Black-box nonlinear observer-based deep reinforcement learning controller with application on Floating Wind Turbines

Hadi Mohammadian KhalafAnsar [a,*], Jafar Keighobadi [a], Mir Mohammad Ettefagh [a], Jafar Tanha [b],

[a] *Faculty of Mechanical Engineering, University of Tabriz, East Azerbaijan.*

[b] *Faculty of Electrical and Computer Engineering, University of Tabriz, East Azerbaijan*.

ABSTRACT

The developments in ocean energy have prompted researchers to investigate the floating offshore wind turbines (FOWTs). Therefore, the need to stabilize this structure is a crucial aspect in control engineering. The presence of disturbances and noise highlights the importance of implementing an intelligent control approach. This paper focuses on the nonlinear FOWT with an online feedback control system utilizing deep reinforcement learning (DRL) algorithms. The inherent characteristics of DRL allow the FOWT to adapt to changing environments, employing two parallel networks known as online-target. An observer system is integrated with direct gain based on the measured outputs from available sensors, demonstrating global asymptotic stability through a Lyapunov function. Furthermore, an agent trained using DQN in the adapted environment requires minimal instances to determine the optimal control policy. Simulation tests conducted in MATLAB exhibit the superior performance of the proposed observer-controller compared to the LQR approach in terms of FOWT stabilization. Additionally, it is shown that the Luenberger observer doesn't perform as effectively as the newly developed observer in presence of uncertainty, unknown disturbances. Finally, the outcomes are compared with the gain scheduling PI control method recommended by Jonkman as a well-known benchmark to validate the accuracy of the simulation results.

Keywords: Floating Offshore Wind Turbine; Multi-body system; Nonlinear control; Machine learning; Deep reinforcement learning; Black-box nonlinear observer;

1. Introduction

1.1. Background and motivation

Deep water is a significant source of wind energy, but traditional concrete foundations are not suitable for deployment in such depths. Floating foundations offer stability and environmental response, offering six degrees of freedom for Floating Offshore Wind Turbines (FOWTs). However, design limitations like blade tip deflection have become more significant. To ensure accuracy, efficient controllers and tuning techniques are essential, especially for large, flexible turbines. Automated optimization methods like Cp-max, HawtOpt, and WISDEM® [1-3] help integrate dynamic aspects into optimization. Regular updates to controllers are necessary for evolving wind turbine designs [4,5].

1.2. Literature review

Floating Offshore Wind Turbines (FOWTs) face complex dynamics that necessitate advanced control strategies to optimize performance. Traditional proportional–integral–derivative (PID) controllers often struggle with uncertainties and disturbances, prompting the exploration of novel approaches including machine learning and deep learning techniques. Enrique et al. [6] combined reinforcement learning (RL) with PID for pitch angle control, enhancing traditional methods, while their use of a radial basis function (RBF) network [7] and adaptive neuro-fuzzy inference systems [8] shows potential in wind power estimation. Deep learning methods like Long Short-Term Memory

* Corresponding author. Tel./Fax: 041-33354153; Mobile Number: 09353813756
E-mail addresses: H.MohammadianKhalafAnsar@tabrizu.ac.ir (Hadi Mohammadian KhalafAnsar); keighobadi@tabrizu.ac.ir (Jafar Keighobadi); ettefagh@tabrizu.ac.ir(Mir Mohammad Ettefagh); tanha@tabrizu.ac.ir(Jafar Tanha)

(LSTM) models and Variational Mode Decomposition (VMD) have also proven effective for wind speed forecasting [9-12]. Hybrid control approaches that integrate adaptive neural networks with PID controllers and inverse plant models are being investigated for improved signal tracking [13].

Recent research has further explored RL-based mechanisms for FOWT control. Zhang et al. [14] applied RL to power systems, and Fernandez-Gauna et al. [15] used RL to adjust controllers for varying wind conditions in a variable-speed FOWT. Abouheaf et al. employed RL in policy iteration and adaptive actor-critic methods for a doubly-fed induction generator FOWT [16], while RL has also been applied to yaw control tasks [17, 18]. Hybrid intelligent controllers combining traditional methods with intelligent approaches have shown promise. For instance, Iqbal et al. proposed a hybrid fuzzy system and model predictive controller [19], and Ngo et al. created a fuzzy logic-based PID controller [20]. Sedighizadeh and Rezazadeh developed an adaptive PID controller tuned with RL [21], and optimization methods like particle swarm optimization (PSO) have been used alongside neural networks [22]. Active control techniques, including neural network-based systems for pitch control [23-26] and hybrid ANFIS and fuzzy systems [27, 28], have demonstrated potential in enhancing FOWT performance.

In addition to controller design, observer design plays a crucial role in estimating unmeasured signals in nonlinear systems [29–32]. However, model-oriented observers often require precise system models, which may be impractical under real-world uncertainties [33–40]. Sliding mode algorithms offer advantages such as finite-time convergence and robustness to uncertainty [41–49], with second-order sliding mode observers addressing chattering issues and enhancing robustness [50–54]. These advancements in control and estimation techniques, including machine learning and sliding mode methods, hold promise for improving the performance and reliability of FOWTs under challenging conditions.

1.3. Paper contributions and organizations

For the past two decades, robustness in control systems amid modeling uncertainties has been a significant research focus [39]. This is often due to simplified dynamic models and approximate physical parameters. This paper addresses gaps in FOWT control, particularly yaw and generator torque modulation, using Deep Reinforcement Learning (DRL) combined with Deep Q-Network (DQN) for real-time control. The proposed DRL approach, supported by extensive simulation data, offers potential for efficient nonlinear control strategies, even amidst environmental constraints and model uncertainties.

The paper introduces a nonlinear observer designed to reconstruct state variables—translation, rotation, and their derivatives—using only translational/orientational measurements. This observer, developed independently of the controller, ensures global asymptotic stability and is readily executable due to its design. Extensive MATLAB simulations demonstrate that the black-box nonlinear observer-based DRL system outperforms traditional controllers, such as the Linear Quadratic Regulator (LQR) and Luenberger observer, in control performance. The DRL system is also compared to the gain scheduling PI control method by Jonkman, confirming its effectiveness.

Novelties of proposed approach to Floating Offshore Wind Turbine Control:

- **Unique DRL Integration:** Optimizes control policies for nonlinear FOWT dynamics.
- **Online Feedback Control:** Uses DRL algorithms (DQN, actor-critic) for real-time adjustments.
- **Black-Box Nonlinear Observer:** Enhances adaptability and robustness using sensor data.
- **Superior Performance:** Outperforms traditional linear methods like LQR.
- **Global Asymptotic Stability:** Achieves stability and rapid convergence with minimal learning instances.

Paper structure is as follows:
- Section 2: FOWT modeling and disturbance handling.
- Section 3: Nonlinear observer design.
- Section 4: Comparison of LQR controller and Luenberger observer.
- Section 5: Conclusion and reflections.

2. Problem formulation of FOWT

2.1. General description

This paper aims to simulate the exact model for the FOWT, a semi-submersible structure with three triangular buoyancy cylinders and a central cylinder for tower control. The proposed controller has a nominal power of 5-MW and a tower height of 87.6 m, a weight of 13.5 kilotons for entire structure, filling a gap in literature on RL applications in FOWT control.

Fig. 1 illustrates the various components of the system, including the aerodynamic force ($\vec{F}_A$), buoyancy force ($\vec{F}_B$), catenary line forces ($\vec{F}_C$), and hydrodynamic drag ($\vec{F}_D$). Additionally, the forces in inertial and body references are denoted as $\mathcal{F}^0$and $\mathcal{F}^b$, respectively. Torque ($\vec{T}_A, \vec{T}_B, \vec{T}_C$, and $\vec{T}_D$) is associated with each force, with T_r representing the rotor-oriented torque. Furthermore, Fig. 2 provides a visual representation of the system's components.

2.1.1. State space model of FOWT

The state space analysis of a system involves considering various forces.

$$f\left(\mathrm{x},\mathbf{u},\mathbf{v},\mathbf{w}\right)=\begin{bmatrix}\dot{\vec{x}}_g\\ \dot{\vec{\theta}}_g\\ \omega_r\\ \omega_g\\ \vec{f}_F\\ \vec{f}_T\\ f_Q\end{bmatrix} \tag{1}$$

where $\vec{f}_F(\vec{x},\vec{u},\vec{v},\vec{w})$ stands for the resultant force:

$$\vec{f}_F=\left(m_g I_{3\times3}+diag\left[\vec{m}_a\right]\right)^{-1}\sum_j \vec{F}_j \tag{2}$$

m_g is the FOWT's mass, $I_{3\times3}$ is the identity matrix, and $\vec{m}_a$ (added mass) is the inertia increase due to fluid displacement during acceleration. The other term in Equation (1), $\vec{f}_T$, represents the resultant torque:

$$\vec{f}_T=\left(\mathrm{R}\mathrm{I}_g^{-1}\mathrm{R}^T\right)\sum_j \vec{T}_j \tag{3}$$

In this mathematical expression, I_g denotes the inertial tensor with respect to the vertical axis, R signifies the transformation matrix, and $\vec{T}_j$ denotes the collective torques acting on the system. The resultant f_Q is derived as:

$$f_Q = \begin{bmatrix} \sum_{k_r} \frac{1}{J_r} Q_{k_r} \\ \sum_{k_g} \frac{1}{J_g} Q_{k_g} \end{bmatrix} \tag{4}$$

where J_r and J_g denote the inertia of the rotor and generator-side shaft, respectively, and Q_{k_r} and Q_{k_g} represent the k_r^{th} and k_g^{th} torque around each shaft.

2.1.2. buoyancy force

The buoyant force equals the weight of fluid displaced by the floating object, as per Archimedes' principle [55]:

$$\vec{F}_{B,i}(x) = \rho_\omega g A_i l_i \hat{e}_3 \tag{5}$$

In the provided equation, ρ_ω represents the density of water, g denotes the acceleration due to gravity, A_i signifies the projection of the cylinder along its height, l_i represents the length of the cylinder, and $\hat{e}_3$ denotes the unit vector aligned with the z-axis direction.

2.1.3. Wave simulation

A 1.75m amplitude, 12s period sine wave yields a three-dimensional wave height:

$$h(\vec{x}_w, t, \alpha) = A\sin\left(\Phi(\vec{x}_w, t, \alpha)\right) \tag{6}$$

where A is the magnitude of the wave oscillation, t stands for the simulation time, α shows the change in the direction of the wave around the z-axis of the inertial coordinates, $\vec{x}_w$ is the spatial position of the wave elevation and the variable Φ, a function of time and place, is computed as follows:

$$\Phi(\vec{x}_w, t, \alpha) = \frac{-\omega^2}{g}\left(\hat{e}_1 \mathrm{R}_z^T(\alpha)\vec{x}_w\right) + \omega t + \varphi \tag{7}$$

where ω, $\mathrm{R}_z^T(\alpha)$, g and φ stand for the rate of recurrence of the wave, the transformation matrix about the z axis, the gravity constant, and the phase angle, respectively.

JONSWAP spectrum, resulting of the Pierson – Moskowitz spectrum valid for undeveloped marine countries, is given by:

$$S_{nn}(\omega)_J = S_{nn}(\omega)_{p_M} \gamma_p^{e^{\frac{(\omega-\omega_p)^2}{2\sigma^2\omega_p^2}}} \tag{8}$$

where γ_p is the ultimate increase factor and σ the spectra ultimate width is:

$$\begin{aligned} \sigma_a &= 0.07 \quad \omega \le \omega_p \\ \sigma_b &= 0.09 \quad \omega > \omega_p \end{aligned} \tag{9}$$

and $S_{nn}(\omega)_{p_M}$ is the Pierson – Moskowitz spectrum with the amount of:

$$S_{nn}(\omega)_{pM} = \frac{H_s^2}{4\pi T_Z^4 \omega^5} \exp\left(\frac{1}{\pi T_Z^4 \omega^4}\right) \tag{10}$$

Where H_s and T_z represent wave height and the average periodic time, respectively. With usage

of Equation (8), the amplitude of wave is calculated as follows.

$$A = \sqrt{2S_{\omega}\left(\omega_i\right)\Delta\omega} \tag{11}$$

Hydrodynamic characteristics such as velocity, acceleration, and pressure will be analyzed:

$$\begin{aligned}
\vec{v}\left(\vec{x}_w,t,\alpha\right) &= \omega e^{\left(\frac{-\omega^2}{g}z\right)}\begin{bmatrix} cos(\alpha)\,sin(\zeta) \\ sin(\alpha)\,sin(\zeta) \\ cos(\zeta) \end{bmatrix} \\
\vec{a}\left(\vec{x}_w,t,\alpha\right) &= \omega^2 e^{\left(\frac{-\omega^2}{g}z\right)}\begin{bmatrix} cos(\alpha)\,cos(\zeta) \\ cos(\alpha)\,cos(\zeta) \\ -sin(\zeta) \end{bmatrix} \\
P_d\left(\vec{x}_w,t,\alpha\right) &= \rho g e^{\left(\frac{-\omega^2}{g}z\right)} sin(\zeta)
\end{aligned} \tag{12}$$

in which z is equal with:

$$z = \hat{e}_3 \mathrm{R}_z^T\left(a\right)\vec{x}_w \tag{13}$$

2.1.4. Wave drag force

The drag force resists body movement in fluid, estimated by the Morrison equation [55]:

$$\vec{F}_{Dt,i} = K_{d,i}\left\|\vec{v}_{t,i}\right\| + K_{a,i}\vec{a}_{t,i} \tag{14}$$

where $K_{d,i}$ is the drag constant, $K_{a,i}$ demonstrates the inertia constant, and $\vec{v}_{t,i}$ and $\vec{a}_{t,i}$ are the crosswise velocities and accelerations, correspondingly, calculated by:

$$\begin{aligned}
\vec{v}_t &= \underline{R}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}\underline{R}^T\vec{\omega}_{rel} \\
\vec{a}_t &= \underline{R}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}\underline{R}^T\dot{\vec{\omega}}_{rel}
\end{aligned} \tag{15}$$

and norm speed is also equal to:

$$\left\|\vec{v}_{t,i}\right\| = \left(\vec{\omega}_{rel}^T R\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\underline{R}^T\vec{\omega}_{rel}\right)^{\frac{1}{2}} \tag{16}$$

The relative wave velocity, $\vec{\omega}_{rel}$, and its derivative are:

$$\begin{aligned}
\vec{\omega}_{rel} &= \vec{\omega} - \dot{\vec{x}}_g - \dot{R}\vec{r}_{gi}^{\,b} \\
\dot{\vec{\omega}}_{rel} &= \dot{\vec{\omega}}
\end{aligned} \tag{17}$$

where $\vec{\omega}$ holds the wave velocity elements, $\dot{\vec{x}}_g$ includes the state variable, $\dot{R}$ is a derivative of the transformation matrix:

$$\dot{R}=\begin{bmatrix} \circ & -\dot{\theta}_z & \dot{\theta}_y \\ \dot{\theta}_z & \circ & -\dot{\theta}_x \\ -\dot{\theta}_y & \dot{\theta}_x & \circ \end{bmatrix} \tag{18}$$

2.1.5. Thrust and drag force of air

The aerodynamic force consists of thrust force and drag force, with an approximation for thrust force applied at the thrust center:

$$\vec{F}_A=\frac{1}{2}\rho A_r C_t(\lambda,\beta)\|\vec{v}_n\|\vec{v}_n \tag{19}$$

C_t is the drift factor, with the variables including the tip speed ratio (TSR), λ, and the pitch angle, β. The ρ and A_r are density and blade area, respectively. Average velocity to the surface of the rotor blades, $\vec{v}_n$, can be computed by:

$$\vec{v}_n=\underline{R}_{eq}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\underline{R}_{eq}^T\vec{v}_{rel} \tag{20}$$

And the norm of this speed is:

$$\begin{aligned} \|\vec{v}_n\| &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\underline{R}_{eq}^T\vec{v}_{rel} \\ R_{eq} &= \underline{R}_y(\theta_{tilt})\underline{R}_z(\gamma)\underline{R} \end{aligned} \tag{21}$$

$\vec{v}_{rel}$ is the relative wind velocity. θ_{tilt} represents the measurement of the angle formed between the rotor axis, and horizontal wind direction and γ stands for the angle between the nacelle and tower.

Wind disturbance is the wind velocity vector relative to the rotor's thrust center:

$$\mathrm{v}=\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}[m/s]$$

TurbSim [56] created a wind disturbance profile resembling real-world conditions, with an average wind speed of 18 m/s and a 20° prevailing angle. This tool creates consistent wind profiles for FAST and the proposed model, requiring a singular 3-D wind velocity vector. The wind vector's orientation in relation to the world frame at a hub height of 90 meters (the original position of the FOWT's center of thrust) is illustrated in Fig. 3. It is assumed that the wind vector consistently acts upon the center of thrust of the FOWT.

Due to FAST software limitations, a single frequency wave was used, its frequency and magnitude determined using the Pierson-Moskowitz Ocean wave spectrum [57], assuming a fully developed wave profile.

$$S(\omega)=\frac{s_\alpha g^2}{\omega^5}\exp\left(s_\beta\left(\frac{\omega_0}{\omega}\right)^4\right),$$

The parameters s_α, g and s_β are constants, whereas ω_0 is determined by:

$$\omega_0=\frac{g}{U_{19.5}}.$$

$U_{19.5}$ represents the wind velocity at a height of 19.5 meters above sea level. Because our reference height is significantly greater, we may estimate $U_{19.5}$ using the power law approximation.

$$U_{19.5} = U_{90}\left(\frac{h_{19.5}}{h_{90}}\right)^{\kappa},$$

In this context, $U_{19.5}$ signifies wind speed at a height of 90 meters, whereas $h_{19.5}$ and h_{90} are the specified heights for reference in 19.5 and 90 meters, correspondingly. The wind power exponent κ, with a quantity of 0.11 in open sea circumstances, is also a crucial element. Based on the aforementioned characteristics, the Pierson-Moskowitz spectrum, exposed in Fig. 4, may be used to calculate significant wave height and peak spectral period for the provided wind circumstances. These resulting values can then be used as inputs for the FAST model.

The wave disturbance gets reduced as a series of *n* wave velocity vectors $\vec{w}_{v,1}\ldots\vec{w}_{v,n}$-[m/s], *n* wave acceleration vectors $\vec{w}_{a,1}\ldots\vec{w}_{a,n}$-$\left[m/s^2\right]$, *n* wave heights $w_{h,1}\ldots w_{h,n}$-$[m]$, and *n* dynamic pressure terms $w_{p,1}\ldots w_{p,n}$-$[Pa]$, all related to the global frame. Subsequently, we will delineate the wave disturbance vector:

$$\mathbf{w} = \left[\vec{w}_{v,1}\ldots\vec{w}_{v,n}, \vec{w}_{a,1}\ldots\vec{w}_{a,n}, w_{h,1}\ldots w_{h,n}, w_{p,1}\ldots w_{p,n}\right]^T$$

If the aerodynamic power is expressed by:

$$P = \frac{1}{2}\rho A_r C_p\left(\lambda,\beta\right)\left\|\vec{v}_n\right\|^3 \tag{22}$$

Where C_p is the power constant. The balance of torque around both the rotor and generator axes results in:

$$\begin{aligned}\vec{\omega}_r &= \frac{1}{J_r}\left(\frac{P}{\omega_r} - k\left(\theta_r - \frac{1}{N_{gr}}\theta_g\right) - b\left(\omega_r - \frac{1}{N_{gr}}\omega_g\right)\right)\\ \vec{\omega}_g &= \frac{1}{J_g}\left(-T_g + \frac{k}{N_{gr}} + \frac{b}{N_{gr}}\left(\omega_r - \frac{1}{N_{gr}}\omega_g\right)\right)\end{aligned} \tag{23}$$

Where N_{gr} is the gear ratio and T_g generator torque.

2.1.6. Drag force of cables

The mooring system connects wind turbine cables to the sea bottom, responding to wind and wave disturbances. The Gaussian model cable involves nonlinear coupling equations, varying depending on the rope's position on the sea bottom or direct interaction.

The vector $\vec{x}_{t,i}$ is the connection point to the FOWT given by:

$$\vec{x}_{t,i} = \vec{x}_{a,i} - \vec{x}_g - \underline{R}\vec{r}^{\,b}_{gci} \tag{24}$$

The vector of $\vec{x}_{t,i}$ is decomposed into its components as:

$$\vec{x}_{th,i}=\frac{\begin{pmatrix}1&0&0\\0&1&0\\0&0&0\end{pmatrix}\left(\vec{x}_{a,i}-\vec{x}_g-\underline{R}\vec{r}^{\,b}_{gci}\right)}{\left\|\vec{x}_{t,i}\right\|}$$

$$y_t=\begin{pmatrix}0\\0\\1\end{pmatrix}\left(\vec{x}_{a,i}-\vec{x}_g-\underline{R}\vec{r}^{\,b}_{gci}\right) \tag{25}$$

Therefore, the drag forces of cables on the *x* and *y* sides are as follows:

$$F_x=\frac{W_c}{\left(1\left\|\vec{x}_{t,i}\right\|\left\|\vec{x}_{t,i}\right\|^2\left\|\vec{x}_{t,i}\right\|^3\left\|\vec{x}_{t,i}\right\|^4\left\|\vec{x}_{t,i}\right\|^5\right)P_c\begin{bmatrix}1\\y_t\end{bmatrix}}\vec{x}_{th,i}$$

$$F_y=W_c\sqrt{\left(\frac{2}{\left(1\left\|\vec{x}_{t,i}\right\|\left\|\vec{x}_{t,i}\right\|^2\left\|\vec{x}_{t,i}\right\|^3\left\|\vec{x}_{t,i}\right\|^4\left\|\vec{x}_{t,i}\right\|^5\right)P_c\begin{bmatrix}1\\y_t\end{bmatrix}}+y_t\right)y_t\begin{bmatrix}0\\0\\-1\end{bmatrix}} \tag{26}$$

with W_c and P_c as the weight of the cable and the 6×2 known constant, respectively.

The FOWT calculation of this paper is based on a model in the National Energy Laboratory. The proposed turbine concept contains three fluctuating cylinders and a central control cylinder. The structure's properties are as Table 1.

The system aims to achieve uniform energy by tracking wind direction, utilizing a controlled trajectory, and maintaining consistent aerodynamic power. The actuator constraint ensures seamless implementation and minimizes instantaneous power changes:

$$\delta P(t)=\frac{\partial P(t)}{\partial x}\delta x+\frac{\partial P(t)}{\partial u}\delta u+\frac{\partial P(t)}{\partial v}\delta v=0 \tag{27}$$

where the power $P(t)$ is obtained from Equation (22). Owing to the low dependency of captured power to state variables variation, by ignoring the state section in Equation (27) and inserting the relevant inputs,

$$\frac{\partial P(t)}{\partial \beta}\delta\beta+\frac{\partial P(t)}{\partial \gamma}\delta\gamma+\frac{\partial P(t)}{\partial v}\delta v=0 \tag{28}$$

the variational quantity of β angle as the control goal is:

$$\delta\beta=-\left(\frac{\partial P(t)}{\partial \gamma}\delta\gamma+\frac{\partial P(t)}{\partial v}\delta v\right)/\left(\frac{\partial P(t)}{\partial \beta}\right) \tag{29}$$

The final control input involves adjusting generator torque based on changes in generator speed, with the strategy being to maintain a consistent rotor speed:

$$\dot{\omega}_r=\frac{1}{J_r}\left(\frac{P}{\omega_r}-\left(N_{GR}\right)T_g\right)=0 \tag{30}$$

which gives:

$$\delta T_g=\frac{1}{N_{GR}\omega_r}\left(\frac{\partial P(t)}{\partial \beta}\delta\beta+\frac{\partial P(t)}{\partial \gamma}\delta\gamma+\frac{\partial P(t)}{\partial v}\delta v\right) \tag{31}$$

With solving Equation (31), the generator torque is computed.

3. Controller and observer design of FOWT

3.1. Drafting the black-box nonlinear observer

Assuming a group of multi-input multi-output (MIMO) nonlinear uncertain dynamical system characterized by:

$$\ddot{x}=h\left(x,\dot{x}\right)+G\left(x,\dot{x}\right)u \tag{32}$$

where $x(t)\in\Re^n$ denotes the states, $u(t)\in\Re^r$ is the control action, and $h(x,\dot{x}):\Re^n\times\Re^n\to\Re^n$ and $G(x,\dot{x}):\Re^n\times\Re^r\to\Re^{n\times r}$ are indeterminate nonlinear functions.

The following assumptions will be required in the proof section.

Assumption 1: $h(x,\dot{x})$ and $G(x,\dot{x})$ are made of C^2 functions.

Assumption 2: The control input $u(t)$ includes C^2 functions and $u(t),\dot{u}(t)\in\ell_\infty$.

Assumption 3: The states are restricted permanently, that is, $x(t),\dot{x}(t)\in\ell_\infty$.

The assumptions above are reasonable in the outline of state observation in nonlinear systems. Our purpose is to propose an observer to approximate the unmeasurable derivative signals $\dot{x}(t)$ only based on the position/orientation measurements. Precisely, let $\hat{x}_1\in\Re^n$ indicate the estimated surge, sway, heave, roll, pitch, and yaw, respectively, and $\hat{x}_2\in\Re^n$ the corresponding derivative of the mentioned state variables. Also, the estimation errors $e(t),\dot{e}(t)\in\Re^n$, respectively, are described by:

$$\begin{aligned} e&=\hat{x}_1-x\\ \dot{e}&=\hat{x}_2-\dot{x} \end{aligned} \tag{33}$$

Therefore, we plan to guarantee that $e(t)\to 0$ and $\dot{e}(t)\to 0$ as $t\to\infty$ only through quantities of $x(t)$.

3.1.1. Observer Formulation

We suggest the subsequent nonlinear observer to explain the above-mentioned dynamical system:

$$\begin{aligned} \dot{\hat{x}}_1&=\hat{x}_2\\ \dot{\hat{x}}_2&=-K_0\mathrm{sgn}\left(e\right)-\left(K_1+I_n\right)\hat{x}_2-K_2e \end{aligned} \tag{34}$$

with $K_0,K_1,K_2\in\Re^{n\times n}$ being diagonal constant matrices, $K_1<K_2$, I_n denoting the $n\times n$ unit matrix, and sgn(·) being well-defined as following:

$$\begin{aligned} sgn(\xi)&=\left[\begin{array}{ccc} sgn(\xi_1) & sgn(\xi_2) & \ldots sgn(\xi_n) \end{array}\right]^T\\ \forall\xi&=\left[\begin{array}{ccc} \xi_1 & \xi_2 & \ldots\xi_n \end{array}\right]^T \end{aligned} \tag{35}$$

where sgn(·) is the typical signum function.

To obtain the error dynamics, we take the derivative of Equation (33) and replace the equivalent of variables:

$$\ddot{e}=-K_0\mathrm{sgn}\left(e\right)-\left(K_1+I_n\right)\hat{x}_2-K_2e-\ddot{x} \tag{36}$$

Assuming the signal $r\in\Re^n$ be declared as follows.

$$r=\dot{e}+e \tag{37}$$

After the time derivative of Equation (37) and replacing Equations (32) and (36), we have

$$\dot{r}=-K_0\mathrm{sgn}(e)-(K_1+I_n)\hat{\dot{x}}_2-K_2e-h(x,\dot{x})-G(x,\dot{x})u+\dot{e} \tag{38}$$

Considering Equation (33), Equation (34) yields to

$$\dot{r}=N_o(x,\dot{x},u)-K_0\mathrm{sgn}(e)-K_1r-(K_2-K_1)e \tag{39}$$

in which

$$N_o(x,\dot{x},u)=-h(x,\dot{x})-G(x,\dot{x})u-(K_1+I_n)\dot{x} \tag{40}$$

Returning to Assumptions 1-3, we conclude that $N_o(x,\dot{x},u),\dot{N}_o(x,\dot{x},u)\in\ell_\infty$.

3.1.2. Globally Asymptotic Convergence Analysis

Lemma 1: $L(t)\in\Re$ as an auxiliary function is defined as follows:

$$L=r^T\left(N_o-K_0\mathrm{sgn}(e)\right) \tag{41}$$

If the matrix K_0, presented in Equation (34), is designated to fulfill the following acceptable condition:

$$K_{0i}>\left\|N_{oi}(x,\dot{x},u)\right\|_\infty+\left\|\dot{N}_{oi}(x,\dot{x},u)\right\|_\infty \tag{42}$$

with the indices $i=1,2,\ldots,n$ being the i-th component of the vector or diagonal matrix and $\|\cdot\|_\infty$ signifies the ℓ_∞ norm, therefore

$$\int_0^t L(\tau)d\tau,,\ \xi_o \tag{43}$$

where the positive constant ξ_o is described as

$$\xi_o=\sum_{i=1}^{n}K_{0i}e_i(0)-e^T(0)N_o(0) \tag{44}$$

Proof: After replacing Equation (37) with Equation (41) and then integrating in time,

$$\begin{aligned}\int_0^t L(\tau)d\tau=&\int_0^t e^T(\tau)\left(N_o(x,\dot{x},u,\tau)-K_0sgn(e)\right)d\tau\\&+\int_0^t\frac{d\left(e^T(\tau)\right)}{d\tau}N_o(x,\dot{x},u,\tau)d\tau\\&-\int_0^t\frac{d\left(e^T(\tau)\right)}{d\tau}K_0sgn(e)\,d\tau\end{aligned} \tag{45}$$

Integrating the second element on the right side of Equation (45), we have

$$\begin{aligned}\int_0^t L(\tau)d\tau=&\int_0^t e^T(\tau)\left(N_o(x,\dot{x},u,\tau)-K_0sgn(e)\right)d\tau+e^T(\tau)N_o(x,\dot{x},u,\tau)\Big|_0^t\\&-\int_0^t e^T(\tau)\frac{d\left(N_o(x,\dot{x},u,\tau)\right)}{d\tau}d\tau-\sum_{i=1}^{n}K_{0i}\left|e_i(\tau)\right|\Big|_0^t\\=&\int_0^t e^T(\tau)\left(N_o(x,\dot{x},u)-\frac{d\left(N_o(x,\dot{x},u)\right)}{d\tau}\right)d\tau-\int_0^t\left(\sum_{i=1}^{n}K_{0i}\left|e_i(\tau)\right|\right)d\tau\\&+e^T(t)N_o(x,\dot{x},u)-e^T(0)N_o(0)-\sum_{i=1}^{n}K_{0i}\left|e_i(t)\right|+\sum_{i=1}^{n}K_{0i}\left|e_i(0)\right|\end{aligned} \tag{46}$$

Now, we obtain the highest amount of the right-hand side of (46) as follows:

$$\int_0^t L(\tau)d\tau,, \int_0^t |e(\tau)|\left(|N_o(x,\dot{x},u,\tau)|+\left|\frac{d(N_o(x,\dot{x},u,\tau))}{d\tau}\right|-K_0\right)d\tau$$
$$+\sum_{i=1}^{n}|e_i(t)|\left(|N_{oi}(x,\dot{x},u)|-K_{0i}\right)+\sum_{i=1}^{n}K_{0i}|e_i(0)|-e^T(0)N_o(0) \quad (47)$$

From Equation (47), it is concluded if K_0 is selected based on Equation (42), so Equation (43) is true. This ends the proof.

Now, the major outcome of the current paper is expressed in the subsequent Theorem.

Theorem 1: The derivative of observer defined by Equation (34) provokes global asymptotic alignment of $e(t)$ and $\dot{e}(t)$, that is, $e(t)\to 0$ and $\dot{e}(t)\to 0$ as $t\to\infty$, as long as the matrices $K_2 > K_1$ and K_0 is chosen to hold the satisfactory condition (42).

Proof: Let the supplementary function $P_o(t)\in\Re$ be described as:

$$P_o(t)=\xi_o-\int_0^t L(\tau)d\tau \quad (48)$$

where ξ_o and $L(t)$ are expressed in prior Lemma. From Lemma 1, it is conceivable to conclude $P_o(t)..0$. Therefore, we express the subsequent function $V_o(t,y):\Re_+\times\Re^{2n}\times\Re_+\to\Re_+$:

$$V_o=\frac{1}{2}r^Tr+\frac{1}{2}e^T(K_2-K_1)e+P_o \quad (49)$$

where $y=\begin{bmatrix}z^T & \sqrt{P_o}\end{bmatrix}^T$ with $z=\begin{bmatrix}r^T & e^T\end{bmatrix}^T$. V_o is a positive-definite Lyapunov function in terms of $e,\dot{e},\sqrt{P_o}$.

The time derivative of Equation (49), along with replacement from Equation (41) and the derivative of Equation (48), results in:

$$\dot{V}_o=r^T\left\{N_o-K_0\mathrm{sgn}(e)\ -K_1r-(K_2-K_1)e\right\}+$$
$$\dot{e}^T(K_2-K_1)e-r^T\left\{N_o-K_0\mathrm{sgn}(e)\right\} \quad (50)$$

Applying Equation (39) to Equation (50), we have

$$\dot{V}_o=-r^TK_1r-e^T(K_2-K_1)e \quad (51)$$

Henceforth, $V_o(t)$ is a positive-definite Lyapunov function which brings about the negative semidefinite time derivative $\dot{V}_o(t)$. Since $\dot{V}_o(t)\equiv 0$ yields to $r\equiv 0$ and $e\equiv 0$, regarding Equation (33), we have $\dot{e}\equiv 0$. Besides, considering Equations (41) and (44), this paper yields $L\equiv 0$ and $\xi_o=0$, that is, $\sqrt{P_o}\equiv 0$. Through LaSalle's theorem, we conclude $e(t)\to 0$ and $\dot{e}(t)\to 0$ as $t\to\infty$. This section finishes the proof.

3.2. Implementation of DRL

DRL is a kind of data-oriented approach working with Markov Decision Process (MDP). Fig. 5 illustrates the design where an agent experiences an observed state $\mathbf{s_t}$ and a reward $\mathbf{r}$ so that it takes action $\mathbf{a_t}$ in a situation, with the purpose of catching the cumulative reward $\sum\mathbf{r}$ during the time.

Section 3.1 describes the estimation of FOWT states using an observer, which is then fed into a designed DRL, which based on reward, ensures structure stability and global stability.

The training technique of the agent is the DQN, a black-box approach. The agent comprises two

neural networks, so-called Q-online, Q_{ϑ}, and Q-target, $Q'_{\vartheta'}$. The Q_{ϑ} network approximates the maximum Q-values of the observation $\boldsymbol{s_t}$, considering the possible actions. On the other hand, $Q'_{\vartheta'}$ computes the maximum Q-values of the observation $\mathbf{s_{t+1}}$ based on the same possible actions. Consequently, main target value is computed as follows.

$$y_j = \begin{cases} r_j & \text{stops at j+1} \\ r_j + \gamma \max_{a'} \mathbf{Q}'_{\vartheta'}\left(s_{j+1}, a; \vartheta'\right) & \text{otherwise} \end{cases} \tag{52}$$

where r_j denotes reward, γ shows the discount factor and ϑ represents the weights of Q-online network. The reward is heuristically defined as:

$$r = \begin{cases} r_1, & |x| \le x_{lim} \\ r_2, & |x| > x_{lim} \end{cases} \tag{53}$$

where x_{lim} is the extreme permissible oscillation for the tower throughout the simulation task, while r_1 and r_2 are:

$$r_1 = \left(A_r \left|x\right|^2 + B_r \left|\theta\right|^2 + C_r \left|\mathrm{u}\right|^2 \right) D_r \tag{54}$$

$$r_2 = r_1 + E_r \tag{55}$$

The optimal action in DRL involves choosing parameters like $A_r = 10^{-1}, B_r = 10^{-2}, C_r = 50, D_r = 10^{-2}, E_r = -10^2$, with rewards and punishments to minimize error. The choice of multiplier depends on the problem's characteristics and tracking performance. Intuition helps determine optimal parameters, but lower multipliers result in slower convergence, while higher multipliers have the opposite effect. The reward function's second element is utilized to implement punishments.

The logic-based reward function provides a clear, easily understood method for assigning rewards or penalties to agents based on defined thresholds, allowing for easier interpretation and debugging of their behavior, ensuring focus on crucial task elements [58].

We addressed the sparsity issue in our DQN algorithm by:

Reward Shaping: Intermediate rewards and continuous penalties.

Hindsight Experience Replay (HER): Learning from near misses.

Intrinsic Rewards: Incentives for exploration.

Pre-Training with Expert Data: Bootstrapping with expert actions.

Supplementary Tasks: Additional auxiliary tasks for feedback.

Multi-step Returns: Efficient distribution of future rewards.

These strategies ensure effective learning and desired control policy.

The DRL initializes its work with arbitrary weights. Next, the agent reacts in the environment to save information in an experience range of dimension D. Each element within the spectrum of the encounter encompasses four distinct components $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')$, where $\mathbf{s}'$ represents the system's observed state after acting $\mathrm{a_t}$. Therefore, to get the optimal Q-value, the Mean Squared Error (MSE) will be minimalized:

$$L_{MSE} = \left(y_j - Q\left(s_j, a_j; \vartheta\right) \right)^2 \tag{56}$$

By backpropagating the derivative of Equation (56) concerning the weights, the update of weights in Q-online is carried out. However, to avoid the divergence in update phase after some epochs, the update is executed for weights in Q-target with application of Polyak-Ruppert Averaging with the

parameter τ as [59]:

$$\vartheta'^{(t)} = \tau\ \vartheta'^{(t-1)} + (1-\tau)\vartheta^{(t)} \tag{57}$$

This procedure is reiterated until the ideal policy is found. The explained process is shown in Fig. 6 for clarification purposes and in Table 2 as a pseudo code for the coding aim.

Fig. 6 carries three nominated actions. This figure actually records a time interval, as action values a_1, a_2, and a_3 indicate distinct steps when the agent can perform. These steps are derived from a stochastic policy rather than any just three consecutive actions in real time. Put another way, every action corresponds to one point of operational control decision: something that an agent might do at a certain state. The distinction between these actions is critical for our understanding of the agent's decision-making process and resultant Q-values.

Deep Q-Networks (DQN) use Q-values to evaluate potential actions within various states, guiding decision-making and balancing exploration and exploitation. The network computes Q-values for possible actions $Q(s,a_1)$, $Q(s,a_2)$, and $Q(s,a_3)$, where s is the state and a_1, a_2, and a_3 are actions. These Q-values serve three main purposes:

1. Action Evaluation: The network assesses potential rewards for each action, helping to select the most profitable one.

2. Improving Policy: Q-learning refines the agent's decision-making by updating the relationship between states and actions, thereby improving the policy over time.

3. Balancing Exploration and Exploitation: The agent must balance exploring new actions and exploiting known strategies to optimize performance in uncertain environments.

The DQN architecture typically includes an input layer, multiple hidden layers, and an output layer. The input layer captures the state representation, hidden layers derive hierarchical features, and the output layer generates Q-values for possible actions. This architecture is distinct in that it starts with fewer neurons in early layers, increasing in deeper layers, which is particularly effective for controlling complex systems like Floating Offshore Wind Turbines (FOWTs).

Key considerations in this DQN architecture include:

1. Handling Nonlinearities and Disturbances: FOWTs operate in nonlinear environments with significant disturbances. Starting with fewer neurons and gradually increasing helps manage these complexities (Table 3).

2. Avoiding Overfitting: Beginning with fewer neurons prevents overfitting in high-dimensional spaces with limited data, while deeper layers accommodate complex features.

3. Empirical Success in DRL Applications: Studies ([60, 61]) show that customized neural networks improve performance in dynamic environments, supporting flexible structures for FOWT control ([62, 63]).

4. Experimental Validation: MATLAB simulations confirm that the proposed DQN architecture outperforms traditional methods like LQR and Luenberger observers under uncertainty.

The DQN's design, including neuron counts and connections, directly impacts the number of

weights in the network. Hyperparameters like learning rates, activation functions, and exploration techniques are crucial for optimizing DQN performance, often requiring extensive tuning. The training process involves iteratively updating network weights based on the temporal difference error between predicted and target Q-values, refining the control policy over time.

Computational complexity is a significant consideration in DQN development, driven by factors like network architecture and dataset size. To mitigate these challenges, optimization techniques such as mini-batch training, parameter sharing, and distributed computing are used. Additionally, hardware accelerators like GPUs can greatly reduce training time by leveraging parallel processing capabilities. Careful consideration of these factors is essential for the practical application of DQNs in real-world scenarios.

Table 3 shows the numerical computing costs based on common scenarios and findings in the field of DRL for FOWT.

3.2.1. Stability analysis

The transition from the current state s_t to the next state s_{t+1} is characterized by the probability $\mathrm{P}(s_{t+1}|s_t,a_t)$, where a_t is the action chosen by the controller from various possible actions. The value of an action can be determined based on the cost function $c(s_t,a_t)=\mathrm{E}_{P(\cdot|s_t,a_t)}s_{t+1}$. Stability of the stochastic system is ensured if $\lim_{t\to\infty}\ \mathrm{E}_{s_t}c_\pi(s_t)=0$ for any initial state s_0. The likelihood of transitioning to the next state is given by $P_\pi(s')=\int_A \pi(a|s)P(s'|s,a)da$. The state distribution at time t, denoted as $\mathrm{P}(s|\rho,\pi,t)$, is recursively defined by $\mathrm{P}(s'|\rho,\pi,t+1)=\int_S P_\pi(s'|s)\mathrm{P}(s|\rho,\pi,t)\,\mathrm{d}a,\forall t\in Z_+$, with $P(s|\rho,\pi,0)=\rho(s)$. Assuming an ergodic policy π with steady-state distribution $q_\pi(s)=\lim_{t\to\infty}P(s|\rho,\pi,t)$, the Region of Attraction (ROA) is the set of initial states s_0 that lead the system to stabilization. Convergence to equilibrium is guaranteed if the system starts within the ROA.

Theorem 2: The casual system is well-defined steady under mean lost definition if a function $L:S\to\mathbb{R}_+$ and non-negative coefficients β_1, β_2 and β_3 are available,

$$\begin{aligned}&\beta_1 c_\pi(s)\le L(s)\le\beta_2 c_\pi(s)\\&\mathrm{E}_{s\sim\mu_\pi}\left(\mathrm{E}_{s'\sim P_\pi}L(s')-L(s)\right)\le-\beta_3\mathrm{E}_{s\sim\mu_\pi}c_\pi(s)\end{aligned}\tag{58}$$

where,

$$\mu_\pi(s)=\lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N}P(s_t=s|\rho,\pi,t)\tag{59}$$

is the unrestricted distribution.

Proof: If the example distribution sequence $\{P(s|\rho,\pi,t),t\in Z_+\}$ reaches $q_\pi(s)$ when t goes to infinity, then according to Abelian theorem, the set $\left\{\frac{1}{N}\sum_{t=0}^{N}P(\rho,\pi,t),N\in Z_+\right\}$ also converges and $\mu_\pi(a)=q_\pi(s)$. Integrated with the form of μ_π, Equation (59) accomplishes that first, on the left-hand-side, $L(s)\le\beta_2 c_\pi(s)$ for all $\beta\in Z$ based on Equation (58). Since the probability density function $P(s|\rho,\pi,t)$ is a restricted function over S for all t, consequently a factor M is obtainable such that

$$P\left(s|\rho,\pi,t\right)L\left(s\right)\le M\beta_2 c_\pi\left(a\right),\forall s\in S,\forall t\in Z_+ \tag{60}$$

Second, the series $\left\{\frac{1}{N}\sum_{t=0}^{N} P(s|\rho,\pi,t)L(s),N\in Z_+\right\}$ approaches element-wise to $q_\pi(s)L(s)$. Considering the Lebesgue's theorem [59], it offers the convergency of a set $f_n(s)$ element-wise to f defining with some integrable function g such that,

$$\begin{aligned}&\left|f_n\left(s\right)\right|\le g\left(s\right),\forall s\in \mathrm{S},\\&\forall n\lim_{n\to\infty}\int_s f_n\left(s\right)ds=\int_S\lim_{n\to+\infty} f_n\left(s\right)\mathrm{ds}\end{aligned} \tag{61}$$

Consequently, the left side of Equation (61) is written:

$$\begin{aligned}&\int_{\mathrm{S}}\lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N}P\left(s|\rho,\pi,t\right)\int_{\mathrm{S}}\left(P_\pi(s'|s)L(s')ds'-L\left(s\right)\right)ds\\&\lim_{N\to\infty}\frac{1}{N}\left(\sum_{t=1}^{N+1}\mathrm{E}_{P(s|\rho,\pi,t)}L\left(s\right)-\sum_{t=0}^{N}\mathrm{E}_{P(s|\rho,\pi,t)}L\left(s\right)\right)\\&\lim_{N\to\infty}\frac{1}{N}\left(\mathrm{E}_{P(s|\rho,\pi,N+1)}L\left(s\right)-\mathrm{E}_{\rho(s)}L\left(s\right)\right)\end{aligned} \tag{62}$$

Hence, considering above-mentioned relations, Equation (62) supposes

$$\begin{aligned}&\lim_{N\to\infty}\frac{1}{N}\left(\mathrm{E}_{P(s|\rho,\pi,N+1)}L\left(s\right)-\mathrm{E}_{\rho(s)}L\left(s\right)\right)\\&\le -\beta_3\lim_{t\to\infty}\mathrm{E}_{P(s|\rho,\pi,t)}c_\pi\left(s\right)\end{aligned} \tag{63}$$

Since $\mathrm{E}_{\rho(s)}L(s)$ is a limited quantity and L is non-negative definite, it leads to,

$$\lim_{t\to\infty}\mathrm{E}_{P(s|\rho,\pi,t)}c_\pi\left(s\right)\le\lim_{N\to\infty}\frac{1}{N}\left(\frac{1}{\beta_3}\mathrm{E}_{\rho(s)}L\left(s\right)\right)=0 \tag{64}$$

Supposing a state $s_0\in\left\{s_0|c_\pi\left(s_0\right)\le b\right\}$ and a positive d are available such that $\lim_{t\to\infty}\mathrm{E}_{P(s|s_0,\pi,t)}c_\pi(s)=d$ or $\lim_{t\to\infty}\mathrm{E}_{P(s|s_0,\pi,t)}c_\pi(s)=\infty$. Since $\rho\left(s_0\right)>0$ for all initial states in $\left\{s_0|c_\pi\left(s_0\right)\le b\right\}$. It follows that $\lim_{t\to\infty}\mathrm{E}_{s_t\sim P(.|\pi,\rho)}c_\pi\left(s_t\right)>0$ which is inconsistent with Equation (64). Thus $\forall s_0\in\left\{s_0|c_\pi\left(s_0\right)\le b\right\}$, $\lim_{t\to\infty}\mathrm{E}_{P(s|s_0,\pi,t)}c_\pi(s)=0$. So, the system is steady in mean lost.

In each sea state, the DRL algorithm adapts to environmental conditions to identify the optimal input actions necessary for consistent energy harnessing from FOWTs. It adjusts the FOWT's position to face the wind, ensures the generator operates at the desired shaft speed, and aligns blade angles with the wind flow. Actions, executed by the yaw motor, generator circuit, and pitch motor units, are periodically reassessed. Rewards accumulate based on control inputs and absolute values of translation and orientation Eqs. (54) and (55). The algorithm computes the average input action over a specified horizon H within a wave cycle where state s_n and action a_{n-1} vary, and then determines a new action a_n by instantaneously modifying the state to s_{n+1}.

State Space: As previously mentioned, the situational factors are expected to represent the position, orientation and their corresponding derivatives so that the assumed DRL state space is:

$$S=\left\{s|s_{j,k,l}=x_{s,j}+\theta_{z,k}+[\dot{x};\dot{\theta}]_l\ \begin{matrix}j=1:J,\\k=1:K,\\l=1:L\end{matrix}\right\} \tag{65}$$

To prevent overfitting, balance data quantity with state representation, considering J, K from FOWT configuration, and L as discrete time sequence values.

Action Space: The action series comprises three levels determined by the chosen state space, as outlined below:

$$A=\left\{a\left|\left(\gamma_{yaw}, T_{generator}, \theta_{pitch}\right)\right.\right\} \tag{66}$$

Equivalent extreme states, i.e., x_{max} and θ_{max}, takes certain measures to prevent the controller from exceeding the limits of the state space.

Reward: In DRL, the reward function aims to maximize performance by giving positive rewards for correct actions and negative rewards for incorrect ones, focusing on state and control actions. Therefore, in the context of the black-box nonlinear observer-oriented DRL control system for FOWT, the compensation function is deemed relevant in relation to the absolute values of states and control actions.

In conclusion, there exist $n_s = J \times K \times L$ states, number of possible combination actions out of Equation (66), which are chosen via Equations (27) through (29), and accumulated rewards according to Equations (54) and (55). The block diagram of the aforementioned plan is seen in Fig. 7. Also, the accumulated reward during training is shown in Fig. 8, which demonstrates the progress of the training process typically and finally takes the maximum reward.

4. Numerical Results and Discussions

Along with the Maximum Energy Tracking controller with nonlinear model, the NREL 5-MW controller defines the gain-scheduling PI controller for FOWT as

$$k_P\left(v\right)=\frac{2J\omega_{r,rat}\zeta_{\mathrm{des}}\omega_{\mathrm{des}}}{N_g\dfrac{\partial P}{\partial\beta}\left(v\right)}$$

and

$$k_I\left(v\right)=\frac{J\omega_{r,rat}\omega^2{}_{\mathrm{des}}}{N_g\dfrac{\partial P}{\partial\beta}\left(v\right)},$$

In this equation, J represents rotor inertia, $\omega_{r,\mathrm{rat}}$ is the rated rotor speed, and N_g is the gearbox ratio. Parameters ζ_{des} and ω_{des} are user-configurable. The term $\partial P/\partial\beta(v)$ indicates wind-speed dependent sensitivity. Fig. 9 shows MATLAB simulations comparing the suggested nonlinear controller with a gain-scheduling PI controller, highlighting minimal differences from NREL's work and validating the proposed strategy.

The proposed adaptive DRL controller outperforms the gain scheduling PI controller in several key areas:

1. Adaptability: The DRL controller flexibly adapts to changing conditions and disturbances, unlike the gain scheduling PI controller, which relies on predetermined gain values, limiting real-time adjustments crucial for offshore floating wind farms.

2. Power Regulation and Stability: DRL maintains stable power output and overall system stability, outperforming the PI controller, particularly under disturbances.

3. Handling Complexity: DRL excels in controlling nonlinear and complex systems like FOWTs, improving performance through its online-target network structure.

4. Robustness: DRL shows superior performance under uncertainties and disturbances, surpassing traditional methods like the LQR and Luenberger observer.

The simulation process of observer is initiated with the deliberate introduction of random initial values for the system states as defined in Equation (34). The numerical values assigned to the system's gain parameters, denoted as K_0, K_1, and K_2, are carefully selected, taking into account certain predefined assumptions and considerations. The amounts of these parameters are as follows.

$$K_0 = \begin{bmatrix} 0.537 & -0.433 & 0.725 & 1.409 & 0.488 & 0.888 \\ 1.833 & 0.342 & -0.063 & 1.417 & 1.034 & -1.147 \\ -2.258 & 3.578 & 0.714 & 0.671 & 0.726 & -1.068 \\ 0.862 & 2.769 & -0.204 & -1.207 & -0.303 & -0.809 \\ 0.318 & -1.349 & -0.124 & 0.717 & 0.293 & -2.944 \\ -1.307 & 3.034 & 1.489 & 1.630 & -0.787 & 1.438 \end{bmatrix}$$

$$K_1 = \begin{bmatrix} 0.325 & 0.319 & 1.093 & -0.006 & -1.089 & -1.491 \\ -0.754 & 0.312 & 1.109 & 1.532 & 0.032 & -0.742 \\ 1.370 & -0.864 & -0.863 & -0.769 & 0.552 & -1.061 \\ -1.711 & -0.030 & 0.077 & 0.371 & 1.100 & 2.350 \\ -0.102 & -0.164 & -1.214 & -0.225 & 1.544 & -0.615 \\ -0.241 & 0.627 & -1.113 & 1.117 & 0.085 & 0.748 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -0.384 & -0.354 & -1.608 & -2.295 & -0.164 & 0.200 \\ 1.777 & -0.392 & 1.393 & 0.209 & -3.866 & -1.089 \\ -1.529 & 2.838 & 1.670 & 1.444 & -0.877 & 0.607 \\ -2.804 & 0.583 & -0.487 & 5.170 & -3.589 & -1.200 \\ -2.844 & 0.395 & 0.431 & -1.333 & 1.680 & 0.979 \\ 0.976 & 3.175 & -2.331 & 0.374 & -1.776 & 1.478 \end{bmatrix} \quad (67)$$

The exact gain levels are crucial for optimal control, ensuring accurate state tracking by minimizing errors between measured and estimated outputs (Figs. 10 and 11). The system effectively reduces estimation errors to zero, although convergence times vary across states. The results confirm the observer's effectiveness, robustness, and reliability, emphasizing the practical benefits of the proposed control technique in real-world applications.

4.1. LQR controller

Optimal control seeks strategies to find the best solutions by optimizing a performance index while adhering to constraints.

In order to determine the equilibrium point, the state variables on the right-hand side of Equation (1) are equated to zero:

$$\begin{aligned} x=\{&3.90m, 1.76m, -9.91m \\ &-0.50^{\circ}, 1.60^{\circ}, -0.00^{\circ} \\ &0.20^{\circ} \\ &0, \cdots, 0 \\ &12.1rpm, 1173.8rpm\} \end{aligned} \quad (68)$$

Linearizing Equation (1) around the equilibrium point Equation (68) results in the derivation of the linear state space model:

$$\dot{x}=Ax+Bu \tag{69}$$

Given the LQR cost function upon the model (69):

$$J=\frac{1}{2}\left(X_f^T S_f X_f\right)+\frac{1}{2}\int_{t_0}^{t_f}\left(X^T QX+U^T RU\right)dt$$
$$S_f, Q\geq 0, R>0 \tag{70}$$

leads to the optimum state feedback control action $u=-kx$ and thus,

$$\dot{x}=(A-Bk)x \tag{71}$$

Within the wind turbine facility discussed in this article, the zero end-point weight denoted as S_f is responsible for producing the subsequent basic configuration:

$$J=\int\left(x^T Qx+u^T Ru\right)dt \tag{72}$$

The specified weighting matrices R and Q are employed to determine the limitations of actuator size and cost.

$$R=\begin{bmatrix}100 & & \\ & 100 & \\ & & 100\end{bmatrix},\quad Q=\begin{bmatrix}1 & & \\ & 1 & \\ & & 1\end{bmatrix} \tag{73}$$

The observation matrix c is derived based on the configuration of the output measurement system.:

$$y=cx=\begin{bmatrix}x_1 & x_2 & x_3 & x_4 & x_5 & x_6\end{bmatrix}^T \tag{74}$$

The Kalman gain matrix k is derived through a specific computational process:

$$k=R^{-1}B^T P \tag{75}$$

The positive definite matrix P is generated by finding the response of the algebraic Riccati matrix problem as follows:

$$A^T P+PA-PBR^{-1}B^T P+Q=0 \tag{76}$$

The outcomes of implementing the previously discussed LQR on FOWT are illustrated in Figs. 12-17.

To compare and assess the impact of measurement noise on the DRL-based control system's performance, white Gaussian noise with various Signal-to-Noise Ratios (SNRs) is introduced. This tests the DRL controller's resilience and accuracy amidst real-world noise. The performance of the DRL controller is contrasted with the standard LQR controller under minimal noise conditions. Figs. 12–17 illustrate how noise affects the tracking errors of the DRL system compared to noise-free conditions and the LQR controller. These figures provide insight into the DRL system's robustness and effectiveness, highlighting its advantages in noisy environments and demonstrating its real-world applicability.

As shown in Figs. 12-17, even in settings with a significant amount of noise, particularly at the lowest SNR, the DRL-based controller successfully regulates noise and mitigates its negative effects. In striking contrast, traditional LQR-based controllers struggle to sustain performance in the face of measurement noise, resulting in considerable differences when compared to the DRL system working in noisy environments. In summary, these data demonstrate the traditional controller's failure to adequately manage these unwanted phenomena, resulting in a significant decrease of system

responsiveness. As a result, the deterioration of system states, which amounts to an extremely small 0.1% tracking error in compared to the noise-free scenario, emphasizes the superiority of the DRL-based controller as the best control system for FOWTs. Notably, the successful control system achieves the required equilibrium point in a short amount of time, usually about 7 seconds. This speedy convergence and ability to sustain accurate control demonstrate the proposed DRL-based controller's speed and efficacy when compared to traditional alternatives. Table 4 contains quantitative data and statistical studies that indicate the system's resilience under different noise levels and disturbance situations.

4.2. Luenberger observer

The Luenberger observer estimates unmeasured states in dynamical systems using available measurements when not all states are observable. A system's dynamics can be described using state-space equations:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{77}$$

Here, x is the state vector, u is the input vector, y is the output vector, A, B, C, and D are the system's matrices, and $\dot{x}$ is the derivative of x relevant to time. The Luenberger observer takes the following form:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{78}$$

The estimated state vector is $\hat{x}$, the observer matrix is L, and the measured output is y.

The observer gain matrix, L, is intended to reduce the error between the real system state (x) and the predicted state. The algebraic Riccati equation is frequently solved during the construction of an L.

The algebraic Riccati equation of the observer matrix L is given by:

$$A^T P + PA - PC^T CP + Q = 0 \tag{79}$$

P solves the Riccati equation, Q is a positive definite matrix for desired observer performance. The Luenberger observer's impact on FOWT is shown in Figs. 18 and 19.

Our investigation reveals several limitations of the Luenberger observer despite its benefits:

1. Sensitivity to Model Mismatches: Errors arise if the system model differs from reality.

2. Limited Applicability to Nonlinear Systems: Designed for linear systems, requiring linearization for nonlinear systems, which can introduce errors.

3. Noisy Output: Susceptible to inaccuracies from measurement noise.

4. Convergence and Stability Issues: Stability and convergence depend on proper observer gain selection.

5. Initial State Estimation: Initial estimate affects convergence time and accuracy.

6. Limited Information from Outputs: States not directly observable from outputs may be inaccurately estimated.

7. Computation Complexity: Implementation can be computationally intensive, especially in real-time applications.

8. Design and Tuning Challenges: Difficulties in selecting appropriate observer gains.

9. Robustness: Lacks inherent resilience to disturbances; may require alternative approaches for improved robustness.

Despite these drawbacks, Luenberger observers are widely used for their practical state estimation capabilities in control engineering.

5. Conclusion and final remarks

The application of Deep Reinforcement Learning (DRL) to Floating Offshore Wind Turbines (FOWTs) introduces a novel approach in renewable energy research. FOWTs are ideal for deep waters where fixed-bottom turbines are impractical, but optimizing control algorithms to maximize energy and maintain stability is challenging. DRL, a machine learning method, addresses this by dynamically adjusting the turbine's pitch (θ) and yaw (ψ) angles based on changing wind conditions and wave dynamics.

In this study, DRL was employed to develop an intelligent control system for FOWTs. Data from simulations and real-world prototypes were used to train and validate the DRL model, which resulted in significant performance improvements:

1. Increased Energy Capture: DRL-controlled FOWTs showed higher energy capture compared to traditional fixed techniques, with increased power output (P) under varying wind conditions.

2. Adaptive Response: The DRL model adapted to environmental changes, optimizing turbine orientation (θ , ψ) for efficiency.

3. Enhanced Stability: The DRL control system improved stability by reducing the impact of waves (w) on the floating platform.

Additionally, a nonlinear observer was designed to reconstruct system derivatives using displacement (d) and orientation (θ) data, reducing the need for specialized modeling. The observer's global asymptotic convergence was verified using Lyapunov's method. The DRL-based control system showed superior performance compared to traditional methods like the LQR controller and gain-scheduling PI control, proving its effectiveness in managing uncertainties and nonlinearities. Future research will focus on the sensitivity of the DRL system and the effects of observer gains K_0, K_1, and K_2 on system response. The MATLAB simulations confirm the DRL system's robustness against noise and its practical application for FOWTs.

Credit author statement

Hadi Mohammadian KhalafAnsar and Jafar Keighobadi: Conceptualization, Methodology, Data testing, Writing, reviewing and editing – original draft. Mir Mohammad Ettefagh and Jafar Tanha reviewing and editing.

Declaration of competing interest

The authors declare no financial interests/personal relationships which may be considered potential competing interests.

Data availability

The data is accessible with request.

Acknowledgments

This work was supported by University of Tabriz.

Technical biography

Hadi Mohammadian KhalafAnsar received his M.Sc. degree in Mechanical Engineering from University of Tabriz, Iran. Since 2020, he has been working toward the Ph.D. degree with the Department of Mechanical Engineering, University of Tabriz. His research interests include Integrated deep learning, Deep reinforcement learning, Adaptive control, Neuro-fuzzy controller, and Floating wind turbine control.

Dr. Jafar Keighobadi received the Ph.D. degree in Mechanical Engineering and Control Systems from Amirkabir University of Technology, Iran, in 2008.

He is currently the Professor of Mechanical Engineering Department at University of Tabriz. His research interests include Artificial intelligence, Estimation and identification, Nonlinear robust control, and GNC.

Dr. Mir Mohammad Ettefagh received the Ph.D. degree in Dynamics &Vibration (Vibration Signal Processing & Condition

Monitoring/SHM) from University of Tabriz, Iran, in 2009.

He is currently the Associate Professor of Mechanical Engineering Department at University of Tabriz. His research interests include Dynamics, Vibration, Modal Analysis, Condition Monitoring, Structural Health Monitoring.

Dr. Jafar Tanha received the Ph.D. degree in Artificial Intelligence from University of Amsterdam, Netherland, in 2013.

He is currently the Assistant Professor of Computer and Electrical Engineering Department at University of Tabriz. His research interests include Advanced Machine Learning, Feature Selection, Classification, Unsupervised Learning, Computational Intelligence, Supervised Learning, Pattern Recognition, Pattern Classification, and Machine Learning.

References

1. Crespo, A. "Computational fluid dynamic models of wind turbine wakes," *Energies*, 16, p. 1772 (2023). DOI: https://doi.org/10.3390/en16041772.
2. Venkatraman, K., Moreau, S., Christophe, J., and Schram, C. "Numerical investigation of h-darrieus wind turbine aerodynamics at different tip speed ratios," *International Journal of Numerical Methods for Heat & Fluid Flow*, 33, pp. 1489–1512 (2023). DOI: https://doi.org/10.1108/hff-09-2022-0562.
3. KhalafAnsar, H.M., and Keighobadi, J. "Adaptive inverse deep reinforcement Lyapunov learning control for a floating wind turbine," *Scientia Iranica*, 0, 0–0 (2023). DOI: https://doi.org/10.24200/sci.2023.61871.7532.
4. Mohammadian KhalafAnsar, H., and Keighobadi, J. "Deep reinforcement learning with immersion- and invariance-based state observer control of wave energy converters," *International Journal of Engineering*, 37, pp. 1085–1097 (2024). DOI: https://doi.org/10.5829/ije.2024.37.06c.05.
5. Sierra-Garcia, J.E., Santos, M., and Pandit, R. "Wind turbine pitch reinforcement learning control improved by pid regulator and learning observer," *Engineering Applications of Artificial Intelligence*, 111, p. 104769 (2022). DOI: https://doi.org/10.1016/j.engappai.2022.104769.
6. Sierra-García, J.E., and Santos, M. "Performance analysis of a wind turbine pitch neurocontroller with unsupervised learning," *Complexity*, 2020, pp. 1–15 (2020). DOI: https://doi.org/10.1155/2020/4681767.
7. Hosseini, E., Aghadavoodi, E., and Fernández Ramírez, L.M. "Improving response of wind turbines by pitch angle controller based on gain-scheduled recurrent ANFIS type 2 with passive reinforcement learning," *Renewable Energy*, 157, pp. 897–910 (2020). DOI: https://doi.org/10.1016/j.renene.2020.05.060.
8. Tang, B., Chen, Y., Chen, Q., and Su, M. "Research on short-term wind power forecasting by data mining on historical wind resource," *Applied Sciences*, 10, p. 1295 (2020). DOI: https://doi.org/10.3390/app10041295.
9. Yao, W., Huang, P., and Jia, Z. "Multidimensional LSTM networks to predict wind speed," *37th Chinese Control Conference (CCC)* (2018). DOI: https://doi.org/10.23919/chicc.2018.8484017.
10. Gu, C., and Li, H. "Review on deep learning research and applications in wind and wave energy," *Energies*, 15, p. 1510 (2022). DOI: https://doi.org/10.3390/en15041510.
11. Liu, H., Mi, X., and Li, Y. "Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM," *Energy Conversion and Management*, 159, pp. 54–64 (2018). DOI: https://doi.org/10.1016/j.enconman.2018.01.010.
12. Sierra-García, J.E., and Santos, M. "Switched learning adaptive neuro-control strategy," *Neurocomputing*, 452, pp. 450–464 (2021). DOI: https://doi.org/10.1016/j.neucom.2019.12.139.
13. García Márquez, F.P., and Peco Chacón, A.M. "A review of non-destructive testing on wind turbines blades," Renewable Energy, 161, pp. 998–1010 (2020). DOI: https://doi.org/10.1016/j.renene.2020.07.145.
14. Zhang, Z., Zhang, D., & Qiu, R. C. "Deep reinforcement learning for power system applications: an overview," *CSEE Journal of Power and Energy Systems*, *6*(1), pp. 213-225 (2019). DOI: https://doi.org/10.17775/cseejpes.2019.00920.
15. Fernandez-Gauna, B., Fernandez-Gamiz, U., and Graña, M. "Variable speed wind turbine controller adaptation by reinforcement learning," *Integrated Computer-Aided Engineering*, 24, pp. 27–39 (2016).

DOI: https://doi.org/10.3233/ica-160531.
16. Abouheaf, M., Gueaieb, W., and Sharaf, A. "Model-free adaptive learning control scheme for wind turbines with doubly fed induction generators," *IET Renewable Power Generation*, 12, pp. 1675–1686 (2018). DOI: https://doi.org/10.1049/iet-rpg.2018.5353.
17. Saenz-Aguirre, A., Zulueta, E., Fernandez-Gamiz, U., Ulazia, A., and Teso-Fz-Betono, D. "Performance enhancement of the artificial neural network-based reinforcement learning for wind turbine yaw control," *Wind Energy*, 23, pp. 676–690 (2019). DOI: https://doi.org/10.1002/we.2451.
18. Tomin, N., Kurbatsky, V., and Guliyev, H. "Intelligent control of a wind turbine based on reinforcement learning," *16th Conference on Electrical Machines, Drives and Power Systems (ELMA)* (2019). DOI: https://doi.org/10.1109/elma.2019.8771645.
19. Iqbal, A., Ying, D., Saleem, A., Hayat, M.A., and Mehmood, K. "Efficacious pitch angle control of variable-speed wind turbine using fuzzy based predictive controller," *Energy Reports*, 6, pp. 423–427 (2020). DOI: https://doi.org/10.1016/j.egyr.2019.11.097.
20. Ngo, Q.-V., Yi, C., and Nguyen, T.-T. "The fuzzy-PID based-pitch angle controller for small-scale wind turbine," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, 11, pp. 135–142 (2020). DOI: https://doi.org/10.11591/ijpeds.v11.i1.pp135-142.
21. Sedighizadeh, M., and Rezazadeh, A. "A modified adaptive wavelet pid control based on reinforcement learning for wind energy conversion system control," *Advances in Electrical and Computer Engineering*, 10, pp. 153–159 (2010). DOI: https://doi.org/10.4316/aece.2010.02027.
22. Sitharthan, R., Karthikeyan, M., Sundar, D.S., and Rajasekaran, S. "Adaptive hybrid intelligent mppt controller to approximate effectual wind speed and optimal rotor speed of variable speed wind turbine," *ISA Transactions*, 96, pp. 479–489 (2020). DOI: https://doi.org/10.1016/j.isatra.2019.05.029.
23. Sarkar, M.R., Julai, S., Tong, C.W., Uddin, M., Romlie, M.F., and Shafiullah, G. "Hybrid pitch angle controller approaches for stable wind turbine power under variable wind speed," *Energies*, 13, p. 3622 (2020). DOI: https://doi.org/10.3390/en13143622.
24. Salem, M.E.M., El-Batsh, H.M., El-Betar, A.A., and Attia, A.M.A. "Application of neural network fitting for pitch angle control of small wind turbines," *IFAC-PapersOnLine*, 54, pp. 185–190 (2021). DOI: https://doi.org/10.1016/j.ifacol.2021.10.350.
25. Ananth, D.V.N. "Artificial neural network based direct torque control for variable speed wind turbine driven induction generator," *International Journal of Computer and Electrical Engineering*, 3, pp. 880–889 (2011). DOI: https://doi.org/10.7763/ijcee.2011.v3.437.
26. Reddak, M., Berdai, A., Nouaiti, A., and Vlasenko, V. "Collaboration of nonlinear control strategy and pitch angle control of dfig equipped wind turbine during all operating regions," *International Journal of Computer Applications*, 179, pp. 16–21 (2018). DOI: https://doi.org/10.5120/ijca2018916531.
27. Fan, Y.-J., Xu, H., and He, Z.-Y. "Smoothing the output power of a wind energy conversion system using a hybrid nonlinear pitch angle controller," *Energy Exploration & Exploitation*, 40, pp. 539–553 (2021). DOI: https://doi.org/10.1177/01445987211041779.
28. Elsisi, M., Tran, M.-Q., Mahmoud, K., Lehtonen, M., and Darwish, M.M.F. "Robust design of anfis-based blade pitch controller for wind energy conversion systems against wind speed fluctuations," *IEEE Access*, 9, pp. 37894–37904 (2021). DOI: https://doi.org/10.1109/access.2021.3063053.
29. Hearns, G. "New directions in non-linear observer design," *International Journal of Adaptive Control and Signal Processing*, 15, pp. 428–428 (2001). DOI: https://doi.org/10.1002/acs.653.
30. Aghannan, N., and Rouchon, P. "An intrinsic observer for a class of lagrangian systems," *IEEE Transactions on Automatic Control*, 48, pp. 936–945 (2003). DOI: https://doi.org/10.1109/tac.2003.812778.
31. Xian, B., de Queiroz, M.S., Dawson, D.M., and McIntyre, M.L. "A discontinuous output feedback controller and velocity observer for nonlinear mechanical systems," *Automatica*, 40, pp. 695–700 (2004). DOI: https://doi.org/10.1016/j.automatica.2003.12.007.
32. Zhang, F., Dawson, D.M., de Queiroz, M.S., and Dixon, W.E. "Global adaptive output feedback tracking control of robot manipulators," *IEEE Transactions on Automatic Control*, 45, pp. 1203–1208 (2000). DOI: https://doi.org/10.1109/9.863607.
33. Atassi, A.N., and Khalil, H.K. "A separation principle for the stabilization of a class of nonlinear systems," *IEEE Transactions on Automatic Control*, 44, pp. 1672–1687 (1999). DOI: https://doi.org/10.1109/9.788534.
34. Floquet, T., Barbot, J.-P., Perruquetti, W., and Djemai, M. "On the robust fault detection via a sliding mode disturbance observer," *International Journal of Control*, 77, pp. 622–629 (2004). DOI: https://doi.org/10.1080/00207170410001699030.
35. Davila, J., Fridman, L., and Levant, A. "Second-order sliding-mode observer for mechanical systems," *IEEE Transactions on Automatic Control*, 50, pp. 1785–1789 (2005). DOI: https://doi.org/10.1109/tac.2005.858636.
36. Nicosia, S., and Tomei, P. "Robot control by using only joint position measurements," *IEEE Transactions on Automatic Control*, 35, pp. 1058–1061 (1990). DOI: https://doi.org/10.1109/9.58537.
37. Berghuis, H., and Nijmeijer, H. "A passivity approach to controller-observer design for robots," *IEEE Transactions on Robotics and Automation*, 9, pp. 740–754 (1993). DOI: https://doi.org/10.1109/70.265918.
38. Battilotti, S., and Lanari, L. "Global set point control via link position measurement for flexible joint robots," *Systems*

& Control Letters, 25, pp. 21–29 (1995). DOI: https://doi.org/10.1016/0167-6911(94)00052-w.
39. Choi, H.H., and Jung, J.-W. "Fuzzy speed control with an acceleration observer for a permanent magnet synchronous motor," *Nonlinear Dynamics*, 67, pp. 1717–1727 (2011). DOI: https://doi.org/10.1007/s11071-011-0099-y.
40. Xiao, M. "The global existence of nonlinear observers with linear error dynamics: a topological point of view," *Systems & Control Letters*, 55, pp. 849–858 (2006). DOI: https://doi.org/10.1016/j.sysconle.2006.04.006.
41. Dawson, D.M., Qu, Z., and Carroll, J.C. "On the state observation and output feedback problems for nonlinear uncertain dynamic systems," *Systems & Control Letters*, 18, pp. 217–222 (1992). DOI: https://doi.org/10.1016/0167-6911(92)90008-g.
42. Pagilla, P.R., and Tomizuka, M. "An adaptive output feedback controller for robot arms: stability and experiments," *Automatica*, 37, pp. 983–995 (2001). DOI: https://doi.org/10.1016/s0005-1098(01)00048-6.
43. Arteaga, M.A., and Kelly, R. "Robot control without velocity measurements: new theory and experimental results," *IEEE Transactions on Robotics and Automation*, 20, pp. 297–308 (2004). DOI: https://doi.org/10.1109/tra.2003.820872.
44. Su, Y.X., Zheng, C.H., Mueller, P.C., and Duan, B.Y. "A simple improved velocity estimation for low-speed regions based on position measurements only," *IEEE Transactions on Control Systems Technology*, 14, pp. 937–942 (2006). DOI: https://doi.org/10.1109/tcst.2006.876917.
45. Korovin, S.K., and Utkin, V.I. "Using sliding modes in static optimization and nonlinear programming," *Automatica*, 10, pp. 525–532 (1974). DOI: https://doi.org/10.1016/0005-1098(74)90053-3.
46. Bartolini, G., Pisano, A., Punta, E., and Usai, E. "A survey of applications of second-order sliding mode control to mechanical systems," *International Journal of Control*, 76, pp. 875–892 (2003). DOI: https://doi.org/10.1080/0020717031000099010.
47. Canudas de Wit, C., and Slotine, J.-J.E. "Sliding observers for robot manipulators," *Automatica*, 27, pp. 859–864 (1991). DOI: https://doi.org/10.1016/0005-1098(91)90041-y.
48. Choi, J.-H., Misawa, E.A., and Young, G.E. "A study on sliding mode state estimation," *Journal of Dynamic Systems, Measurement, and Control*, 121, pp. 255–260 (1999). DOI: https://doi.org/10.1115/1.2802463.
49. Xiong, Y., and Saif, M. "Sliding mode observer for nonlinear uncertain systems," *IEEE Transactions on Automatic Control*, 46, pp. 2012–2017 (2001). DOI: https://doi.org/10.1109/9.975511.
50. Ahmed-Ali, T., and Lamnabhi-Lagarrigue, F. "Sliding observer-controller design for uncertain triangular nonlinear systems," *IEEE Transactions on Automatic Control*, 44, pp. 1244–1249 (1999). DOI: https://doi.org/10.1109/9.769383.
51. Levant, A. "Robust exact differentiation via sliding mode technique," *Automatica*, 34, pp. 379–384 (1998). DOI: https://doi.org/10.1016/s0005-1098(97)00209-4.
52. Pisano, A., and Usai, E. "Output-feedback control of an underwater vehicle prototype by higher-order sliding modes," *Automatica*, 40, pp. 1525–1531 (2004). DOI: https://doi.org/10.1016/j.automatica.2004.03.016.
53. Sira-Ramirez, H. "Dynamic second-order sliding mode control of the hovercraft vessel," *IEEE Transactions on Control Systems Technology*, 10, pp. 860–865 (2002). DOI: https://doi.org/10.1109/tcst.2002.804134.
54. Alvarez, J., Orlov, I., and Acho, L. "An invariance principle for discontinuous dynamic systems with application to a coulomb friction oscillator," *Journal of Dynamic Systems*, Measurement, and Control, 122, pp. 687–690 (2000). DOI: https://doi.org/10.1115/1.1317229.
55. Keighobadi, J., Mohammadian KhalafAnsar, H., and Naseradinmousavi, P. "Adaptive neural dynamic surface control for uniform energy exploitation of floating wind turbine," *Applied Energy*, 316, p. 119132 (2022). DOI: https://doi.org/10.1016/j.apenergy.2022.119132.
56. National Renewable Energy Laboratory (NREL). "Filtration + separation," 38(4), p. 46 (2001). DOI: https://doi.org/10.1016/s0015-1882(01)80290-1. Available at: www.nrel.gov.
57. Pierson, W.J., and Moskowitz, L. "A proposed spectral form for fully developed wind seas based on the similarity theory of s. a. kitaigorodskii," *Journal of Geophysical Research*, 69, pp. 5181–5190 (1964). DOI: https://doi.org/10.1029/jz069i024p05181.
58. Matheron, G., Perrin, N., and Sigaud, O. "The problem with ddpg: understanding failures in deterministic environments with sparse rewards," *arXiv preprint* arXiv:1911.11679 (2019). DOI: https://doi.org/10.48550/arxiv.1911.11679.
59. van der Vaart, H.R., and Yen, E.H. "Weak sufficient conditions for fatou's lemma and lebesgue's dominated convergence theorem," *Mathematics Magazine*, 41, pp. 109–117 (1968). DOI: https://doi.org/10.1080/0025570x.1968.11975853.
60. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. "Proximal policy optimization algorithms," *arXiv preprint* arXiv:1707.06347 (2017). DOI: https://doi.org/10.48550/arxiv.1707.06347.
61. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. "Continuous control with deep reinforcement learning," *arXiv preprint* arXiv:1509.02971 (2015). DOI: https://doi.org/10.48550/arxiv.1509.02971.
62. Manrique Escobar, C.A., Pappalardo, C.M., and Guida, D. "A parametric study of a deep reinforcement learning control system applied to the swing-up problem of the cart-pole," *Applied Sciences*, 10(24), p. 9013 (2020). DOI: https://doi.org/10.3390/app10249013.
63. Kővári, B., Hegedüs, F., and Bécsi, T. "Design of a reinforcement learning-based lane keeping planning agent for

automated vehicles," *Applied Sciences*, 10(20), p. 7171 (2020). DOI: https://doi.org/10.3390/app10207171.

List of figures:

List of Tables:

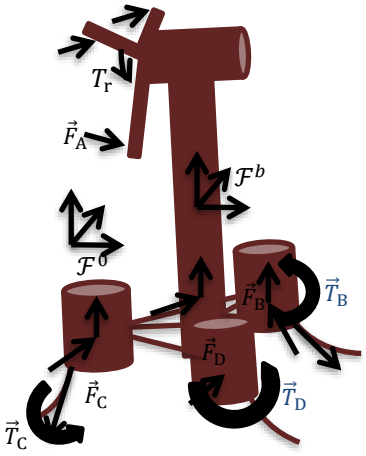


Fig. 1. Overall Force Diagram of the nonlinear Model

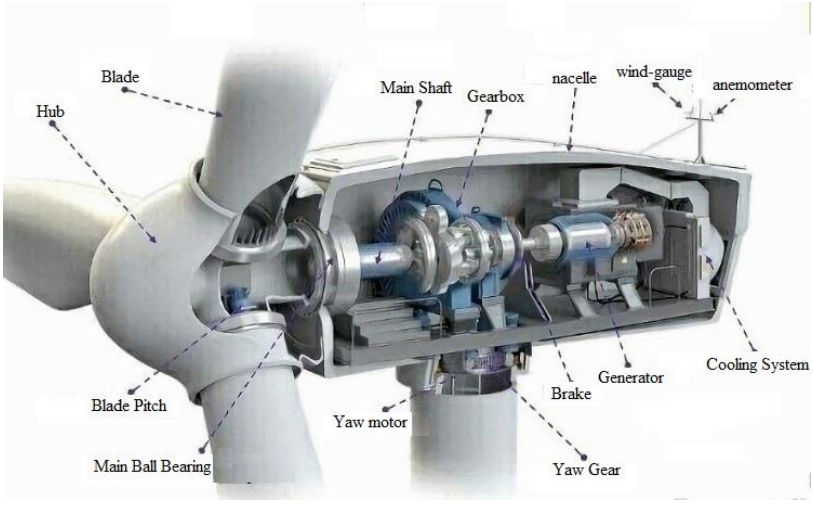


Fig. 2. Components of the system under control.

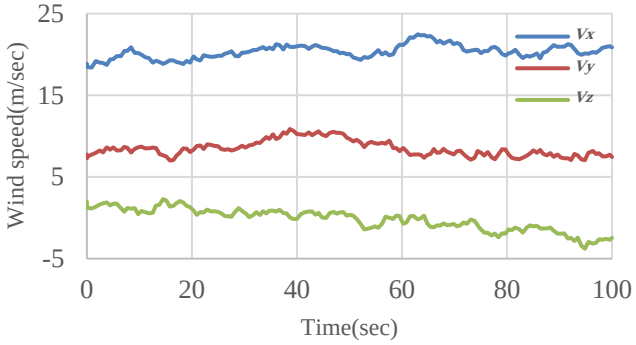


Fig. 3. Wind trajectory of the paper [11]

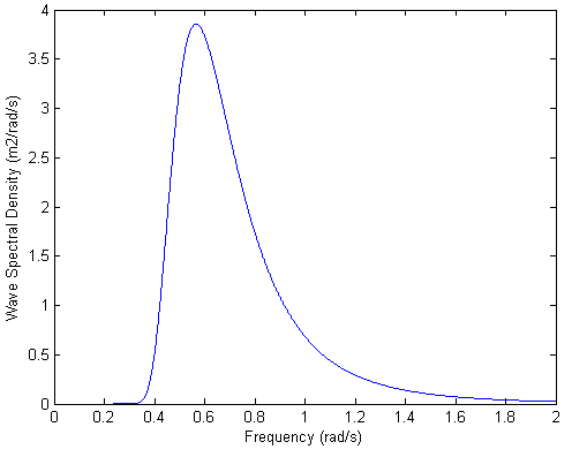


Fig. 4. Pierson-Moskowitz spectrum

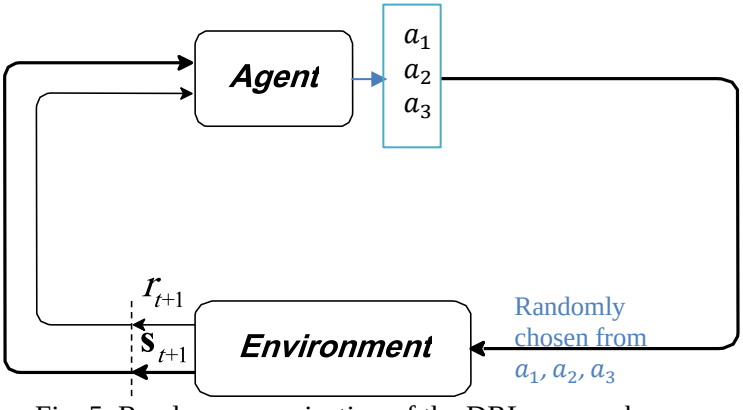


Fig. 5. Roadmap organization of the DRL approach.

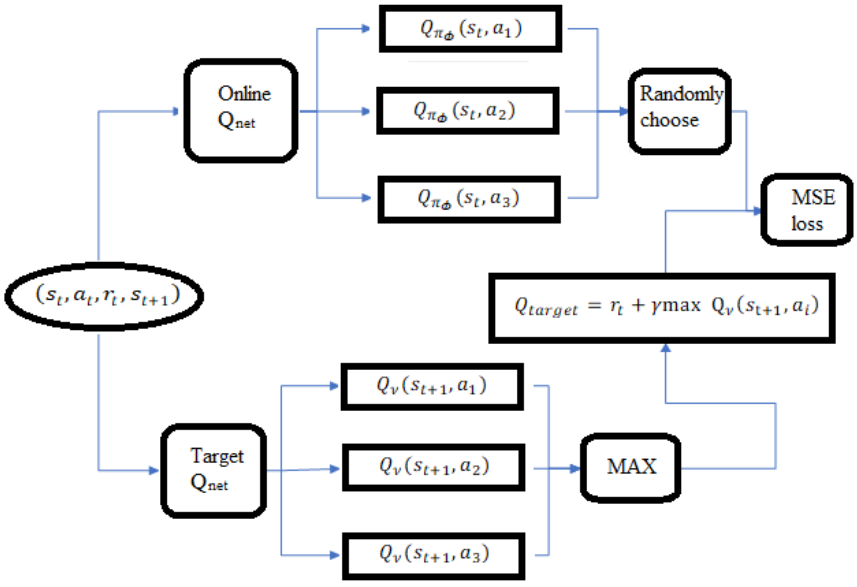


Fig. 6. Workflow scheme of DQN in DRL approach.

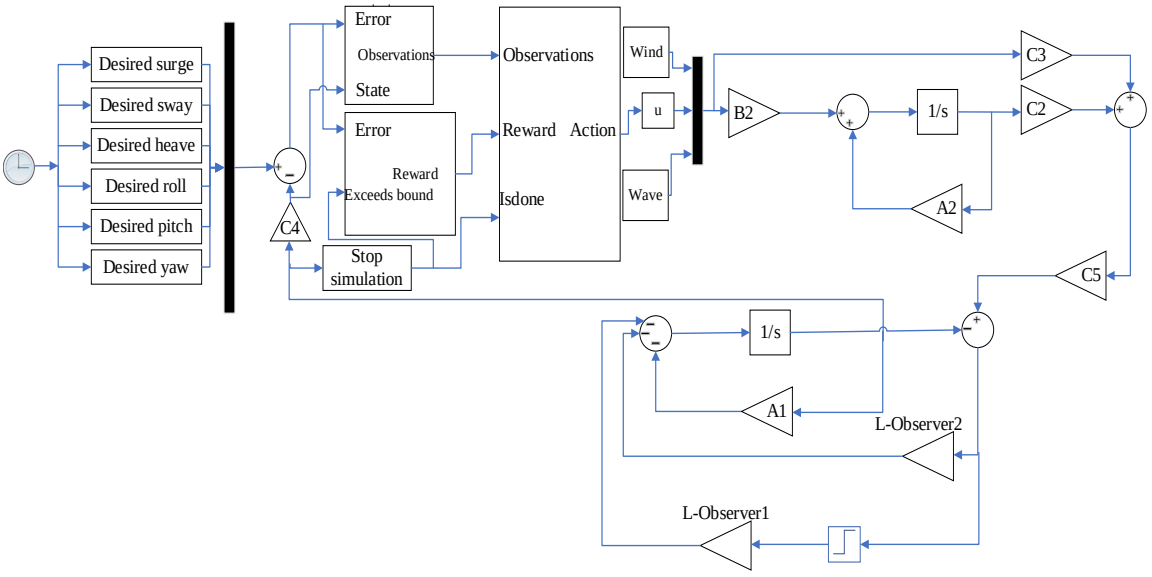


Fig. 7. Schematic representation of the comprehensive observer and controller system in MATLAB.

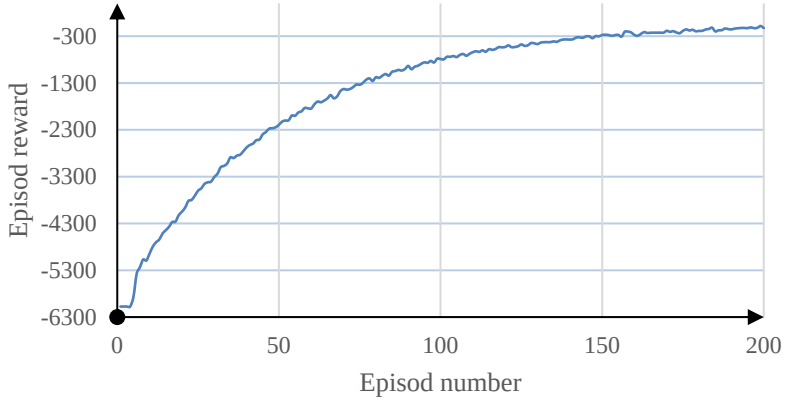
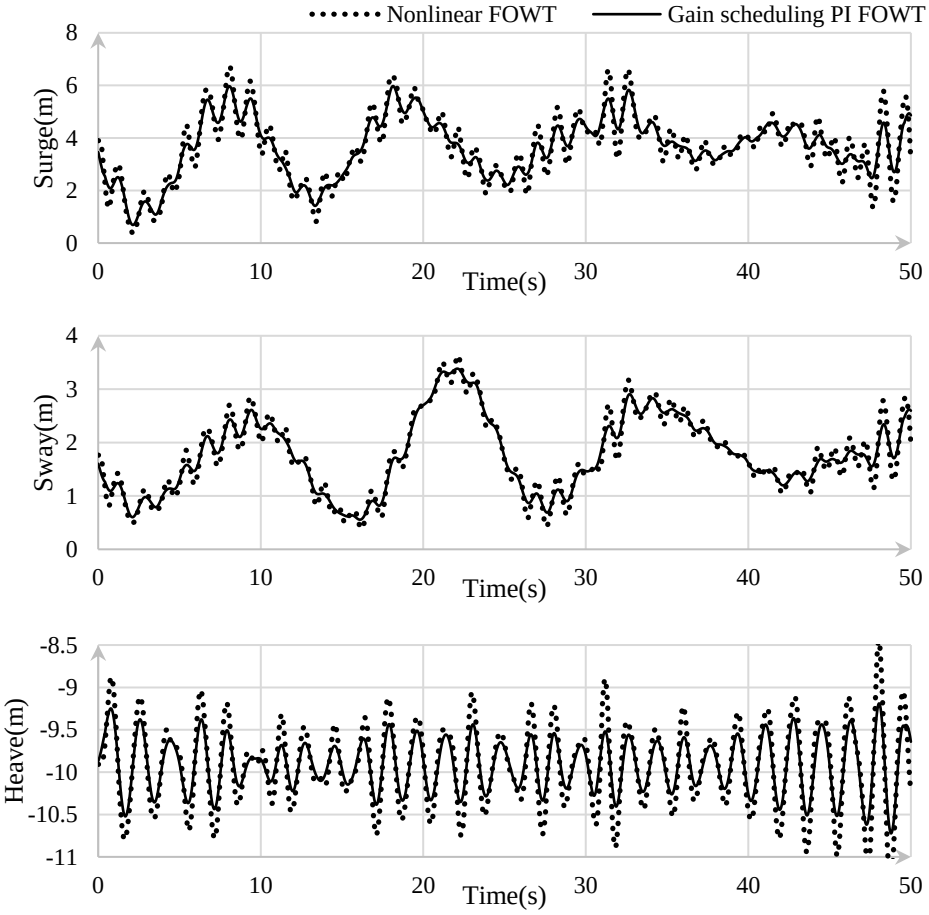
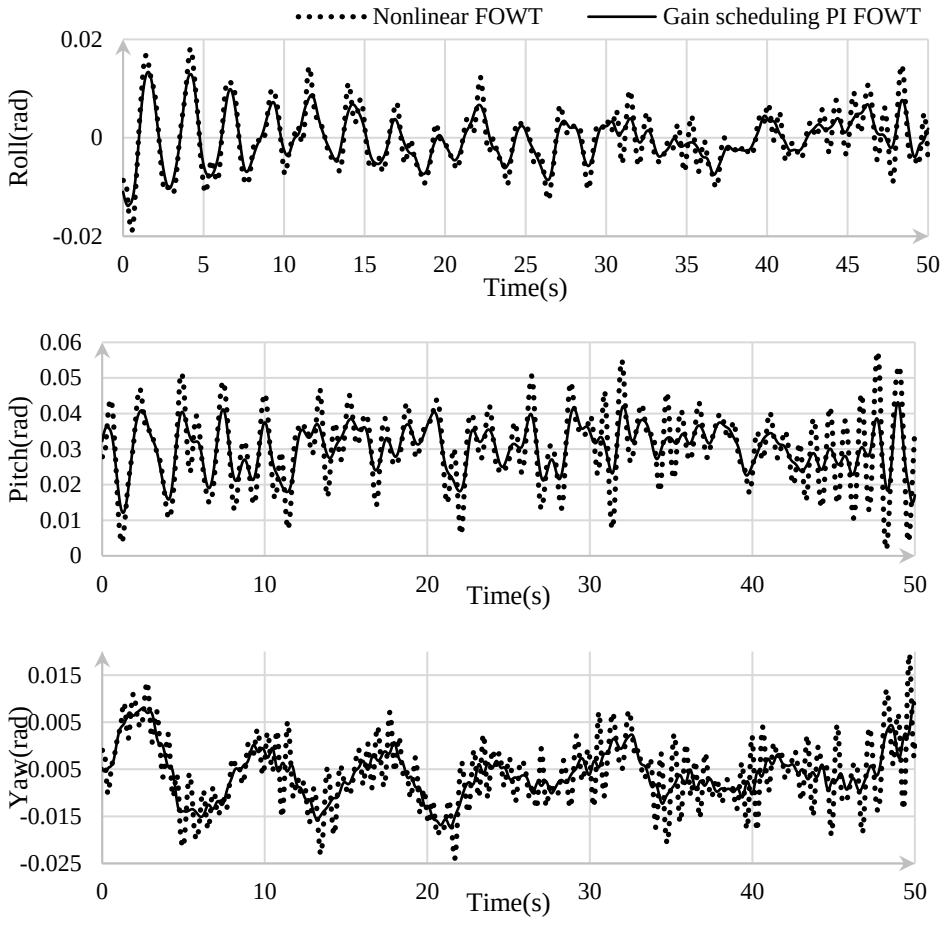


Fig. 8. Cumulated reward during simulation of DRL.



a)



b)

Fig. 9. Desired system simulation in MATLAB for a) translational, b) rotational states

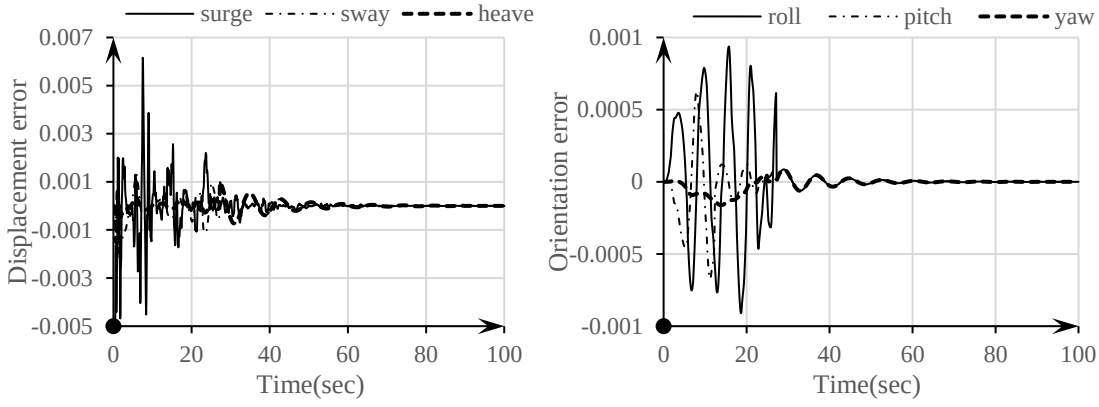


Fig. 10. Displacement and orientation estimation error

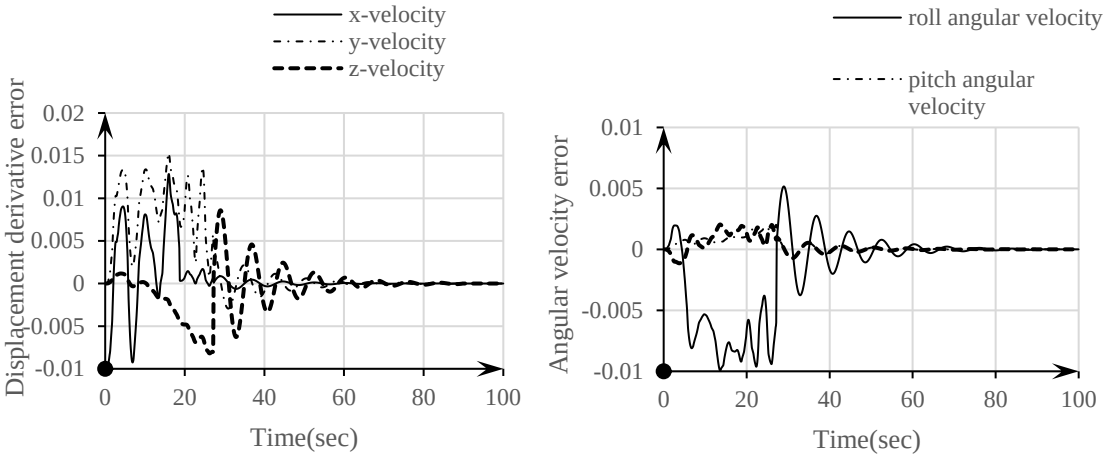


Fig. 11. Linear and angular velocity estimation error

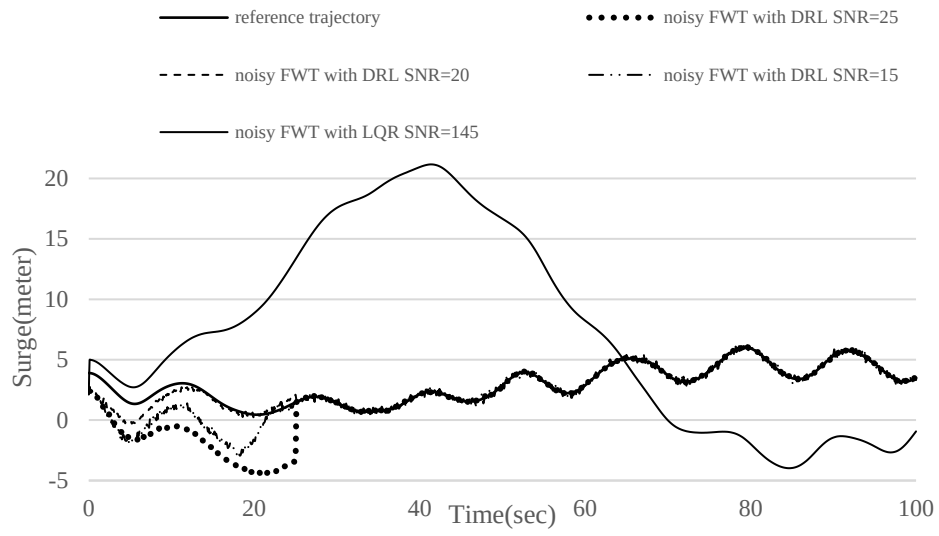


Fig. 12. Assessing the performance of DRL controller of the surge with noise

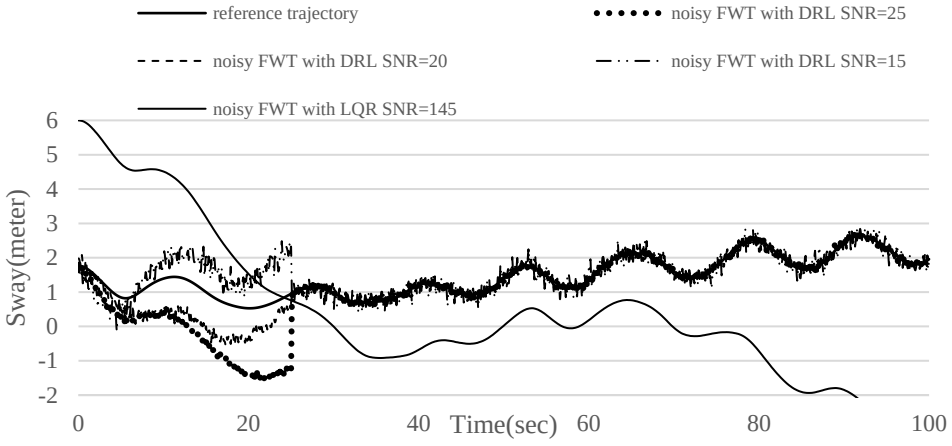


Fig. 13. Assessing the performance of DRL controller on sway with noise

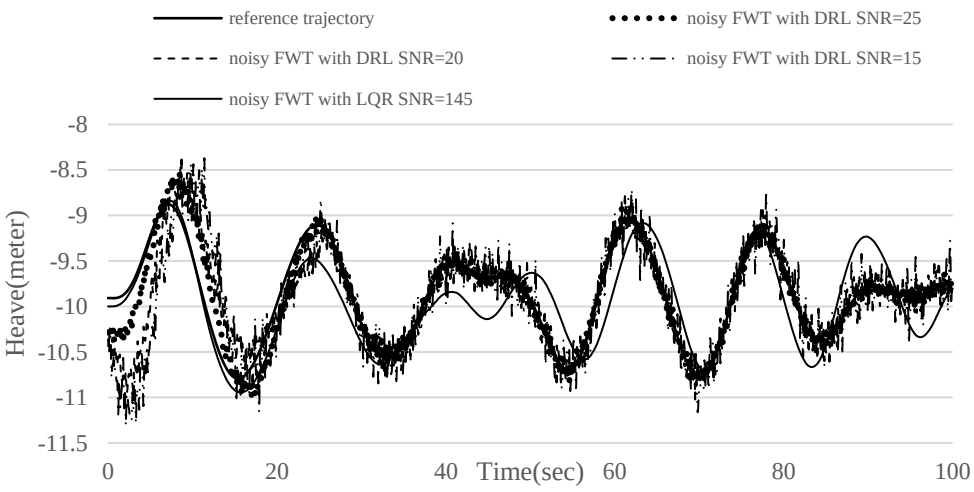


Fig. 14. Assessing the performance of DRL controller on heave with noise

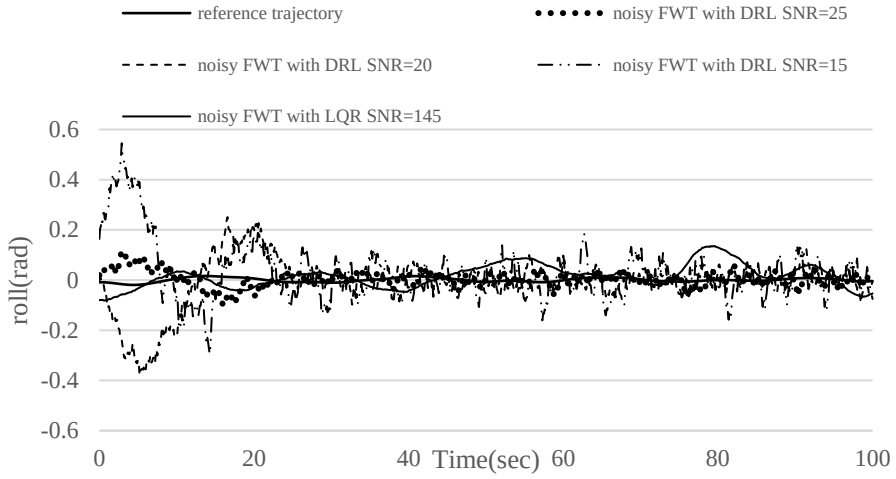


Fig. 15. Assessing the performance of DRL controller on roll with noise

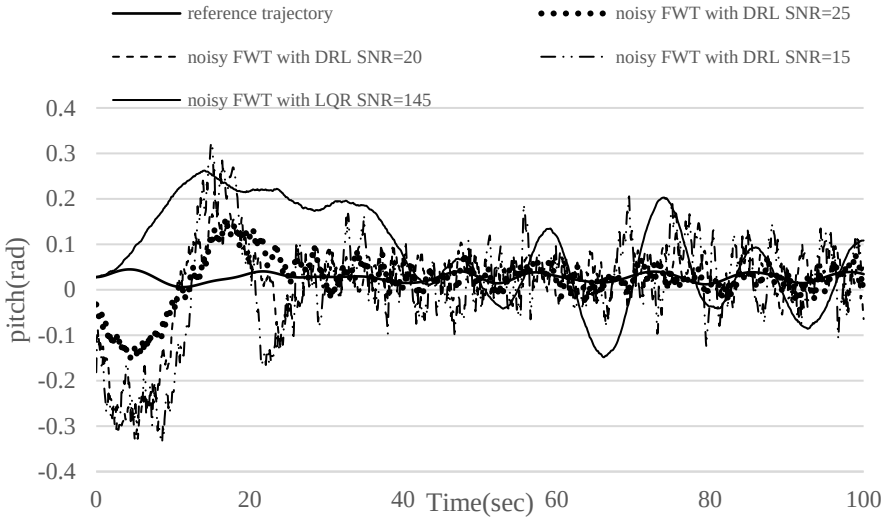


Fig. 16. Assessing the performance of DRL controller on pitch with noise

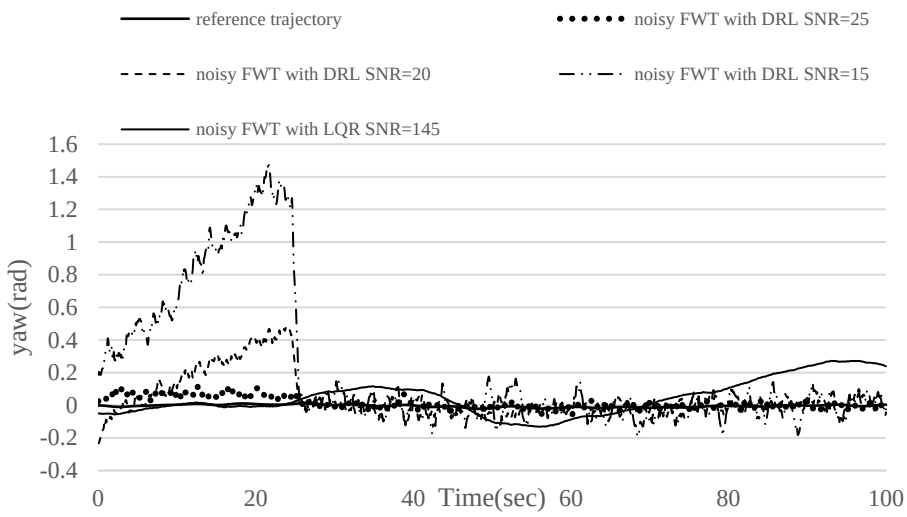


Fig. 17. Assessing the performance of DRL controller on yaw with noise

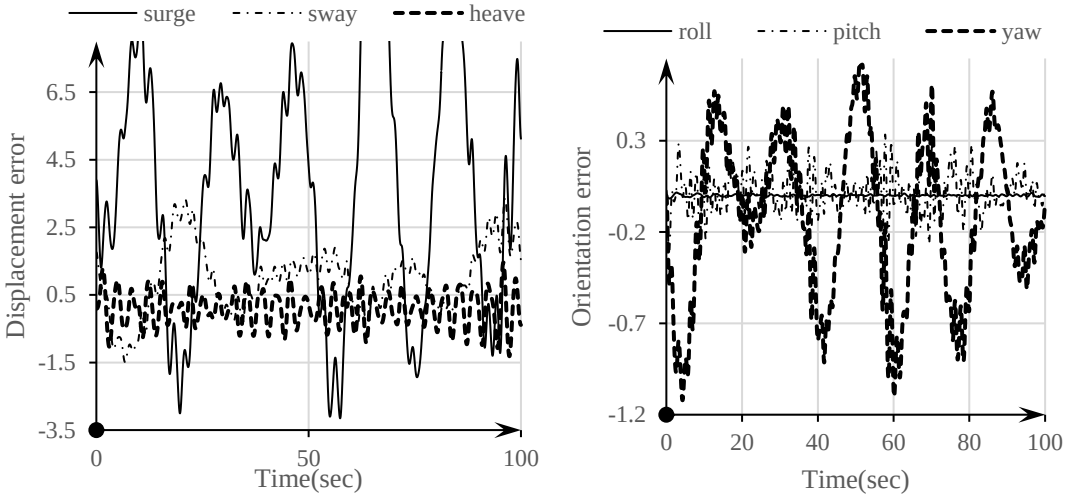


Fig. 18. Displacement and Orientation estimation error using Luenberger observer

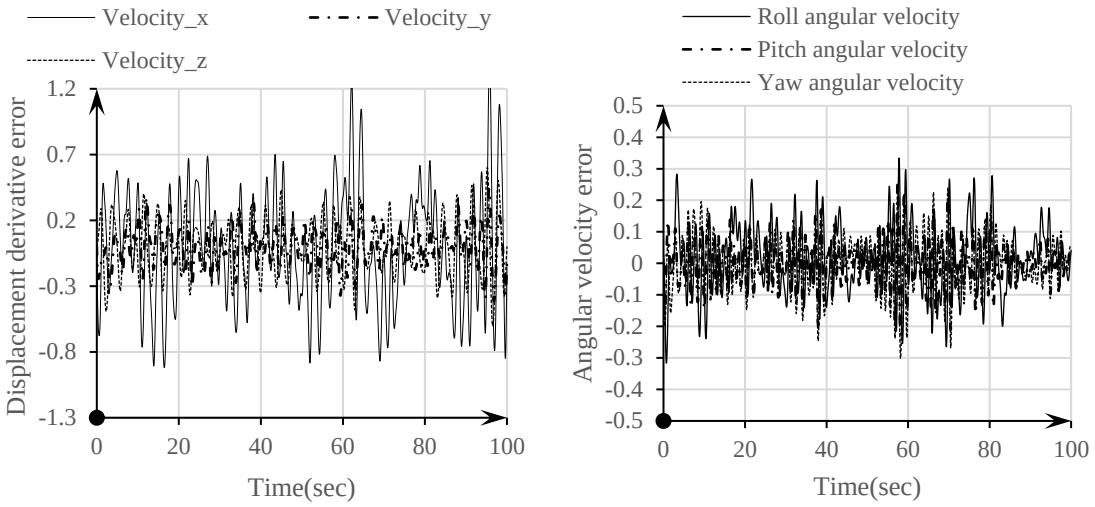


Fig. 19. Linear and angular velocity estimation error using Luenberger observer

Table 1. FOWT's properties

Property	Sign	Value	Unit
Water density	ρ	1025	kg/m^3
Physical mass	m_g	14,072,718	kg
Inertia around x-axis	I_{xx}	1.695e10	kg.m^2
Inertia around y-axis	I_{yy}	1.695e10	kg.m^2
Inertia around z-axis	I_{zz}	1.845e10	kg.m^2
Air density	ρ_a	1.225	kg/m^3
Effective rotor radius	R_r	62.94	m
Distance vector from FOWT's center to thrust center	$\vec{r}_{gt}$	$\begin{bmatrix} -5 \\ 0 \\ 99.889 \end{bmatrix}$	m
Rotor Inertia	J_r	3.5444e7	kg.m^2
Generator Inertia	J_g	5.34116e2	kg.m^2
Driveshaft stiffness on rotor side	k_r	8.676e8	N.m/rad
Driveshaft damping on rotor side	b_r	6.215e6	N.m.s/rad
Gear ratio	N_{gr}	97	-

Table 2. Pseudo code of the training of the DQN algorithm

Algorithm 1: deep Q-learning
Initialize action-online network Q_ϑ with random weights ϑ
Initialize action-target network $Q'_{\vartheta'}$ with weights $\vartheta'=\vartheta$
For interval $=1,K$ do
Reset system observation $s_1=\{x_1\}$ and preprocessed sequence $\phi_1=\phi(s_1)$

For t=1, T do

Generate a random number between zero and one i.e., k

If $k < \varepsilon$

From given state, pick a random action a_t with possibility ε

else

Select $a_t = \text{argmax}_a Q\left(\phi\left(s_t\right), a; \vartheta\right)$

Perform action a_t on environment and detect reward r_t and state x_{t+1}

Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess state $\phi_{t+1} = \phi\left(s_{t+1}\right)$

Store experience in Replay Buffer $\left(\phi_t, a_t, r_t, \phi_{t+1}\right)$ in buffer D

Sample a random batch $\left(\phi_j, a_j, r_j, \phi_{j+1}\right)$ of experiences from Replay Buffer D

Set the target value

$$y_j = \begin{cases} r_j & \text{stops at j+1} \\ r_j + \gamma \max_{a'} \mathbf{Q}'_{\vartheta'}\left(s_{j+1}, a; \vartheta'\right) & \text{otherwise} \end{cases}$$

Execute a gradient descent step on $\left(y_j - Q\left(\varphi_j, a_j; \vartheta\right)\right)^2$ regarding the network parameters ϑ

Every M steps reset $\mathrm{Q}'_{\vartheta'} = Q$

End For

End For

Table 3. Computational complexity and time for DRL featured FOWT

Aspect	Description	Numerical Example
Model Complexity	Number of parameters in the DNN architecture	1-10 million parameters
	Depth of the DNN (number of layers)	5-20 layers
Training Complexity	Number of training epochs	100-1000 epochs
	Size of training dataset	10,000-1,000,000 samples
	Computational power (e.g., GPUs, TPUs)	High-performance computing cluster
Inference Complexity	Forward pass time for a single input	1-100 milliseconds
	Inference time variability (due to input size, network architecture)	10-50 milliseconds (for real-time control)

Computational Time (Training)	Total training time	1-2 weeks (on a high-performance computing cluster)
	Average time per epoch	10-100 minutes
Computational Time (Inference)	Average time per inference	1-10 milliseconds
Computational Resources	Number of GPUs/TPUs used for training	4-16 GPUs/TPUs
	Memory requirements for training data and model parameters	100 GB - 1 TB

Table 4. Statistical specifications of tracking errors of adaptive observer-oriented DRL controller with noise compared with under-noise LQR.

State Variables	Optimal LQR with noise SNR=145		Adaptive black-box observer-oriented DRL controller with noise SNR=15	
	Mean	Standard Deviation	Mean	Standard Deviation
Surge(m)	4.467	9.219	-0.526	1.0618
Sway(m)	-1.0614	2.4535	0.1105	0.3833
Heave(m)	-0.762	0.845	-0.0463	0.3934
Roll(deg)	0.0276	0.1178	0.0152	0.0469
Pitch(deg)	0.0573	0.1210	-0.0216	0.1187
Yaw(deg)	0.4960	0.4039	0.2066	0.1065