Faulty Wind Farm Simulation: An Estimation/Control-Oriented Model

Yerai Peña-Sanchez
(e-mail: yerai.pena@ehu.eus)
Mathematics Department, Euskal Herriko Unibertsitatea (EHU/UPV), 48940 Leioa, Spain.

Markel Penalba
Fluid Mechanics Department, Mondragon University, 20500 Arrasate, Spain.
Ikerbasque, Basque Foundation for Science, 48009 Bilbao, Spain.

Vincenzo Nava
TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain
Basque Centre for Applied Mathematics (BCAM), 48009 Bilbao, Spain

ABSTRACT: Wind energy is experiencing a rapid growth over the last decades. Wind turbines (WTs) are often placed together in wind farms in order to reduce their associated electricity generation cost. However, a large percentage of such cost is related to the operation and maintenance of the WTs. To reduce such costs, fault detection and isolation (FDI) and fault tolerant control (FTC) methods have become popular over the last decade. Most works develop FDI and FTC strategies for single WTs or WT subsystems. The present paper introduces a Simulink-based simulator able to simulate WT farms with the capacity to recreate different fault scenarios on the subsystems composing the WTs, in order to be used by researchers to develop FDI and FTC strategies for wind farms. This work shows a case study illustrating the effects that different faults have on a wind farm.

1 INTRODUCTION

Considering the ever-increasing worldwide energy demand, and the undeniable environmental impact associated with the combustion of fossil fuels, modifying the current energy mix is one of the most crucial challenges of this century. To this end, renewable energies (REs) represent a vast energy source, able to supply affordable, clean and sustainable energy. For instance, the annual global energy generation potential of offshore wind energy is estimated (IEA 2019) in approximately 420,000TWh, which represents more than 18 times the current global electricity demand. To reduce the cost of the electricity generation and make wind energy economically more appealing, wind turbines (WTs) are often placed together in farms. Additionally, in the last years there is a trend to install wind farms offshore, due to the better wind conditions. In fact, while most of the offshore wind farms were installed in shallow water, they are moving further into deeper water (Cho, Gao, & Moan 2018).

Due to the harsh environment where such offshore WTs operate, the different components of the turbines fail more frequently than for the onshore analogous, requiring increased maintenance schedules. Such faults represent a double drawback since, apart from the high maintenance cost, no energy is generated during the downtime. In fact, the operation and maintenance (O&M) cost associated to an offshore wind farm is estimated to be approximately 30% of the total income (for 20 years of operating life), as opposed to 10-15% for an onshore farm (Sørensen & Sørensen 2012, McMillan & Ault 2007). To reduce such costs, fault detection and isolation (FDI) and fault tolerant control (FTC) methods have become more popular over the last decade (Habibi, Howard, & Simani 2019). The fault information provided by the FDI system can be used to identify the faulty subsystem, while the FTC system allows keeping (if possible) the performance of the wind turbine at the desired level, while reducing the fault consequences. The interested reader is referred to (Habibi, Howard, & Simani 2019, Saha & Singh 2019, Gao & Liu 2021) and to (Habibi, Howard, & Simani 2019, Huang, Wu, Liu, Gao, & He 2017, Pourmohammad & Fekih 2011, Gao & Sheng 2018) for comprehensive reviews on FDI and FTC strategies applied to WTs, respectively.

Most of the works in the literature apply FDI and FTC strategies to WTs or WT subsystems (such as generator, drive-train, or pitch subsystems). However, since WTs are installed in farms, it is important to use wind farm simulations —as opposed to single WTs or WT subsystem simulations— to develop FDI and FTC strategies. By simulating the whole farm, extra measurements from other WTs can be feed to the FDI and FTC systems, which should reinforce the quality of the results. Additionally, it is important to differentiate when high- and mid- or low-fidelity model simulators need to be used. While high-fidelity models are developed to assess how WTs behave in different specific scenarios, low- and mid-fidelity models are suitable for estimation/control strategies development since, even though they do not characterise the exact behavior of WTs, they are able to represent the underlying behavior of WTs with a lower computational complexity. By way of example, in (Odgaard, Stoustrup, & Kinnaert 2009, Odgaard, Stoustrup, & Kinnaert 2013), the authors develop an openly-available WT model using simple descriptions of all WT subsystems so that non WT-experts can directly employ it and also serves as a benchmark to test FDI and FTC strategies. Another noteworthy example is the *SimWindFarm* (SWF) simulator (Grunnet, Soltani, Knudsen, Kragelund, & Bak 2010, Soltani, Knudsen, & Bak 2009), a wind farm simulator that was developed as part as the European-Union-funded project AEOLUS[1]. SWF uses a simple description of the subsystems of the WTs (similar to that used by (Odgaard, Stoustrup, & Kinnaert 2009)), to provide a wind farm model that can be used to develop wind farm control strategies. Finally, another control-oriented wind farm model is presented in (Boersma, Doekemeijer, Vali, Meyers, & van Wingerden 2018), called *WindFarmSimulator* (WFSim), with a higher fidelity description of the dynamics of the wind farm than that considered for the SWF, but also considerably slower.

This paper introduces a new wind farm simulator able to recreate faults on different components of the WTs composing the farm, termed *faulty wind farm simulator* (FWFS). The FWFS is based on the SWF simulator and, hence, is coded in Simulink (Documentation 2020). It allows the user to select any wind farm layout and pick at what moment the faults (one or several at the same time) occur. As for the SWF, the model used in the FWFS is simple enough so that estimation/control developers can use it to develop their FDI and FTC strategies while, at the same time, it represents the main dynamics of real wind farms. It should be noted that the simplicity of the model is not only appealing from a code-comprehension point of view, but also to simulate a large number of possible scenarios to test the FDI and FTC strategies[2].

The remainder of this paper is organised as follows: Section 2 introduces the Simulink model defining the dynamics of the farm, while Section 3 presents the fault scenarios considered for the simulator. Section 4 shows some examples of how the behavior of the WTs in the farm is affected when faults occur and, finally, Section 5 draws some conclusions and some possible future lines of research.

2 WIND FARM MODEL

The FWFS toolbox is a Matlab/Simulink-based simulator able to recreate wind farms, composed of any desired WT number and any layout. As mentioned in the introduction, the FWFS is an extension of the SWF simulator and, as such, they share the same code structure, where five main parts are differentiated: (i) the turbines block, where the simulation of the WTs composing the farm is carried out; (ii) the wind field block, where the wind affecting each WT is calculated; (iii) the farm control block, where the farm-level control is computed; (iv) the network part, which is composed of three blocks (network operator, grid, and network load), where the simulation of the grid is carried out; and (v) a post-processing block which, if clicked, provides several plots of the obtained results.

As the SWF toolbox, the objective of the FWFS is to be used for the design of estimators/controllers for WT farms and, therefore, some simplifications are assumed to reduce the computational burden of the model. In particular, the main assumptions (As) are:

A1 Constant (mean) wind direction and speed. In order to change the mean wind speed value, a new wind field (and wind farm) needs to be generated. Additionally, since the wind is always assumed to go parallel to the x axis, a new wind farm needs to be generated (with a rotated layout) in order to test how the farm behaves with other wind directions.

A2 The wind field is computed as a 2D plane at the hub height. Therefore, no wind shear or tower shadow effects are considered.

A3 Constant turbine yaw is considered, since the wind direction is assumed constant.

In the following subsections a description of the mentioned parts composing the FWFS are provided. However, since the main changes with re-

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

[2] This is particularly important for data-driven FDI strategies, where a big number of simulations needs to be run in order to train the FDI strategy.

spect to the SWF toolbox are within the turbines block, the other parts are very briefly described here, and the interested reader is referred to (Grunnet, Soltani, Knudsen, Kragelund, & Bak 2010) for a more comprehensive description of those parts.

2.1 *Wind field*

The wind field block computes the wind acting in each WT of the farm as a combination of a previously computed ambient wind field and the effect of the wakes generated by the WTs. To create such ambient wind field, the user needs to choose the mean wind speed, turbulence intensity, length and width of the wind field, and the size of the 2D grid. Figure 1 shows a wind field grid example with length and width of 400m and 300m, respectively, a grid size of 25m (both for x and y), and two WTs.

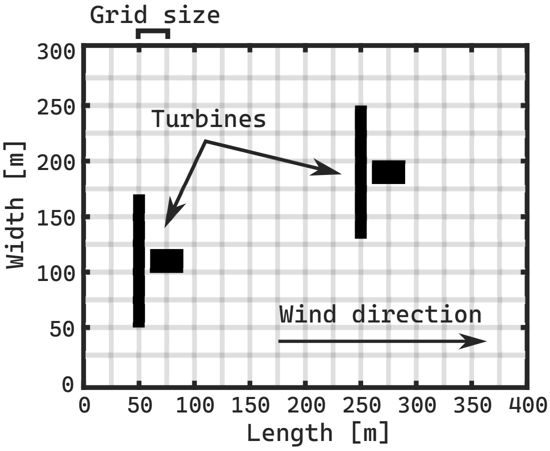

Figure 1: Wind field grid example.

With the input parameters defined by the user, the toolbox generates a wind field using Veers algorithm (Veers 1988), with a spectrum according to the recommendations in IEC (International Electrotechnical Commission) 61400-3 (Quarton et al. 2005) concerning offshore WTs, which states that IEC 61400-1 (Madsen & Risø 2008) values can be used for non-site-specific wind conditions. Additionally, Taylor's frozen turbulence hypothesis for inviscid flow (Davidson 2015) is assumed, which reduces the computational effort required to generate the wind field and simplifies the equations to calculate the wake effects that WTs have on the wind field.

Regarding wake effects, for a given WT, they can be purely defined by the behaviour of the WTs upwind. Additionally, due to the constant mean wind direction and the frozen turbulence assumption, those upwind and downwind turbines can be defined from the beginning, which simplifies the calculation of wake effects (Grunnet, Soltani, Knudsen, Kragelund, & Bak 2010). Three wake effects are considered: deficit, expansion, and center. The deficit quantifies the downwind speed decrease after going through a WT, the expansion describes the area behind the turbine affected by the wake, and the center defines the lateral position of the wake area.

2.2 *Turbines*

The turbines subsystem is where the simulation of the WTs composing the farm is carried out. It is composed of n_t subsystems, with n_t being the number of WTs in the farm, and in each one of them the dynamics of the specific WT are separated into five sections: (i) the control part, where the reference pitch angle (β_r) and generator reference power (P_r) are computed; (ii) the aerodynamics part, where the pitch subsystem is simulated, the aerodynamic forces are computed, and the motion of the tower due to the wind is calculated; (iii) the electrical part, where the drive train and generator dynamics are simulated; (iv) the sensing part, where the simulated variables are sampled in order to recreate the effect of real sensors; and (v) the subsystems fault section, where the signal containing information about the subsystems faults (defined by the user) is loaded and sent to the corresponding subsystems. Figure 2 shows a simplified block diagram of how the aforementioned parts of the WT are interconnected, where P_rf denotes the reference power commanded from the farm control, v_rot the wind speed over the rotor, β the pitch angle, ω_rot the rotational speed of the rotor, P_r the power reference for the generator, β_r the reference pitch angle, τ_rot the torque of the rotor, ω_gen the rotational speed of the generator, and the symbol $\hat{\{\cdot\}}$ stands for the measured value of $\{\cdot\}$.

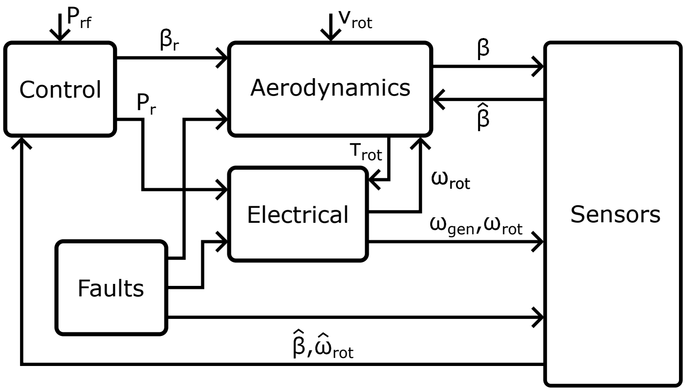

Figure 2: Simplified block diagram of the WT description in FWFS.

As SWF, FWFS provides the data for the NREL 5MW baseline turbine (Jonkman, Butterfield, Musial, & Scott 2009), but it is relatively simple to include more WT models if needed.

One of the differences with respect to SWF is that a separate pitch system is defined for each blade in order to recreate faults separately. To this end, it is assumed that the torque of each blade corresponds to a third of the total torque given by the three turbines. Hence, the aerodynamic effects of the WTs are computed using lookup tables as

$$\tau_{\mathrm{s}} = \sum_{i=1}^{3} \frac{1}{6} v_{\mathrm{rot}}^{3} \rho A C_{\mathrm{p}}(\lambda, \beta_i) \omega_{\mathrm{rot}}^{-1}$$
$$F_{\mathrm{t}} = \sum_{i=1}^{3} \frac{1}{6} v_{\mathrm{rot}}^{2} \rho A C_{\mathrm{t}}(\lambda, \beta_i), \quad (1)$$

where τ_{s} denotes the torque on the shaft, ρ the density of the air, $A = \pi r^2$ the area of the blades, C_{p} the power coefficient, $\lambda = r\omega_{\mathrm{rot}}/v_{\mathrm{rot}}$ the tip-speed ratio, r the radius of the turbine blades, F_{t} the tower thrust, and C_{t} the thrust coefficient. Note that the coefficients C_{p} and C_{t} are two lookup tables that can be obtained from the geometry of the blades. Additionally, it should be noted that the description of the aerodynamics shown in Equation (1) is only valid for small differences between the blades pitch angles (Odgaard, Stoustrup, & Kinnaert 2013).

An additional difference with respect to SWF is the definition of the pitch actuator system. While in SWF a first order model is defined, a (slightly) more realistic second order model is employed in FWFS in order to recreate faults more realistically. In particular, the pitch actuator model is the same as that in (Cho, Gao, & Moan 2018). The model of the remaining WT subsystems (drivetrain, generator, etc.) are left as defined in the original SWF toolbox (see (Grunnet, Soltani, Knudsen, Kragelund, & Bak 2010)).

2.3 *Controller and grid*

In the farm controller and network parts of the code, the same example codes for farm controller, network controller and grid models provided in the original SWF simulator are provided. Such provided controllers/models are just simple example codes, since the idea is for the developers to replace them with the controllers/models of their application case. The provided farm controller is a simple algorithm that proportionally distributes the estimated current production capacity and the total power demand from the network operator among all the WTs of the farm. Regarding the network operator, three different modes are included: (i) the absolute mode, where the farm power output is explicitly specified by the user; (ii) the delta mode, where a specific amount of reserve power has to be always ensured; and (iii) the frequency mode, where the objective is to keep the frequency of the grid constant.

2.4 *Post processing*

The post processing block provides three different tools to examine some of the obtained results: (i) a fatigue tool, based on (Buhl 2008), that computes rainflow counts and damage equivalent loads for shaft torsion and tower bending; (ii) a visualization tool that shows the wake that a turbine creates, which is useful to visualize the effect of that WT downwind; and (iii) another visualization tool showing the animation of the complete wind field wind speed, where interactions between the different turbines can be spotted.

Additionally, FWFS is provided with a Matlab script where the different system faults are defined and, once the Simulink simulation is finished, some of the obtained variables (such as generated electricity, blade pitch angles, etc.) are shown.

3 FAULT SCENARIOS

In this section the different fault scenarios considered in FWFS are introduced. Only faults that allow the turbine to keep operating (although in a sub-optimal mode) are considered. In other words, critical faults that require a complete closing down of the WT are not considered, since the objective of FWFS is to be used to develop FDI and FTC strategies that allow the operation of the WTs despite the presence of faults. The considered faults can be divided into three categories: sensor faults, actuator faults, and system faults.

3.1 *Sensor faults*

Sensor faults are important to consider since the control strategies of wind farms (in general) and individual WTs depend on measurements of some of the variables (as shown in Figure 2) in order to compute the reference values for the actuators to follow. Therefore, a sensor fault can render in a wrong reference signal that can damage the WT if it is not effectively dealt with. In particular, three different sensor faults are considered in FWFS:

- **Stuck sensor:** the measurement signal provided by the sensor is fixed and does not change with the measured variable: $\hat{x}(t+t_{\mathrm{f}}) = x(t_{\mathrm{f}})$, where t_{f} is the time instant where the fault occurs.

- **Constant offset:** the output of the sensor has an offset with respect to the original signal: $\hat{x}(t) = x(t) + x_{\mathrm{off}}$, for any $t > t_{\mathrm{f}}$, with x_{off} an offset value chosen by the user.

- **Precision degradation:** the noise on the measurements obtained from the sensor increases: $\hat{x}(t) = x(t) + \epsilon(t)$, for any $t > t_{\mathrm{f}}$, where $\epsilon(t)$ is a random number (with a standard deviation selected by the user).

The sensor faults are considered to happen in two different variables: pitch angle and rotor angular speed. Both variables are needed by the WT controller and, therefore, an incorrect detection of those faults could have fatal consequences.

3.2 *Actuator faults*

As for the sensor faults, three different fault types are differentiated in the actuators: stuck actuators, offset actuator, and loss of effectiveness of the actuator. The first two are analogous to those introduced for the sensors faults; i.e. the actuator gets stuck and does not change after the fault occurs (analogous to stuck sensor case), or the actuator acts on the system with a constant error with respect to the reference signal (analogous to offset sensor case). Regarding the third actuator fault considered, the loss of effectiveness refers to actuators that do not behave as expected and (generally) need more time to reach the commanded value. Such fault scenario is simulated by varying the model of the actuator system to make it act slower.

Actuator faults are considered in the blade pitch and generator subsystems, which are the main actuators that the control strategies use to optimise WTs' behaviour. Therefore, it is important to inform the controller of any actuator fault so that it can alter the control strategy if needed, in order to avoid any major problem in the WTs.

3.3 *System faults*

System faults refer to changes in the dynamics of specific subsystems. In FWFS, the only altered subsystem (so far) is the drive-train that connects the blades with the generator. Although faults in the drive-train are usually gradual, it is important to consider them since they affect the connection between the generator and the blades and, hence, it is crucial to have FDI and FTC strategies that are aware of this dynamic changes and can alter the pitch and torque references accordingly.

4 CASE STUDY

This section provides some examples of how the considered faults alter the behaviour of the WTs in the farm, emphasizing the importance of accurate FDI and FTC strategies.

4.1 *Description*

As an example, a wind farm composed of five WTs is selected, with the WTs arranged in a layout as shown in Figure 3. The simulations are 2000 s long, with a wind of 12 m/s mean speed, and a turbulence intensity of 0.1.

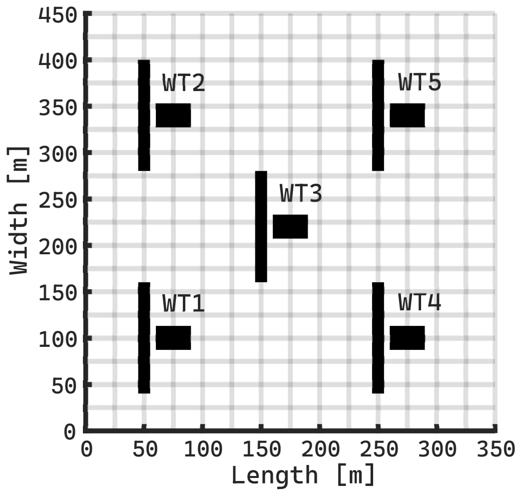

Figure 3: Diagram of the wind farm example.

4.2 *Nominal behaviour*

In order to define the nominal behaviour of the farm, this section discusses the obtained results when no fault occur. Figure 4 shows the pitch angles of the five WTs of the farm over the last 500 s of simulation. Note that a single pitch angle is shown for each turbine because, since no fault is considered, the pitch angle of the three blades is the same. The generated energy over the 2000 s for the five WTs is $E = \{2364, 2317, 2222, 1888, 1850\}$ kWh, 10642 kWh in total. Figure 4 shows that, while the first two wind turbines (WT1 and WT2) are constantly changing their pitch angles, the last two turbines in the farm (WT4 and WT5) keep their pitch angle fixed at zero degrees for long periods of time. This is because, while the wind that affect WT1 and WT2 is undisturbed and, hence, has a 12 m/s mean speed, the wind affecting WT4 and WT5 is disturbed from the wake created by the other upwind WTs and, therefore, has a mean speed below 12 m/s. Thus, WT4 and WT5 operate in the control region 2, where the controller acts in the generator torque (τ_{gen}) rather than in the blade pitch angles (Grunnet, Soltani, Knudsen, Kragelund, & Bak 2010). The effect of the wakes can also be observed on the generated energy, since WT1 and WT2 generate 25% more energy than WT4 and WT5.

4.3 *Faulty case*

By way of example, lets consider faults on the blade pitch subsystem (actuator and sensors) of several WTs of the farm. Table 1 shows the considered faults, in which WT of the farm are considered, and the time windows in which they occur. Note that, in the case of WT2, the offset added to the commanded pitch is set to three degrees.

Figure 5 shows the blades pitch angles of WT1 when the actuator of one of the blade's pitch system gets stuck. Undoubtedly, the effect of this kind

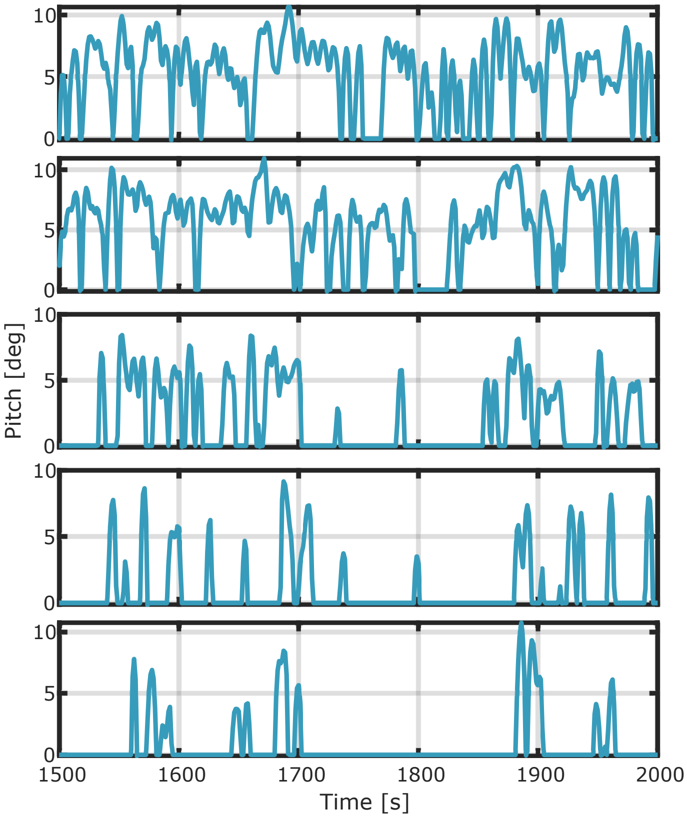

Figure 4: Blade pitch angles of the five WTs for a 500 s time-window with no faults. Turbines 1 to 5 from top to bottom.

Table 1: Pitch subsystem faults of the considered example.

Fault type	WT	time [s]
Stuck actuator	1	1550-1750, 1765-1950
Offset actuator	2	1550-1950
Stuck sensor	4	1550-1720, 1735-1950

of fault depends on the angle in which the actuator gets stuck. To illustrate such effect, the fault is induced in two different time windows. In the first one (from 1550 to 1750 s), the actuator gets stuck at (approximately) 8 degrees while, in the second one (from 1650 to 1950 s), the actuator gets stuck at zero degrees. Usually, the only problem this kind of fault creates is that the force exerted on the three blades is different and, hence, an unbalanced force is created on the turbine, which could severely harm the rotor. However, since when the actuator is stuck the pitch controller sends meaningless pitch reference commands, if the actuator gets unstuck in the wrong moment, as shown at $t = 1950$ s in Figure 5, the blade could try to follow such meaningless pitch commands and harm the WT. It should be noted that the energy loss on the two windows where the fault is induced are 0.5 and 2.1 kWh, respectively.

Regarding the fault in WT2, as shown in Figure 6, when the actuator provides a pitch angle which has an offset with respect to the commanded β, no critical effects can be observed. However, the effect of this fault could be worse for bigger offset values. In fact, the main problem of this type of fault is that the forces in the three blades are unbalanced

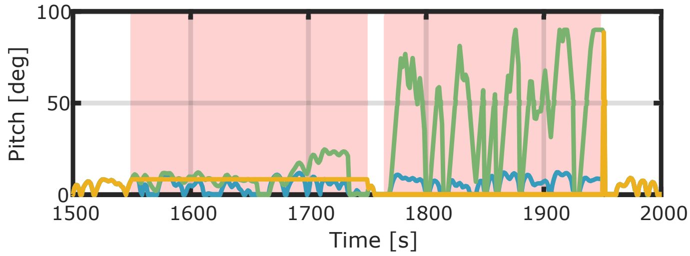

Figure 5: Blade pitch angles of WT1 for a 500 s time-window with a stuck pitch actuator fault. Blue line denotes β for the blades with no faults, while yellow and green lines denote β, β_r of the faulty blade, respectively. The red area indicates the time window in which the fault occurs.

which, as mentioned before, could severely harm the rotor. The energy loss in WT2 due to the fault is 0.5 kWh.

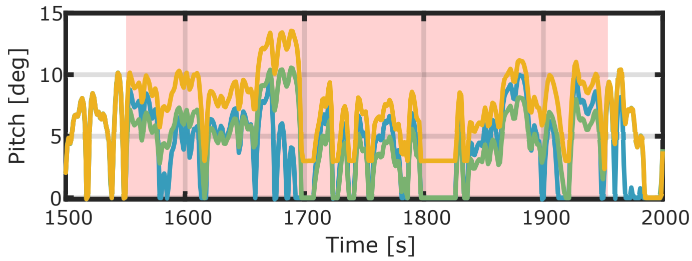

Figure 6: Blade pitch angles of WT2 for a 500 s time-window with an offset pitch actuator fault. Blue line denotes β for the blades with no faults, while yellow and green lines denote β, β_r of the faulty blade, respectively. The red area indicates the time window in which the fault occurs.

Finally, Figure 7 shows the effects of a stuck blade pitch sensor on WT4. As for the stuck actuator case, the effect of the fault depends on the value in which it gets stuck. To illustrate that, two different fault time windows are considered: from 1550 to 1720 s the sensor gets stuck at $\hat{\beta} = 0$ degrees, and from 1735 to 1950 s the sensor gets stuck at $\hat{\beta} \approx 3.5$ degrees. Since the pitch controller does not know that the sensor is stuck, during the first time window it increases the commanded pitch value to take it out from zero, and the blade follows the command, reaching more than 25 degrees, which would largely unbalance the forces on the three blades. Regarding the second time window, since the reference angle is (most of the time) lower than the value in which the sensor is stuck, no critical effects can be observed. Note that the energy loss due to the two faulty windows is 3.2 and 0.1 kWh, respectively.

In total, the energy loss due to the faults defined in Table 1 is 8.5 kWh. One could think that such a loss is insignificant, but considering that the faults only happen in 3 of the 5 turbines of the farm, and just for (approximately) 6.5 minutes (out of 33), it is a considerable loss. However, the most important problem regarding blade pitch subsystem faults is that they result in an off-balanced thrust on the blades, which can severely harm several parts of the Wt (such as the rotor, bearings etc.).

Finally, the time required to run the simulation

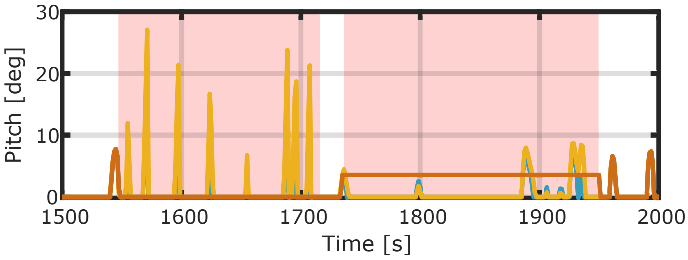

Figure 7: Blade pitch angles of WT4 for a 500 s time-window with an stuck pitch sensor fault. Blue line denotes β for the blades with no faults, while yellow and orange lines denote β, $\hat{\beta}$ of the faulty blade, respectively. The red area indicates the time window in which the fault occurs.

should be mentioned. In this case, using a laptop with a 12th gen. Intel i7 processor and 16GB of RAM, it takes around 20-25 s to run the 2000 s of simulation, i.e. ≈100 s of simulation per second, which makes it a relatively quick model suitable for the development of FDI and FTC strategies.

5 CONCLUSIONS

The present paper introduces a toolbox that can be used to generate and simulate wind farms able to recreate faults in different subsystems of the turbines composing the farm. It is shown, through an example, that the behaviour of the WTs effectively changes due to the faults and that the effect of some of the considered faults is higher than others. Additionally, some of the risks of not handling these faults are pointed out, highlighting the need of efficient FDI and FTC strategies.

The presented toolbox enables to develop estimation/control strategies for faulty (or non-faulty) wind farms, boosting the creation of FDI and FTC strategies which will reduces the electricity generation cost of wind energy.

Some of the following steps to upgrade the FWFS toolbox are: (i) validate the model against other existing WT fault simulators; (ii) include more faults in different subsystems and, specifically, critical faults so that developers/researchers working in that area can also use the toolbox; and (iii) extend the model to be able to simulate farms in different scenarios, such as floating offshore farms.

6 ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Maire Sklodowska - Curie grant agreement No. 101034297.

REFERENCES

Boersma, S., B. Doekemeijer, M. Vali, J. Meyers, & J.-W. van Wingerden (2018). A control-oriented dynamic wind farm model: WFSim. *Wind Energy Science 3*(1), 75–95.

Buhl, M. L. (2008). *MCrunch user's guide for version 1.00.* National Renewable Energy Laboratory Denver, CO.

Cho, S., Z. Gao, & T. Moan (2018). Model-based fault detection, fault isolation and fault-tolerant control of a blade pitch system in floating wind turbines. *Renewable Energy 120*, 306–321.

Davidson, P. A. (2015). *Turbulence: an introduction for scientists and engineers.* Oxford university press.

Documentation, S. (2020). Simulation and model-based design.

Gao, Z. & X. Liu (2021). An overview on fault diagnosis, prognosis and resilient control for wind turbine systems. *Processes 9*(2), 300.

Gao, Z. & S. Sheng (2018). Real-time monitoring, prognosis, and resilient control for wind turbine systems. *Renewable Energy 116*, 1–4.

Grunnet, J. D., M. Soltani, T. Knudsen, M. N. Kragelund, & T. Bak (2010). Aeolus toolbox for dynamics wind farm model, simulation and control. In *Proceedings of the European Wind Energy Conference and Exhibition (EWEC), Warsaw, Poland.*

Habibi, H., I. Howard, & S. Simani (2019). Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review. *Renewable energy 135*, 877–896.

Huang, S., X. Wu, X. Liu, J. Gao, & Y. He (2017). Overview of condition monitoring and operation control of electric power conversion systems in direct-drive wind turbines under faults. *Frontiers of Mechanical Engineering 12*(3), 281–302.

IEA (2019). Offshore wind outlook 2019. Technical report, International Energy Agency, Paris.

Jonkman, J., S. Butterfield, W. Musial, & G. Scott (2009). Definition of a 5-MW reference wind turbine for offshore system development. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Madsen, P. H. & D. Risø (2008). Introduction to the IEC 61400-1 standard. *Risø National Laboratory, Technical University of Denmark*.

McMillan, D. & G. W. Ault (2007). Quantification of condition monitoring benefit for offshore wind turbines. *Wind engineering 31*(4), 267–285.

Odgaard, P. F., J. Stoustrup, & M. Kinnaert (2009). Fault tolerant control of wind turbines–a benchmark model. *IFAC Proceedings Volumes 42*(8), 155–160.

Odgaard, P. F., J. Stoustrup, & M. Kinnaert (2013). Fault-tolerant control of wind turbines: A benchmark model. *IEEE Transactions on control systems Technology 21*(4), 1168–1182.

Pourmohammad, S. & A. Fekih (2011). Fault-tolerant control of wind turbine systems-a review. In *Proceedings of the IEEE Green Technologies Conference (IEEE-Green), Baton Rouge, USA*, pp. 1–6. IEEE.

Quarton, D. et al. (2005). An international design standard for offshore wind turbines: IEC 61400-3. *Bristol, UK: Garrad Hassan and Partners, Ltd*.

Saha, C. & A. K. Singh (2019). 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), Jharkhand, India*, pp. 87–95. Springer.

Soltani, M., T. Knudsen, & T. Bak (2009). Modeling and simulation of offshore wind farms for farm level control. In *Proceedings of the European Offshore Wind Conference and Exhibition (EOW), Stockholm, Sweden.*

Sørensen, J. D. & J. N. Sørensen (Eds.) (2012). *Wind Energy Systems: Optimising Design and Construction for Safe and Reliable Operation*, Volume 1. Woodhead Publishing Series in Energy.

Veers, P. S. (1988). Three-dimensional wind simulation. Technical report, Sandia National Labs., Albuquerque, NM (USA).