Development and validation of a health-aware floating offshore wind farm simulation platform: FOWLTY

Peña-Sanchez, Yerai; Penalba, Markel; Knudsen, Torben; Nava, Vincenzo; Pardo, David

Published in:
Wind Energy and Engineering Research

DOI (link to publication from Publisher):
10.1016/j.weer.2024.100008

Publication date:
2024

Document Version
Publisher's PDF, also known as Version of record

2 (2024) 100008

Contents lists available at ScienceDirect

Wind Energy and Engineering Research

journal homepage: https://www.sciencedirect.com/journal/wind-energy-and-engineering-research

Development and validation of a health-aware floating offshore wind farm simulation platform: FOWLTY

Yerai Peña-Sanchez [a,*], Markel Penalba [a,b], Torben Knudsen [c], Vincenzo Nava [d,e], David Pardo [f,e,b]

[a] Fluid Mechanics Department, Mondragon University, 20500 Arrasate, Spain
[b] Ikerbasque, Basque Foundation for Science, 48009 Bilbao, Spain
[c] Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark
[d] TECNALIA, Basque Research and Technology Alliance (BRTA), 48169, Derio, Spain
[e] Basque Center for Applied Mathematics (BCAM), 48001, Bilbao, Spain
[f] Department of Mathematics, University of the Basque Country, 48940 Leioa, Spain

ARTICLE INFO

Keywords:
Health-aware floating offshore wind farm simulation
Model validation
Fault detection and isolation
Fault-tolerant control
FOWLTY

ABSTRACT

Offshore wind energy is now an important game player towards the achievement of United Nations' Sustainable Goals 7 (clean and affordable energy) and 13 (climate change) due to its huge potential but, given the harsh environment in which the turbines operate, they lead to more frequent and severe component faults compared to onshore farms. These faults result in increased maintenance and operational costs, posing a significant challenge to profitability. To mitigate these costs and maintain energy production during faults, fault detection and isolation (FDI) and fault-tolerant control (FTC) strategies have gained traction. Effective simulation tools are crucial for the development of these strategies. FOWLTY, a new simulator introduced in this paper, which derives its name from 'floating offshore wind farm simulator' incorporating the term 'faulty', allows users to recreate faults in wind turbine subsystems. It simplifies simulation to facilitate the development of control and estimation strategies, offering a balance between accuracy and computational efficiency. FOWLTY is compared to OpenFAST considering different scenarios, demonstrating its suitability for fault detection, diagnosis and impact mitigation through control. Additionally, a case study considering a farm composed of eight floating wind turbines is provided to show the effects of different faults.

1. Introduction

The increasing global demand for energy, combined with the catastrophic environmental impact of traditional fossil fuel combustion, presents a significant challenge for this century. To address this challenge, renewable energies, such as onshore and offshore wind energy, offer a promising solution, providing affordable, clean and sustainable energy. In fact, offshore wind energy alone has enormous potential, with an estimated annual global generation capacity of 420,000 TWh, which is more than 18 times the current global electricity demand [1]. To make wind energy economically viable, wind turbines (WTs) are often installed together in offshore wind farms. In recent years, offshore wind farms have become more popular due to more consistent wind conditions, and the difficulties of finding further suitable and available onshore locations. Furthermore, offshore wind farms are now being installed in deeper waters, breaking the historical trend of installing them in shallow waters [2]. Overall, the development of the offshore wind energy sector and the expansion of offshore wind farms to deeper water further from shore offer a promising way to address the United Nations' sustainable development goals [3] of meeting the growing energy demand, while reducing greenhouse gas emissions and, thus, the most critical environmental impacts of climate change.

However, offshore wind farms face a significantly harsher environment that leads to more frequent and severe component faults compared to their onshore counterparts. As a consequence, higher maintenance needs and associated costs are expected in offshore wind farms. Such more frequent and severe faults result in increased downtime of wind turbines, reducing the final energy generation and increasing operation and maintenance (O&M) costs.

For offshore wind farms, these O&M costs are estimated to be about 30% of the total cost (rising up to 40% under extreme scenarios) [4,5], compared to the 10%–15% of an onshore farm. Note that such increased maintenance has a double negative effect due to the cost of the maintenance itself, but also due to the less energy generated due to the increased downtime, [6]. To mitigate these costs and increase energy generation despite faults, fault detection and isolation (FDI),

* Corresponding author.
E-mail address: ypena@mondragon.edu (Y. Peña-Sanchez).

https://doi.org/10.1016/j.weer.2024.100008
Received 13 May 2024; Received in revised form 2 December 2024; Accepted 2 December 2024
Available online 8 December 2024

Nomenclature

α	Position in defined DoFs
β	Blade pitch angle [rad]
λ	Tip-speed ratio
μ_∞	Radiation added mass coefficient at infinite frequency
ω	Angular velocity [rad/s]
ρ_a	Air density [kg/m^3]
τ	Torque [Nm]
wc	Wake centre
wd	Wake deficit
we	Wake expansion
A	Swept area of the blades [m^2]
C_P	Coefficient of power
C_T	Coefficient of thrust
C_{opt}	Optimal power coefficient
d	Distance of the turbine downwind [m]
f_ex	Wave excitation force or torque
f_hyd	Hydrodynamic forces or torques
f_h	Hydrostatic force or torque
f_moo	Mooring force or torque
f_rad	Radiation force or torque
m	Mass matrix [kg]
n_dt	Drive train gear ratio
P	Power [W]
R	Rotor radius [m]
t_g	Generator constant
tg	Generator time constant [s]
v_w	Wind speed [m/s]
DoF	Degree of Freedom
FDI	Fault Detection and Isolation
FOWT	Floating Offshore Wind Turbine
FTC	Fault-Tolerant Control
NRMSE	Normalised Root Mean Squared Error
O&M	Operation and Maintenance
WT	Wind Turbine

and fault tolerant control (FTC) methods have gained popularity in the last decade [7]. While FDI systems are used to detect and isolate any possible fault, FTC systems aim to maintain performance at the desired level and reduce their impact. Thus, by employing appropriate FDI and FTC strategies, downtime and maintenance needs are decreased, improving the robustness of the energy generation in offshore WTs, keeping the final energy cost as cheap as possible [8,9]. The interested reader is referred to [7,7,10–14] for comprehensive reviews on FDI and FTC strategies, respectively, applied to WTs.

Note that, in addition to FDI and FTC methods, other approaches can also be used to decrease the electricity production costs of wind farms. Two notable examples are condition monitoring and optimal control of wind farms. Condition monitoring involves using measurements from different parts of the WTs to estimate the health of their components in real-time. This information can then be used to optimise maintenance strategies, which can result in lower maintenance costs and reduced downtime. Meanwhile, advanced wind turbine farm control strategies take into account the interactions between wind turbines to calculate the optimal reference values for each turbine. An example of such a technique is wake steering control, which involves adjusting the yaw angles of the turbines in a wind farm to reduce the wake effects on downwind turbines [15]. This can lead to more efficient energy generation, lower structural requirements and higher turbine performance.

To develop the aforementioned strategies, it is of paramount importance to be able to simulate WT farms that can characterise the interactions between the turbines. To this end, various simulation techniques can be found in the literature, including high-, mid-, and low-fidelity models. High-fidelity models excel at accurately assessing WT behaviour in specific scenarios. However, when it comes to developing estimation and control strategies, low- and mid-fidelity models are more practical. These models might not precisely replicate the WT's exact behaviour but can effectively represent its underlying dynamics with significantly lower computational complexity. The most widely used mid-fidelity simulators within the community is OpenFAST [16], which is usually considered as a benchmark model when comparing the results from other models. Among the OpenFAST tools, the specific tools to simulate wind farms are FAST.farm, a code with a fidelity similar to OpenFAST, and SOWFA, which is a CFD model coupled with OpenFAST. Using OpenFAST tools to develop FDI/FTC strategies (or, in general, control strategies) might not be the best option for two main reasons: (i) recreating faults in the different subsystems is far from intuitive without a deep understanding of the OpenFAST code structure and advanced coding skills, and (ii) the computational time required to compute a simulation in the case of farms is prohibitive. To this end, several other models can be found in the literature. By way of example, in [17,18] the authors develop an openly-available WT model using a simple description of all the subsystems of the WT so that non-WT-experts can use the model without difficulty. Hence, this model can be used as a benchmark to test FDI and FTC strategies by any interested researcher/developer. However, its use is very limiting since, as it is, it is only able to characterise a single wind scenario and it is (virtually) impossible to find the files online nowadays. Another noteworthy example is the *SimWindFarm* (SWF) simulator [19,20], a wind farm simulator that was developed as part of the European-Union-funded project AEOLUS,[1] also using a simple description of the WT subsystems and providing a wind farm model that can be used to develop wind farm control strategies. The required files to run SWF are available online, although they need some changes on the code to be used on the latest Simulink versions, due to outdated Matlab functions it requires. Another control-oriented wind farm model is presented in [21], called *WindFarmSimulator* (WFSim), which has a higher fidelity description of the dynamics of the wind farm than that considered for the SWF, but, because of that, it is considerably slower. Finally, one of the more recently developed simulators is the Matlab for offshore floating wind turbine simulator (MOST), as presented by Cottura et al. in [22]. While MOST provides a simple WT model within a control-friendly Matlab/Simulink environment, it does not offer capabilities for simulating wind farms or including faults.

Considering the simulation tools available in the literature to simulate wind farms introduced before, one could notice that there is no toolbox able to recreate faults on different components on a wind farm. The only tool is the single turbine benchmark case presented in [17,18] but, as mentioned before, it provides just a specific WT case with a single wind case and cannot be used to simulate different scenarios or turbines. Furthermore, given its publication date, it is difficult to obtain it online anymore. Thus, there is no simulation tool able to simulate floating offshore wind farms and recreate faults on the different subcomponents.

This paper presents a new wind farm simulator called FOWLTY (derived from **F**loating **O**ffshore **W**ind farm simula**T**or and incorporating the term faulty) that can simulate faults in the most critical components of WTs that make up the farm. The FOWLTY platform is based on the SimWindFarm (SWF) simulator and, hence, is implemented in Simulink. FOWLTY is the only simulator available in the literature

[1] https://ict-aeolus.eu/index.html

providing the possibility of simulating an offshore floating wind farm able to recreate faults on different subsystems. FOWLTY enables the user to choose any wind farm layout and select the initialisation and duration of one or multiple faults. In addition, FOWLTY is designed to be straightforward for the implementation of estimation and control strategies (FDI and FTC approaches, for example) for non-experts in the wind energy field, while also accurately representing the essential dynamics of real wind farms. The simplicity of the model considered in the FOWLTY platform is not only beneficial for ease of comprehension but also for running numerous scenarios to test the developed strategies in a timely manner. Thus, one of the main objectives of FOWLTY is to provide a platform to be used by experts who have limited knowledge/understanding of the WT sector to advance the development of FDI and FTC strategies. Finally, given its nature, the FOWLTY platform can also be used for additional purposes that benefit from quick and reliable simulations, such as layout optimisation or energy estimation, among others.

Results of this study demonstrate that FOWLTY provides significant computational advantages over high-fidelity tools like OpenFAST, particularly when simulating larger wind farms. While OpenFAST experiences exponential growth in computational time with increasing farm size, FOWLTY exhibits a linear increase, making it a more scalable solution. Additionally, FOWLTY's ability to recreate a wide variety of faults in critical wind turbine components allows users to analyse the effects of these faults on key system measurements, such as power generation, blade loads and operational efficiency. These features make FOWLTY an effective tool for developing control and estimation strategies that require rapid testing and iteration, without sacrificing accuracy in fault scenarios.

The remainder of this paper is organised as follows: Section 2 presents the Simulink model that defines the main dynamics of the wind farm. Section 3 describes the fault scenarios considered by the FOWLTY platform. Section 4 delves into a code-to-code validation of FOWLTY with OpenFAST under different scenarios. Section 5 provides examples of how the behaviour of the WTs in the farm is affected when faults occur. Finally, in Section 6, some conclusions are drawn and future improvements are discussed.

2. Wind farm model

FOWLTY is a Matlab/Simulink-based numerical platform designed to replicate wind farms of any size and configuration. Since it is based on the previous SWF simulator [19], FOWLTY shares a similar code structure that consists of four primary blocks:

i. the *Wind turbines* block, responsible for simulating the behaviour of the individual WTs;
ii. the *Wind field* block, which calculates the wind effects on each WT;
iii. the *Farm controller* block, which implements farm-level control (note that this differs from the WT control block implemented in the *Wind turbine* block, as explained later in Section 2.2); and
iv. the *Network* block, which includes three sub-blocks (*i.e* network operator, grid, and network load) that simulate the grid behaviour.

In addition, there is a fifth block for the post-processing of the results. This block can be executed once the simulation is completed to display wake and wind field animations. Additionally, when the simulation finishes, the variables are automatically imported to Matlab to compute the final quantities, such as mean power production and power losses due to the different faults, and to plot the results in an organised manner.

Fig. 1 shows the four main blocks composing the top layer of the FOWLTY platform. In order to understand the operation of the FOWLTY platform, let us define the main interconnections between the four blocks composing the platform (illustrated in Fig. 1 by means of grey arrows):

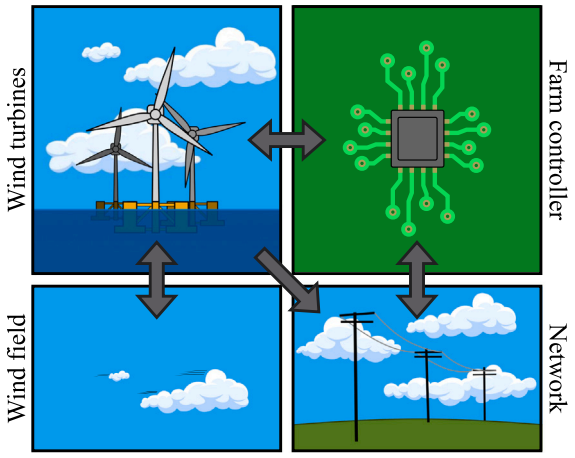

Fig. 1. The four main blocks composing the top layer of FOWLTY.

- The ambient **wind field** is the main input for the wind turbine block, causing the rotation of the WT rotor and generating power. In turn, the wind field is also altered due to the wake effects generated by WTs on the farm. Hence, the relationship between the wind turbines and wind field blocks is bidirectional.
- The **wind turbines** block communicates with the network block by sending the information about the generated power that represents the integration of the power into the grid.
- The **network** block commands a certain power to the farm controller based on the available power estimated by the farm controller.
- The **farm controller** uses measurements collected at different points of the WTs and the power commanded by the network to calculate the reference power for each WT, delivering the control action to each of the WTs.

The main objective of the FOWLTY platform is to be used for the design of wind farm estimators/controllers and, hence, certain assumptions (A) of the aero-hydrodynamic interactions can be relaxed, reducing the associated computational burden:

A-1 The wind direction and (mean) speed are constant along the whole simulation, with the direction parallel to the WT rotor axis. To test the effects of different wind directions, a new wind farm needs to be generated with a rotated layout.

A-2 The wind field is computed as a 2D plane at the hub height and, hence, does not consider 3D effects like wind shear or tower shadow.

A-3 The turbine yaw is constant, as the wind direction is also assumed to be constant.

In the following subsections, a description of the aforementioned four blocks composing FOWLTY (see Fig. 1) is provided.

2.1. Wind field

In the wind field block of the FOWLTY platform, the speed of the wind at the position of each turbine within the farm is calculated as a combination of the (pre-computed) ambient wind field and the wake effect generated by upwind turbines. The ambient wind field is computed using the Veers algorithm [23], which considers a spectrum generated following the recommendations described by the International Electrotechnical Commission (IEC) 61400-1 [24]. Such spectrum, as noted by [25], can be used as non-site-specific wind conditions for offshore wind turbines. In addition, FOWLTY assumes Taylor's frozen turbulence hypothesis for inviscid flow [26], simplifying the equations

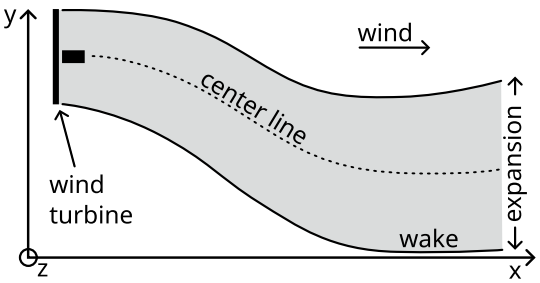

Fig. 2. Sketch of the (top view) wake generated by a WT.

used to compute wake effects and reducing the computational effort. The wind speed is generated using the Kaimal spectrum [27] with parameters as specified by Grunnet et al. [19].

In FOWLTY, the wake effects on a given wind turbine are solely determined by the operating conditions of upwind turbines. Since a constant (mean) wind direction and Taylor's frozen turbulence assumption are considered the characterisation of upwind and downwind turbines is done prior to the simulation, further simplifying wake effect calculations [19]. The wake effects encompass three different phenomena: deficit, expansion, and centre. Deficit quantifies the decrease in the wind speed behind a turbine, expansion represents the area behind the turbine affected by the wake, and centre specifies the lateral position of the wake. For a better understanding, Fig. 2 provides a sketch of the wake generated by a wind turbine, highlighting in grey the area affected by the wake.

In order to calculate the wake effect of a given WT, first, the affected area behind such a turbine is calculated. As introduced in [19], the expansion of the wake generated by WT i is calculated according to [28] as

$$\mathrm{we}_i(d) = \sqrt{4R^2 + dR}, \tag{1}$$

with R being the radius of the WT rotor and d the downwind distance from turbine i. Then, the wake centre at a distance d and time t is computed as a function of its value at $t - t_s$, and the lateral wind speed at that position is given as:

$$\mathrm{wc}_i(d,t) = \mathrm{wc}_i(d,t-t_s) + \hat{v}^{y}_{\mathrm{w},i}(d, W_i(d,t-t_s)), \tag{2}$$

where $W_i(d,t-t_s)$ refers to all the points at $x = d$ and y between $\mathrm{wc}_i(d,t-t_s) - \mathrm{we}_i(d)$ and $\mathrm{wc}_i(d,t-t_s) + \mathrm{we}_i(d)$, t_s is the sampling time, $\hat{v}^{y}_{\mathrm{w},i}(d, W_i(d,t-t_s))$ the average lateral (y-axis) wind speed over the area specified by $x = d$ and $y = W_i(d,t-t_s)$. Thus, the effect that the WT i has on the ambient wind speed (at any point of the generated wake) can be obtained as

$$v_{\mathrm{w},i}(p,t) = \mathrm{wd}_i(d,t) u_x(p,t), \tag{3}$$

$\mathrm{wd}_i(d,t)$ being the wake deficit generated by the WT i at a d distance and time t, and $u_x(p,t)$ the (undisturbed) ambient wind speed at the same position and time. According to [28], such a deficit can be computed, as a function of the turbine's current behaviour, as

$$\mathrm{wd}_i(d,t) = 1 - 0.5 C_{\mathrm{T,i}}(t)\left(1 + \frac{d}{4R}\right) - 1, \tag{4}$$

where $C_{\mathrm{T},i}(t)$ is the coefficient of thrust of WT i at time t.

Finally, the actual wind speed at any point of the farm can be computed by multiplying all the wake deficit contributions of all the WTs as

$$v_{\mathrm{w}}(x,y,t) = u_x(x,y,t) \Pi_{i=1}^{n_{\mathrm{t}}} \mathrm{wd}_i(d_i,t), \tag{5}$$

where n_{t} is the number of turbines in the farm. Despite the simplicity of the considered wake model, it captures the most critical aspects of the wake effects [28].

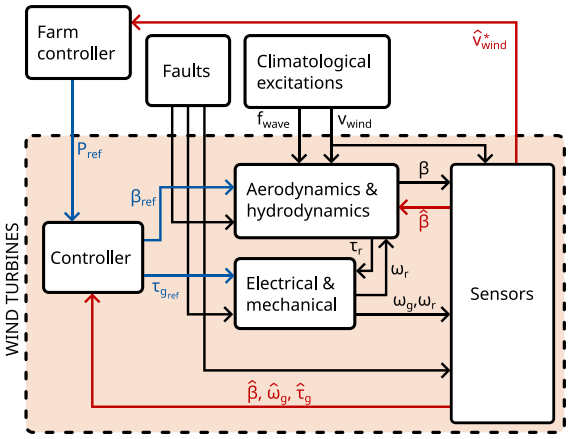

Fig. 3. Simplified block diagram of the WT subsystems.

2.2. Wind turbines

The wind turbines block contains a separate system for each WT in the farm including four (interconnected) subsystems for each WT:

i. the aero-hydrodynamic subsystem, which defines the dynamics of the blade pitch mechanism, calculates aerodynamic forces, and computes the motion of the floating platform due to the effect of wind and waves;
ii. the mechanical/electrical subsystem, which simulates the drive train and generator dynamics;
iii. the control subsystem, which computes reference optimal power trajectory by defining the required pitch angles based on a predefined control law; and
iv. the sensing subsystem, which samples measurable variables from different subsystems to recreate real sensor effects.

Fig. 3 provides a simplified diagram illustrating the interconnection among the different parts within the turbine subsystem, where β denotes the pitch angle, P the power, f_e the excitation force, ω the rotational speed, and τ the torque. The symbol $\{\hat{\cdot}\}$ stands for the measured value of the variable $\{\cdot\}$, the subscripts $\{\cdot\}_{\mathrm{ref}}$ and $\{\cdot\}_{\mathrm{ref_f}}$ denote the reference values of the WT or the farm controller, respectively, and $\{\cdot\}_{\mathrm{r}}$ and $\{\cdot\}_{\mathrm{g}}$ the variables referring to the rotor and generator subsystem, respectively. Note that the fault subsystem shown in Fig. 3 loads and sends the corresponding signal containing information about the fault initiation, duration, and severity (as defined by the user) in each subsystem.

Note that FOWLTY considers two different control levels. On the one hand, the controller at the farm level computes the power to be requested to each WT in the farm (further detailed in Section 2.3). On the other hand, the controller at the turbine level generates the reference values for the blade pitch angle (β_{r}) and the generator torque (τ_{r}) using the reference power from the controller at the farm level and the information from the different subsystems of the farm (see Fig. 3).

2.2.1. Wind turbine controller subsystem

As introduced in [19,29], the control strategy at the WT level is classified into two regions based on wind speed: partial-load and full-load. In the partial-load region, the blade pitch remains constant at zero (the pitch angle providing the maximum coefficient of power, C_{P}), and the generator power reference is determined as a function of the generator speed (see [29]), given optimal tip-speed-ratio. In contrast, in the full-load region, the generator power reference is held constant at the rated power, and the rotor speed is controlled to rotate at nominal

speed by adjusting the pitch angle of the blades, using a PI controller defined as follows,

$$\beta_{\mathrm{r}}(t) = k_{\mathrm{p}}\omega_{\mathrm{e}}(t) + k_{\mathrm{i}} \int \omega_{\mathrm{e}}(t)dt, \tag{6}$$

where $\omega_{\mathrm{e}}(t)$ is the error between the rated and actual rotational speed of the generator, and k_{p} and k_{i} are, respectively, the proportional and integral gains, computed as shown in [29]. Note that, in the Simulink code, a saturation term is included to avoid the integrator windup.

2.2.2. Aerodynamic and hydrodynamics subsystem

The *aerodynamic and hydrodynamic* subsystem shown in Fig. 3 is composed of the turbine aerodynamics and floating platform and mooring hydrodynamics. As introduced in [17], it is assumed that the thrust force (f_{t}) and torque exerted on the blades by the wind can be divided by three to obtain the contribution of each blade, in order to reflect the effect that a blade with a different angle has on the power production. Thus, the aerodynamic forces are computed as

$$\begin{aligned} \tau_{\mathrm{r}}(t) &= \sum_{i=1}^{3} \frac{1}{6} v_{\mathrm{w_r}}^3(t)\rho_{\mathrm{a}} A C_{\mathrm{P}}(\lambda, \beta_i)\omega_{\mathrm{rot}}^{-1}(t) \\ f_{\mathrm{t}}(t) &= \sum_{i=1}^{3} \frac{1}{6} v_{\mathrm{w_r}}^2(t)\rho_{\mathrm{a}} A C_{\mathrm{T}}(\lambda, \beta_i), \end{aligned} \tag{7}$$

where $v_{\mathrm{w_r}}$ is the relative wind speed affecting the turbine (obtained by subtracting the nacelle speed to the incoming wind speed), $A = \pi R^2$ the swept area of the blades, ρ_{a} the density of the air, $\lambda = r\omega_{\mathrm{rot}}/v_{\mathrm{r_r}}$ the tip-speed ratio, and C_{P} and C_{T} the power and thrust coefficients, respectively, defined by using two lookup tables obtained based on the geometry of the blades. Note that Eq. (7) is only valid for small differences between the pitch angles of the blades (see [18]).

The hydrodynamic effects on the floating platform where WTs sit are defined using the well-known Cummins' equation [30], considering multiple degrees-of-freedom (DoFs), defined as

$$\left(\boldsymbol{m} + \boldsymbol{\mu}_{\infty}\right)\ddot{\alpha}(t) = \boldsymbol{f}_{\mathrm{e}}(t) - \boldsymbol{f}_{\mathrm{hyd}}(t) - \boldsymbol{f}_{\mathrm{t}}(t), \tag{8}$$

where $\boldsymbol{m}$ is the mass/inertia matrix of the system, $\boldsymbol{\mu}_{\infty}$ the radiation added mass coefficient at infinite frequency, α the position of the device in the defined DoFs, $\boldsymbol{f}_{\mathrm{t}}(t)$ the transformation of the thrust force to the different hydrodynamic DoFs, and $\boldsymbol{f}_{\mathrm{e}}(t)$ the excitation force/torque due to the incoming waves. In Eq. (8), $\boldsymbol{f}_{\mathrm{hyd}}(t)$ refers to the sum of all the hydrodynamic forces arising from the motion of the platform. In the present model, three different forces are considered to define $\boldsymbol{f}_{\mathrm{hyd}}(t)$:

i. the radiation force as $\boldsymbol{f}_{\mathrm{rad}}(t) = \mathrm{k}_{\mathrm{rad}} \star \dot{\alpha}(t)$, where $\star$ denotes a convolution integral operation and $\mathrm{k}_{\mathrm{rad}}$ is the radiation force impulse response;
ii. the hydrostatic force as $\boldsymbol{f}_{\mathrm{h}}(t) = s_{\mathrm{h}}\alpha(t)$, with s_{h} the hydrostatic stiffness due to buoyancy;
iii. and the mooring force as $\boldsymbol{f}_{\mathrm{mo}}(t) = s_{\mathrm{mo}}\alpha(t) + d_{\mathrm{h}}\dot{\alpha}(t)$, with s_{mo} and d_{mo} the mooring stiffness and damping coefficients, respectively.

The calculation of the platform motion has been simplified by making several assumptions. First, a state-space model is used to approximate the radiation convolution term shown in Eq. (8). Second, only surge and pitch DoFs are considered since, considering the simplifying assumptions of the model, they are the only DoFs affecting the WT performance. This is because, on the one hand, since wind speed is calculated only at the hub height, computing the heaving motion of the platform will not affect the performance of the WT. On the other hand, since wind and waves are assumed to come from a single direction and no yaw mechanism is considered, it is unnecessary to consider the sway, roll, and yaw motions of the platform. However, if required, the implementation of the hydrodynamic model in FOWLTY allows the users to consider extra (or different) DoFs. Third, no hydrodynamic interaction effect is considered between platforms within the wind farm due to the large inter-device distances usually assumed between turbines. This means that waves radiated and diffracted from one platform are assumed to have no impact on the motion of the other platforms.

2.2.3. Electrical and mechanical subsystem

Finally, the *electrical and mechanical* block in Fig. 3 defines the behaviour of (i) the pitch actuator, (ii) the tower deflection, (iii) the electrical generator, and (iv) the drive train that connects the rotor to the generator. The pitch actuator system, which is defined separately for each blade so that it is possible to recreate faults on the different blades of the WT individually, is defined as

$$\ddot{\beta}(t) + 2d_{\mathrm{p}}\omega_{\mathrm{n_p}}\dot{\beta}(t) + \omega_{\mathrm{n_p}}^2\beta(t) = \omega_{\mathrm{n_p}}^2\beta_{\mathrm{ref}}(t), \tag{9}$$

with $\omega_{\mathrm{n_p}}$ and d_{p} as the natural frequency and damping of the pitch system, respectively.

The tower deflection dynamics are defined using a spring–damper system [19] as

$$\ddot{x}_{\mathrm{twr}}(t) = \frac{f_{\mathrm{t}}(t) - k_{\mathrm{twr}}x_{\mathrm{twr}}(t) - b_{\mathrm{twr}}\dot{x}_{\mathrm{twr}}(t)}{m_{\mathrm{twr}}}, \tag{10}$$

with m_{twr}, k_{twr} and b_{twr} as the mass, spring constant, and damping of the tower, respectively. The drive train model is defined as

$$\begin{aligned} \imath_{\mathrm{r}}\dot{\omega}_{\mathrm{r}}(t) &= \tau_{\mathrm{r}}(t) - s_{\mathrm{s}}\phi_{\mathrm{s}}(t) - d_{\mathrm{s}}\dot{\phi}_{\mathrm{s}}(t), \\ \imath_{\mathrm{g}}\dot{\omega}_{\mathrm{g}}(t) &= \tau_{\mathrm{g}}(t) + \frac{1}{n_{\mathrm{dt}}}\left(s_{\mathrm{s}}\phi_{\mathrm{s}}(t) + d_{\mathrm{s}}\dot{\phi}_{\mathrm{s}}(t)\right), \\ \dot{\phi}_{\mathrm{s}}(t) &= \omega_{\mathrm{r}}(t) - \frac{\omega_{\mathrm{g}}(t)}{n_{\mathrm{dt}}}, \end{aligned} \tag{11}$$

where $\imath$ is the inertia, s the stiffness, d the damping, n_{dt} the gear ratio of the drive train, ϕ_{s} the torsion of the shaft, and the subscripts $\{\}_{\mathrm{s}}$, $\{\}_{\mathrm{g}}$ and $\{\}_{\mathrm{r}}$ denote shaft-, generator- and rotor-related parameters, respectively (as in Fig. 3). Finally, the dynamics of the electric generator are defined using a first-order system as

$$t_{\mathrm{g}}\dot{\tau}_{\mathrm{g}}(t) = \frac{P_{\mathrm{ref}}(t)}{\omega_{\mathrm{g}}(t)} - \tau_{\mathrm{g}}(t), \tag{12}$$

with t_{g} as the generator time constant.

2.3. Farm controller and network

In the provided *farm-level controller*, the power reference of each WT is defined proportionally to the estimated available power at each WT, which, for a given turbine i, is computed as:

$$P_{\mathrm{a},i}(t) = \frac{1}{2}\rho_{\mathrm{a}} A \hat{v}_{\mathrm{w_r}}^3(t) C_{\mathrm{P},i}^{\mathrm{opt}}(t), \tag{13}$$

where $C_{\mathrm{P},i}^{\mathrm{opt}}(t)$ is the maximum efficiency that the turbine could achieve at that moment (which depends on β and ω_{g}). Hence, for a given demanded power (P_{d}) from the network, the farm controller calculates the reference power for each WT as

$$P_{\mathrm{ref},i}(t) = \frac{P_{\mathrm{d}}(t)P_{\mathrm{a},i}(t)}{\sum_{i=1}^{n_{\mathrm{t}}} P_{\mathrm{a},i}(t)}. \tag{14}$$

In terms of the *network operator*, there are different pre-defined modes, as originally defined in the *SimWindFarm* toolbox. By default, the absolute mode is selected, where the farm power output is specified by the user and is set to $P_{\mathrm{ref}} = 0.8 \times n_{\mathrm{t}} \times P_{\mathrm{nom}}$, P_{nom} being the reference power of the considered wind turbine. Additionally, the other working modes of the network operator include the delta mode, the balanced mode, the rate limiter, and the frequency control mode. For more information on such additional working modes, the interested reader is referred to [19].

3. Fault scenarios

This section describes how the considered faults are induced in FOWLTY. It is important to clarify that only non-critical faults that permit wind turbines to remain operational, though with a lower performance, are considered. Critical faults that require the turbines to be stopped are not included, aligning with the objective of the benchmark to develop FDI and FTC strategies that enable turbine operation in the presence of faults. However, the authors acknowledge the possibility

of incorporating critical faults in future versions of FOWLTY to test different control techniques.

Hence, the considered faults are classified into two groups: sensor faults and actuator faults, which are discussed in the following subsections. On the current version of FOWLTY, sensor faults can occur in blade pitch angle sensors, as well as in rotor and generator speed measuring sensors. On the other hand, actuator faults can occur in the blade pitch actuator and electric generator. These specific components were chosen because they all are required by the control system of the wind turbine and, therefore, a fault in any of them can have a significant impact on the turbine's performance if not appropriately addressed.

3.1. Sensor faults

Typically, control strategies rely on system measurements to compute the control input. In this case, as shown in Fig. 3, both the wind farm control strategy and the individual wind turbine control strategy utilise measurements of other variables to compute reference values. Therefore, it is crucial to consider potential faults in the system's sensors to develop FDI strategies that prevent turbine shutdown in the presence of such faults. In FOWLTY, five different sensor faults can be recreated:

Stuck sensor: This fault assumes that the measurement signal provided by the sensor becomes fixed and does not change with the actual measured variable: $\hat{x}(t + t_f) = x(t_f)$, where t_f is the time when the fault occurs.

Constant offset: The sensor signal has an offset relative to the original measured signal: $\hat{x}(t) = x(t) + x_{off}$, for any $t > t_f$, where x_{off} is an offset value determined by the user.

Constant gain: The signal from the sensor is a scaled version of the original measured signal but multiplied by a constant gain: $\hat{x}(t) = \delta_{gain} x(t)$, for any $t > t_f$, where δ_{gain} is a gain value determined by the user.

Diverging sensor: The sensor signal diverges from the actual measured signal, and the error between both signals increases with time: $\hat{x}(t) = x(t) + x_{div}(t)$, where $x_{div}(t)$ is an error that increases with time with a slope defined by the user.

Precision degradation: The signal-to-noise ratio of the measured signal decreases, resulting in increased noise in the sensor measurements[2]: $\hat{x}(t) = x(t) + \epsilon(t)$, for any $t > t_f$, where $\epsilon(t)$ is a random number with a standard deviation chosen by the user.

As mentioned at the beginning of the section, the sensor faults are assumed to affect three different sensor systems: blade pitch angle ($\hat{\beta}$), rotor angular speed ($\hat{\omega}_r$), and generator angular speed ($\hat{\omega}_g$) sensors. As shown in Fig. 3, such measurements are essential for the wind turbine controller and, thus, a fault on any of them could have severe consequences on the WTs' performance.

3.2. Actuator faults

As for the sensor faults, five different actuator faults can be simulated in FOWLTY. Four of the considered actuator faults are analogous to the sensor faults discussed in Section 3.1: the first is the stuck actuator, which remains fixed at a certain value after the fault occurs; the second is the constant offset, which introduces a constant error in the actuator relative to the reference signal; the third one is the constant gain fault, introducing a gain with respect to the reference in the signal provided by the actuator; and the fourth one is the diverging actuator, where the error between actual and reference signals of the actuator increase in time with a given slope defined by the user. In addition to these faults, FOWLTY includes an additional type of fault:

Change of dynamics: This fault causes the actuator to operate slower than expected, resulting in a longer time to reach the reference values. To simulate this fault, the parameters of the dynamical model of the actuators, such as the pitch actuator or the generator systems (see Eq. (9)), are adjusted. To this end, a constant value, denoted as α_g or α_p, is multiplied by the generator time constant t_g or the natural frequency of the pitch system ω_{n_p} to simulate the loss of effectiveness.

Noted that, as introduced in Section 2, the considered wind turbines do not include a yaw mechanism. Therefore, the main actuators used by the controllers to optimise the behaviour of the wind turbines are the pitch actuator and electric generator. Thus, by detecting these actuator faults, the control strategies can adjust the provided references to mitigate any major problems that may arise in the wind turbines.

4. Code validation

In this section, the results obtained with the FOWLTY platform are compared to those obtained with the well-known and established OpenFAST simulation tool [16]. To that end, three of the example test cases provided in OpenFAST are considered:

TC1: A single onshore turbine, to verify the aerodynamics and control effects.

TC2: A single offshore floating turbine, to verify the effects of the additional feature related to the hydrodynamics of the floating platform.

TC3: An onshore farm with three turbines, to verify the interactions between the turbines composing the farm.

Note that, in ***TC3***, the OpenFAST version able to run wind farms FAST.farm is used but, for the sake of consistency, it is referred to as OpenFAST in this paper.

The comparison is carried out using four different mean wind speeds (8, 12, 16, and 20 m/s), with one 5000 s-long simulation for each wind speed. The difference between FOWLTY and OpenFAST is quantified for different variables by means of the normalised root mean squared error (NRMSE) and is denoted as e combined with the subindex referring to the variable under analysis. By way of example, e_β stands for the difference of the blade pitch angle β between FOWLTY and OpenFAST, and is computed as

$$e_\beta = \sqrt{\frac{\sum_{i=1}(\beta_{OF}(i) - \beta_{FOWLTY}(i))^2}{\sum_{i=1}^{n_d} \beta_{OF}(i)^2}}, \tag{15}$$

with β_{OF} and β_{FOWLTY} the pitch angle obtained using OpenFAST and FOWLTY, respectively.

Note that, in all the cases, the considered turbine is the 5MW reference wind turbine designed by NREL [29]. Additionally, for ***TC2***, the considered floating platform is the spar-type platform introduced in [31], which is provided within the example cases in OpenFAST. In order to have the same wind files in both simulators, the wind from OpenFAST (generated using TurbSim) is adapted to be used in the FOWLTY platform. Since OpenFAST uses a 3D wind field and FOWLTY a 2D wind field, such adaptation is carried out by averaging the wind speeds from OpenFAST at the different heights affecting the blades of the turbine. Finally, it should also be mentioned that all the files used to run the aforementioned cases are provided as supplementary material of the present paper to ease the reproduction of the results.

[2] Note that, without this fault, measurements are considered to be perfect, with no noise. A functionality to add measurement noise will be added in a future version of FOWLTY.

Table 1
Obtained NRMSE values for ***TC1***.

v_w [m/s]	8	12	16	20
e_β	–	31.6	11.1	7.4
e_{τ_r}	12.7	7.4	5.5	8.1
e_{ω_r}	4.2	2	2.5	3.2
$e_{F_{thrust}}$	24.9	13.5	16.8	21.5
e_{P_g}	15.3	7.2	2.6	3.2
e_{τ_g}	11.5	6.8	3.1	3.6
e_{ω_g}	4.2	2	2.5	3.2

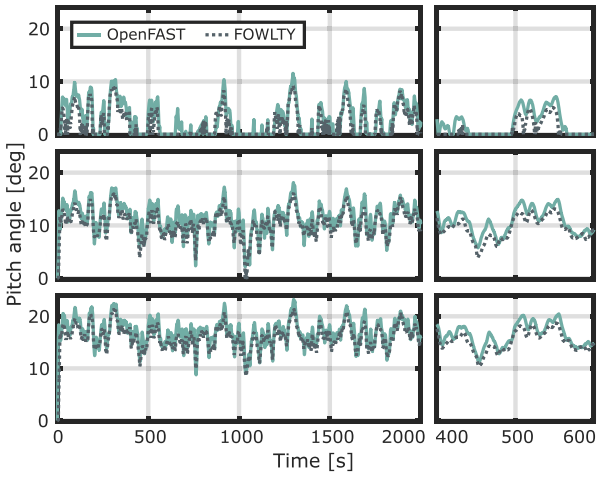

Fig. 4. Blade pitch angle for the single onshore turbine WT (***TC1***) with wind speeds (from top to bottom) of 12, 16, and 20 m/s, and a zoomed area on the right plots.

4.1. TC1: single onshore turbine

Table 1 shows the results obtained for ***TC1***. Note that no result for e_β with 8 m/s is shown since, for such wind speed, the pitch angle is zero. The highest difference (31.6%) happens on the pitch blade for 12 m/s wind speed since, with that speed the turbine is regulating the pitch angle at low angles, small errors amplify because of the considered performance index (see Eq. (15)). For instance, in this scenario, the most substantial disparity occurs at approximately 700 s, amounting to 6 degrees. However, it is crucial to note that the root mean square (rms) value of the error is around 1 degree. This implies that, on average, the difference between both signals is relatively minor.

In fact, Fig. 4 shows that, despite the error values shown in Table 1, the differences between the pitch angles obtained with the two codes are negligible.

As shown in Table 1, after the e_β with 12 m/s wind speed, the thrust force is the variable with the highest differences (up to 25%). This is, indeed, the main difference between the two codes since, even though the results from the two simulation codes show similar behaviour, a constant offset difference between the two thrust forces can be spotted (with an amplitude depending on the wind case). However, if needed, such offset difference could be minimised by a fine-tuning of the C_T look-up table (see Eq. (7)). By way of example, Table 2 shows how the obtained results improve if the thrust force from the C_T look-up table is (manually) tuned.[3] No other variable is shown since they are not affected by such a tuning.

For illustrative purposes, Fig. 5 shows the thrust force for the case obtaining the largest difference (8 m/s) with and without the mentioned tuning, in order to analyse similarities between FOWLTY and OpenFAST results. However, one could notice that OpenFAST is able to capture some high-frequency components that are not present in the results from the FOWLTY platform.

[3] The superscript $\{\cdot\}^t$ stands for the obtained value of the variable $\{\cdot\}$ after the tuning.

Table 2
Obtained NRMSE values for ***TC1*** with the tuned nacelle thrust force.

v_w [m/s]	8	12	16	20
$e_{F_{thrust}}$	24.9	13.5	16.8	21.5
$e_{F^t_{thrust}}$	6.7	9.2	11.9	15.1

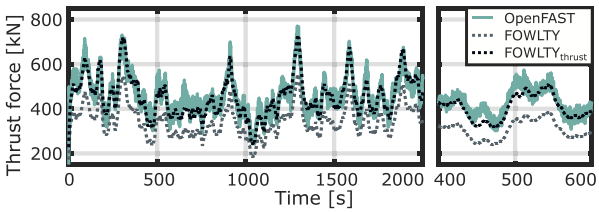

Fig. 5. Nacelle thrust force for the single onshore turbine WT (***TC1***) with 8 m/s wind speed, and a zoomed area on the right plot.

Since the variables affecting the turbine measurements and control strategies show a good agreement with OpenFAST (see Table 1), the FOWLTY platform can be considered verified for the development of FDI and FTC strategies of onshore single turbines. In general, the FOWLTY platform exhibits strong agreement on the generated power, which is the main variable utilised in this study to quantify the effect of the faults, since loads on the different subsystems are not computed. Hence, large errors in other variables do not result in large errors in power generation, which highlights the suitability of FOWLTY for evaluating FDI and FTC strategies.

4.2. TC2: single offshore floating turbine

As mentioned in the introduction, this case considers a single 5MW NREL reference turbine [29] floating on top of a spar-type floating platform [31]. First of all, it should be noted that the hydrodynamic coefficients used to define the floating platform model in FOWLTY (see Section 2.2) are either obtained from [31] or computed using the boundary element method solver Nemoh [32]. Additionally, the model for the mooring is described using a damping and a stiffness term (as introduced in [31]), in order to ensure that the same mooring lines model is implemented in both codes.

Table 3 shows the results obtained for ***TC2*** with and without the thrust force tuning introduced in Section 4.1. The results are similar to those obtained for the onshore case, with the highest difference obtained for the blade pitch angle with a 12 m/s wind speed and the rest of the results around the 10%. The main differences arise also due to discrepancies in the thrust force, although exacerbated by the fact that the thrust force affects the motion of the floating platform leading to large errors in e_{x_p} and e_{θ_p}. However, if the thrust coefficient is fine-tuned as in ***TC1***, the results are improved significantly (see the last three rows of Table 3).

By way of example, Figs. 6 and 7 show the worst and best case scenarios (20 and 8 m/s, respectively) for the platform motion in surge and pitch, respectively. In general, it can be appreciated that the results obtained without tuning the nacelle thrust have an offset with respect to the results from OpenFAST, which makes sense since, as shown in Eq. (8), the nacelle thrust force is an input of the floating system. However, once the nacelle thrust force is tuned to provide results more similar to those of OpenFAST, the offset on the platform motion is reduced, improving the obtained results (see Table 3).

From Figs. 6 and 7, one could notice that, for both platform surge and pitch motions, the results obtained for 8 m/s are more similar than those obtained for 20 m/s. This difference is, in part, due to the larger

Table 3
Obtained NRMSE values for ***TC2***.

v_w [m/s]	8	12	16	20
e_β	–	47.3	17.9	9.6
e_{τ_r}	14.5	11.3	13.8	15.8
e_{ω_r}	4.5	6.4	8.2	8.5
e_{P_g}	16.7	12.7	13.7	14.7
e_{τ_g}	12.7	10	12.9	14.0
e_{ω_g}	4.5	6.4	8.2	8.5
$e_{F_{thrust}}$	30.1	21.5	35.7	35.6
e_{x_p}	30.3	24.5	39.1	40.6
e_{θ_p}	30.9	30.5	45.6	46.3
$e_{F^c_{thrust}}$	10.8	13.4	20.5	19.4
$e_{x^t_p}$	13.1	14.1	22.3	20.2
$e_{\theta^t_p}$	13.2	21.6	32.1	30.5

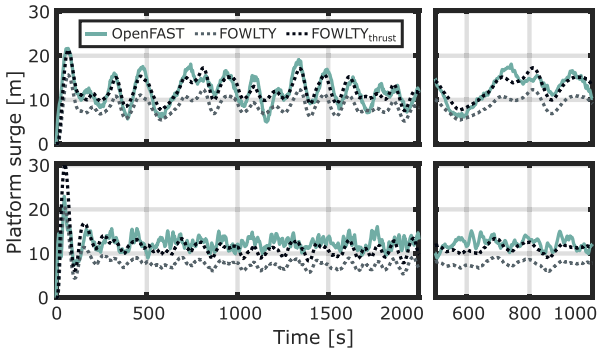

Fig. 6. Platform surge motion for the single offshore floating WT (***TC2***) with 8 (top) and 20 m/s (bottom) wind speeds, and a zoomed area on the right plots.

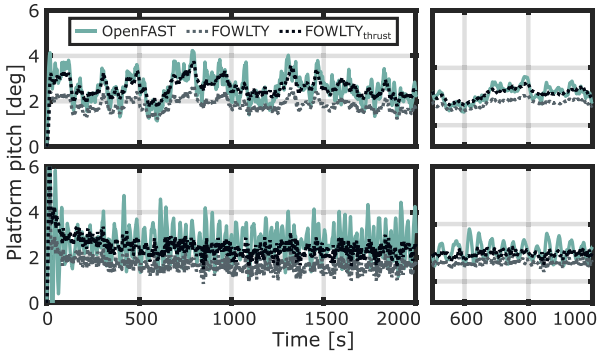

Fig. 7. Platform pitch motion for the single offshore floating WT (***TC2***) with 8 (top) and 20 m/s (bottom) wind speeds, and a zoomed plot on the right.

amount of high-frequency components in the results from OpenFAST, as opposed to those obtained from FOWLTY.

The results obtained for the platform surge motion (Fig. 6) are more similar than in the case of the pitch motion (Fig. 7), due to the faster dynamics of the latter, which are not well captured in FOWLTY.

4.3. TC3: three onshore turbines farm

This subsection presents the results for the last comparison case analysed in this study. The considered wind farm is composed of three onshore WTs (see [29]) arranged in line with the wind direction, as illustrated in Fig. 8, with an inter-turbine distance of 1000 m (approximately 7.75 times the diameter of the WT). It should be noted that, in FOWLTY, the wind speed is computed from the OpenFAST files only for the first wind turbine. Then, the calculation of the wind speed arriving

Table 4
Obtained NRMSE values for ***TC3***.

v_w [m/s]		8	12	16	20
e_β	WT1	–	27.1	7.8	2.9
	WT2	–	100.3	15.5	4.2
	WT3	–	128.8	16.5	4.1
e_{τ_r}	WT1	3.1	4.6	3.2	4.1
	WT2	34.4	12.6	3.7	4.1
	WT3	23.4	13.6	3.9	4.1
e_{ω_r}	WT1	1	1.6	1.8	1.9
	WT2	5.4	4.2	2	2
	WT3	3.7	4.5	2	2
$e_{F_{thrust}}$	WT1	19.4	16.3	20.7	22.2
	WT2	12	19.2	23.3	22.6
	WT3	17.5	17	22.8	22.4
e_{P_g}	WT1	4	5.1	1.4	0.7
	WT2	40.5	15.4	2.2	0.7
	WT3	26.2	16.6	2.9	0.7
e_{τ_g}	WT1	2.9	4.6	2.1	1.9
	WT2	34.3	12.6	2.7	1.9
	WT3	23.3	13.6	3.2	2
e_{ω_g}	WT1	1	1.6	1.8	1.9
	WT2	5.4	4.2	2	2
	WT3	3.7	4.5	2.1	2

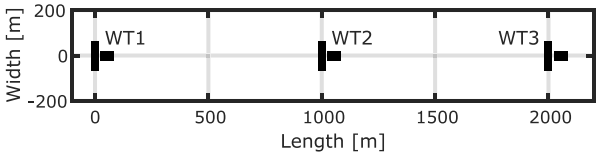

Fig. 8. Sketch of the wind farm considered in ***TC3***, with wind travelling from left to right.

at turbines downwind is performed (internally) using the propagation model considered in FOWLTY, taking into account wake effects (see Section 2.1).

Similarly to the previous test cases, the differences in the results obtained by the two simulation tools are shown in Table 4. In this case, the three WTs composing the farm are considered separately. One could notice that the results obtained for WT1 are similar to the results obtained in the single onshore turbine case ***TC1*** (see Table 1). In general, the results obtained for WT2 and WT3 are poorer than those obtained for WT1, which is mainly due to the differences in the propagation and wake models considered by the two simulation codes.

Fig. 9 shows the wind speed measured at the nacelle of WT3 for wind speeds of 8 and 20 m/s. The plot exclusively displays the wind measured at WT3, as it represents the most challenging scenario among the three turbines. A crucial observation is the increased high-frequency content in the wind from OpenFAST compared to FOWLTY. This discrepancy arises from the fact that FOWLTY models the wind field as a 2D plane at the nacelle height, while OpenFAST considers a 3D wind field. Consequently, the noise level in FOWLTY is mitigated since the wind is generated by averaging the wind speed from OpenFAST at various blade heights. Beyond such high-frequency content, Fig. 9 indicates a diminishing agreement between the wind simulations of both platforms as the wind speed decreases, with the 8 m/s case exhibiting the least concordance among all scenarios.

Among the results shown in Table 4, the biggest difference between the two simulators is obtained for the blade pitch angles of WT2 and WT3. In order to visually quantify how large such differences are, Fig. 10 shows the blade pitch angle at the three turbines for a 12 m/s wind speed. One could notice that the results from both simulators are similar, which seems to contradict the results shown in Table 4. However, such differences occur because pitch values are constantly around zero, which highly affects the obtained NRMSE measure.

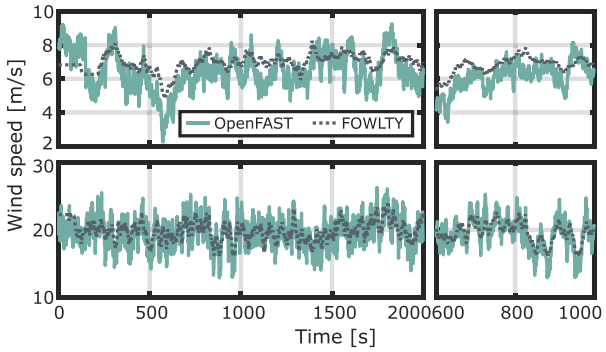

Fig. 9. Wind speed measured at the nacelle of WT3 of ***TC3*** for 8 m/s (top) and 20 m/s (bottom), with a zoomed area on the right plots.

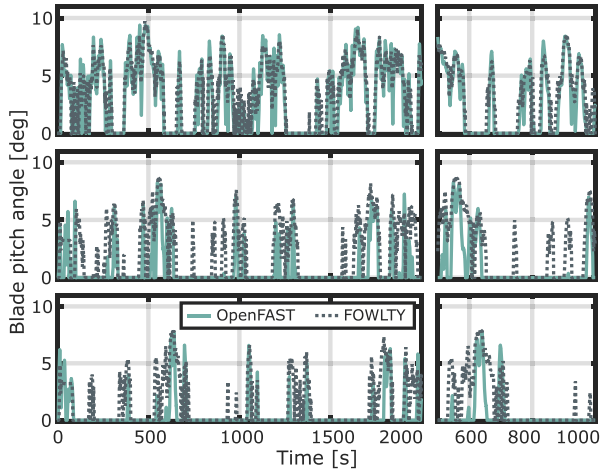

Fig. 10. Blade pitch angle of (from top to bottom) WT1, WT2, and WT3 for ***TC3*** with a 12 m/s wind speed.

Regarding power generation, Fig. 11 shows the generator power of the three WTs on the farms for an 8 m/s wind speed. It can be appreciated that the power generated by WT1 is virtually the same for the two simulators (4% of difference), but differences increase for WT2 and WT3. Such differences decrease when the wind speed increases, which is probably due to the discrepancies between the wind speeds arriving at each WT, as shown earlier in Fig. 9.

4.4. Computational times and usability

In this subsection, more general aspects of the computational times and usability of the two tools are discussed. First of all, Table 5 shows the computational times required by both codes to run the considered test cases. Note that such simulation times have been measured on a laptop with an Intel Core i7-8550U processor and 16 GB RAM. In addition to the required computational time, Table 5 shows the relative time required to run each case (first, compared to ***TC1*** with the same simulator and, second, compared to FOWLTY for the same test case). Thus, it can be observed that even though for the simplest case ***TC1*** OpenFAST is not much slower than FOWLTY (only 2.41 times slower), that difference increases with the complexity of the case (up to 5.36 times slower for ***TC3***). That is because, while FOWLTY does not get much slower with the complexity of the case (simulating ***TC3*** is 2.57 times slower than ***TC1***), such complexity highly affects the time required by OpenFAST (simulating ***TC3*** is 5.72 times slower than ***TC1***). Hence, the results shown in Table 5 clearly suggest that FOWLTY is a more viable option to develop control/estimation strategies (where

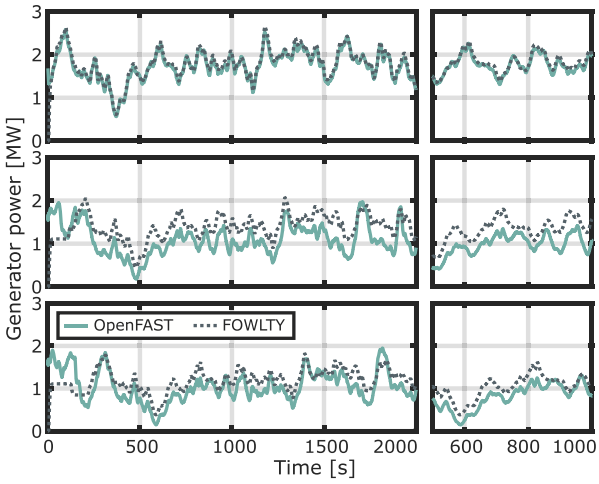

Fig. 11. Generator power obtained by (from top to bottom) WT1, WT2, and WT3 for ***TC3*** with a 8 m/s wind speed.

Table 5
Time (in seconds) required by both simulation codes (averaged over the four wind speeds) to run the considered cases. Additionally, the relative time required to run each case is shown in parenthesis: compared to TC1 with the same simulator on the left and compared to the fastest simulator for the same TC on the right.

	FOWLTY	OpenFAST
TC1	370 (1 - 1)	891 (1 - 2.41)
TC2	475 (1.28 - 1)	1706 (1.91 - 3.59)
TC3	950 (2.57 - 1)	5094 (5.72 - 5.36)

a large number of simulations are required) and, especially, for large farms.

While only the computational times required to run the cases are shown here, the computational time required to generate a wind case is also much larger for OpenFAST (using TurbSim in the present case) than for FOWLTY. Furthermore, the robustness of the code against initial conditions and time/space discretisation is also higher for the FOWLTY platform. In this sense, it should be mentioned that both simulators have been run with the same time step (0.05 s), while (at least) FOWLTY could use a higher time step, making the simulations quicker and achieving the same level of fidelity. Regarding the robustness against initial conditions, it should be highlighted that the initial conditions for OpenFAST needed adjustments to run the scenarios outlined in the previous sections, unlike the FOWLTY platform, where no modifications to the initial conditions were necessary. In fact, the authors observe that minor discrepancies in initial conditions can have a substantial impact on the simulation outcomes of OpenFAST, significantly affecting the results or even leading to non-converging simulations. Such a discrepancy in the sensitivity to initial conditions is understood to be due to the different model complexity levels considered in the two simulation tools. While FOWLTY considers simpler numerical models for the different components, OpenFAST includes more complex, non-linear models, which can introduce greater sensitivity to initial conditions and potentially lead to numerical instabilities.

Finally, the results demonstrate that FOWLTY is a reliable tool to develop fault detection and diagnosis techniques and health-aware control strategies. Some benefits can be highlighted compared to OpenFAST, such as the significantly lower computational costs and the full flexibility to recreate faults, which are of particular interest for the development of FDI and FTC strategies. Indeed, the objective of the FOWLTY platform is not to replace OpenFAST or any other higher-fidelity simulation tool. In fact, once the required FDI and FTC strategies are developed using FOWLTY, they should be rigorously tested using higher fidelity simulation codes (such as OpenFAST) for further validation and accuracy assessment.

Table 6
Description of the considered faults.

	Description	Subsystem	Type	WT	Time [s]
F1	Air on hydraulic pitch circuit	Pitch actuator	Change of dynamics	1	150–200
F2	Pitch sensor connection problems	Pitch sensor	Constant gain ($\delta_{\text{gain}} = 0$)	2	250–300
F3	Generator encoder slippage or wear	Generator sensor	Drift	4	350–400
F4	Electric generator incorrect functioning	Generator	Offset	7	450–500
F5	Hydraulic problems in pitch circuit	Pitch actuator	Stuck	8	550–600

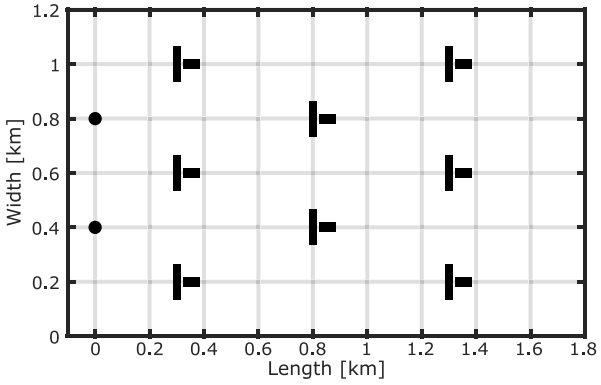

Fig. 12. Sketch of the considered offshore floating wind farm, with the measuring masts depicted with black dots and the wind travelling from left to right.

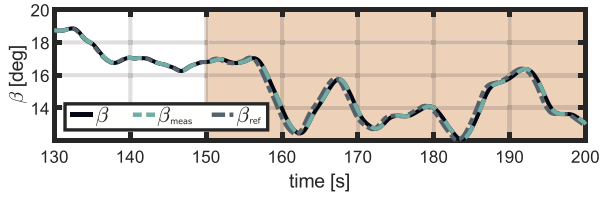

Fig. 13. Actual, measured, and reference blade pitch angle of WT1 during **F1**.

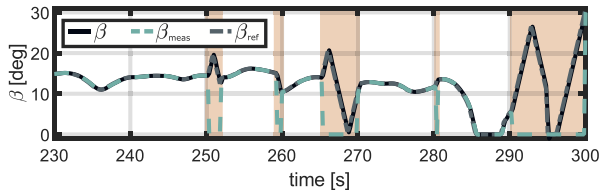

Fig. 14. Actual, measured, and reference blade pitch angle of WT2 during **F2**.

5. Faulty case study

In this section, an example offshore wind farm is presented to illustrate the impact of the considered faults on the behaviour of the wind turbines. By showing such an impact, the case study highlights the importance of implementing adequate FDI and FTC strategies that are able to minimise it. In particular, an offshore floating wind farm with eight WWTs is considered here, including two measuring masts upwind of the farm. The farm is arranged as shown in Fig. 12, with an inter-turbine distance of (approximately) three and four rotor diameters with respect to the turbines on the same line and the turbines behind, respectively. The grid size of the farm is 1200x1800 m (with a grid cell size of 10 m), the simulation is 600 s long with a time step of 0.2 s, and a 17 m/s wind speed with a 10 % turbulence intensity. Note that OpenFAST is not used in this section because replicating the specific faults considered in this study, if possible at all, would require significant coding adjustments and customisation, which are currently beyond the scope of the study.

In order to demonstrate the impact of different faults on the turbines' behaviour, this study investigates five distinct faults, modelled by means of the faults specified in Section 3) that are selected to emulate common real-world faults encountered in WTs:

F1 simulates **changes in the dynamics** of the hydraulic pitch system caused by air entering the circuit. This type of fault, as detailed in [18,33], results in a slower response time of the pitch actuator system.

F2 represents a fault related to the connection or cable of the pitch sensor. It is modelled as a brief, **constant-gain** (with a value of 0) fault that occurs over short time periods.

F3 reproduces the behaviour of a generator encoder **drifting** from its true measured speed due to wear and tear.

F4 simulates a malfunctioning electric generator that introduces a torque different from its reference value. This fault is defined with a **constant offset** value, as described in [18].

F5 replicates a pitch system that gets **stuck** due to friction and becomes immobile.

An overview of these considered faults is provided in Table 6.

Fig. 13 shows the blade pitch angle of WT1 during **F1**. While the actual pitch closely tracks the reference value with minimal error during regular operation, during **F1**, the actual pitch exhibits a delay relative to its reference value. The energy loss on WT1 during **F1**, in comparison to a fault-free simulation, is approximately 0.01%. While it may initially appear insignificant, the pitch angle differences between the three blades can result in an imbalanced force on the rotor, potentially leading to significant issues, especially fatigue.

In the case of **F2**, the fault is reproduced during short periods of time, as shown in Fig. 14. One could notice that, when the connection of the sensor fails, the WT controller thinks that the pitch angle is zero and tries to correct it by increasing the reference angle, which is followed by the actual blade. As in **F1**, the energy loss in **F2** is not substantial (approximately 0.04%), but given the nature of the reference values shown in Fig. 14, it may result in unbalanced forces on the rotor or pitch system. Additionally, depending on the wind speed, having such high pitch angles may result in over-speeding of the generator, which could cause major issues. These situations could be avoided (or mitigated) with an appropriate FDI strategy.

As shown in Fig. 15, even though there is a fault on the generator speed sensor, the speeds obtained at the healthy and faulty simulations are (virtually) the same. This is because, when **F3** occurs, the WT controller (incorrectly) detects a change in the generator speed and adjusts the control action to maintain the measured generator speed on the optimal trajectory. However, due to the disparity between measured and actual speeds, the generator does not rotate at the optimal speed, leading to a reduction in generated power (see bottom plot of Fig. 15). In this case, the energy loss concerning the healthy simulation is approximately 1.6%.

As illustrated in Fig. 16, **F4** introduces an offset in the torque applied by the generator with respect to the torque applied during the healthy simulation. In this specific case, an offset of 1 kNm is introduced, resulting in an energy loss of (approximately) 2.7%.

Finally, Fig. 17 shows the pitch angle of one of the blades on WT8 during **F5**. Despite the reference pitch value's attempt to initiate blade

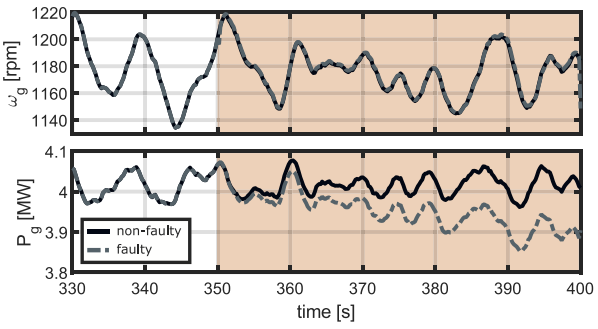

Fig. 15. Generator speed (top) and power (bottom) obtained by WT4 for a non-faulty simulation and during **F3**.

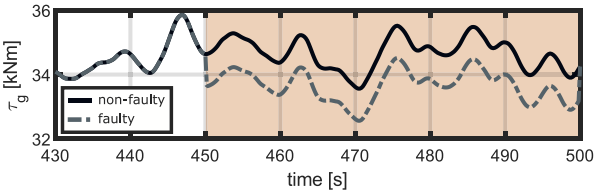

Fig. 16. Generator torque obtained by WT7 for a non-faulty simulation and during **F4**.

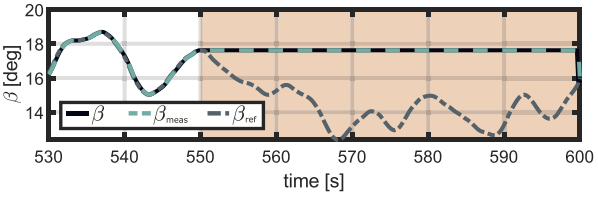

Fig. 17. Actual, measured, and reference blade pitch angle of WT8 during **F5**.

movement, the blade remains stuck at the angle it was when the fault occurred, resulting in an energy loss of 0.1%. As explained for **F1** and **F2**, the primary concerns with this fault are regarding to its impacts on other system components rather than the energy loss itself.

As mentioned before, FOWLTY is a platform for simulating fault scenarios and facilitating the development of estimation and control techniques for fault detection and impact mitigation. To this end, effects that do not directly impact the immediate behaviour of the turbines, such as the aforementioned unbalanced forces on the blades, are not included in the model. This approach maintains the simplicity and user-friendliness of the simulator. However, it should be highlighted that the goal of FOWLTY is not to replicate every detail of the effects that faults have on wind turbines, for which higher fidelity models should be employed.

6. Conclusions

The study presents a novel health-aware floating offshore wind farm simulator, referred to as FOWLTY, which enables users to easily replicate faults in various subsystems and components of wind turbines (WTs). FOWLTY builds upon the existing simulator *SimWindFarm* and, like its predecessor, employs a simplified representation of the WTs. This simplification is intentional, aiming to enhance comprehensibility and reduce computational complexity, particularly when developing fault detection and diagnosis techniques and health-aware control strategies for mitigating the impact of the faults.

FOWLTY is validated against the well-established OpenFAST simulator across three distinct scenarios that cover the whole range of capabilities of the tool: a single onshore WT, a single floating offshore WT, and a three-turbine onshore wind farm. Acknowledging the different ambitions of the FOWLTY and OpenFAST tools, the primary goal of FOWLTY is to generate similar results for the variables that influence wind turbine performance under faulty conditions, offering a quicker and more user-friendly simulation environment tailored for control/estimation strategy development. The results demonstrate that FOWLTY consistently yields results similar to OpenFAST for most of the simulation variables in all analysed cases. Moreover, it achieves this while requiring 2.4 to 5.4 times less simulation time, thereby confirming its practical usefulness for the scope it has been developed. Results demonstrate that the computational complexity of OpenFAST significantly increases as the number of turbines grows, while FOWLTY offers a linear increase in computational time with farm size, making it much more scalable for larger wind farms.

In addition to the validation, a case study featuring an eight-turbine floating offshore wind farm was conducted, simulating five different realistic faults. The results highlight the need for computationally efficient tools for the analysis of potential faults, and the development of fault detection and isolation (FDI) or fault-tolerant control (FTC) techniques that will enable mitigating the impact of faults on wind farm operations. FOWLTY has demonstrated to provide a framework that combines computational efficiency, ease of use and fault simulation capabilities that make it a valuable tool for early-stage development of FDI and FTC algorithms, which can later be validated using more detailed models like OpenFAST.

CRediT authorship contribution statement

Yerai Peña-Sanchez: Writing – review & editing, Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis. **Markel Penalba:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Torben Knudsen:** Writing – review & editing, Software, Resources, Conceptualization. **Vincenzo Nava:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **David Pardo:** Writing – review & editing, Supervision, Funding acquisition.

Download FOWLTY

FOWLTY can be downloaded from https://github.com/MGEP-Fluidos/FOWLTY.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yerai Pena-Sanchez reports financial support was provided by European Commission. Markel Penalba, Vincenzo Nava, David Pardo reports financial support was provided by Spanish ministry. Markel Penalba, Vincenzo Nava, David Pardo reports financial support was provided by Basque Government. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Yerai Peña-Sanchez is funded by the European Union's Horizon 2020 research and innovation program under the Maire Sklodowska - Curie grant agreement No. 101034297. Markel Penalba is funded by the research project TED2021-132767A-I00 funded by the Spanish Ministry of Science and Innovation and the Economic Development, Sustainability and Environment Department of the Basque Government under Grant No. KK-2024/00047 (ELKARTEK). David Pardo is funded by the Spanish Ministry of Science and Innovation project with references

PDC2021-121093-I00 (MCIN/AEI / 10.13039/501100011033/Next Generation EU); and the Basque Government, Spain through the Elkartek project BEREZ-IA (KK-2023/00012), and the Consolidated Research Group MATHMODE (IT1456-22) given by the Department of Education, and the Euskampus Foundation through the ORLEG-IA project. Additionally, both Vincenzo Nava and David Pardo received funding from the Spanish Ministry of Science and Innovation projects with references TED2021-132783B-I00, PID2019-108111RB-I00 (FEDER/AEI); the "BCAM Severo Ochoa" accreditation of excellence CEX2021-001142-S/ MICIN/AEI/10.13039/501100011033; the Spanish Ministry of Economic and Digital Transformation with Misiones Project IA4TES (MIA.2021.M04.008/NextGenerationEU PRTR); and the Basque Government, Spain, through the BERC 2022–2025 program.

Data availability

Data will be made available on request.

References

[1] IEA, Offshore Wind Outlook 2019, Tech. Rep., International Energy Agency, Paris, 2019.
[2] S. Cho, Z. Gao, T. Moan, Model-based fault detection, fault isolation and fault-tolerant control of a blade pitch system in floating wind turbines, Renew. Energy 120 (2018) 306–321.
[3] UN, The Sustainable Development Goals Report 2023: Special Edition, Tech. Rep., United Nations, 2023.
[4] J. McMorland, C. Flannigan, J. Carroll, M. Collu, D. McMillan, W. Leithead, A. Coraddu, A review of operations and maintenance modelling with considerations for novel wind turbine concepts, Renew. Sustain. Energy Rev. 165 (2022) 112581.
[5] M. Centeno-Telleria, H. Yue, J. Carrol, J.I. Aizpurua, M. Penalba, O&M-aware techno-economic assessment for floating offshore wind farms: A geospatial valuation off the North Sea and the Iberian Peninsula, Appl. Energy (ISSN: 03062619) 371 (2024).
[6] Z. Feng, M. Liang, Y. Zhang, S. Hou, Fault diagnosis for wind turbine planetary gearboxes via demodulation analysis based on ensemble empirical mode decomposition and energy separation, Renew. Energy 47 (2012) 112–126.
[7] H. Habibi, I. Howard, S. Simani, Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review, Renew. Energy 135 (2019) 877–896.
[8] Z. Hameed, Y. Hong, Y. Cho, S. Ahn, C. Song, Condition monitoring and fault detection of wind turbines and related algorithms: A review, Renew. Sustain. Energy Rev. 13 (1) (2009) 1–39.
[9] G.d.P. Leite, A.M. Araújo, P.A.C. Rosas, Prognostic techniques applied to maintenance of wind turbines: a concise and specific review, Renew. Sustain. Energy Rev. 81 (2018) 1917–1925.
[10] C. Saha, A.K. Singh, A review article on fault-tolerant control (FTC) and fault detection isolation (FDI) schemes of wind turbine, in: Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems, MCCS 2017, Springer, 2019, pp. 87–95.
[11] Z. Gao, X. Liu, An overview on fault diagnosis, prognosis and resilient control for wind turbine systems, Processes 9 (2) (2021) 300.
[12] S. Huang, X. Wu, X. Liu, J. Gao, Y. He, Overview of condition monitoring and operation control of electric power conversion systems in direct-drive wind turbines under faults, Front. Mech. Eng. 12 (3) (2017) 281–302.
[13] S. Pourmohammad, A. Fekih, Fault-tolerant control of wind turbine systems-a review, in: Proceedings of the IEEE Green Technologies Conference, IEEE-Green, Baton Rouge, USA, IEEE, 2011, pp. 1–6.
[14] Z. Gao, S. Sheng, Real-time monitoring, prognosis, and resilient control for wind turbine systems, Renew. Energy 116 (2018) 1–4.
[15] M.F. Howland, S.K. Lele, J.O. Dabiri, Wind farm power optimization through wake steering, Proc. Natl. Acad. Sci. 116 (29) (2019) 14495–14500.
[16] B. Jonkman, R. Mudafort, A. Platt, E. Branlard, M. Sprague, J. Jonkman, G. Hayman, G. Vijayakumar, M. Buhl, H. Ross, et al., OpenFAST/openfast: OpenFAST v3. 1.0, 2022, Zenodo [code] 10.
[17] P.F. Odgaard, J. Stoustrup, M. Kinnaert, Fault tolerant control of wind turbines–a benchmark model, IFAC Proc. Vol. 42 (8) (2009) 155–160.
[18] P.F. Odgaard, J. Stoustrup, M. Kinnaert, Fault-tolerant control of wind turbines: A benchmark model, IEEE Trans. Control Syst. Technol. 21 (4) (2013) 1168–1182.
[19] J.D. Grunnet, M. Soltani, T. Knudsen, M.N. Kragelund, T. Bak, Aeolus toolbox for dynamics wind farm model, simulation and control, in: Proceedings of the European Wind Energy Conference and Exhibition, EWEC, Warsaw, Poland, 2010.
[20] M. Soltani, T. Knudsen, T. Bak, Modeling and simulation of offshore wind farms for farm level control, in: Proceedings of the European Offshore Wind Conference and Exhibition, EOW, Stockholm, Sweden, 2009.
[21] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, J.-W. van Wingerden, A control-oriented dynamic wind farm model: Wfsim, Wind Energy Sci. 3 (1) (2018) 75–95.
[22] L. Cottura, R. Caradonna, A. Ghigo, R. Novo, G. Bracco, G. Mattiazzo, Dynamic modeling of an offshore floating wind turbine for application in the Mediterranean Sea, Energies 14 (1) (2021) 248.
[23] P.S. Veers, Three-Dimensional Wind Simulation, Tech. Rep., Sandia National Labs., Albuquerque, NM (USA), 1988.
[24] P.H. Madsen, D. Risø, Introduction to the IEC 61400-1 Standard, Risø National Laboratory, Technical University of Denmark, 2008.
[25] D. Quarton, et al., An International Design Standard for Offshore Wind Turbines: IEC 61400-3, Garrad Hassan and Partners, Ltd, Bristol, UK, 2005.
[26] P.A. Davidson, Turbulence: An Introduction for Scientists and Engineers, Oxford University Press, 2015.
[27] J.C. Kaimal, J. Wyngaard, Y. Izumi, O. Coté, Spectral characteristics of surface-layer turbulence, Q. J. R. Meteorol. Soc. 98 (417) (1972) 563–589.
[28] N.O. Jensen, A Note on Wind Generator Interaction, vol. 2411, Citeseer, 1983.
[29] J. Jonkman, S. Butterfield, W. Musial, G. Scott, Definition of a 5-MW Reference Wind Turbine for Offshore System Development, Tech. Rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2009.
[30] W. Cummins, The Impulse Response Function and Ship Motion, Tech. Rep. 9, Schiffstechnik, 1962, pp. 101–109.
[31] J. Jonkman, Definition of the floating system for phase iv of oc3, Tech. Rep., National Renewable Energy Lab. (NREL), Golden, CO (United States, 2010.
[32] LHEEA, NEMOH-Presentation, Laboratoire de Recherche en Hydrodynamique Énergetique et Environnement Atmospherique, 2017, https://goo.gl/yX8nFu. [Accessed Sepember 2022].
[33] L. Rakoto, J. Schorsch, M. Kinnaert, Modelling hydraulic pitch actuator for wind turbine simulation under healthy and faulty conditions, IFAC-PapersOnLine 48 (21) (2015) 577–582.