Blade Pitch Control of Floating Offshore Wind Turbine Systems Using Super-Twisting Algorithm and Recurrent RBF Neural Network

Flavie Didier[1], Yong-Chao Liu[1], Salah Laghrouche[1], Daniel Depernet[1]
[1]*Energy Department, FEMTO-ST Institutes (UMR 6174)*
Université Franche-Comté, UTBM, CNRS
Belfort, France
flavie.didier@utbm.fr, cn.yong-chao.liu@ieee.org, salah.laghrouche@utbm.fr, daniel.depernet@utbm.fr

***Abstract*—This paper presents an approach to enhance the performance of floating offshore wind turbines mounted on semi-submersible platforms through the integration of a Super-Twisting Sliding Mode Collective Blade Pitch Controller (STSM-CBPC) with a Recurrent Radial Basis Function Neural Network (RRBFNN). The proposed CBPC, developed based on a refined nonlinear control-oriented model, leverages the RRBFNN as an adaptive estimator to address lumped uncertainties and external disturbances, when operating above the rated wind speed. The recurrent neural network features a dual feedback loop structure. The internal feedback loop, operating on the hidden layer, and the external feedback loop, enabling the transmission of the output signal back into the input signal, collectively contribute to a comprehensive capture of the system's state information. To ensure convergence, adaptive laws governing the neural network are derived through Lyapunov analysis, ensuring real-time updates to the RRBFNN parameters. Simulation results demonstrate the superior performance of the proposed CBPC over the baseline gain scheduling proportional integral controller for regulating rotor speed and mitigating platform motion.**

***Index Terms*—Super-twisting sliding mode control, recurrent neural network, floating offshore wind turbine**

I. Introduction

In the face of intensifying concerns surrounding global warming, the imperative for meeting climate targets, and the ever-growing demand for sustainable energy solutions, the field of renewable technologies is witnessing rapid evolution. Among these advancements, Floating Offshore Wind Turbines (FOWTs) stand out as a pioneering innovation within the realm of wind energy. Leveraging the distinctive advantage of a floating platform, FOWTs enable the strategic placement of turbines at greater distances from coastlines, facilitating extensive deployment with minimal visual impact. While this strategic positioning holds the promise of enhanced power generation, it also introduces a spectrum of control challenges. The operational range of wind turbines, segmented into regions based on incident wind speed, necessitates tailored control strategies. Region III, characterized by wind speeds surpassing the rated value, becomes a focal point where the primary objective is to maintain generated power at its rated value. However, the inherent nonlinear dynamics of FOWT systems, compounded by hydrodynamic and mooring dynamics due to the floating platform, add a layer of complexity into effective control mechanisms. Notably, the manifestation of the negative damping phenomenon in Region III poses the risk of instability in the platform's pitching motion, demanding precision in power regulation coupled with the minimization of platform motion.

While conventional control methodologies, originally designed for fixed wind turbines, have been adapted for FOWT systems using controllers such as Gain Scheduling Proportional Integral (GSPI) [1], their limitations in terms of sensitivity to external disturbances and limited robustness against unmodeled dynamics promote the exploration of more sophisticated alternatives. Various linear control strategies, employing model linearization around specific operating points, have been proposed. However, their efficacy is degraded when the turbine operates away from the defined operating point. Addressing these challenges, a more promising alternative involves the adoption of nonlinear controllers. The Super-Twisting algorithm-based Sliding Mode (STSM) controller emerges as a compelling choice, demonstrating significant performance while mitigating the chattering issues associated with standard Sliding Mode Control (SMC). In tackling Region III control challenges in FOWT systems, researchers have developed a first-order SMC based on a reduced model [2]. Additionally, an STSM controller has been successfully implemented [3], [4], leveraging a linear model generated by OpenFAST [5]. Embracing a continuous approach with adaptive methods, an adaptive version of the STSM controller has demonstrated superior performance in rotor speed control, power regulation, and the platform pitch motion reduction compared to traditional GSPI approaches [6].

Although the STSM controller demonstrates significant efficacy, challenges arising from the bounded sign function necessitate innovative solutions. Slower convergence of the sliding variable when distant from the origin necessitates an

increase in gains, exacerbating the occurrence of chattering effects. To address this limitation, introducing an artificial neural network as an estimator becomes a pivotal strategy to enhance the STSM controller in managing uncertainties and disturbances inherent in the model. Leveraging on the capability of neural networks to approximate a diverse range of nonlinear functions, their integration with the STSM controller strengthens the system's robustness. The resultant composite controller design achieves control objectives and disturbance attenuation through the feedback regulation mechanism of the controller and the neural network estimator, respectively. Recurrent Neural Networks (RNNs) are particularly investigated for their capacity to approximate dynamic mappings. Thus, RNNs have attracted considerable attention as a popular and effective approximator in the control of dynamic systems. Additionally, the Radial Basis Function Neural Network (RBFNN) stands out as a well-explored class known for its straightforward. Therefore, the synergistic combination of both structures, RNN and RBFNN, enhances the overall approximation ability. Prior research has explored the integration of neural networks with Sliding Mode Control (SMC) for nonlinear systems. For instance, in [7], a robust adaptive SMC strategy based on RBFNN is presented for time-varying systems. In another instance [8], an SMC scheme based on an RBFNN is proposed for robotic manipulators, where the RBF network approximates the nonlinear dynamics of the robot. Moreover, a neural network-based SMC employing dual feedback loop recurrent structure has been investigated for MEMS gyroscopes in [9].

In this paper, a Recurrent RBFNN (RRBFNN), harnessing the dynamic approximation capabilities of RNNs and the structured simplicity of RBFNNs, is proposed as an indispensable complement to the STSM controller. This synergistic integration aims to enhance robustness, resulting in a composite controller design capable of achieving control objectives and mitigating disturbances.

The paper is structured as follows: In Section II, a refined nonlinear Control-Oriented Model (COM) is presented, specifying the essential controlled dynamics. Section III introduces the structure of the RRBFNN estimators. The design of the RRBFNN-based STSM-CBPC (RRBFNN-STSM-CBPC), detailing the formulation of adaptive laws for the RRBFNN is given in Section IV. Section V validates the proposed CBPC through a comparative analysis against the standard STSM-CBPC and the Baseline controller. Finally, Section VI provides a conclusive overview of the paper.

II. Floating Offshore Wind Turbine Model

In this study, the NREL OC4-DeepCwind 5 MW semi-submersible Floating Offshore Wind Turbine (FOWT) [10] is considered. The control laws are derived based on the Homer nonlinear COM [11], which serves as a foundational framework for designing nonlinear controllers [12]. This section offers a condensed overview of the chosen model, presenting dynamic reformulations essential for crafting effective nonlinear controllers.

A. Refined Control-Oriented Model

In this study, the nacelle yaw motion is neglected, and the semi-submersible FOWT is treated as a single rigid body. The state vector encompasses position and orientation states, denoted as x_p and θ_p respectively, including their derivatives. Additional states, namely rotor azimuth angle θ_r and rotor speed ω_r, are incorporated, leading to the following state vector x:

$$x = [x_p, \theta_p, \theta_r, \dot{x_p}, \dot{\theta_p}, \omega_r]^\top \tag{1}$$
$$with \quad x_p = [x, y, z] \quad and \quad \theta_p = [\theta_x, \theta_y, \theta_z].$$

The control input vector u for operating in Region III is composed of the blade pitch angle β, as the generator torque T_g is fixed at its rated value here.

$$u = \beta \tag{2}$$

The nonlinear COM underwent simplifications through the adoption of a one-mass rigid shaft model to capture the drive-train dynamics $\dot{\omega}_r$. The resulting simplified equation is expressed as follows:

$$\dot{\omega}_r = \frac{1}{J_l}(\frac{P}{\omega_r} - n_g T_g) \tag{3}$$

where n_g represents the gearbox ratio, and J_l denotes the low-speed shaft equivalent inertia. The aerodynamic power P is defined as:

$$P = \frac{1}{2}\rho_a \pi R_r^2 C_p(\lambda, \beta) \left\|v_n\right\|_2^3 \tag{4}$$

where ρ_a is the air density, R_r is the effective rotor radius, $\mathbf{v}_n$ is the equivalent velocity vector normal to the face of the rotor, C_p is the power coefficient–a highly nonlinear function of the tip speed ratio $\lambda = \frac{R_r \omega_r}{\|\mathbf{v}_n\|_2}$ and β. $\|\cdot\|_2$ denotes the Euclidean norm of a vector.

Additionally, a sole thrust force denotes as F is employed to encapsulate all forces arising from the interaction between the wind and the FOWT. This force is applied at the central thrust point within the hub.

$$F = \frac{1}{2}\rho_a \pi R_r^2 C_t(\lambda, \beta) \left\|v_n\right\|_2 v_n \tag{5}$$

where C_t represents the thrust coefficient, a deeply nonlinear function of variables λ and β.

The resulting equation of motion are then presented as follow:

$$\begin{aligned}\begin{bmatrix} \ddot{x}_m \\ \ddot{\theta} \\ \dot{\omega}_r \end{bmatrix} &= \begin{bmatrix} (m_g I_3 + m_a)^{-1} F \\ R(\theta) I_\theta^{-1} R(\theta)^\top (T_A + T_B + T_C + T_D) \\ \frac{1}{J_l}\left(\frac{P}{\omega_r} - n_g T_g\right) \end{bmatrix} \\ &= f(x, u, v, w)\end{aligned} \tag{6}$$

where, m_g is the total mass of the FOWT while m_a represents the hydrodynamic added mass vector. T_A, T_B, T_C, T_D are the aerodynamic torque vector, the buoyancy torque vector, the mooring line torque vector, the hydrodynamic drag and inertial torque vector, respectively, I_θ denotes the inertia tensor of the FOWT, $R(\theta)$ is the simplified rotation matrix, and $f = (x, u, v, w)$ constitutes the nonlinear function vector describing

the equations of motion. Here, v and w are the wind and wave disturbance vectors, respectively.

B. Model Dynamics Adjustments for Control Design

In Region III, the principal control objectives involve regulating the generator power to its rated value. Considering T_g fixed, this objective is equivalent to maintaining the rotor speed at its rated value ω_{ref}, expressed as the tracking error e_r. Simultaneously, a secondary imperative for efficient operation involves ensuring stability in the platform pitch movement by driving the platform pitch rate ω_y to zero, expressed as e_y:

$$e_r = \omega_r - \omega_{ref}, \quad e_y = \omega_y - 0 = \omega_y. \tag{7}$$

Since both dynamics share the same control input β, a challenge arises in the form of an under-actuated control problem. To address this, the rated rotor speed ω_{rd} is adapted from a fixed value to a linear function ω_{rd}^* correlated with the platform pitch rate:

$$\omega_{ref}^* = \omega_{rd}(1 - k_y\omega_y) \tag{8}$$

where k_y is a positive constant. Consequently, the control objective for the FOWT in Region III is to drive the following composite tracking error e to zero:

$$e = \omega_r - \omega_{ref}^* = \omega_r - \omega_{ref}(1 - k_y\omega_y) = e_r + k_l e_y \tag{9}$$

where $k_l = k_y\omega_{ref}$ is a positive constant scalar.

However, achieving these control objectives is impeded by the intricate coupling between states and control inputs in ω_r- and $\dot{\theta}_y$-dynamics. Hence, a reformulation of these dynamics is essential for the effective design of the control system. The thrust and power coefficients, denoted as $C_p(\lambda, \beta)$ and $C_t(\lambda, \beta)$, are modeled through polynomial functions:

$$C_p(\lambda, \beta) = g_{cp}\beta + f_{cp}, \quad C_t(\lambda, \beta) = g_{ct}\beta + f_{ct} \tag{10}$$

where g_{cp}, g_{ct}, f_{cp} and f_{ct} are polynomial functions with respect to λ.

This leads to refined dynamics for ω_r and $\dot{\theta}_y$:

$$\dot{\omega}_r = \ddot{\theta}_r = g_r\beta + D_r, \quad \dot{\omega}_y = \ddot{\theta}_y = g_y\beta + D_y \tag{11}$$

where ω_r denotes the platform pitch rate, and g_r, g_y are nonlinear functions of ω_r and ω_y respectively. The lumped uncertainties and external disturbances for each dynamics are represented by D_r and D_y, defined as follows:

$$\begin{cases} g_r = \frac{\rho_a \pi R_r^2}{2J_l\omega_r} g_{cp} \left\|\mathbf{v}_n\right\|_2^3 \\ D_r = \frac{\rho_a \pi R_r^2}{2J_l\omega_r} f_{cp} \left\|\mathbf{v}_n\right\|_2^3 - \frac{n_g}{J_l} T_g \end{cases} \tag{12}$$

$$\begin{cases} g_y = d_\theta g_{ct} \left\|\mathbf{v}_n\right\|_2^2 (\theta_x^2 + \theta_y^2 + \theta_z^2 + 1) \\ D_y = d_\theta f_{ct} \left\|\mathbf{v}_n\right\|_2^2 (\theta_x^2 + \theta_y^2 + \theta_z^2 + 1) + D_{\theta_y} \end{cases} \tag{13}$$

where D_{θ_y} and d_θ are expressed as detailed in [13].

Assumption 1. D_r and its derivative $\dot{D}_r$ are bounded with $|D_r| \leq \lambda_{r1}$ and $|\dot{D}_r| \leq \rho_{r2}$, where λ_{r1} and λ_{r2} are two positive constants.

Assumption 2. D_y and its derivative $\dot{D}_y$ are bounded with $|D_y| \leq \lambda_{y1}$ and $|\dot{D}_y| \leq \lambda_{y2}$, where λ_{y1} and λ_{y2} are two positive constants.

III. Recurrent Radial Basis Function Neural Network Estimator

This section presents the overall architecture of the considered RRBFNN, chosen for its significant ability to approximate unknown nonlinear functions in dynamic systems.

A. RRBFNN Structure

RNN structures are categorized into two classes based on the nature of feedback signals they employ, either the internal signal or the output signal as the feedback signal. The former structure is known as the internal feedback loop, while the later structure is named external feedback loop. The considered RRBFNN's structure, as depicted in Fig. 1, comprises an input layer, a hidden layer, and an output layer. This design leverages both internal and external feedback loops, enabling the capture of both internal states and output information. Moreover, the hidden layer adopts the well-established Gaussian RBFNN structure. The signal propagation within the neural network

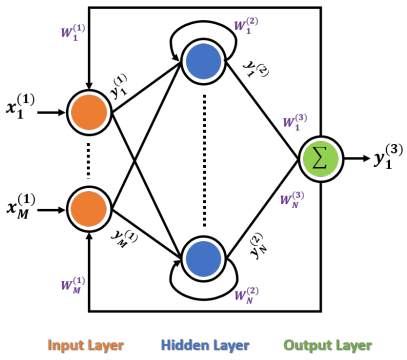

Fig. 1. RRBFNN structure

unfolds as follows:

Input Layer: The output signal $y_i^{(1)}$ of the i-th neuron of the input layer of M neurons is expressed as:

$$y_i^{(1)} = \frac{x_i^{(1)} W_i^{(1)} exy^{(3)}}{eexy^{(3)}} \tag{14}$$

where $X = [x_1^{(1)} \dots x_M^{(1)}]$ is the input signal vector where $x_i^{(1)}$ is the input value for the i-th neuron. The output vector is denoted as $Y^{(1)} = [y_1^{(1)} \dots y_M^{(1)}]$, $exy^{(3)}$ and $eexy^{(3)}$ represents the output signals of the RRBFNN output layer in the previous and pre-previous time steps, respectively. The input weight vector denoted as $W^{(1)} = [W_1^{(1)} \dots W_M^{(1)}]$ connects the output signal to the input layer of the external recurrent loop.

Hidden Layer: The output signal $y_i^{(2)}$ of the i-th neurons of the hidden layer composed of N neurons with Gaussian activation function is expressed as:

$$y_i^{(2)} = \exp(-\theta_i), \quad with \tag{15}$$

$$\theta_i = \sum_{j=1}^{M} \frac{(y_j^{(1)} + W_i^{(2)} exy_i^{(2)} - c_{ji})^2}{b_{ji}^2} \tag{16}$$

where the output vector of the hidden layer is $Y^{(2)} = [y_1^{(2)} \dots y_N^{(2)}]$, and $W^{(2)} = [W_1^{(2)} \dots W_N^{(2)}]$ is the weight vector of the internal recurrent loop, $exy_i^{(2)}$ is the output signal of the hidden neurons at the previous time step. $C = [c_{11} \dots c_{M1} \dots c_{1N} \dots c_{MN}]$ and $B = [b_{11} \dots b_{M1} \dots b_{1N} \dots b_{MN}]$ represent the centers and width vectors of the Gaussian function respectively.

Output Layer: The output signal $y^{(3)}$ of the neural network is expressed as:

$$y^{(3)} = W^{(3)\top} Y^{(2)} = \sum_{i=1}^{N} W_i^{(3)} y_i^{(2)} \tag{17}$$

where $W^{(3)} = [W_1^{(3)} \dots W_N^{(3)}]$ is the weight vector connecting the hidden layer with the output neuron.

In this neural network structure, the weight vectors $(W^{(1)}, W^{(2)}, W^{(3)})$, and the Gaussian parameters (C, B), are adjusted online to converge towards a precise approximation. Consequently, the RRBFNN output is given by:

$$\begin{aligned} &Y^{(3)}(X, W^{(3)}, W^{(2)}, W^{(1)}, C, B) = \\ &W^{(3)\top} Y^{(2)}(X, W^{(2)}, W^{(1)}, C, B). \end{aligned} \tag{18}$$

B. RRBFNN Estimator

The RRBFNN can be employed to approximate an unknown nonlinear function f. The output of the neural network, aimed at capturing f and denoted $\hat{f}$, is given by:

$$\hat{f} = W^{(3)T} Y^{(2)}(X, W^{(2)}, W^{(1)}, C, B) = W^{(3)T} Y^{(2)}. \tag{19}$$

Based on the universal approximation property, there exist optimal weight vectors and Gaussian parameters, $W^{(3)*}, W^{(2)*}, W^{(1)*}, C^*, B^*$, for the estimation of f such that:

$$\begin{aligned} f &= W^{(3)*\top} Y^{(2)*}(W^{(2)*}, W^{(1)*}, C^*, B^*) + \epsilon^* \\ &= W^{(3)*\top} Y^{(2)*} + \epsilon^* \end{aligned} \tag{20}$$

where ϵ^* is the minimal approximation error of the neural network. Thus the approximation error between f and $\hat{f}$ is defined as:

$$\begin{aligned} f - \hat{f} &= W^{(3)*\top} Y^{(2)*} + \epsilon^* - W^{(3)\top} Y^{(2)} \\ &= (W^{(3)} + \tilde{W}^{(3)})^\top (Y^{(2)} + \tilde{Y}^{(2)}) - W^{(3)\top} Y^{(2)} + \epsilon^* \\ &= \tilde{W}^{(3)\top} Y^{(2)} + W^{(3)\top} \tilde{Y}^{(2)} + \tilde{W}^{(3)\top} \tilde{Y}^{(2)} + \epsilon^* \end{aligned} \tag{21}$$

where, $\tilde{W}^{(3)} = W^{(3)*} - W^{(3)}$ and $\tilde{Y}^{(2)} = Y^{(2)*} - Y^{(2)}$.

Based on the multi-variable expression of $\tilde{Y}^{(2)}$, Taylor expansion linearization is performed to obtain a partial linear function of the adaptive parameters:

$$\begin{aligned} \tilde{Y}^{(2)} = &\frac{\partial \tilde{Y}^{(2)}}{\partial W^{(2)}} \tilde{W}^{(2)} + \frac{\partial \tilde{Y}^{(2)}}{\partial W^{(1)}} \tilde{W}^{(1)} + \frac{\partial \tilde{Y}^{(2)}}{\partial C} \tilde{C} \\ &+ \frac{\partial \tilde{Y}^{(2)}}{\partial B} \tilde{B} + \delta \end{aligned} \tag{22}$$

where $\tilde{W}^{(2)} = W^{(2)*} - W^{(2)}$, $\tilde{W}^{(1)} = W^{(1)*} - W^{(1)}$, $\tilde{C} = C^* - C$, $\tilde{B} = B^* - B$, δ is a high order term, and the partial derivatives are defined as:

$$\frac{\partial \tilde{Y}^{(2)}}{\partial Z} = \left[\frac{\partial \tilde{y_1}^{(2)}}{\partial Z}^\top \quad \dots \frac{\partial \tilde{y_N}^{(2)}}{\partial Z}^\top \right]^\top \tag{23}$$

with $Z = W^{(3)}, W^{(2)}, W^{(1)}, C, B$, respectively.

Substituting (22) into the expression of the approximation error:

$$\begin{aligned} f - \hat{f} &= \tilde{W}^{(3)\top} Y^{(2)} + W^{(3)\top} \tilde{Y}^{(2)} + \tilde{W}^{(3)\top} \tilde{Y}^{(2)} + \epsilon^* \\ &= \tilde{W}^{(3)\top} Y^{(2)} + W^{(3)\top} \Big(\frac{\partial \tilde{Y}^{(2)}}{\partial W^{(2)}} \tilde{W}^{(2)} + \frac{\partial \tilde{Y}^{(2)}}{\partial W^{(1)}} \tilde{W}^{(1)} \\ &+ \frac{\partial \tilde{Y}^{(2)}}{\partial C} \tilde{C} + \frac{\partial \tilde{Y}^{(2)}}{\partial B} \tilde{B} + \delta \Big) + \tilde{W}^{(3)\top} \tilde{Y}^{(2)} + \epsilon^* \\ &= \tilde{W}^{(3)\top} Y^{(2)} + W^{(3)\top} \Big(\frac{\partial \tilde{Y}^{(2)}}{\partial W^{(2)}} \tilde{W}^{(2)} + \frac{\partial \tilde{Y}^{(2)}}{\partial W^{(1)}} \tilde{W}^{(1)} \\ &+ \frac{\partial \tilde{Y}^{(2)}}{\partial C} \tilde{C} + \frac{\partial \tilde{Y}^{(2)}}{\partial B} \tilde{B} \Big) + \varepsilon_l^* \end{aligned} \tag{24}$$

where $\varepsilon_l^* = W^{(3)\top} \delta + \tilde{W}^{(3)\top} \tilde{Y}^{(2)} + \epsilon^*$ is the lumped approximation error of the neural network.

IV. Adaptive Control Design

In this section, the synthesis of a composite control strategy involving the STSM control and the presented RRBFNN is detailed as illustrated in Fig. 2. The resulting controller, denotes

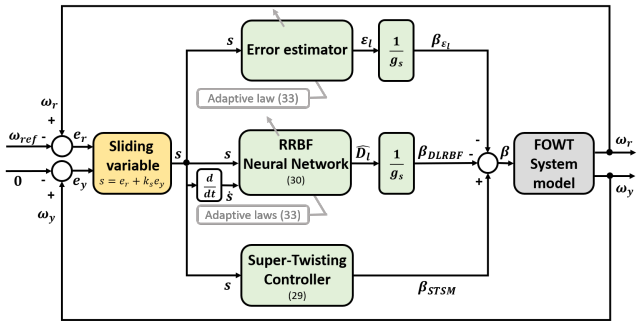

Fig. 2. Bloc diagram of the RRBFNN-STSM-CBPC

as RRBFNN-STSM-CBPC, is expressed by the following control law:

$$\beta = \beta_{\text{STSM}} + \beta_{\text{RRBF}} + \beta_{\varepsilon_l} \tag{25}$$

where β_{STSM} represents the standard STSM-CBPC, while β_{RRBF} is the contribution from the neural network, with β_{ε_l} the error estimation.

A. Super-Twisting Sliding Mode Control Design

The STSM controller, widely recognized for its robust performances in handling first-order systems amid model uncertainties and external disturbances, is introduced in this subsection.

Based on the formulation of the composite tracking error (9) the sliding variable s and its time derivative are defined as follows:

$$s = \omega_r - \omega_{ref} + k_y \omega_{ref} \omega_y = e_r + k_l e_y \tag{26}$$

$$\dot{s} = \dot{e_r} + k_l \dot{e_y} = \dot{\omega_r} + k_l \dot{\omega_y}. \tag{27}$$

Substituting (11) into (27) yields the time derivative expression:

$$\dot{s} = (g_r \beta + D_r) + k_l (g_y \beta + D_y) = g_l \beta + D_l \tag{28}$$

where $g_l = g_r + k_l g_y$ and $D_l = D_r + k_l D_y$ represents the unknown lumped uncertainties and external disturbance.

According to Assumption 1 and Assumption 2, $|D_l| \leq \lambda_{l1}$ and $|\dot{D_l}| \leq \lambda_{l2}$ where $\lambda_{l1} = \lambda_{r1} + k_l \lambda_{y1}$ and $\lambda_{l2} = \lambda_{r2} + k_l \lambda_{y2}$ are two positive constants.

The control law β_{STSM} is then given by:

$$\beta_{STSM} = \frac{1}{g_l}(-k_1 \sqrt{|s|} sgn(s) - k_2 \int sgn(s) d\tau). \tag{29}$$

Here, $k_1 > 0$, $k_2 > 0$ are the STSM controller gains, and $sgn(\cdot)$ is the signum function.

To enhance the robustness and control performance of the presented standard STSM-CBPC controller, the RRBFNN is integrated to attenuate disturbances.

B. Design of RRBFNN-STSM-CBPC

This subsection outlines the design of the adaptive RRBFNN-STSM-CBPC.

The primary objective of the neural network is to capture and estimate the unknown lumped disturbance denoted as D_l. The input to the RRBFNN is chosen as $x = [s, \dot{s}]$, and the output of the RRBFNN corresponds to the estimated value $\hat{D}_l$. Thus, based on (19) and (20), the expression of $\hat{D}_l$ and D_l are given by:

$$\hat{D}_l = Y^{(3)} = W^{(2)\top} Y^{(2)}, \quad D_l = W^{(2)*\top}(n) Y^{(2)} + \epsilon^* \tag{30}$$

where the difference $\tilde{D}_l = D_l - \hat{D}_l$ is defined as (23).

The control laws of the neural network estimator β_{RRBF} and the error estimator are formulated as:

$$\beta_{\text{RRBF}} = -\frac{1}{g_l}\hat{D}_l, \quad \beta_{\varepsilon_l} = -\frac{1}{g_l}\varepsilon_l. \tag{31}$$

Substituting (25) into (28), the s-dynamics can be described as:

$$\dot{s} = -k_1 \sqrt{|s|} sgn(s) - k_2 \int sgn(s) d\tau - \hat{D}_l - \varepsilon_l + D_l. \tag{32}$$

The following theorem is used to update online the parameters of the neural networks $(W^{(1)}, W^{(2)}, W^{(3)}, C, B)$, and the lumped neural network error ε_l, by adaptive laws derived from Lyapunov stability analysis to ensure the asymptotic stability of the closed-loop system.

Theorem 1. Regarding the system (32), considering $k_1 > 0$ and $k_2 > 0$, if the learning laws are selected as (33), s will converge to the origin asymptotically.

$$\begin{aligned}
\dot{\tilde{W}}^{(3)\top} &= -\frac{k_2 sgn(s) Y^{(2)}}{\eta_1} \\
\dot{\tilde{W}}^{(2)\top} &= -\frac{k_2 sgn(s) W^{(3)\top} \frac{\partial \tilde{Y}^{(2)}}{\partial W^{(2)}}}{\eta_2} \\
\dot{\tilde{W}}^{(1)\top} &= -\frac{k_2 sgn(s) W^{(3)\top} \frac{\partial \tilde{Y}^{(2)}}{\partial W^{(1)}}}{\eta_3} \\
\dot{\tilde{C}}^{\top} &= -\frac{k_2 sgn(s) W^{(3)\top} \frac{\partial \tilde{Y}^{(2)}}{\partial C}}{\eta_4} \\
\dot{\tilde{B}}^{\top} &= -\frac{k_2 sgn(s) W^{(3)\top} \frac{\partial \tilde{Y}^{(2)}}{\partial B}}{\eta_5}, \quad \dot{\varepsilon}_l = \frac{k_2 sgn(s)}{\eta_6}
\end{aligned} \tag{33}$$

where η_1, η_2, η_3, η_4, η_5, η_6 are positive learning rates.

Proof. The system (32) is converted to the following form.

$$\begin{cases} \dot{s} = -k_1 \sqrt{|s|} \text{sgn}(s) + (D_l - \hat{D}_l) - \varepsilon_l + \phi \\ \dot{\phi} = -k_2 \text{sgn}(s) \end{cases} \tag{34}$$

The following Lyapunov candidate function [14] is chosen for the system (34).

$$\begin{aligned}
V = {} & k_2 |s| + \frac{1}{2}\phi^2 + \frac{1}{2} tr(\tilde{W}^{(3)\top} \eta_1 \tilde{W}^{(3)}) \\
& + \frac{1}{2} tr(\tilde{W}^{(2)\top} \eta_2 \tilde{W}^{(2)}) + \frac{1}{2} tr(\tilde{W}^{(1)\top} \eta_3 \tilde{W}^{(1)}) \\
& + \frac{1}{2} tr(\tilde{C}^{\top} \eta_4 \tilde{C}) + \frac{1}{2} tr(\tilde{B}^{\top} \eta_5 \tilde{B}) + \frac{1}{2\eta_6} \Delta \varepsilon_l^2
\end{aligned} \tag{35}$$

where $\Delta \varepsilon_l = \varepsilon_l - \varepsilon_l^*$. Based on (24), the time derivative of V can be calculated as:

$$\begin{aligned}
\dot{V} = {} & k_2 \dot{s} sgn(s) + \phi \dot{\phi} + \dot{\psi} \\
= {} & k_2 (-k_1 \sqrt{|s|} sgn(s) + (D_l - \hat{D}_l) - \varepsilon_l + \phi) sgn(s) \\
& - \phi k_2 sgn(s) + \dot{\psi} \\
= {} & -k_1 k_2 \sqrt{|s|} + k_2 sgn(s) \tilde{W}^{(3)\top} Y^{(2)} \\
& + k_2 sgn(s) W^{(3)\top} (\frac{\partial \tilde{Y}^{(2)}}{\partial W^{(2)}} \tilde{W}^{(2)} + \frac{\partial \tilde{Y}^{(2)}}{\partial W^{(1)}} \tilde{W}^{(1)} \\
& + \frac{\partial \tilde{Y}^{(2)}}{\partial C} \tilde{C} + \frac{\partial \tilde{Y}^{(2)}}{\partial B} \tilde{B}) - k_2 sgn(s) \Delta \varepsilon_l + \dot{\psi}
\end{aligned} \tag{36}$$

with $\dot{\psi}$ expressed as:

$$\begin{aligned}
\dot{\psi} = {} & tr(\dot{\tilde{W}}^{(3)\top} \eta_1 \tilde{W}^{(3)}) + tr(\dot{\tilde{W}}^{(2)\top} \eta_2 \tilde{W}^{(2)}) \\
& + tr(\dot{\tilde{W}}^{(1)\top} \eta_3 \tilde{W}^{(1)}) + tr(\dot{\tilde{C}}^{\top} \eta_4 \tilde{C}) + tr(\dot{\tilde{B}}^{\top} \eta_5 \tilde{B}) \\
& + \frac{1}{\eta_6} \dot{\Delta \varepsilon_l} \Delta \varepsilon_l.
\end{aligned} \tag{37}$$

Considering the properties of matrix trace and substituting (33) and (37) into (36) leads to:

$$\dot{V} = -k_1 k_2 \sqrt{|s|}. \tag{38}$$

Since $\dot{V} \leq 0$, $\dot{V}$ is negative semi-definite. Hence the asymptotic convergence of s to the origin can be demonstrated. The proof is completed. □

V. Simulations Results

To assess the effectiveness of the RRBFNN-STSM-CBPC, a co-simulation study was conducted using both Simulink and OpenFAST. The performance of the proposed CBPC was evaluated in comparison to the standard STSM-CBPC, maintaining identical gain values (k_1, k_2), and alongside a GSPI controller [1].

OpenFAST is utilized to simulate the NREL OC4-DeepCwind 5 MW semi-submersible FOWT. The wind and wave profiles are defined by a mean wind speed of 20 m/s with 12% turbulence, a wave height of 3 meters, and a peak wave period of 12 seconds. The control parameters are chosen as $k_1 = k_2 = 2$ representing the controller gains of (29), $k_y = 0.05$, and $g_l = -2.6296$. The input of the RRBFNN is denoted as $x = [s \quad \dot{s}]$. The hidden layer of the network consists of $N = 5$ neurons, and the parameters of the Gaussian function are specified as $c = [[-1 \quad -0.5 \quad 0 \quad 0.5 \quad 1]; [-1 \quad -0.5 \quad 0 \quad 0.5 \quad 1]]$, $b = 2$, $\eta_1 = 10$, $\eta_2 = 10$, $\eta_3 = 100$, $\eta_4 = 100$, $\eta_5 = 10$, and $\eta_6 = 2$.

The simulation results, depicted in Fig. 3, provide a comparative analysis of the performance among the RRBFNN-STSM-CBPC, STSM-CBPC, and the GSPI controller.

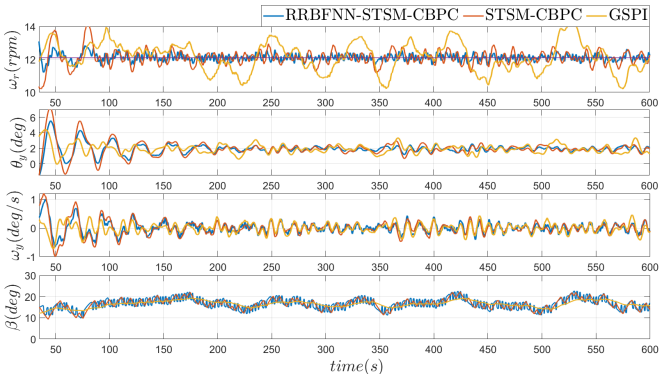

Fig. 3. Simulations results

TABLE I
Mean errors and standard deviation

Controllers	**Mean** ω_r [rpm]	**STD** ω_r [rpm]	**Mean** θ_y [deg]	**STD** θ_y [deg]
GSPI	12.1642	0.8915	1.8906	0.4965
STSM-CBPC	12.1104	0.4767	1.9016	0.6153
RRBF-STSM-CBPC	12.1083	0.2192	1.9097	0.4326

Table I presents the means and standard deviations (STD) values. The RRBFNN-STSM demonstrates superior performance in regulating rotor speed to its rated value, achieving a mean speed closer to the rated value compared to both the STSM and the GSPI controller. Additionally, the STD of the rotor speed with RRBFNN-STSM is nearly 4.5 times lower than that of the GSPI and more than half that of the STSM. The platform pitch angle θ_y exhibits lower STD values with the proposed CBPC, meeting control objectives more effectively than the STSM controller. However, the RRBFNN-STSM introduces a more aggressive blade pitch angle, resulting in wider variations compared to the GSPI.

VI. Conclusion

This paper presents the design of a RRBFNN-STSM-CBPC tailored for a FOWT mounted on a semi-submersible platform. Through the integration of RRBFNN to approximate lumped uncertainties and external disturbances, the composite controller is designed to achieve dual objectives: regulating the rotor speed to its rated value and minimizing platform pitching motion. By combining the robust characteristics of SMC with the dynamic estimation capabilities of recurrent neural networks, the proposed controller offers an adaptive solution. Simulation results demonstrate the effectiveness of the proposed CBPC in handling the intricate dynamics and uncertainties associated with FOWTs operating in Region III.

Acknowledgment

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

References

[1] J. Jonkman, "Dynamics modeling and loads analysis of an offshore floating wind turbine," National Renewable Energy Laboratory, Golden, Colorado, USA, Tech. Rep. NREL/TP-500-41958, Dec. 2007.

[2] K. H. O. Bagherieh and R. Horowitz, "Nonlinear control of floating offshore wind turbines using input/output feedback linearization and sliding control," in *ASME Dynamic Systems and Control Conference*, San Antonio, Texas, USA, Oct. 2014.

[3] C. Zhang, E. Tahoumi, S. Gutierrez, F. Plestan, and J. Deleon-Morales, "Adaptive robust control of floating offshore wind turbine based on sliding mode," in *IEEE 58th Conference on Decision and Control*, Nice, France, Dec. 2019.

[4] F. P. C. Zhang, S. V Gutierrez and J. D. Leon-Morales, "Adaptive super-twisting control of floating wind turbines with collective blade pitch control," *IFAC-PapersOnLine*, vol. 52, pp. 6936–6941, Aug. 2019.

[5] J. M. Jonkman and M. L. Buhl, "Fast user's guide," in *Nat. Renewable Energy Lab.*, U.S. Dept. Energy, Golden, CO, USA, 2005.

[6] C. Zhang and F. Plestan, "Adaptive sliding mode control of floating offshore wind turbine equipped by permanent magnet synchronous generator," *Wind Energy*, vol. 24, pp. 754–769, Jan. 2021.

[7] J. Fei and H. Ding, "Adaptive sliding mode control of dynamic system using rbf neural network," *Nonlinear Dyn*, vol. 70, no. 2, pp. 1563–1573, 2012.

[8] L. Wang, T. Chai, and L. Zhai, "Neural-network-based terminal sliding-mode control of robotic manipulators including actuator dynamics," *IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3296–3304, 2009.

[9] J. Fei and C. Lu, "Adaptive sliding mode control of dynamic systems using double loop recurrent neural network structure," *IEEE Trans. Neural Netw. and Learn. Syst.*, vol. 29, no. 4, pp. 1275–1286, 2018.

[10] A. Robertson, J. M. Jonkman, M. Masciola, A. G. H. Song, A. Coulling, and C. Luan, "Definition of the semisubmersible floating system for phase ii of oc4," National Renewable Energy Laboratory, Golden, Colorado, USA, Tech. Rep. NREL/TP-5000-60601, Sep. 2014.

[11] J. R. Homer and R. Nagamune, "Physics-based 3-d control-oriented modeling of floating wind turbines," *IEEE Trans. Control Syst. Technol*, vol. 26, no. 1, pp. 14–26, 2018.

[12] 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, 2022.

[13] Y.-C. Liu, H. Basbas, and S. Laghrouche, "Robust blade pitch control of semi-submersible floating offshore wind turbines based on the modified super-twisting sliding-mode algorithm," *Under Review*, 2024.

[14] 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," *J. Frankl. Inst*, vol. 358, no. 1, pp. 400–427, 2021.