Data–Driven Wake Steering Control for a Simulated Wind Farm Model

Silvio Simani[1,*], Saverio Farsoni[1], and Paolo Castaldi[2]

[1]Department of Engineering, Ferrara University, Ferrara, FE, 44122 Italy

[2]Department of Electrics and Informatics, Bologna University, Bologna, BO, 40121 Italy

Abstract: Upstream wind turbines yaw to divert their wakes away from downstream turbines, increasing the power produced. Nevertheless, the majority of wake steering techniques rely on offline lookup tables that translate a set of parameters, including wind speed and direction, to yaw angles for each turbine in a farm. These charts assume that every turbine is working well, however they may not be very accurate if one or more turbines are not producing their rated power due to low wind speed, malfunctions, scheduled maintenance, or emergency maintenance. This study provides an intelligent wake steering technique that, when calculating yaw angles, responds to the actual operating conditions of the turbine. A neural network is trained live to determine yaw angles from operating conditions, including turbine status, using a hybrid model and a learning-based method, *i.e.* an active control. The proposed control solution does not need to solve optimization problems for each combination of the turbines' non-optimal working conditions in a farm; instead, the integration of learning strategy in the control design enables the creation of an active control scheme, in contrast to purely model-based approaches that use lookup tables provided by the wind turbine manufacturer or generated offline. The suggested methodology does not necessitate a substantial amount of training samples, unlike purely learning-based approaches like model-free reinforcement learning. In actuality, by taking use of the model during back propagation, the suggested approach learns more from each sample. Based on the flow redirection and induction in the steady state code, results are reported for both normal (nominal) wake steering with all turbines operating as well as defective conditions. It is a free tool for optimizing wind farms that The National Renewable Energy Laboratory (USA) offers. These yaw angles are contrasted and checked with those discovered through the resolution of an optimization issue. Active wake steering is made possible by the suggested solution, which employs a hybrid model and learning-based methodology, through sample efficient training and quick online evaluation. Finally, a hardware-in-the-loop test-bed is taken into consideration for assessing and confirming the performance of the suggested solutions in a more practical setting.

Keywords: Fault diagnosis, Neural network, Data– driven approach, Model–based scheme, Wind farm simulator, Hardware–in–the–loop test–rig.

1. INTRODUCTION

Wake steering can increase the net power produced by a wind farm by yawing up-stream turbines to redirect their wake away from downstream turbines, as shown *e.g.* in [1-3] for turbines in commercial wind farms. These works proposed the use of a lookup table that for each Wind Speed (WS), Wind Direction (WD), and Turbulence Intensity (TI) sequence provides the yaw angles to maximise net power. This table is generated offline by solving an optimisation problem for each WS, WD and TI sequence exploiting a wake model. For their wake models, *e.g.* the works [1-3] suggested in particular the employment of the lifting line [4]), Gauss Curl–Hybrid [5], and Gaussian [6] models, respectively.

One common key aspect not considered by these works is the faulty condition (and the failure, *i.e.* the shutdown) of one or more turbines in a farm, which often occurs due to low wind speed (wind speed below the cut–in speed), anomalous working conditions, routine maintenance, or emergency maintenance, as remarked *e.g.* in [7, 8]. Therefore, by simply applying a lookup table, which does not consider the turbine healthy status, can lead to non- optimal control, such as yawing a turbine to direct its wake away from a downstream turbine that is not working properly, thus decreasing the power of the upstream turbine without gaining any power increase from the downstream turbine. To account for turbine conditions, the lookup table approach would need to optimise over turbine efficiency and availability; this could be prohibitively expensive for a large wind farm due to the large number of possible combinations of turbine availability.

One possible approach to take into account turbine efficiency reduction or even their shutdown due to faults or failures while avoiding optimisation over all combinations of turbine availability is through learning by using real–time data. Learning– based control, sometimes in connection with Reinforcement Learning (RL) methods, has recently demonstrated interesting properties, and it may be considered for the problem of

*Address correspondence to this author at the Department of Engineering, Ferrara University, Ferrara, FE, 44122 Italy; Tel: +390532974844;
E-mail: silvio.simani@unife.it

E-ISSN: 2409-9694/23

wake steering when all wind turbines are active. For example, the works [9] and [10] exploited RL for maximising wind farm power generation. On the other hand, [11, 12] applied RL to the related problem of power tracking, thus matching the output power of the wind farm asa function of the electricity grid.

Note that a key challenge with purely learning–based methods is that they can require a long time to train; therefore, hybrid model– and learning–based solutions, such as the solutions proposed in [13], showed to train faster than RL. To this end, hybrid methods have also shown interesting features for real–world problems, such as building control, but have not been tested on wind farm control.

It is worth noting that the use of Machine Learning (ML) tools, such as Deep Learning (DL) is available in many cases [14], for example in computer vision [15], speech recognition [16]. Thus, researches are involved in fault detection [17, 18] of Wind Turbines (WT)s, as for in this paper. With respect to other situations, it is normal to have data from healthy conditions and only a few sequencesof the faulty ones [19]. In this paper, a scheme was exploited to extend the knowledge acquired from different WTs and extend it to different. WTs can be similar, but with different working conditions, asshown *e.g.* in [20, 21]. The development of Things (IoT) can allow the collection of data from remote and different tools, such as Supervisory Control AndData Acquisition (SCADA) devices.

Where as typical ML models are predicated on the idea that both the training and testing data belongto the same data distribution, TL aims to improve learners' performance by transferring information from a related domain. For example, in ML, the concept of Transfer Learning (TL) is influenced by how people learn, which involves applying priorknowledge to solve difficulties. For instance, if a person can ride a bike, learning to drive a car willgo more swiftly than starting from scratch with no prior driving experience. As will be seen in thepaper, TL allows ML models to transfer learned information from source domains to a target domain in order to enhance the effectiveness of the target learning function, even though both the source and target domains have different data distributions [22]. Moreover, it is possible to transmit data samples from the source domain to the target model to enhance learning [23]. There are still significant research gaps that need to be filled, despite the fact that TL is becoming more and more popular and has been used in various sectors, including defect detection.

The methods that are currently accessible only use one computer as a source [24]. In order to learn features from the source and target domains together, domain adaptation is proposed in the study as a fault detection technique. A support vector machine classifier is then used to predict problems [25]. In order to forecast bearing inner race, ball, and outer race problems under fluctuating working conditions, a TL approach was applied. In order to classify gear pitting flaws, the publication [26] suggested an enhanced deep neural network optimized by a particle swarm optimization methodology anda regularization method.

Although labeling data is a challenging operation when fault data is unbalanced, the information can still be sent to a destination without labeled data. When the labeled data was unavailable for the target, the greatest mean discrepancy was employedto reduce the difference between the source and target domains. Several DL models, such as the sparse autoencoder [27] or the Convolutional Neural Network (CNN) [28], are employed for condition recognition in addition to the domain adaptation utilizing the maximum mean discrepancy. In order to extract transferable features from the unprocessed vibration data, the article suggested a feature-based CNN [29].

Some other researches, which combine TL with DL for fault diagnosis, focuses on TL that allow data to be combined across numerous devices. To cope with big data sets and capture nonlinear trends from diverse measures, an approach such as DL is required rather than shallow machine learning models. TL with DL is used to transfer gained knowledge from vast fault history WT to scarce fault history WT, which is insufficient to train typical MLmodels. In fact, in contrast to prior research by the authors [30], this paper employs DL and TL methods for fault diagnostic application to WT SCADA data. Other well–established traditional approaches showing the application of artificial intelligence tools for fault diagnosis for power plants and other industrial applications can be found *e.g.* in [31-35].

In particular for this paper, it proposes a Hybrid Model– and Learning–Based (HMLB) approach to wake steering for the first time, with the aim of man-aging wind turbine working conditions that can varydue to their availability after faults and shutdown. Note that this HMLB scheme was developed earlier for dynamic control, whilst this work develops a similar approach for steady–state conditions. This method could also be seen as model–based or whitebox RL, which is able to

take into account possible faults and failures affecting the wind farm.

Therefore, the solution considered in this paper relies on Neural Networks (NN)s to describe the nonlinear dynamic analytical link between the windfarm measurement and the control signal. The chosen network architecture belongs to the Nonlinear Auto-Regressive with eXogenous (NARX) input prototype, which can describe dynamic relationships along time. The training of the NN exploits standard training algorithm, that processes the data acquired from the process [36]. The developed strategies are verified by means of a high–fidelity simulator, which describes the normal and the faulty behaviour of a wind farm plant. The achieved performances are assessed in the presence of uncertainty and disturbance effects, thus validating the robustness features of the proposed schemes. The effective-ness of the proposed solutions is also verified and validated using a more realistic test–bed consisting of a Hardware–In–the–Loop experimental tool. Thisshould to serve a more realistic application of the proposed schemes. It is worth noting the main contributions of this work. With respect to previous investigations by the authors, see *e.g.* [37], this pa- per develops an Active and Learning–Based Control (ALBC) strategy that is verified and validated also with respect to a real–time test–bed.

The manuscript is organised as follows. SectionII presents the HMLB solution for the steady– state setting considered in this paper, including how it applies for real–time (online) control and offline training. The training phase of the ALBC approach requires the generation of data through a model; therefore, a learning–enabled version of the wake model is presented in Section III. Section IV recalls the wind park simulator. This solutionis exploited for the active wake steering of a wind farm in Section V. A more realistic validation of the achieved results is addressed in Section VI using the Hardware–In–the–Loop (HIL) tool. Finally, Section VII summarises the achievements of the paper and highlights possible directions of future works.

2. HYBRID MODEL– AND LEARNING–BASED CONTROL METHOD

The active control method exploited here [13]is an ALBC method, originally developed for dynamic control. The same approach can be applied in steady–state conditions, specifically for active wake steering. This approach could also represent a model–based or white box RL. In particular, Section II-A explains how this approach can be used for real–time (online) active control of a particular wind farm, whilst Section II-B shows the offline training mechanism.

A. Online Active Control

In real–time conditions, the active control strategy passes the current state and the exogenous inputs to a policy, $\mathcal{P}^*_\theta$, with optimised parameters θ^*, which generates the control action. In this steady– state case, the considered policy takes in only the exogenous inputs: the Wind Speed (WS), v_w, the Wind Direction (WD), φ, and the turbine status. The policy is trained for a particular wind farm, where the number of turbines and their location is fixed. The vector indicates whether each turbine in the farm is active, inactive, or faulty. The policy generates a vector, γ, with a yaw angle, γ_i, for each i–th turbine in the farm. Thus, ALBC requires to solve the problem defined by the relation of Eq. (1):

$$\gamma = \mathcal{P}_{\theta^*}(v_w, \phi, \mathcal{S}) \tag{1}$$

The policy $\mathcal{P}^*_\theta$ is obtained by using a NN, as recalled in this section. The optimised parameters, ϑ^*, represent the weights and biases of the NN, obtained during the training as described in Section II-B.

This study proposes a different data–driven approach, based on NNs, which is exploited to implement the fault diagnosis block. This section briefly recalls their general structure and properties, which are used to implement the policy generator for the signals $\gamma(t)$.

Therefore, a NN is realised in order to reproduce the behaviour of the policy relation of Eq. (1) using a proper set of input and output measurements. The NN structure consists of different layers of neurons [36], modelled as a static function f. This function is described by an activation function with multiple inputs properly weighted by unknown parameters that determine the learning capabilities of the whole network.

A categorisation of these learning structures concerns the way in which their neurons are connected each others. This work proposes to use feed– forward network, also called multi-layer perceptron, where the neurons are grouped into unidirectional layers. Moreover, this multi-layer perceptron is provided with a tapped delay line, which

is a feed– forward network whose inputs come from a delay line. This study proposes to use this solution, defined as 'quasi–static' NN, as it represents a suit- able tool to predict dynamic relationships between the input–output measurements and the considered policy function $\mathcal{P}^*_\theta$. In this way, a NARX description is obtained, since the nonlinear (static) network is fed by the delayed samples of the system inputs and outputs selected by the analysis tool described in Section IV. Indeed, if properly trained, the NARX network can estimate the current (and the next) control vector $\gamma(t)$ on the basis of the selected past measurements of system inputs and outputs $u_l(t)$ and $y_j(t)$, which represent the wind turbine status $\mathcal{S}$. Other inputs of the NN are the WS and WD, *i.e.*, v_w and φ. Therefore, with reference to the policy generator, which is used to design the yaw vector γ, this NARX network is described by the relationof Eq. (2):

$$\gamma(t) = F(u_j(\cdot),\, y_l(\cdot),\, v_w(\cdot),\, \phi(\cdot)) \tag{2}$$

where $u_j(\cdot)$ and $y_l(\cdot)$ are the generic j–th and l–th components of the measured inputs and outputs **u** and **y**, respectively, that are selected via the sensitivity analysis table recalled in Section IV. They represent the status $\mathcal{S}$ of the wind turbines of the wind farm, thus including possible fault (efficiency reduction) and failure (shutdown) conditions. $F(\cdot)$ is the function realised by the static NN, which depends on the layer architecture, the number of neurons, their weights and their activation functions. The NARX network is thus used as active policy estimator $\mathcal{P}^*_\theta$. t is the time step, whilst the signals $u_j(\cdot)$, $y_l(\cdot)$, $v_w(\cdot)$, and $\varphi(\cdot)$ have a number of delays n_d that have to be properly selected.

The design parameters of the optimised vector θ^* also includes the number of neurons and the numberof delays of the network inputs and outputs, while the value of the weights of each neuron are derived from the network training from the data acquired from the wind farm.

Note finally that a sigmoidal activation function for the input and the hidden layers are exploited. The output layer use linear activation functions. Therefore, the output of the NN needs to be con- strained between the yaw angles $[0^o \div 25^o]$ by using a saturation block before feeding the wind turbine model. In this way, the constrain of the yaw to be positive prevents the control to generate unachievable or fatigue-causing changes betweenpositive and negative yaw. Moreover, keeping the yaw below 25^o limits loads of the turbines. The wind speed and wind direction are normalised by a maximum wind speed and wind direction and the output is scaled up by the maximum yaw, 25^o, to ensure the inputs and outputs of the NN are in the range between 0 and 1, thus facilitating the NNtraining. These tasks are automatically performed by training the NN in the Matlab and Simulink environments, as shown in Section V. Moreover, the ALBC method described in this section is the same as that in model–free RL. The difference between the ALBC approach proposed here and model–free RL is in the training method, as described in SectionII-B.

B. Offline Training

During training, the ALBC scheme samples the exogenous inputs. For each batch of samples, the ALBC scheme runs a forward pass of the policy on those inputs to generate a control action, calculates the loss from the inputs and control by using the model, runs a backward pass of the policy to get gradients of the loss with respect to the policy parameters, and takes a gradient step to update the policy parameters in the direction that decreases the loss. The ALBC strategy has been implemented in the learning framework [38], which exploits the algorithm for gradient–based optimisation addressedin [39]. The loss function $\mathcal{V}$ in Eq. (3) is defined as negative and represents the average power produced by the turbines:

$$\mathcal{V} = \frac{1}{N_t} \sum_{t=1}^{N_t} W_i\,(v_w, \phi, \mathcal{S}) \tag{3}$$

Where $W_i\ (v_w,\ \varphi,\ \mathcal{S})$ represents the power generated from the i–th turbine given the WS v_w, the WD φ, and the yaw γ of all turbines in the farm; N_t indicates the number of wind turbines that are able to contribute to the power generation. This power iscomputed using the Gaussian wake model and the wind farm simulator addressed in Sections III and IV, respectively.

A key aspect of the proposed ALBC, and whereit differs from model–free RL, is that during the backward pass of the training phase, the ALBC generates the predicted output signals through the model to get the exact gradients. This is in con- trast to model–free RL, which does not exploit the model and thus needs to estimate the gradient from samples. To use the ALBC for wake steering, aversion of the wake model able to predict future measurements with a learning–enabled strategy is required. Its implementation is presented in Section III.

3. GAUSSIAN WAKE MODEL

The section describes the wake model considered in the wind farm simulator recalled in Sectionsec: wind-farm-description. Therefore, the complete wind farm model consists of three main submodels: the wind and wake models, the plant model, and the controller model, interacting as sketched in Figure **1**.

The design of the proposed ALBC requires the generation of proper signals by means of the Gaussian wake model derived in [6] and explained more in general in [40]. The Gaussian wake model has been implemented in the Flow Redirection and In-duction in Steady State (FLORIS) software package [41], and included by the authors in the former wind farm simulator developed in Matlab and Simulink [42]. However, this version cannot easily be used with existing learning packages [38]. Therefore, a learning–enabled implementation of FLORIS is required, which has been obtained by the authors by using a prediction model based on a NN, as suggested in Section II-A for the policy generator.

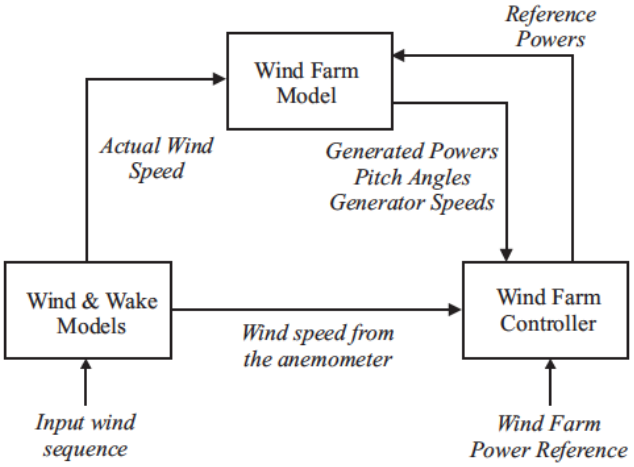

Figure 1: Block diagram of the wind and wake models of the windfarm simulator.

The solution achieved has been validated with respect to FLORIS. In particular, Figure **2** illus-trates a hub height slice of the flow field fromthe modified version of FLORIS for the validation case considered in this work: a 9 turbine wind farmwith incoming hub height wind speed v_w, incoming hub height wind direction φ, and yaw angles γ as described in Section IV. This flow field and theunderlying values of wind speed on the rotor disk match those from FLORIS.

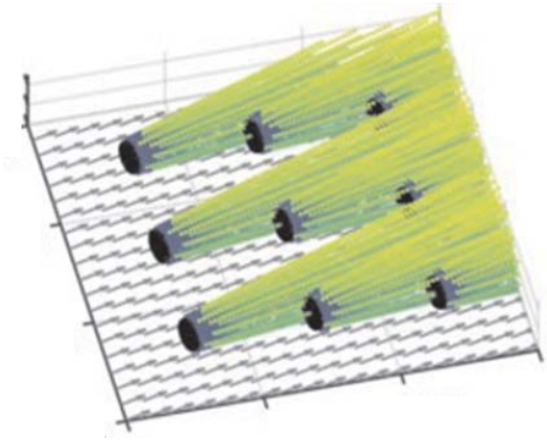

Figure 2: Hub height slice of the flow field from FLORIS depicted in Matlab.

The improved version of FLORIS is compatible with any learning method implemented in Matlab and Simulink, including existing packages for RL and user–implemented hybrid methods, such as the development of the ALBC scheme proposed in this paper.

4. WIND FARM SIMULATOR

This benchmark model implements a simple wind farm with 9 wind turbines that are arranged in a square grid layout [42]. The distance between the wind turbines in both directions are 7 times the rotordiameter, L. Two measuring masts are located in front of the wind turbines, one in each of the wind directions φ considered in this benchmark model, *e.g.* 0^o and 45^o. The wind speed is measured by these measuring masts and they are located in a distance of 10 times

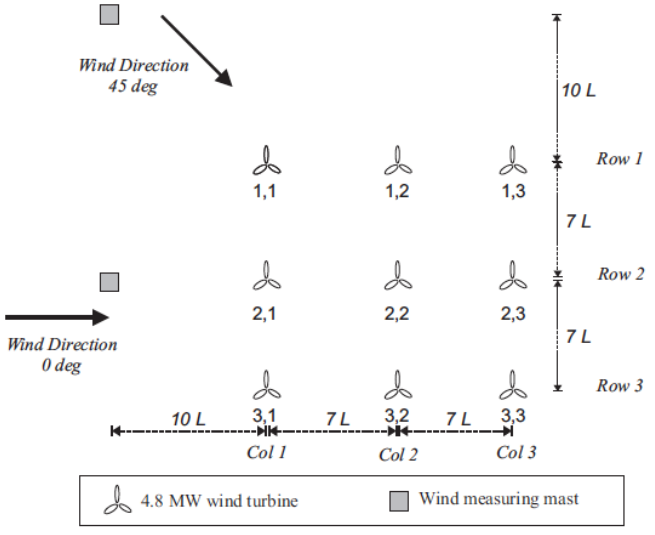

Figure 3: Layout of the wind farm with 9 wind turbines.

L in front of the wind farm.The wind turbines of the farm are defined by their row and column indices in the coordinate system illustrated in Figure **3**, which sketches layout of the wind farm with the 9 turbines of the square grid and the masts along the wind directions. It is worth noting that the original simulator described in [42] has been modified by the authors in order to includethe wake model recalled in Section III.

The farm uses generic 4.8 MW wind turbines, which were described in detail [43]. The turbine is a three bladed horizontal axis, pitch controlled variable speed wind turbine. Each of the wind turbines are described by simplified models including control logics, variable parameters and 3 states. The i–th wind turbine model generates the electrical power, $P_{ig}(t)$, the collective pitch angle, $\beta_i(t)$, and the generator speed, $\omega_{ig}(t)$. Note that the wind farmsimulator has been modified by the authors in order to include the control of the wind turbine yaw angles $\gamma_i(t)$. The second control input is thus representedby the pitch angle $\beta_i(t)$ modified by the baseline wind farm controller [42].

The two scenarios with different wind directions but driven both by the same wind speed sequence $v_w(t)$ (possibly with a time shift) are considered. The wind sequence contains a wind speed v_w in- creasing from 5 m/s. to 15 m/s, and with a peak value of about 23 m/s. In this benchmark model a very simple wind farm controller is used, which provides the wind turbine controllers with a power reference $P_{i\ ref}(t)$. More details on wind farm modelconsidered in this paper can be found in [42]. It is worth noting that the wind farm considered here could be seen as simplistic model. However, the work [42] describes how the simulator can fit realistic wind farms.

With these assumptions, the complete continuous–time description of the wind farmunder diagnosis has the following form:

$$\begin{cases} \dot{x}_c(t) &= f_c\left(x_c(t), u(t)\right) \\ y(t) &= x_c(t) \end{cases} \tag{4}$$

where $u(t) = [v_w(t),\ \beta_i(t)]^T$ and $y(t) = x_c(t) = [\omega_{ig}(t),\ P_{ig}(t)]^T$ are the input and the monitored output measurements, respectively. The subscript i indicates the measurement from the i–th wind turbine of the wind farm $(i = 1, \ldots, 9)$. $f_c(\cdot)$ represents the continuous–time nonlinear function describing the model of the plant under investigation.

In this benchmark three faults are considered that influence the measured variables from the wind turbine, *i.e.* $\beta_i(t)$, $\omega_{ig}(t)$, and $P_{ig}(t)$. It is also assumed that the considered faults can be detected at a wind farm level by comparing the performance from other wind turbines in the wind farm, but they are difficult to detect at a wind turbine level. Moreover, these three faults affect different wind turbines at different times, as described in moredetail in [42].

The remainder of this section describes the relations among the fault cases considered above, and the monitored measurements acquired from the wind park benchmark, in the presence of uncertainty and measurement errors. Moreover, Table **1** shows the fault effect distribution in the case of single fault occurrence, with respect to the acquired inputs and outputs of the wind park simulator.

Table 1: Wind Park Fault Scenarios

Fault Scenario	Affected WT(s) (row, col)	Affected Measurements
Fault 1	(3, 3)	$\{v_w(t),\ \omega_9(t),\ P_{4\,g}\}$
Fault 2	(3, 2), (3, 3)	$\{v_w(t),\ \beta_2(t),\ P_{6\,g}\}$
Fault 3	(3, 2), (3, 3), (2, 3)	$\{v_w(t),\ \beta_3(t),\ P_{7\,g}\}$

Table **1** was obtained by performing a fault sensitivity analysis. In practice, Table **1** is thus built by selecting the most sensitive measurement (u_i or y_j) with respect to the simulated fault conditions. Obviously, when different fault conditions have been considered with respect to the scenario of this work, different measurements will probably be taken into account. These conditions define uniquely the wind turbine status S in the wind park.

A proper analysis has been performed in order to verify and test the robustness of the considered strategy with respect to uncertainty and disturbance effects affecting the considered benchmark. A Monte Carlo tool could be useful at these stage, as the efficacy of the diagnosis depends on both the model approximation capabilities and the measurements errors.

In particular, the benchmark includes realistic wind turbine uncertainties that have been considered by modelling some meaningful variables as Gaussian stochastic processes around the nominal values and with standard deviations corresponding to the realistic minimal and maximal error values of Table **2**.

Table 2: Benchmark Uncertainties

Parameter	Value	Standard Deviation
ρ	$1.225 Kg/m^3$	±25%
J	$7.794 \times 10^6 Kg/m^3$	±35%
C_p	C_{p0}	±35%

Note finally that the wind park simulator ad- dressed in this section will be replaced with a more realistic test–bed described in Section VI, which will be exploited for verifying and validating the final performances of the proposed solutions.

5. SIMULATION RESULTS

The wind farm system is considered as described in Section IV consisting of 9 National Renewable Energy Laboratory (NREL) reference turbines [44]. The atmospheric conditions are constant, with windshear of 0.12, wind veer of 0.0, and turbulence intensity of 0.06. The incoming hub height wind direction can vary according two directions, as re- called in Section IV, the incoming hub height wind speed ranges from $3m/s$ to $24m/s$. The turbine status describes the working conditions of the turbines that can be active, inactive (shutdown), or faulty. These conditions are provided *e.g.* by a Fault Detection and Isolation (FDI) scheme already proposed by the authors *e.g.* in [45]. Figure **4** shows the different working conditions of the wind turbines of the park, when a set of turbine statuses, each one with nine, eight, seven and six turbines are considered. They are described by means of their row and column indices in the layout matrix.

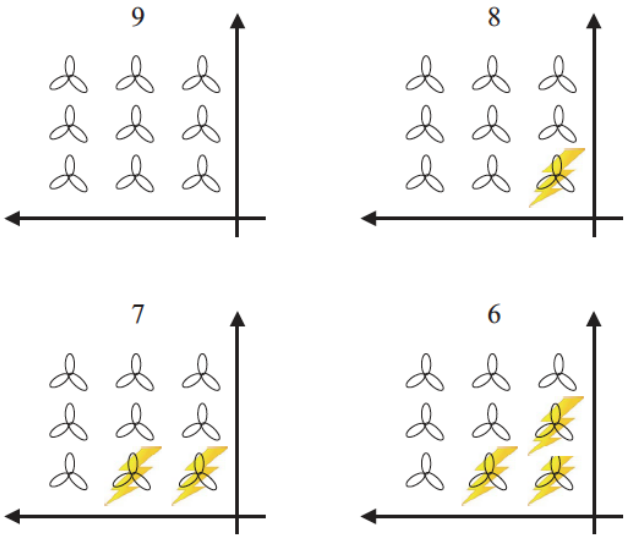

Figure 4: The considered working conditions of the wind farm.

The wind farm benchmark model considered in this work was proposed earlier in [44] and modified by the same authors who developed the wind turbine benchmark model. It consists of nine wind turbines arranged in a squared grid of three rows and three columns. The distance between two adjacent turbines is seven times the rotor diameter R. Two measuring masts (anemometers) are placed in front of the first line of turbines, at a distance of ten times R, providing the measurements of the undisturbed wind speed. The considered turbines are 4.8MW three– blades HAWT, represented by a simpler model, with respect to the previously described turbine. Each of them is provided with a controller, but also a wind farm controller is included in the benchmark model. Common fault scenarios can be simulated. The complete wind farm model consists of three main submodels: the wind and wake model, the plant model, and the controller model, interacting assketched in Figure **1**. The layout of the wind farm with nine turbines of the square grid and the masts along the wind directions are sketched in Figure **4**.

The distance between the wind turbines in both directions is 7 times the rotor diameter, L. Two measuring masts are located in front of the wind turbines, one in each of the wind directions consid ered in this benchmark model, *e.g.* 0^o and 45^o. The wind speed is measured by these measuring masts which are located in a distance of 10 times L infront of the wind farm. The wind turbines of the farm are defined by their row and column indices in the coordinate system, as illustrated in Figure **4**. The farm uses generic 4.8 MW wind turbines, which are three–bladed horizontal axis, pitch–controlled variable–speed wind turbines. Each of the wind turbines is described by simplified models including control logics, variable parameters and 3 states.

The ALBC method has been trained as described in Section II-B. Based on the NN properly trained as suggested in Section II-A, the control policy was estimated with one hidden layer with 30 neurons and an input layer with 15 neurons. A number of 4 delays has been exploited. An adaptive learningrate was used in the learning algorithm.

Table **3** shows the loss function of Eq. (3) for each turbine status vector in the test set, where the faulty turbines are highlighted, also according to Table **1**, labelled by the number of fault–free turbines. The four turbine statuses in the test set are thus illustrated in Figure **4**.

Table 3: Values of the Loss Function of EQ. (3) Labelled by the Number of Fault–Free Turbines N_t

Reference Loss Function	Average Loss Function	N_t
- 751	- 755	9
- 735	- 731	8
- 726	- 728	7
- 714	- 717	6

The loss in the test set is computed according to Eq. (3). Table **3** shows that for all turbine status cases in the test set, the averaged loss approximates quite accurately the reference optimal loss. The reference loss is the loss for the yaw angles foundby optimisation using FLORIS for just the fault– free turbines, as described in [41]. This is the (approximately) optimal loss value, which is correct up to the accuracy of the FLORIS optimisation method.

The trained ALBC policy proposed in this work is applied to the test set. Table **4** shows the net wind farm power for each turbine status in the test set, averaged over the wind speeds in the test set both for the ALBC and the approximate optimal power from FLORIS for the fault–free turbines. The power obtained using the baseline controller is reported, which sets all the turbines to have zero yaw (relative to the incoming wind direction). The power achieved via the developed ALBC closely matches the optimal power for each of the turbine statuses, indicating how this ALBC method is able to adapt effectively to turbine status. The power obtained by both the ALBC and the optimal power are higher than the one under baseline control, indicating the importance of wake steering for this test case.

Table 4: Wind Farm Power (KW) for Each Wind Turbine Status in Simulation

Fault–Free Turbines N_t	ALBC Method	Optimal Power	Baseline Controller
6	4321	4334	3981
7	4987	5001	4498
8	5521	5567	4987
9	6123	6145	5426

Table **5** summarises the net wind farm power for each wind speed in the test set, averaged over the turbine statuses in the test set for the ALBC, the optimal power, and baseline control. The power obtained by the ALBC closely matches the optimal power and is higher than the baseline power for each of the turbine statuses, indicating that the ALBC identifies the correct relationship between wind speed and yaw angles.

Table 5: Wind Farm Power (Kw) for Each Wind Speed V_w in the Test Set

Wind Speed v_w (m/s)	ALBC Scheme	Optimal Power	Baseline Controller
4	987	996	961
5	1998	2001	1876
6	3112	3124	3001
7	5882	5898	5679
8	8299	8321	8012

To further validate the proposed ALBC, the generated yaw angles γ are analysed with respect to the changes in the wind speed v_w and number of fault–free turbines N_t. Tables **6** and **7** report the yaw angles (^{o}) assigned to the bottom middle (3, 2) and center (2, 2) turbines in the farm, respectively. The tables show the yaw angles for the developed ALBC and the optimal power for the cases from the test set where nine and eight turbines (N_t) are fault– free. In Table **6**, the yaw angles for the ALBC and optimal power are quite similar. The ALBC works properly for all wind speeds in the 9 turbine case, the bottom middle turbine (3, 2) should yaw to steer the wake from the downstream turbine, but that in the 8-turbine case, this solution does not lead to any benefit since the downstream turbine (3, 3) is faulty.

Table 6: Yaw Angles (^{O}) Computed for the Bottom Middle Turbine (3, 2)

Wind Speed v_w (m/s)	ALBC Scheme $N_t = 9$	ALBC Scheme $N_t = 8$	Optimal Power $N_t = 9$	Optimal Power $N_t = 8$
4	19.8	0.8	19.7	0.8
5	19.6	0.7	19.6	0.9
6	18.7	0.7	20.3	0.8
7	19.8	0.65	19.7	0.9
8	19.6	0.7	20.1	0.8

In contrast to the bottom middle turbine (3, 2), the center (2, 2) turbine should yaw for both the 9 and 8 active turbine cases, since in both cases the turbine downstream from the center turbine is active. The yaw angles in Table **7** show that the ALBC yaws effectively the center turbine in both the nine and eight turbine cases, as desired.

Table 7: Yaw Angles (^{o}) Computed for the Center Turbine (2, 2)

Wind Speed v_w (m/s)	ALBC Scheme N_t = 9	ALBC Scheme N_t = 8	Optimal Power N_t = 9	Optimal Power N_t = 8
4	19.2	19.1	18.75	19.3
5	18.7	18.8	18.8	18.9
6	18.5	18.6	20.5	18.5
7	18.2	18.3	18.6	18.6
8	17.6	17.8	18.6	18.6

The yaw angles obtained with the ALBC and the ones corresponding to the optimal power in Table **7** are more different than those in Table **6**. In order to investigate how significant this difference is, the optimal yaw angles for the case with $N_t = 9$ fault–free turbines are considered with $v_w = 6m/s$, then replace the angle for the center turbine with that from the ALBC, and calculate the resulting wind farm power. It can be shown that using the yaw from the ALBC decreases the power produced by about 0.5kW, which is less than 0.01% of the total power. If instead the center turbine does not yaw at all, *i.e.* the yaw is 0^o, then the total power decreases by 1.5%. This indicates that yawing the center turbine (2, 2) is important to maximise the power, but also that the power function has a plateau around a yaw of $18^o \div 21^o$.

Finally, the training procedure, the power production, and the yaw angles reported in this section indicate that the developed ALBC solution is able to find the yaw angles that maximise the total power production of the wind farm. This result is valid for varying wind speed and different turbine status considering faulty conditions.

6. HARDWARE–IN–THE–LOOP VALIDATION

The HIL test–rig has been implemented in order to verify and validate the proposed solutions in more realistic real–time working conditions. These experimental tests aim at validating the achieved results obtained in simulations, considering the al- most real conditions that the wind turbine systems under analysis may deal with, during their working situations.

The set-up of the test–rig, represented in Figure **5**, consists of three interconnected components:

Simulator: the model of the wind farm, including the wind and the wake models, have been implemented in the Matlab, Simulink and Lab VIEW environments, and consider factors such as disturbance, measurement noise and uncertainty, according to the system described in Section IV. This software tool runs on an industrial CPU and allows also the real–time monitoring and the ALBC method of the simulated system parameters.

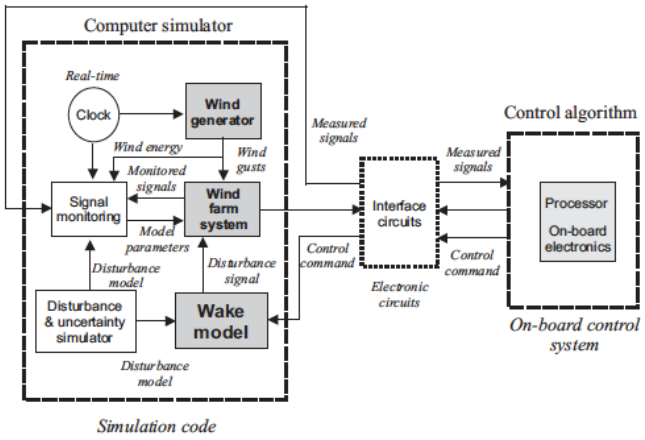

Figure 5: The wind farm block diagram of the HIL testrig.

On board electronics: The control schemes have been implemented in the AWC 500 sys- tem, which features standard wind turbines specifications, but it may fit the requirements of a wind park. This element receives the signals relative to the main controlled outputs. Then, it processes the control algorithm, possibly including the fault estimation and diagnosis module, and produces the command signals transmitted to the wind farm simulator.

Interface circuits: they carry out the communication between the simulator and the on board electronics, receiving the output signals from the simulator and transmitting the signal generated by the control algorithm.

Table **8** summarises the results obtained using this real–time HIL set–up.

Table 8: Wind Farm Power (Kw) for Each Wind Turbine Status With the Hil Test–Bed

Fault–Free Turbines N_t	ALBC Method	Optimal Power	Baseline Controller
6	4327	4338	3987
7	4992	5008	4502
8	5528	5572	4991
9	6129	6153	5433

It is worth observing the consistency of the almost real–time test of Table **8** with respect to the results

reported in Section V. Although the average performances seem to be better than those obtained using the HIL platform, some issues have to be taken into account. Indeed, the numerical accuracy of the on–board electronics, which involves float calculations is more restrictive than the CPU of the simulator. Moreover, also the A/D and D/A conversions can motivate possible deviations. Note that real situations do not require to transfer data from a computer to the on board electronics, so that this error is not actually introduced. However, the obtained deviations are not critical and the developed control strategies can be also considered in real applications.

Some concluding remarks can be finally drawn here. This paper considered a wake-steering controller to allow the use of preview wind direction information, with the goal of improving yaw tracking in dynamic wind conditions. Using data from an offshore wind power plant, it was developed a method for generating realistic wind direction measurements to evaluate preview-based wake steering. Simulations were performed for a nive-turbine array using the FLORIS model to determine power production for different control scenarios. With perfect preview information, preview-based wake steering was found to increase energy production significantly more than standard wake steering, with a preview time providing the most benefit. The optimal preview time was derived to depend on the yaw controller dynamics and wind conditions. For controllers that yaw less frequently, longer preview times will likely be needed to overcome controller lag. But for more responsive controllers, or when operating in highly variable wind conditions, shorter preview times should help ensure the yaw position does not lead the wind direction too much.

With realistic preview measurement accuracy, onthe other hand, no improvement in wake-steering performance was observed. However, the wind direction coherence model was used to determine preview measurement accuracy represents the average coherence for a variety of atmospheric conditions. More research is needed to determine how longitudinal wind direction coherence depends on atmospheric conditions as well as terrain. Particular sites or wind conditions may be more favorable for preview-based wake steering.

To fully evaluate the benefits of preview-enabled wake steering with realistic measurement accuracy, more sophisticated control strategies should be explored. For example, rather than waiting for thewind turbine's existing yaw controller to implement wake steering, additional performance gains could be made by yawing the turbine more frequently, (*e.g.*, at fixed time intervals). Further, model predictive control approaches could be used to explicitly optimize the control actions based on the wind direction preview.

In addition to yaw controller improvements, more effective methods for estimating the approaching wind directions should be investigated. For instance, some works presented a consensus approach for estimating local wind directions through information exchange between wind turbines that could improve forecasting accuracy. Moreover, remote- sensing devices, such as scanning devices, could be used to measure the approaching wind conditions over a large area.

7. CONCLUSION

In this study, an intelligent wake steering technique was introduced that calculates yaw anglesbased on the actual operating parameters of the turbine. It specifically made use of a learning- based hybrid model, or active control, where a neural network was trained online to calculate yaw angles from operating conditions like wind turbine status. In actuality, wake steering is frequently used to yaw upstream wind turbines in order to divert their wakes away from downstream turbines, so boosting the power generated. The majority of wake steering techniques, however, rely on lookup tables that are acquired offline and that translate a setof parameters, including wind speed and direction, to yaw angles for each turbine in a farm. These charts assume that every turbine is working well, however they may not be very accurate if one or more turbines are not producing their rated power due to low wind speed, malfunctions, scheduled maintenance, or emergency maintenance. The pro- posed control method did not rely on optimization problems, such as those applied to each combinationof the wind farm turbines, unlike solely model- based systems that often use lookup tables provided by the wind turbine manufacturer or developed offline. This is how the incorporation of a learning approach into the control architecture enabled the creation of an active control architecture. On the other hand, the created methodology did not need a lot of training samples, unlike purely learning- based methods like model-free reinforcement learning. By taking advantage of the model during the back propagation learning procedure, the suggested strategy learned more from each sample. Based on flow redirection and induction in steady state code, results were taken into account

for both standard (nominal) wake steering (all turbines functioning) and with problematic turbines (lower efficiency or shutdown). The National Renewable Energy Laboratory (USA) has made this code an open-source tool for optimizing wind farms. These yaw angles were compared to those discovered by solving an optimization problem, and those results were validated. The proposed method, which made useof a hybrid model- and learning-based approach, was demonstrated to enable active wake steering through sample-efficient training and quick online evaluation. Finally, a hardware-in-the-loop test-bed was taken into consideration for evaluating and con- firming the performance of the suggested solutions in a more practical setting. The use of the proposed methods with data from actual wind farms and installations will be considered in subsequent research. When considering future research directions, the challenges associated to testing and implementing new control in modern wind farms need to be highlighted, with investment costs in the billion- euro range. Direct testing of new ideas on the full scale is simply not possible, and instead a careful proof-of-concept and validation strategy is required. To this end, it is foreseen that in the coming years, the sequence of large-eddy simulations, wind tunnel experiments, and small field campaigns will play an ever larger role. Each of these faces its own challenges, respectively related to model bias, scale similarity, and establishing statistical significance. Moreover, for actual commercial implementation, additional issues arise, such as controller safety and proof of commercial value for different sites, among others.

FUNDING

This work was not financially supported by any grant.

REFERENCES

[1] M. F. Howland, J. B. Quesada, J. J. P. Martinez, F. P. Larranaga, N. Yadav, J. S. Chawla, V. Sivaram, and J. O. Dabiri, "Collective wind farm operation based on a predictive model increases utility-scale energy production," Nature Energy, vol. 7, no. 1, pp. 818-827, Aug. 2022. https://doi.org/10.1038/s41560-022-01085-8

[2] P. Fleming, J. King, E. Simley, J. Roadman, A. Scholbrock, P. Murphy, J. K. Lundquist, P. Moriarty, J. Fleming, K. van Dam, C. Bay, R. Mudafort, D. Jager, J. Skopek, M. Scott, B. Ryan, C. Guernsey, and D. Brake, "Continued results from a field campaign of wake steering applied at a commercial wind farm - Part 2," Wind Energy Science, vol. 5, no. 3, pp. 945-958, July 2020. https://doi.org/10.5194/wes-5-945-2020

[3] "Initial results from a field campaign of wake steering applied at a commercial wind farm - Part 1," Wind Energy Sci- ence, vol. 4, no. 2, pp. 273-285, May 2019. https://doi.org/10.5194/wes-4-273-2019

[4] C. Shapiro, D. F. Gayme, and C. Meneveau, "Modelling yawed wind turbine wakes: A lifting line approach," Journal of Fluid Mechanics, vol. 841, no. 1, pp. 1-12, Feb. 2018. https://doi.org/10.1017/jfm.2018.75

[5] J. King, P. Fleming, R. King, L. A. Martinez-Tossas, C. Bay, R. Mudafort, and E. Simley, "Control-oriented model for sec-ondary effects of wake steering," Wind Energy Science, vol. 6, no. 3, pp. 701-714, May 2021. https://doi.org/10.5194/wes-6-701-2021

[6] M. Bastankhah and F. Porte-Agel, "Experimental and theoreti- cal study of wind turbine wakes in yawed conditions," Journal of Fluid Mechanics, vol. 806, no. 1, pp. 506-541, Oct. 2016. https://doi.org/10.1017/jfm.2016.595

[7] C. Dao, B. Kazemtabrizi, and C. Crabtree, "Wind turbine reliability data review and impacts on levelised cost of energy," Wind Energy, vol. 22, no. 1, pp. 1848-1871, July 2019. https://doi.org/10.1002/we.2404

[8] M. Costa, J. A. Orosa, D. Vergara, and P. Fernandez-Arias, "New tendencies in wind energy operation and maintenance," Applied Sciences, vol. 11, no. 1386, pp. 1-26, Feb. 2021. https://doi.org/10.3390/app11041386

[9] H. Dong, J. Xie, and X. Zhao, "Wind farm control technologies: from classical control to reinforcement learning," Progress in Energy, vol. 4, no. 3, pp. 1-19, June 2022. https://doi.org/10.1088/2516-1083/ac6cc1

[10] P. Stanfel, K. Johnson, C. J. Bay, and J. King, "Proof-of-concept of a reinforcement learning framework for wind farm energy capture maximization in time-varying wind," Journal of Renewable and Sustainable Energy, vol. 13, no. 4, pp. 1-14, Aug. 2021. https://doi.org/10.1063/5.0043091

[11] H. Zhao, J. Zhao, J. Qiu, G. Liang, and Z. Y. Dong, "Cooper-ative wind farm control with deep reinforcement learning and knowledge-assisted learning," IEEE Transactions on Industrial Informatics, vol. 16, no. 11, pp. 6912-6921, Nov. 2020. https://doi.org/10.1109/TII.2020.2974037

[12] J. Arroyo, C. Manna, F. Spiessens, and L. Helsen, "Reinforced model predictive control (RL-MPC) for building energy man- agement," Applied Energy, vol. 309, no. 1, pp. 1-16, Mar. 2022. https://doi.org/10.1016/j.apenergy.2021.118346

[13] J. Drgona, K. Kis, A. Tuor, D. Vrabie, and M. Klauco, "Differentiable predictive control: Deep learning alternative to explicit model predictive control for unknown nonlinear systems," Journal of Process Control, vol. 116, no. 1, pp. 80-92, Aug. 2022. https://doi.org/10.1016/j.jprocont.2022.06.001

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015. https://doi.org/10.1038/nature14539

[15] P. Gavali and J. S. Banu, "Deep convolutional neural network for image classification on cuda platform," in Deep Learning and Parallel Computing Environment for Bioengineering Systems, A. K. Sangaiah, Ed. Academic Press, 2019, pp. 99-122. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128167182000130 https://doi.org/10.1016/B978-0-12-816718-2.00013-0

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Processing Magazine, vol. 29,

no. 6, pp. 82-97, 2012.
https://doi.org/10.1109/MSP.2012.2205597

[17] Y. Lei, F. Jia, J. Lin, S. Xing, and S. X. Ding, "An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data," IEEE Transactions on Industrial Electronics, vol. 63, no. 5, pp. 3137-3147, 2016.
https://doi.org/10.1109/TIE.2016.2519325

[18] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," Mechanical Systems and Signal Processing, vol. 100, pp. 439-453, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327017303369
https://doi.org/10.1016/j.ymssp.2017.06.022

[19] L. Chen, G. Xu, Q. Zhang, and X. Zhang, "Learning deep representation of imbalanced scada data for fault detection of wind turbines," Measurement, vol. 139, pp. 370-379, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224119302386
https://doi.org/10.1016/j.measurement.2019.03.029

[20] T. Verstraeten, A. Nowe, J. Keller, Y. Guo, S. Sheng, and J. Helsen, "Fleetwide data-enabled reliability improvement of wind turbines," Renewable and Sustainable Energy Reviews, vol. 109, pp. 428-437, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032119301522
https://doi.org/10.1016/j.rser.2019.03.019

[21] C. Li, S. Zhang, Y. Qin, and E. Estupinan, "A systematic review of deep transfer learning for machinery fault diagnosis," Neurocomputing, vol. 407, pp. 121-135, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220306123
https://doi.org/10.1016/j.neucom.2020.04.045

[22] F. Jamil, T. Verstraeten, A. Nowe, C. Peeters, and J. Helsen, "A deep boosted transfer learning method for wind turbine gearbox fault detection," Renewable Energy, vol. 197, pp. 331-341, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S096014812201134X
https://doi.org/10.1016/j.renene.2022.07.117

[23] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in Artificial Neural Networks and Machine Learning - ICANN 2018, V. Kurkova, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds. Cham: Springer International Publishing, 2018, pp. 270-279.
https://doi.org/10.1007/978-3-030-01424-7_27

[24] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," IEEE Transactions on Industrial Electronics, vol. 64, no. 3, pp. 2296-2305, 2017.
https://doi.org/10.1109/TIE.2016.2627020

[25] R. Zhang, H. Tao, L. Wu, and Y. Guan, "Transfer learning with neural networks for bearing fault diagnosis in changing working conditions," IEEE Access, vol. 5, pp. 14 347-14 357, 2017.
https://doi.org/10.1109/ACCESS.2017.2720965

[26] M. R. Bhuiyan and J. Uddin, "Deep transfer learning models for industrial fault diagnosis using vibration and acoustic sensors data: A review," Vibration, vol. 6, no. 1, pp. 218-238, 2023. [Online]. Available: https://www.mdpi.com/2571-631X/6/1/14
https://doi.org/10.3390/vibration6010014

[27] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 1, pp. 136-144, 2019.
https://doi.org/10.1109/TSMC.2017.2754287

[28] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," IEEE Transactions on Industrial Electronics, vol. 66, no. 9, pp. 7316-7325, 2019.
https://doi.org/10.1109/TIE.2018.2877090

[29] B. Yang, Y. Lei, F. Jia, and S. Xing, "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings," Mechanical Systems and Signal Processing, vol. 122, pp. 692- 706, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327018308367
https://doi.org/10.1016/j.ymssp.2018.12.051

[30] S. Farsoni, S. Simani, and P. Castaldi, "Fuzzy and Neural Net-work Approaches to Wind Turbine Fault Diagnosis," Applied Sciences, vol. 11, no. 11, pp. 1-21, May, 29 2021, ISSN: 2076-3417.
https://doi.org/10.3390/app11115035

[31] M. S. Shaker and R. J. Patton, "Active sensor fault tolerant output feedback tracking control for wind turbine systems via T-S model," Engineering Applications of Artificial Intelligence, vol. 34, no. 2014, pp. 1-12, Sept. 2014.
https://doi.org/10.1016/j.engappai.2014.04.005

[32] D. Xu, B. Jiang, and P. Shi, "Nonlinear actuator fault estimation observer: An inverse system approach via a t-s fuzzy model," International Journal of Applied Mathematics and Computer Science, vol. 22, no. 1, pp. 183-196, March 2012.
https://doi.org/10.2478/v10006-012-0014-9

[33] R. Isermann, "On fuzzy logic applications for automatic control, supervision and fault diagnosis," IEEE Trans. on Sys. Man. and Cyber. Part A: Sys. & Humans, vol. 28, no. 2, pp. 221-235, 1998.
https://doi.org/10.1109/3468.661149

[34] V. Palade, R. J. Patton, F. J. Uppal, J. Quevedo, and S. Daley, "Fault diagnosis of an industrial gas turbine using neuro-fuzzy methods," in IFAC'02. Balcelona, Spain: 15th IFAC World Congress on Automatic Control, July, 21-26 2002, pp. 471- 476.
https://doi.org/10.3182/20020721-6-ES-1901.01632

[35] F. J. Uppal, R. J. Patton, and V. Palade, "Neuro-fuzzy based fault diagnosis applied to an electro-pneumatic valve," in IFAC'02. Balcelona, Spain: 15th IFAC World Congress on Automatic Control, July, 21-26 2002, pp. 477-482.
https://doi.org/10.3182/20020721-6-ES-1901.01633

[36] J. Korbicz, J. M. Koscielny, Z. Kowalczuk, and W. Cholewa, Eds., Fault Diagnosis: Models, Artificial Intelligence, Applications, 1st ed. London, UK: Springer-Verlag, February, 12 2004, ISBN: 3540407677.
https://doi.org/10.1007/978-3-642-18615-8

[37] S. Simani and S. Farsoni, Fault Diagnosis and Sustainable Control of Wind Turbines: Robust data-driven and model-based strategies, 1st ed., ser. Mechanical Engineering. Oxford (UK): Butterworth-Heinemann - Elsevier, Jan. 4th 2018, ISBN: 9780128129845.

[38] P. D. Dueben, M. G. Schultz, M. Chantry, D. J. Gagne II, D. M. Hall, and A. McGovern, "Challenges and benchmark datasets for machine learning in the atmospheric sciences: Definition, status, and outlook," Artificial Intelligence for the Earth Systems, vol. 1, no. 3, pp. 1-11, Oct. 2022.
https://doi.org/10.1175/AIES-D-21-0002.1

[39] T. Homem-De-Mello, "Variable-sample methods for stochastic optimization," ACM Transactions on Modeling and Computer Simulation, vol. 13, no. 2, pp. 108-133, Apr. 2003.
https://doi.org/10.1145/858481.858483

[40] J. Annoni, P. Fleming, A. Scholbrock, J. Roadman, S. Dana, C. Adcock, F. Pore-Agel, S. Raach, F. Haizmann, and D. Schlipf, "Analysis of control-oriented wake modeling tools using lidar field results," Wind Energy Science, vol. 3, no. 2, pp. 819-831, Nov. 2018.
https://doi.org/10.5194/wes-3-819-2018

[41] L. E. Andersson, O. Anaya-Lara, J. O. Tande, K. O. Merz, and L. Imsland, "Wind farm control - Part I: A review on

control system concepts and structures," IET Renewable Power Generation, vol. 15, no. 10, pp. 2085-2108, Jan. 2021. https://doi.org/10.1049/rpg2.12160

[42] P. F. Odgaard and J. Stoustrup, "Fault Tolerant Wind Farm Control - a Benchmark Model," in Proceedings of the IEEE Multiconference on Systems and Control - MSC2013, Hyderabad, India, August 28-30 2013, pp. 1-6. https://doi.org/10.1109/CCA.2013.6662784

[43] "Unknown Input Observer Based Scheme for Detecting Faults in a Wind Turbine Converter," in Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, vol. 1, no. 1. Barcelona, Spain: IFAC - Elsevier, June 30 - July 3 2009, pp. 161-166. https://doi.org/10.3182/20090630-4-ES-2003.00027

[44] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault-Tolerant Control of Wind Turbines: A Benchmark Model," IEEE Transactions on Control Systems Technology, vol. 21, no. 4, pp. 1168-1182, July 2013, ISSN: 1063-6536. https://doi.org/10.1109/TCST.2013.2259235

[45] S. Simani, S. Farsoni, and P. Castaldi, "Residual generator fuzzy identification for wind farm fault diagnosis," in Proceedings of the 19th World Congress of the International Federation of Automatic Control - IFAC'14, vol. 19, no. 1, IFAC & South Africa Council for Automation and Control. Cape Town, South Africa: IFAC, 24-29 August 2014, pp. 4310-4315, invited paper for the special session "FDI and FTC of Wind Turbines in Wind Farms" organised by P. F. Odgaard and S. Simani. https://doi.org/10.3182/20140824-6-ZA-1003.00052

Received on 02-03-2023 Accepted on 30-03-2023 Published on 07-04-2023

DOI: https://doi.org/10.31875/2409-9694.2023.10.02