonlinear Model Predictive Control for Floating Offshore Wind Turbines*, IFAC-PapersOnLine, vol. 52, no. 16, pp. 382–387, 2019, doi: https://doi.org/10.1016/j.ifacol.2019.11.810.

[21] T. Wakui, A. Nagamura, and R. Yokoyama, *Stabilization of power output and platform motion of a floating offshore wind turbine-generator system using model predictive control based on previewed disturbances*, Renewable Energy, vol. 173, pp. 105–127, 2021, doi: https://doi.org/10.1016/j.renene.2021.03.112.

[22] J. Chen, Z. Hu, G. Liu, and D. Wan, *Coupled aero-hydro-servo-elastic methods for floating wind turbines*, Renewable Energy, vol. 130, pp. 13–27, 2019, doi: https://doi.org/10.1016/j.renene.2018.06.060.

[23] C. Gao, X. Zhang, Y. Li, and W. Shi, *Coupled aero-servo-elastic method for floating offshore wind turbine wake analysis*, Ocean Engineering, vol. 307, p. 118108, Sep. 2024. doi: 10.1016/j.oceaneng.2024.118108.

[24] V. I. Utkin, *Sliding Modes in Optimization and Control Problems*, Springer-Verlag, New York, 1992.

[25] H. Basbas, Y.-C. Liu, S. Laghrouche, M. Hilairet, and F. Plestan, *Review on Floating Offshore Wind Turbine Models for Nonlinear Control Design*, Energies, vol. 15, no. 15, article 5477, 2022. DOI: https://doi.org/10.3390/en15155477.

[26] G. Betti, M. Farina, G. A. Guagliardi, A. Marzorati, and R. Scattolini, *Development of a control-oriented model of floating wind turbines*, IEEE Transactions on Control Systems Technology, vol. 22, no. 1, pp. 69–82, 2013.

[27] F. Lemmer, *Low-Order Modeling, Controller Design and Optimization of Floating Offshore Wind Turbines*, Doctoral Thesis, University of Stuttgart, 2018.

[28] J. R. Homer and R. Nagamune, *Physics-based 3-D control-oriented modeling of floating wind turbines*, IEEE Transactions on Control Systems Technology, vol. 26, no. 1, pp. 14–26, 2017.

[29] A. Robertson, *Definition of the semisubmersible floating system for phase II of OC4*, NREL Technical Report NREL/TP-5000-60601, 2014.

[30] H. Basbas, H. Obeid, S. Laghrouche, M. Hilairet, and F. Plestan, *Model-Based Super-Twisting Controller for a Tensioned-Leg-Platform Floating Offshore Wind Turbine*, in Proc. IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society, pp. 1–6, 2022.

[31] H. Basbas, H. Obeid, S. Laghrouche, M. Hilairet, and F. Plestan, *Barrier Function Based-Adaptive Super-Twisting Algorithm for Floating Offshore Wind Turbine*, in Proc. 2022 16th International Workshop on Variable Structure Systems (VSS), pp. 166–171, 2022.

[32] H. Basbas, H. Obeid, S. Laghrouche, M. Hilairet, and F. Plestan, *Comparative study of three high order sliding mode model based design for a floating wind turbine robust control*, in Proc. 2022 IEEE 61st Conference on Decision and Control (CDC), pp. 4294–4299, 2022.

[33] C. Zhang, E. Tahoumi, S. Gutierrez, F. Plestan, and J. De León-Morales, *Adaptive robust control of floating offshore wind turbine based on sliding mode*, in Proc. 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 6936–6941, 2019. DOI: 10.1109/CDC40024.2019.9029231.

[34] C. Zhang, S. V. Gutierrez, F. Plestan, and J. De León-Morales, *Adaptive super-twisting control of floating wind turbines with collective blade pitch control*, IFAC-PapersOnLine, vol. 52, no. 4, pp. 117–122, 2019. IFAC Workshop on Control of Smart Grid and Renewable Energy Systems (CSGRES 2019). DOI: 10.1016/j.ifacol.2019.08.165.

[35] Y. Shtessel, M. Taleb, and F. Plestan, *A novel adaptive-gain supertwisting sliding mode controller: Methodology and application*, Automatica, vol. 48, no. 5, pp. 759–769, 2012. DOI: 10.1016/j.automatica.2012.02.024.

[36] C. Zhang and F. Plestan, *Adaptive sliding mode control of floating offshore wind turbine equipped by permanent magnet synchronous generator*, Wind Energy, vol. 24, no. 7, pp. 754–769, 2021. DOI: 10.1002/we.2601.

[37] M. Taleb, A. Marie, C. Zhang, M. A. Hamida, P. E. Testelin, and F. Plestan, *Adaptive nonlinear control of floating wind turbines: new adaptation law and comparison*, in Proc. IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society, pp. 1–6, 2021. DOI: 10.1109/IECON48115.2021.9589421.

[38] Hou, Z.-S., and Wang, Z., *From model-based control to data-driven control: Survey, classification and perspective*, Information Sciences, vol. 235, 2013, pp. 3–35.

[39] Kane, Michael B., *Machine Learning Control for Floating Offshore Wind Turbine Individual Blade Pitch Control*, 2020 American Control Conference (ACC), pp. 237-241, 2020. doi: 10.23919/ACC45564.2020.9147912.

[40] Roh, Chan, *Deep-Learning-Based Pitch Controller for Floating Offshore Wind Turbine Systems with Compensation for Delay of Hydraulic Actuators*, Energies, vol. 15, no. 9, article 3136, 2022. doi: 10.3390/en15093136.

[41] N. Kayedpour, A. E. Samani, J. D. M. De Kooning, L. Vandevelde, and G. Crevecoeur, *Model Predictive Control With a Cascaded Hammerstein Neural Network of a Wind Turbine Providing Frequency Containment Reserve*, IEEE Transactions on Energy Conversion, vol. 37, no. 1, pp. 198–209, 2022, doi: 10.1109/TEC.2021.3093010.

[42] J. Chen, Z. Hu, G. Liu, and D. Wan, *Data-driven predictive control for floating offshore wind turbines based on deep learning and multi-objective optimization*, Ocean Engineering, vol. 266, Art. no. 112820, 2022, doi: 10.1016/j.oceaneng.2022.112820.

[43] P. Chen, L. Song, J.-h. Chen, and Z. Hu, *Simulation annealing diagnosis algorithm method for optimized forecast of the dynamic response of floating offshore wind turbines*, Journal of Hydrodynamics, vol. 33, no. 2, pp. 216–225, Apr. 2021, doi: 10.1007/s42241-021-0033-9.

[44] I. Ahmad, F. M'zoughi, P. Aboutalebi, I. Garrido, and A. J. Garrido, *A regressive machine-learning approach to the non-linear complex FAST model for hybrid floating offshore wind turbines with integrated oscillating water columns*, Scientific Reports, vol. 13, 2023, doi: 10.1038/s41598-023-28703-z.

[45] I. Ahmad, F. M'zoughi, P. Aboutalebi, I. Garrido, and A. J. Garrido, *Fuzzy logic control of an artificial neural network-based floating offshore wind turbine model integrated with four oscillating water columns*, Ocean Engineering, vol. 269, Art. no. 113578, 2023, doi: 10.1016/j.oceaneng.2022.113578.

[46] Chen, Peng, Chen, Jiahao, and Hu, Zhiqiang, *Software-in-the-Loop Combined Reinforcement Learning Method for Dynamic Response Analysis of FOWTs*, Frontiers in Marine Science, vol. 7, 2021. doi: 10.3389/fmars.2020.628225.

[47] Chen, Peng and Hu, Zhi Qiang, *A Study on Key Disciplinary Parameters of Artificial Intelligent-Based Analysis Method for Dynamic Response Prediction of Floating Offshore Wind Turbines*, Journal of Offshore Mechanics and Arctic Engineering, vol. 145, no. 1, article 010906, 2022. doi: 10.1115/1.4055993.

[48] Chen, P., Song, L., Chen, J.-h., and Hu, Z., *Simulation Annealing Diagnosis Algorithm Method for Optimized Forecast of the Dynamic Response of Floating Offshore Wind Turbines*, Journal of Hydrodynamics, vol. 33, no. 2, pp. 216-225, 2021. doi: 10.1007/s42241-021-0033-9.

[49] Xie, Jingjie, Dong, Hongyang, and Zhao, Xiaowei, *Power Regulation and Load Mitigation of Floating Wind Turbines via Reinforcement Learning*, IEEE Transactions on Automation Science and Engineering, vol. 21, no. 3, pp. 4328-4339, 2024. doi: 10.1109/TASE.2023.3295576.

[50] H. Obeid, S. Laghrouche, and L. Fridman, *Dual layer barrier functions based adaptive higher order sliding mode control*, International Journal of Robust and Nonlinear Control, vol. 31, no. 9, pp. 3795–3808, 2021.

[51] F. Plestan, Y. Shtessel, V. Bregeault, and A. Poznyak, *New methodologies for adaptive sliding mode control*, International Journal of Control, vol. 83, pp. 1907–1919, 2010.

[52] Y. Shtessel, M. Taleb, and F. Plestan, *A novel adaptive-gain super-twisting sliding mode controller: Methodology and application*, Automatica, vol. 48, no. 5, pp. 759–769, 2012. DOI: 10.1016/j.automatica.2012.02.024.

[53] S. Laghrouche, M. Harmouche, Y. Chitour, H. Obeid, and L. M. Fridman, *Barrier function-based adaptive higher order sliding mode controllers*, Automatica, vol. 123, p. 109355, 2021. DOI: 10.1016/j.automatica.2021.109355.

[54] T. N. Truong, A. T. Vo, and H. J. Kang, *Neural network-based sliding mode controllers applied to robot manipulators: A review*, Neurocomputing, vol. 562, p. 126896, 2023. DOI: 10.1016/j.neucom.2023.126896.

[55] S. Zhang, P. Yang, L. Kong, W. Chen, Q. Fu, and K. Peng, *Neural networks-based fault tolerant control of a robot via fast terminal sliding mode*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 7, pp. 4091–4101, 2019. DOI: 10.1109/TSMC.2019.2955695.

[56] X. Zhao, Z. Liu, B. Jiang, and C. Gao, *Switched Controller Design for Robotic Manipulator via Neural Network-based Sliding Mode Approach*, IEEE Transactions on Circuits and Systems II, vol. 70, no. 2, pp. 561–565, 2023. DOI: 10.1109/TCSII.2022.3192026.

[57] Y. Wu, H. Fang, T. Xu, and F. Wan, *Adaptive Neural Fixed-time Sliding Mode Control of Uncertain Robotic Manipulators with Input Saturation and Prescribed Constraints*, Neural Processing Letters, vol. 54, no. 5, pp. 3829–3849, 2022. DOI: 10.1007/s11063-022-10786-6.

[58] M. Rahmani and M. H. Rahman, *Adaptive neural network fast fractional sliding mode control of a 7-DOF exoskeleton robot*, International Journal of Control Automation Systems, vol. 18, no. 1, pp. 124–133, 2020. DOI: 10.1007/s12555-019-0003-3.

[59] M. Hall, *AeroDyn v15 User's Guide and Theory Manual*, NREL Report, 2015. Available: https://www.nrel.gov/wind/nwtc/assets/pdfs/aerodyn-manual.pdf.

[60] National Renewable Energy Laboratory, *OpenFAST Documentation, Release 3.4.0*, 2023, pp. 6–8, https://openfast.readthedocs.io/en/main/.

[61] M. A. Lackner, *Controlling Platform Motions and Reducing Blade Loads for Floating Wind Turbines*, Wind Engineering, vol. 33, no. 6, pp. 541–553, 2009. DOI: 10.1260/0309-524X.33.6.541.

[62] X. Chen, W. Shen, M. Dai, Z. Cao, J. Jin, and A. Kapoor, *Robust adaptive sliding-mode observer using RBF neural network for lithium-ion battery state of charge estimation in electric vehicles*, IEEE Transactions on Vehicular Technology, vol. 65, no. 4, pp. 1936–1947, 2015.

[63] Y. Li, W. Sun, and D. Yu, *RBFNN-based global fast terminal sliding mode control for fully controlled doubly fed induction generator*, Journal of the Franklin Institute, vol. 361, no. 17, p. 107196, 2024. DOI: 10.1016/j.jfranklin.2024.107196.

[64] S.-J. Huang, K.-S. Huang, and K.-C. Chiou, *Development and application of a novel radial basis function sliding mode controller*, Mechatronics, vol. 13, no. 4, pp. 313–329, 2003.

[65] S. Mahjoub, F. Mnif, N. Derbel, and M. Hamerlain, *Radial-basis-functions neural network sliding mode control for underactuated mechanical systems*, International Journal of Dynamics and Control, vol. 2, pp. 533–541, 2014.

[66] J. Fei and H. Ding, *Adaptive sliding mode control of dynamic system using RBF neural network*, Nonlinear Dynamics, vol. 70, no. 2, pp. 1563–1573, 2012. DOI: 10.1007/s11071-012-0556-2.

[67] L. Wang, T. Chai, and L. Zhai, *Neural-network-based terminal sliding-mode control of robotic manipulators including actuator dynamics*, IEEE Transactions on Industrial Electronics, vol. 56, no. 9, pp. 3296–3304, 2009. DOI: 10.1109/TIE.2008.2011350.

[68] Z. Feng and J. Fei, *Super-twisting sliding mode control for micro gyroscope based on RBF neural network*, IEEE Access, vol. 6, pp. 64993–65001, 2018.

[69] Y.-C. Liu, S. Laghrouche, D. Depernet, A. N'Diaye, A. Djerdir, and M. Cirrincione, *Super-twisting sliding-mode observer-based model reference adaptive speed control for PMSM drives*, Journal of the Franklin Institute, vol. 360, no. 2, pp. 985–1004, 2023. DOI: 10.1016/j.jfranklin.2022.12.014.

[70] W. Tang and Y. Cai, *A high order sliding mode control scheme based on adaptive radial basis function neural network*, in Proc. 2011 50th IEEE Conference on Decision and Control and European Control Conference, pp. 6343–6348, 2011.

[71] Y.-C. Liu, S. Laghrouche, A. N'Diaye, and M. Cirrincione, *Hermite neural network-based second-order sliding-mode control of synchronous reluctance motor drive systems*, Journal of the Franklin Institute, vol. 358, no. 1, pp. 400–427, 2021. DOI: 10.1016/j.jfranklin.2020.10.029.

[72] 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 (NREL), Technical Report NREL/TP-500-38060, Golden, Colorado, 2009. Available: https://www.nrel.gov/docs/fy09osti/38060.pdf.

[73] N. Kelley and B. Jonkman, *TurbSim User's Guide: Version 1.5*, NREL Report, 2009.

[74] J. C. Kaimal, J. C. Wyngaard, Y. Izumi, and O. R. Coté, *Spectral characteristics of surface-layer turbulence*, Quarterly Journal of the Royal Meteorological Society, vol. 98, no. 417, pp. 563–589, 1972. DOI: 10.1002/qj.49709841707.

[75] W. J. Pierson Jr. and L. Moskowitz, *A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii*, Journal of Geophysical Research (1896-1977), vol. 69, no. 24, pp. 5181–5190, 1964. DOI: 10.1029/JZ069i024p05181.

[76] M. J. Mirzaei, M. A. Hamida, F. Plestan, *Neural network-based supertwisting control for floating wind turbine in region III*, IFAC-PapersOnLine, vol. 56, no. 2, pp.336–341, 2023. DOI: 10.1016/j.ifacol.2023.10.1590.

[77] Y-C. Liu, H. Basbas, S. Laghrouche, *Robust blade pitch control of semi-submersible floating offshore wind turbines based on the modified super-twisting sliding-mode algorithm*, Journal of the Franklin Institute, vol. 361, no. 18, pp. 107279, 2024. DOI: 10.1016/j.jfranklin.2024.107279.

[78] Leroy, V., Delacroix, S., Merrien, A., Bachynski-Polić, E.E., and Gilloteaux, J.-C., *Experimental investigation of the hydro-elastic response of a spar-type floating offshore wind turbine*, Ocean Engineering, vol. 255, 2022, pp. 111430. DOI: https://doi.org/10.1016/j.oceaneng.2022.111430.

[79] Bonnefoy, F., Leroy, V., Mojallizadeh, M.R., Delacroix, S., Arnal, V., and Gilloteaux, J.-C., *Multi-dimensional hybrid software-in-the-loop modeling approach for experimental analysis of a floating offshore wind turbine in wave tank experiments*, Ocean Engineering, vol. 309, 2024, pp. 118390. DOI: https://doi.org/10.1016/j.oceaneng.2024.118390.

[80] Jassmann, U., Monti, A., and Abel, D., *Hardware-in-the-loop wind turbine system test benches and their usage for controller validation*, . No. RWTH-2018-230830. Lehrstuhl und Institut für Regelungstechnik, 2018.