



Deep learning and fuzzy logic to implement a hybrid wind turbine pitch control

J. Enrique Sierra-Garcia¹ · Matilde Santos²

Received: 5 March 2021 / Accepted: 8 July 2021 / Published online: 19 July 2021
© The Author(s) 2021, corrected publication 2021

Abstract

This work focuses on the control of the pitch angle of wind turbines. This is not an easy task due to the nonlinearity, the complex dynamics, and the coupling between the variables of these renewable energy systems. This control is even harder for floating offshore wind turbines, as they are subjected to extreme weather conditions and the disturbances of the waves. To solve it, we propose a hybrid system that combines fuzzy logic and deep learning. Deep learning techniques are used to estimate the current wind and to forecast the future wind. Estimation and forecasting are combined to obtain the effective wind which feeds the fuzzy controller. Simulation results show how including the effective wind improves the performance of the intelligent controller for different disturbances. For low and medium wind speeds, an improvement of 21% is obtained respect to the PID controller, and 7% respect to the standard fuzzy controller. In addition, an intensive analysis has been carried out on the influence of the deep learning configuration parameters in the training of the hybrid control system. It is shown how increasing the number of hidden units improves the training. However, increasing the number of cells while keeping the total number of hidden units decelerates the training.

Keywords Hybrid system · Deep learning · Fuzzy control · Neural networks · Pitch control · Wind turbines

1 Introduction

The use of coal for energy production has proven to be one of the main contributors to climate change. Thermal plants are a source of carbon dioxide emissions. Even today, these sources of energy are widespread. For example, in Spain only in 2016 coal produced 60% of all carbon dioxide emissions. This air pollution is estimated to cause more than 500,000 asthma attacks each year around the world [1]. Therefore, to slow climate change and achieve the Paris Climate Goals [2], fossil-based energy sources must be replaced by renewable energy such as wind, hydro and solar.

The use of wind energy increases year after year. It is currently the second most used energy source after

hydroelectric [3]. It seems that this growing trend will continue in the future until it becomes the main source of energy generation in 2050 [4]. To contribute to this positive and sustainable trend, research on wind energy and wind turbines must continue and take a leap forward.

Among other energy efficiency goals, controlling wind turbines (WT) remains a challenge for engineers. Its main difficulty from the control point of view comes from the fact that it must meet several objectives simultaneously. First of all, the control is designed to reach and stabilize the generated power at its nominal value. In turn, safety must be guaranteed under all operating conditions [5]. Furthermore, the fatigue and the vibrations of the structure should be minimized since it has been shown that the control influences the stability of the turbine [6].

The control of the turbine is carried out by means of different control actions, mainly the pitch angle, the angular speed of the generator, and the yaw angle. The pitch angle turns the surface of the blade that faces the wind so that the greater the area swept, the more mechanical power is generated. It is used to regulate the output power around its rated value. On the other hand, the

✉ J. Enrique Sierra-Garcia
jesierra@ubu.es

¹ Electromechanical Engineering Department, University of Burgos, 09006 Burgos, Spain

² Institute of Knowledge Technology, Complutense University of Madrid, 28040 Madrid, Spain

rotor angular speed is controlled in order to find the optimal power curve. Finally, the yaw control turns the entire turbine to follow the direction of the wind.

This work focuses on the control of the pitch angle of the blades. Although different solutions have been proposed, it remains an open challenge due to the nonlinearity of the wind turbine, its complex dynamics and the coupling between the internal variables. This control is especially critical for floating offshore wind turbines (FOWT), subjected to strong external loads due to harsh weather conditions, both wind and waves.

These environmental disturbances make significant differences between the wind measured by the anemometer sensor and the wind that effectively attacks the surface of the blades, the latter being eventually transformed to mechanical power. This is called effective wind. As it is well known, the power output and the response of the WT depends directly on the real wind speed; thus, working with the accurate wind information is key to develop efficient WT controllers.

In this work, we develop a WT pitch control architecture that combines fuzzy pitch control and deep learning neural networks used to estimate the effective wind. The main contributions can be summarized as follows:

- The development of an efficient hybrid control architecture that combines fuzzy logic pitch control and deep learning neural networks to estimate the wind input of the WT subjected to disturbances. This goal includes the following.
- Forecasting the future effective wind in WTs by deep learning techniques.
- Estimation of current effective wind WT speed by deep techniques.
- Combination of forecasting and estimation of effective wind as input of the fuzzy logic controller.

The operation of the hybrid control strategy has been simulated and compared with an intelligent fuzzy controller without the deep learning module, and with a conventional PID controller. The simulation results show how including the effective wind obtained with the deep learning module improves the control performance. Indeed, for low and medium wind speeds, an improvement of 21% is obtained respect to the PID controller, and 7% respect to the standard fuzzy controller. In addition, an intensive study has been carried out on the influence of the deep learning configuration parameters in the training performance of the system.

To the best of our knowledge, there are not previous works where fuzzy logic and deep learning techniques are used together for pitch control of wind turbines subjected to external disturbances. In addition, one of the main novelties here proposed is the combination of forecasting

and estimation of effective wind to feed the inputs of the pitch controller. These contributions have been proved quite useful to get a better control of the wind turbine.

The paper is organized as follows. Section 2 presents a brief state of the art. In Sect. 3, the mathematical model of the wind turbine is described. The design of the hybrid controller, the fuzzy controller and the deep learning module, is described in Sect. 4. Simulation results are discussed in Sect. 5. The paper ends with the conclusions and future works.

2 Related works

Due to the characteristics and nature of the control of wind turbines, that are strongly nonlinear and coupled systems, intelligent techniques for pitch control have been applied. In fact, intelligent control techniques such as fuzzy control and neural networks have been successfully used for complex systems in the energy field and in many other areas [7–9]. Intelligent controllers that use neural networks inside are usually called neuro-controllers [10].

Fuzzy logic has been previously used to control the pitch angle of wind turbines. To mention some works, authors in [11] design a hierarchical fuzzy logic pitch controller to solve the nonlinear system effects produced by atypical winds. It is compared to a conventional PID pitch regulator. In the work by Moodi et al., [12], a robust H_∞ observer-based fuzzy controller is designed to control the turbine using the estimated wind speed. A nonlinear Takagi–Sugeno fuzzy model is introduced for a variable speed, variable pitch wind turbine. Two artificial neural networks are used to accurately model the aerodynamic curves. In that paper, in addition to rotor dynamics, blade and tower dynamics are taken into account. In [13], the pitch angle is controlled by a PID whose parameters are tuned by a fuzzy logic inference system. The effectiveness of the method is tested by the simulation of a small wind turbine. In [14], pitch and yaw control of a small wind turbine are addressed. A fuzzy model was used to describe a wind turbine whose output is controlled by changing the blades angle of attack and rotating the nacelle to a position facing the wind.

A hybrid control that combines a fuzzy system and a conventional model predictive controller is developed in [15]. The aim of the fuzzy model predictive pitch controller is to minimize the loading effect on the wind turbine as well as to maximize the extracted power output. The fuzzy logic controller works very efficiently by encountering the system nonlinearity, while the model predictive controller helps the system to become more stable. Sitharthan et al. (2020) propose a hybrid MPPT control strategy that estimates the effective wind speed and the optimal rotor speed of a wind power generation system to track the maximum

power. It uses the particle swarm optimization (PSO) algorithm to enhance a radial basis function neural network [16]. The paper by Sarkar et al. 2020 presents a robust PID pitch angle control system for the rated wind turbine power at a wide range of simulated wind speeds [17]. In addition, ant colony optimization (ACO), particle swarm optimization (PSO), and classical Ziegler–Nichols (Z–N) algorithms have been used for tuning the PID controller parameters to obtain a rated stable output power with fluctuating wind speeds.

The papers on intelligent pitch control of large-scale wind turbines are scarcer. An interesting approach is presented in Rubio et al. [18], where a fuzzy-logic-based pitch control system for a 5 MW wind turbine installed in an OC4 WT semi-submersible platform is presented. The fuzzy controller has as input the instantaneous value of the wind speed, filtered and normalized according to the nominal speed, and gives the pitch reference. The paper by Abdelbaky et al. [19] proposes constrained fuzzy-receding horizon pitch control for a variable speed wind turbine. The controller guarantees the nominal stability, and the model is converted to a simple online quadratic optimization problem that requires less computational time to be solved. A 5 MW offshore wind turbine simulation model is used to validate the results of the mathematical model. With a different aim, fuzzy logic is also applied to develop a rule-based turbine selection methodology in [20]. The proposal analyzes several scenarios in conjunction with the turbine selection model.

Fault detection in WTs is a topic of current interest [21, 22]. It is one of the applications where deep learning is being explored in the wind energy field, together with forecasting and model identification. For example, Khan et al. combine deep learning and principal component analysis to forecast wind power for large-scale wind turbines [23]. Deep learning is also used for wind forecasting in [24]. In this case, Wavelet Packet Transform (WPT) preprocesses the signals. Fu uses deep learning to monitor the condition of the gearbox bearing [25]. Li proposes the use of Deep Small-World Neural Network on the basis of unsupervised learning to detect the early failures of wind turbines [26]. The paper by [27] analyzes the input and output data of wind farm based on deep neural networks, develops an intelligent model with an Extreme Learning Machine, and predicts some parameter of the wind turbine. From a wider perspective, a recent overview of deep reinforcement learning for power system applications can be found in Zhang et al. [28]. Interestingly, in Lin et al., 2020, deep learning is applied to investigate the major driven force on the mooring line tension of a FOWT model [29].

As it has been shown in this brief state of art, there are few works that exploit deep learning in the wind energy

field. But this strategy has been mainly used to forecast wind and to detect failures in WTs. In addition, although there are previous studies that apply fuzzy pitch control, they work with the measured wind without considering it may not be the one that reaches the rotor. In this work, we propose a novel hybridization of these techniques, using deep learning to predict the effective wind and using this estimation as input of the fuzzy pitch controller. As shown, hybrid intelligent controllers can be considered a promising technique for dealing with the complex problem of controlling wind turbines.

3 Mathematical model of the wind turbine

In this work, a small model of a 7 kW wind turbine is used. The mathematical model is given by Eqs. (1–9). The development of these expressions is further explained in [30, 31].

$$\dot{I}_a = \frac{1}{L_a} (K_g \cdot K_\phi \cdot w - (R_a + R_L)I_a), \tag{1}$$

$$f_{\text{blade}}(s) = \frac{\beta \cdot s + \sqrt{2}}{\beta^2 \cdot s^2 \left(\sqrt{\left(\frac{\alpha}{2}\right) + \sqrt{\alpha}} \right) \cdot \beta \cdot s + \sqrt{2}} \cdot \frac{\gamma \cdot s + 1/\tau}{s + 1/\tau}, \tag{2}$$

$$v_{\text{ef}} = f_{\text{blade}}(v_M + \text{dist}), \tag{3}$$

$$\lambda = (w \cdot R)/v_{\text{ef}}, \tag{4}$$

$$\lambda_i = \left[\left(\frac{1}{\lambda + c_8} \right) - \left(\frac{c_9}{\theta^3 + 1} \right) \right]^{-1}, \tag{5}$$

$$C_p(\lambda_i, \theta) = c_1 \left[\frac{c_2}{\lambda_i} - c_3\theta - c_4\theta^{c_5} - c_6 \right] e^{-\frac{c_7}{\lambda_i}}, \tag{6}$$

$$\dot{w} = \frac{1}{2 \cdot J \cdot w} (C_p(\lambda_i, \theta) \cdot \rho \pi R^2 \cdot v^3) - \frac{1}{J} (K_g \cdot K_\phi \cdot I_a + K_f \cdot w), \tag{7}$$

$$\ddot{\theta} = \frac{1}{T_\theta} [K_\theta(\theta_{\text{ref}} - \theta) - \dot{\theta}], \tag{8}$$

$$P_{\text{out}} = R_L \cdot I_a^2, \tag{9}$$

where L_a is the armature inductance (H), K_g is a dimensionless constant of the generator, K_ϕ is the magnetic flow coupling constant (V•s/rad), R_a is the armature resistance (Ω), R_L is the resistance of the load (Ω), considered in this study as purely resistive, w is the angular rotor speed (rad/s), I_a is the armature current (A), λ is the tip-speed ratio which is dimensionless, and $[\alpha, \beta, \gamma, \tau]$ is the set of values of the filter that the blades implement.

The power coefficient C_p (6) depends on the characteristics of the specific wind turbine; J is the rotational

inertia ($\text{Kg}\cdot\text{m}^2$), R is the radius or blade length (m), ρ is the air density (Kg/m^3), K_f is the friction coefficient ($\text{N}\cdot\text{m}/\text{rad}/\text{s}$), K_θ and T_θ are dimensionless parameters of the pitch actuator, v_{ef} is the effective wind velocity in the blades (m/s), v_M is the wind velocity measured by the anemometer sensor, and dist is the external disturbance. In this approach, the pitch reference θ_{ref} is the manipulated variable and the output power P_{out} is the controlled variable.

It is worth to remark that the effective wind, v_{ef} , is not equal to the measured wind, v_M . As formalized by (3–4), the effective wind is the one really transformed into mechanical power by the generator. The disturbances that affect the WT, especially in FOWTs, such as waves and currents, distort the measurements making larger the differences between v_{ef} and v_M . Another source of this mismatching is due to the effect of the blades in the wind. Here, this effect is modeled by a filter (2).

Considering regular waves, the external disturbance has been modeled as a sinusoidal signal with white Gaussian noise (10).

$$\text{dist} = A_d \cdot \sin\left(\frac{2\pi}{T_d} \cdot t\right) + A_d \cdot K_{\text{ns}} \cdot \text{rand}(t) + C_d \quad (10)$$

where A_d is the amplitude of the disturbance in m/s , T_d (s) is the period of the wave, C_d is a constant, K_{ns} is a coefficient to adjust the signal noise ratio, and $\text{rand}()$ denotes the random function.

The wind produces ocean waves with periods between 0.1 and 300 s [32]. So, we have considered these values in the experiments. Storms and earthquakes produce waves with longer periods, but they have not been tested in this work.

The parameters of the wind turbine used during the simulation experiments are shown in Table 1, extracted from [30].

4 Hybrid control strategy design

4.1 Architecture of the hybrid controller

The architecture of the hybrid control strategy is shown in Fig. 1. It is composed of a fuzzy logic controller (FLC) and a module that obtains the effective wind speed using a deep learning system. The FLC receives the power reference, P_{ref} , and the current output power, P_{out} , to calculate the error (11) and generate the pitch reference, θ_{ref} . Besides, this intelligent controller also receives the effective wind speed, V_{DL} , calculated by the deep learning module. The output of the FLC control, i.e., the pitch reference θ_{ref} , is the signal that feeds the WT pitch actuator.

$$P_{\text{err}}(t_i) = P_{\text{ref}}(t_{i-1}) - P_{\text{out}}(t_{i-1}) \quad (11)$$

The wind turbine operation is determined by the wind speed measured by the sensors V_M , the external disturbance dist , and the pith reference generated by the FLC, θ_{ref} , that are the inputs to the WT model. The outputs are the output power, P_{out} , that is sent to the FLC, and the generator speed, w , its acceleration \dot{w} , the current I_a , and the pitch angle θ .

The deep learning module (DLM) receives as inputs the four outputs of the WT model, namely, w, \dot{w}, I_a, θ , and the measured wind speed, V_M . The DLM estimates the effective wind speed, V_{DL} . As it will be shown in the discussion of the results, the use of this effective wind instead of the measured wind produces relevant improvements in the control performance.

The effective wind calculated by the DLM is obtained as a combination of the estimation of the current effective wind, V_{ACT} , and the prediction of the future effective wind, V_{FUT} . Estimation refers to the calculation of the effective current wind speed, and prediction refers to the future effective wind speed. Estimation and prediction are linearly weighted to obtain the effective wind, V_{DL} used as input of the FLC (15).

The operation of this control architecture can be formalized by expressions (12–14).

$$\text{ACT}(t_i) = f_{\text{DL-ACT}}(w(t_{i-1}), \dot{w}(t_{i-1}), I_a(t_{i-1}), \theta(t_{i-1}), V_M(t_{i-1}), V_{\text{ACT}}(t_{i-1})), \quad (12)$$

$$V_{\text{FUT}}(t_i) = f_{\text{DL-FUT}}(w(t_{i-1}), \dot{w}(t_{i-1}), I_a(t_{i-1}), \theta(t_{i-1}), V_M(t_{i-1}), V_{\text{FUT}}(t_{i-1})), \quad (13)$$

$$V_{\text{DL}}(t_i) = K_{\text{ACT}} \cdot V_{\text{ACT}}(t_i) + K_{\text{FUT}} \cdot V_{\text{FUT}}(t_i), \quad (14)$$

where K_{ACT} and K_{FUT} are constants used to adjust the relationship between estimation and prediction. Both constants are in the range [0–1] and must fulfill the relationship $K_{\text{ACT}} + K_{\text{FUT}} = 1$.

The output range of the FLC is $[-\frac{\pi}{4}, \frac{\pi}{4}]$ (rad). However, the input range of the wind turbine is $[0, \frac{\pi}{2}]$ (rad). Hence, a bias of $\frac{\pi}{4}$ (rad) is included to adapt the output of the FLC to the input of the WT (15).

$$\theta_{\text{ref}}(t_i) = \frac{\pi}{4} - f_{\text{FLC}}(P_{\text{err}}(t_i), V_{\text{DL}}(t_i)), \quad (15)$$

4.2 Fuzzy logic controller (FLC)

The fuzzy controller is implemented by a Takagi–Sugeno structure with two inputs, P_{err} and V_{DL} , and one output, θ_{ref} . The input P_{err} is assigned 3 Gaussian fuzzy sets, Negative, Zero and Positive, uniformly distributed in the range $[-250, 250]$ W, and width 87.5. The speed V_{DL} is defined by 3 uniformly distributed Gaussian fuzzy sets in

Table 1 Parameters of the wind turbine model

Parameter	Description	Value/Units
L_a	Inductance of the armature	13.5 mH
K_g	Constant of the generator	23.31
K_ϕ	Magnetic flow coupling constant	0.264 V/rad/s
R_a	Resistance of the armature	0.275 Ω
R_L	Resistance of the load	8 Ω
J	Inertia	6.53 kg m ²
R	Radius of the rotor	3.2 m
ρ	Density of the air	1.223 kg/m ³
K_f	Friction coefficient	0.025 N m/rad/s
$[c_1, c_2, c_3]$	C_p constants	[0.73, 151, 0.58]
$[c_4, c_5, c_6]$	C_p constants	[0.002, 2.14, 13.2, 18.4]
$[c_7, c_8, c_9]$	C_p constants	[18.4, -0.02, -0.003]
$[K_\theta, T_\theta]$	Pitch actuator parameters	[0.15, 2]
$[\alpha, \beta, \gamma, \tau]$	Blade filter constants	[0.55, 0.832, 1.17, 9]
K_{ns}	Noise-signal coefficient	0.5
$[K_{ACT}, K_{FUT}]$	Actual-future coefficients	[0.5, 0.5]

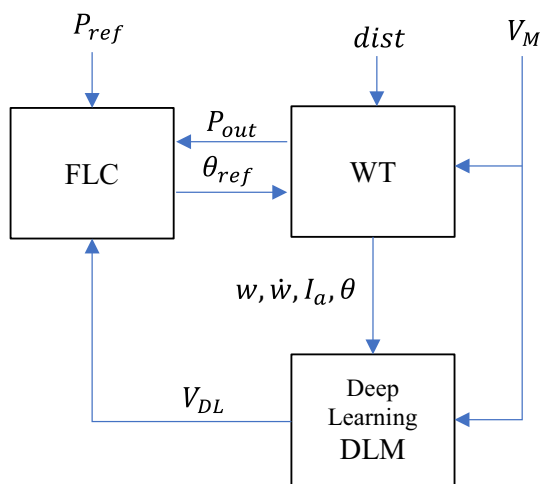


Fig. 1 Architecture of the hybrid controller

the interval [12.25, 13] m/s, the width is 0.175. Its labels are Low, Medium and High. The output is a singleton that can take 3 values: $-\pi/4$, 0, and $\pi/4$ (rad). The configuration of the fuzzy system has been obtained by trial and error. Figure 2 shows the fuzzy sets of the inputs.

The fuzzy rule base is as follows:

- If $P_{err} = \text{Neg}$ and $V_{DL} = \text{Low}$ then out = 0
- If $P_{err} = \text{Neg}$ and $V_{DL} = \text{Med}$ then out = $-\pi/4$
- If $P_{err} = \text{Neg}$ and $V_{DL} = \text{High}$ then out = $-\pi/4$
- If $P_{err} = \text{Zero}$ and $V_{DL} = \text{Low}$ then out = 0
- If $P_{err} = \text{Zero}$ and $V_{DL} = \text{Med}$ then out = 0
- If $P_{err} = \text{Zero}$ and $V_{DL} = \text{High}$ then out = 0
- If $P_{err} = \text{Pos}$ and $V_{DL} = \text{Low}$ then out = $\pi/4$

- If $P_{err} = \text{Pos}$ and $V_{DL} = \text{Med}$ then out = $\pi/4$
- If $P_{err} = \text{Pos}$ and $V_{DL} = \text{High}$ then out = 0

These fuzzy rules have been obtained by the experience and knowledge about how the system works. If the power error is positive, and so the output power is below the rated value, it is necessary to decrease the pitch reference. On the contrary, if the power error is negative, an increment of the pitch reference is the best option.

These simple rules are modified by the action of the wind. If the power is low but the wind speed is high, it is not so necessary to decrement the pitch reference because the wind itself tends to increase the power. On the other hand, if the power is high but the wind speed is low, the power itself tends to decrease making not so necessary to increase the pitch reference.

The nonlinear control surface of the fuzzy control is shown in Fig. 3. It is possible to observe how low wind speeds and positive power errors produce a pitch reference close to zero, in order to increase the output power and thus, to reduce the error. On the other hand, high winds and negative power errors originate a pitch reference close to feather (90°), to reduce the power and the error.

4.3 Deep learning module (DLM)

The scheme of the DLM is shown in Fig. 4. A virtual sensor provides an estimation of the effective wind; it is obtained based on available signals of the wind turbine (w, \dot{w}, I_a, θ) and on the measured wind speed, V_M . A detailed description of the virtual sensor can be found in

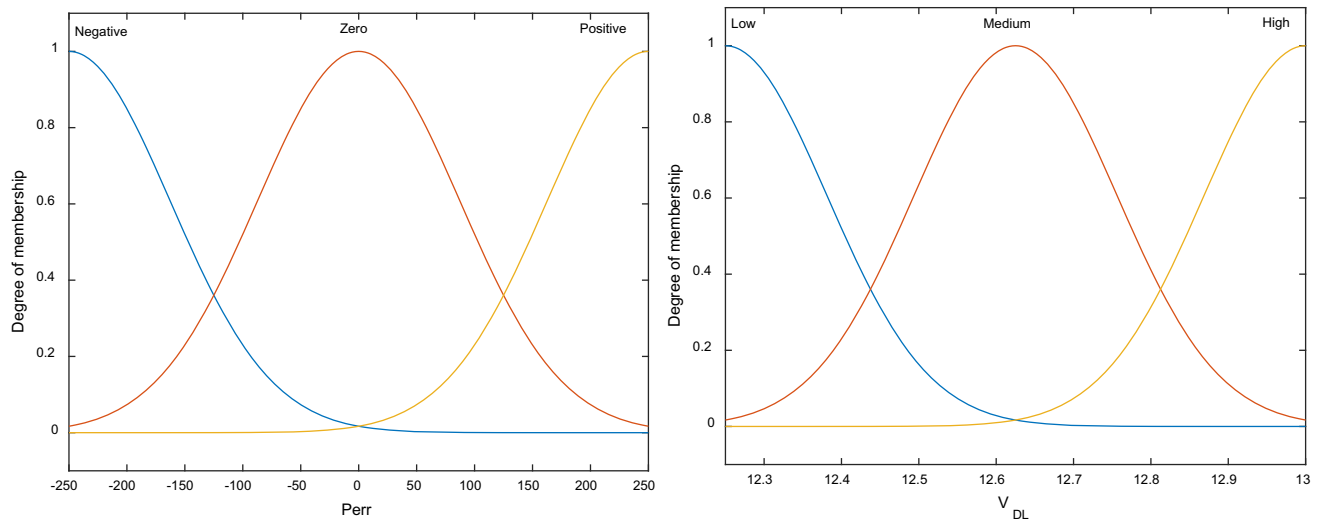


Fig. 2 Fuzzy sets of the inputs of the FLC: power error (left) and effective wind speed (right)

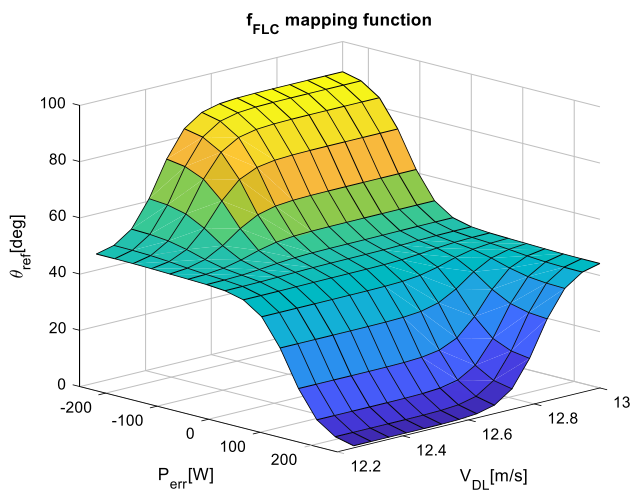


Fig. 3 Control surface of the fuzzy logic controller

[33]. The output of this virtual sensor is used as labeled information to train the deep learning networks.

The DLM has two neural networks. One is used to predict the effective wind in the next control period, V_{FUT} . It has a memory block to delay the input signals ($w, \dot{w}, I_a, \theta, V_M$). Therefore, the signal values at time $(i - 1)$ are related to the outputs at time i . These signals go through this block when the switch SW_1 is in training mode, that is, $SW_1 = \text{down}$.

The other neural network is in charge of the estimation of the effective wind at current control time, i.e., V_{ACT} . The outputs of the WT block feed this neural network directly, without applying any delay. Both values, prediction and estimation of the wind speed, are combined to obtain the effective wind, V_{DL} , that is the input of the intelligent FLC.

The recurrent long short-term memory neural networks (LSTM), commonly used in the deep learning field, are

formed by a set of recurrently connected units, which are usually called memory blocks or hidden units [34]. Each one is made up of a cell, an input gate, an output gate, and a forget gate. The cell acts as the memory, and the gates regulate the flow of learning and forgetting information within the unit. The LSTM structure is able to learn long time dependencies between step times of a sequence of data, and it has been widely used for classification and prediction.

The deep learning NNs of Fig. 4 are based on these LSTM layers and have the architecture shown in Fig. 5. They are composed of several long short-term memory (LSTM) layers in cascade. During the simulation experiments, the number of LSTM structures is varied to analyze its influence in the training. Between each two LSTMs, there is a DROP layer. Connected to the output of the last DROP layer, there is a fully connected layer (FULL). Finally, the last layer of the network is a regression.

The DROP layer randomly sets input elements to zero with certain probability. In the experiments, this probability has been also varied to analyze its influence. This layer contributes to avoid the network overfitting [35].

In the fully connected layer, all the inputs are multiplied by a set of weights, and a bias is added. Finally, the regression layer computes the mean squared error of the loss function and it is used for the training.

The list of parameters of the DLM that is used in the simulations is shown in Table 2. The third column represents the range of values of the parameter. These parameters have been initially obtained by trial and error. Then, they have been varied in the experiments presented in the next sections to study how they affect the training performance of the deep learning module.

Fig. 4 Deep learning module

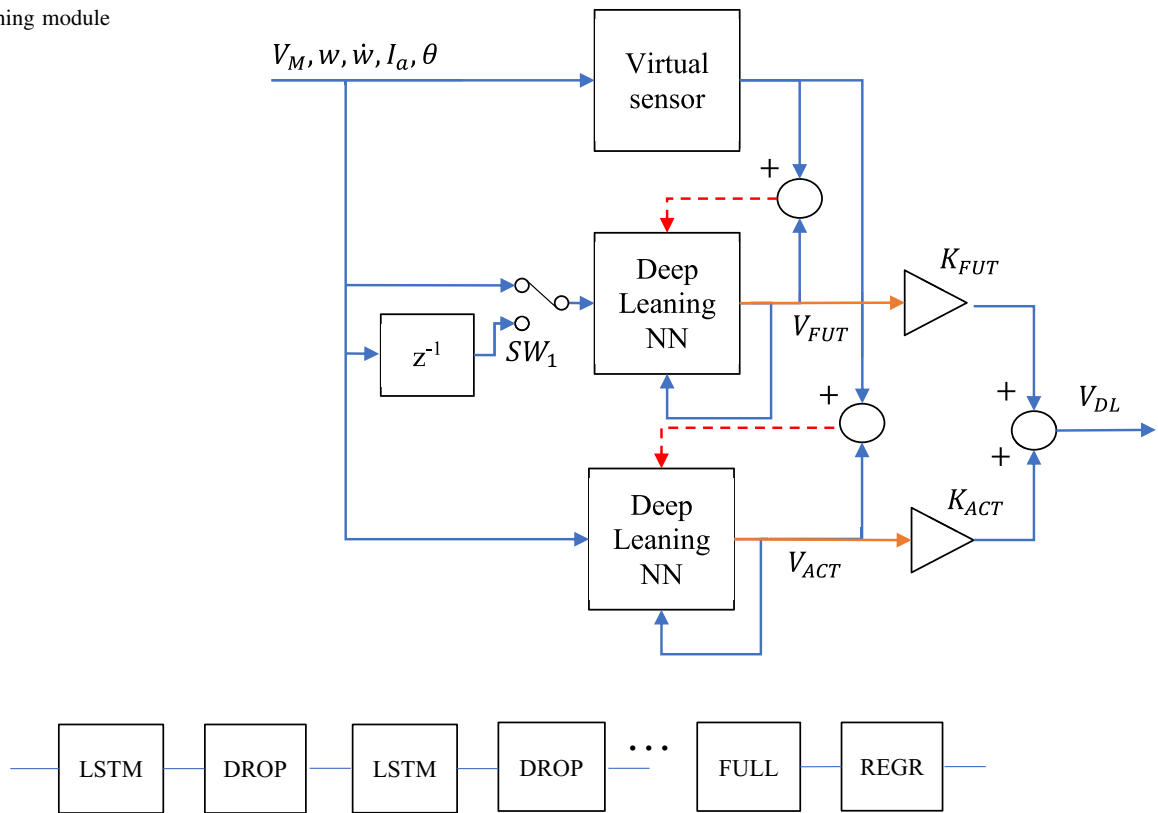


Fig. 5 Architecture of the deep learning neural networks (variant number of layer in the experiments)

Table 2 Parameters of the deep learning module

Parameter	Description	Range
nH	Number of hidden units	25–350
nL	Number of LSTM cells	1–10
δ	Dropout coefficient	0.1–0.9
bs	Batch size	10–100

5 Results and discussion

This hybrid control strategy has been applied to the model of a small wind turbine. Simulation results have been obtained using Matlab/Simulink software. The duration of each simulation is 350 s. The deep learning module is trained during the first 100 s. The sample time, T_s , is variable in order to reduce the discretization error, and its maximum value is set to 10 ms. The parameters of the disturbance: amplitude, constant value, and period (A_d, C_d, T_d), are constant and set to (0.3,0.3,30), respectively.

In order to evaluate the performance of the hybrid controller, it has been compared with a PID (16) and a fuzzy logic controller (FLC) without the deep learning

strategy (17). In this case, the FLC receives the wind speed measured by the anemometer, V_M , instead of the combination of estimated and future wind speed. The power P_{err} is calculated by (11).

$$\theta_{PID}(t_i) = \frac{\pi}{4} - K_p \left[P_{err}(t_i) + K_d \cdot \frac{d}{dt} P_{err}(t_i) + K_i \cdot \int P_{err} \right] \tag{16}$$

$$\theta_{FLC-MW}(t_i) = \frac{\pi}{4} - f_{FLC}(P_{err}(t_i), V_M(t_{i-1})), \tag{17}$$

where $[K_p, K_d, K_i]$ are the tuning parameters of the PID, that have been obtained by trial and error. Their values are $[\pi/4000, 0.2, 0.1]$, respectively.

5.1 Performance of the controller

The performance of the controller is tested with different wind profiles, that are: random with three different ranges: [11.75–12.25], [12.25–12.6], and [12.6–13.25] m/s; a sinusoidal signal with amplitude 0.17 m/s, period of 30 s, and an average value of 12.4 m/s; a square wave signal, and a sawtooth signal, both with the same amplitude of 0.17 m/s and average of 12.4 m/s, but a period of 50 s.

In these experiments, the disturbance has an amplitude of 0.3 m/s, a period of 30 s, and a constant value of 0.3 m/

s. The deep neural networks are composed by 2 LSTM of 200 hidden units. The drop coefficient is set to 0.2, the batch size is 35, and the “adam” solver algorithm is used.

Figure 6 shows the output power obtained with different pitch control strategies, when the wind profile is random within the range [12.25–12.6] m/s. In Fig. 6, left, the output without any disturbance is shown, and on the right, with disturbance. The blue line represents the output power when the pitch reference is set to 0° ; the red one when the reference is permanently set to 90° . The purple line shows the results of the hybrid controller, FUZ-DW. The green line represents the output when the FLC without DLM is applied, so the wind speed is directly measured by the anemometer (FUZ-MW). The yellow line is the PID regulator output. Finally, the black-dashed line represents the rated power.

It is possible to observe how the blue (pitch reference = 0°) and red (pitch reference = 90°) lines limit the signals, because they correspond to the maximum and minimum values of the output power, respectively. The error obtained by the strategies FUZ-DW and FUZ-MW is smaller than the PID error. Without disturbance, the performance of FUZ-DW and FUZ-MW is similar (Fig. 6, left). However, a clear difference between them appears when the disturbance is considered (Fig. 6, right). As expected, this improvement of the performance is produced after the DLM module is trained, that is, from $t = 100$ s onwards.

Figure 7 shows the output power when different control strategies are used, and the wind profile is sinusoidal. The color code is the same as in Fig. 6. As expected, the sinusoidal shape of the wind is noticeable in the signals. Again, the performance of FUZ-DW and FUZ-MW is better than the performance of the PID. But in this case,

this difference is even larger than for the random wind (Fig. 7, left). On the other hand, with disturbance, FUZ-DW clearly improves the output of the FUZ-MW (Fig. 7, right), since the disturbance is considered when the pitch reference is calculated.

Figure 8 presents the same comparison when the wind has a ramp shape. Without disturbance, the fuzzy controller works better than the PID, providing smaller overshoot and shorter settling time. This improvement is less noticeable when the disturbance affects the WT, but still the error is smaller. With disturbance, the performance of FUZ-DW is slightly better than with FUZ-MW. This small improvement may be due to the fact that the ramp is saturated at around $t = 160$ s and, in this experiment, the neural networks have not been trained for this type of saturation.

In addition to these figures, numerical results have been also obtained. Tables 3, 4, 5 show the root mean squared error (RMSE), the mean value (Mean), and the standard deviation (STD) of the output power. The columns represent the results with the PID, the FUZ-DW (FLC + DLM), and FUZ-MW (FLC without DLM). The best results have been boldfaced. The last two rows of these tables represent the average and the standard deviation of the metrics for all wind profiles.

According to Table 3, for almost all wind profiles the RMSE is smaller when the FUZ-DW control strategy is applied. The only exception is with random wind in the range 12.6–13.25 m/s, where the PID regulator works slightly better. This specific wind range of 12.6–13.25 m/s is already high, and if we add the disturbances, it is even higher. That is, it makes the power get values over the rated power almost all the time. The best strategy in this situation is to keep the pitch at the feather position, i.e., 90° . With this wind speed, the power error is mostly negative, and the

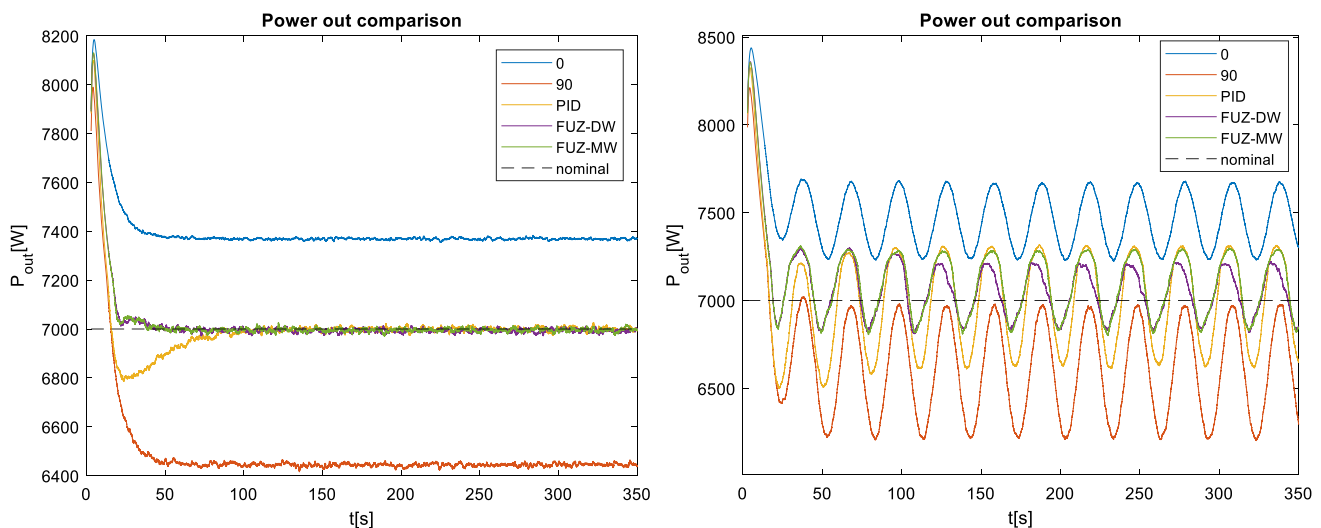


Fig. 6 Comparison of the output power with different control strategies for random wind without disturbance (left) and with disturbance (right)

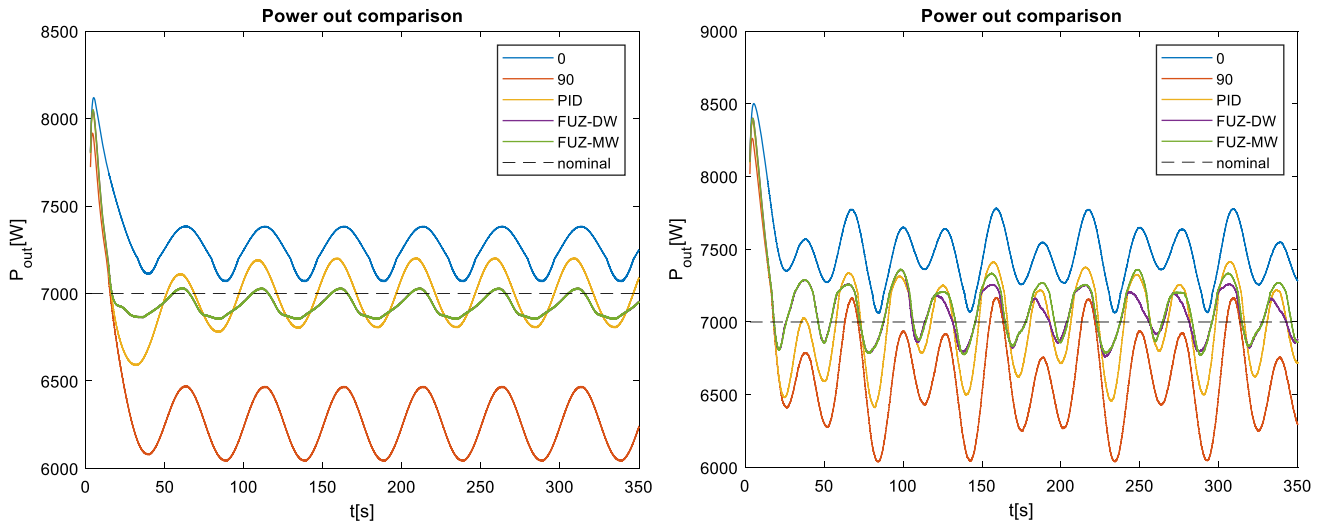


Fig. 7 Comparison of power output with different control strategies for sinusoidal wind profile without disturbance (left) and with disturbance (right)

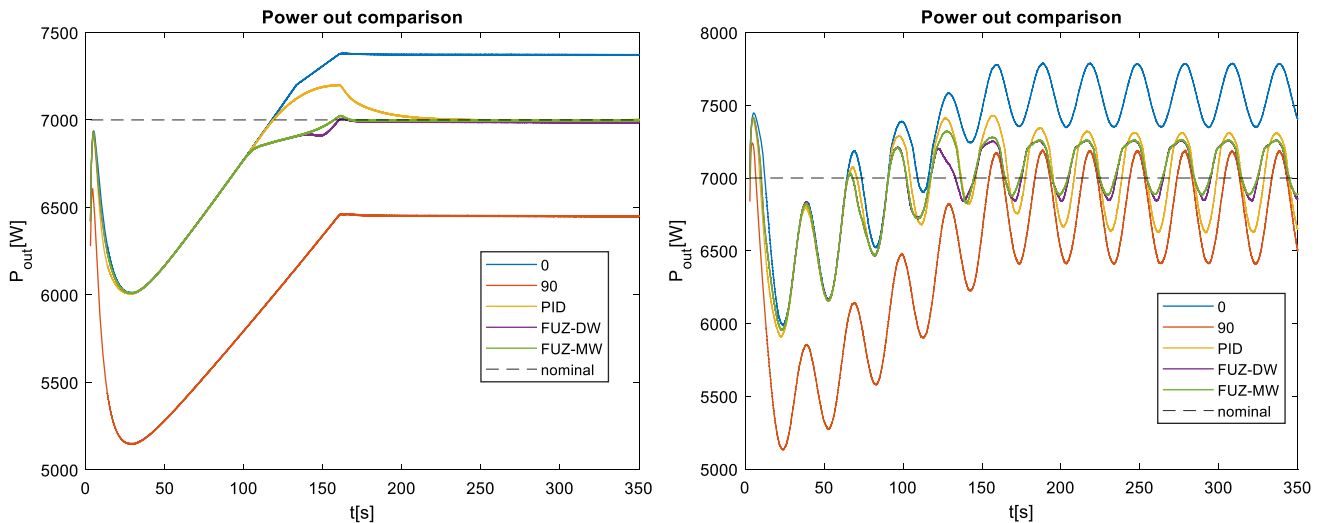


Fig. 8 Comparison of power output with different control strategies for a ramp-shape wind without disturbance (left) and with disturbance (right)

PID controller keeps the blades in feather position due to the integral term. The short time the power error is positive is not enough to compensate the accumulated negative error and so, the pitch does not change. This explains why, in this case, the PID controller gives a lower RMSE than the other techniques, as shown in Table 3. Another interesting result is that the fuzzy logic controller provides better results than the PID. With the FLC, the largest error appears for random wind in the range 12.6–13.25 m/s. It may be explained because of the positive average value of the disturbance; since $C_d = 0.3$, it makes the WT work closer to the limits of the operation for high wind. As the smallest RMSE is obtained with the FUZ-DW strategy for almost all wind profiles, the average value is also the smallest one. If the results of the different experiments are

compared, it can be seen how the PID gives the largest but the most homogeneous values. Thus, the standard deviation obtained with the PID controller is the smallest one.

However, in almost all cases the mean value with the PID is the best (Table 4), and thus its corresponding average value (Table 4) is also the best. This strategy also provides the most regular results. A possible explanation is that the PID tends to provide a more symmetric power error, and thus, the positive values compensate the negative ones, and the mean value is closer to the reference. The power error, defined by (11), can be positive or negative. That is why the mean value of the output power is affected by the sign of the values.

In addition to have a small RMSE and a mean value close to the nominal power, it is also desirable that the

Table 3 Comparison of the RMSE of the output power [W] for different control strategies and wind profiles

Wind profile	PID	FUZ-DW	FUZ-MW
Constant 12.25	302,1556	234,3243	265,2781
Constant 12.75	320,5676	292,5434	292,6662
Random 11.75–12.25	307,9626	257,1144	269,0553
Random 12.25–12.6	305,1875	237,9616	260,9566
Random 12.6–13.25	334,3611	351,9683	351,8860
Sinusoidal	328,9357	247,0831	265,4648
Ramp	358,9158	312,9285	315,1404
Square wave	332,0909	261,4136	273,5319
Sawtooth	309,8832	244,1759	262,7312
Average value	322,2289	271,0570	284,0789
Standard deviation	17,2801	37,69,889	29,1807

Bold values indicate the best results

Table 4 Comparison of the Mean output power [kW] for different control strategies and wind profiles

Wind profile	PID	FUZ-DW	FUZ-MW
Constant 12.25	6,9964	7,0362	7,0904
Constant 12.75	7,0313	7,1856	7,1869
Random 11.75–12.25	6,9652	6,9145	6,9310
Random 12.25–12.6	7,0069	7,0822	7,1144
Random 12.6–13.25	7,1651	7,2690	7,2692
Sinusoidal	7,0032	7,0813	7,1112
Ramp	6,9273	6,9407	6,9608
Square wave	7,0112	7,0812	7,1063
Sawtooth	7,0053	7,0828	7,1132
Average value	7,0124	7,0748	7,0981
Standard deviation	0,0611	0,1029	0,0970

Bold values indicate the best results

output power does not have large oscillations. The STD measures this deviation. In almost all cases, the hybrid FUZ-DW control gives the smallest STD; on the contrary, the PID gives the largest one. This explains why the smallest average value in Table 5 is obtained with the FUZ-DW strategy. As in previous cases, the most regular results are given by the PID controller that has the smallest standard deviation.

5.2 Analysis of the influence of the disturbance parameters

In the previous experiments, the parameters of the disturbance, i.e., amplitude, gain, and period, A_d , C_d , T_d , were constant and set to (0.3,0.3,30), respectively. Now, in order to evaluate the robustness of the proposal, these parameters

Table 5 Comparison of the STD of the output power [W] for different control strategies and wind profiles

Wind profile	PID	FUZ-DW	FUZ-MW
Constant 12.25	302,1388	213,463	224,2909
Constant 12.75	318,9964	225,9429	225,036
Random 11.75–12.25	304,0997	242,5667	260,1064
Random 12.25–12.6	305,1047	223,2451	234,4127
Random 12.6–13.25	290,6526	226,7607	226,3661
Sinusoidal	328,9183	233,2074	240,9406
Ramp	351,5165	307,3091	312,7199
Square wave	331,8928	248,3857	251,9187
Sawtooth	309,8344	229,6215	236,9886
Average value	315,9060	238,9447	245,8644
Standard deviation	17,7098	26,0535	26,2948

Bold values indicate the best results

have been varied to see their influence on the response. The wind profile is sinusoidal as in the experiment of Fig. 7. The MSE for the different control strategies is shown in Fig. 9. The blue line represents the MSE with the PID, the red one with the FUZ-DW, and the yellow line is the response of the fuzzy controller with measured wind but without learning.

As it is possible to see in Fig. 9, the FLC gives smaller MSE than the PID for all the disturbance amplitudes. Moreover, the performance of the FUZ-DW is better than the FUZ-MW in all cases. This improvement is larger for medium amplitudes. If the amplitude is small, the disturbance has a small impact too, and thus the improvement of the FUZ-DW scheme is smaller. When the amplitude is very large, the WT operates closer to the limits, and the results of the FUZ-DW and FUZ-MW control strategies are similar, and thus the improvement is also smaller.

On the other hand, when C_d is very large, the performance of the PID is better than the fuzzy controller. This happens around $C_d = 0.66$ for the FUZ-DW, and around $C_d = 0.44$ for the FUZ-MW control. Thus, it could be said that FUZ-DW is more robust. Moreover, the MSE when the hybrid FUZ-DW control is applied is smaller than with any other control strategy, no matter the value of the C_d .

It is also remarkable how the FLC is less sensible than the PID to the variations of the period of the disturbance. Moreover, it provides smaller MSE than the PID controller for all the tested periods. For disturbances with small periods, the performance of the PID, FUZ-DW and FUZ-MW controllers is similar. This may be explained due to the fact that the blades act as a low-pass filter that reduces the effect of these frequencies. On the other hand, the improvement of the hybrid FUZ-DW control in comparison with the FUZ-MW controller tends to increase with the

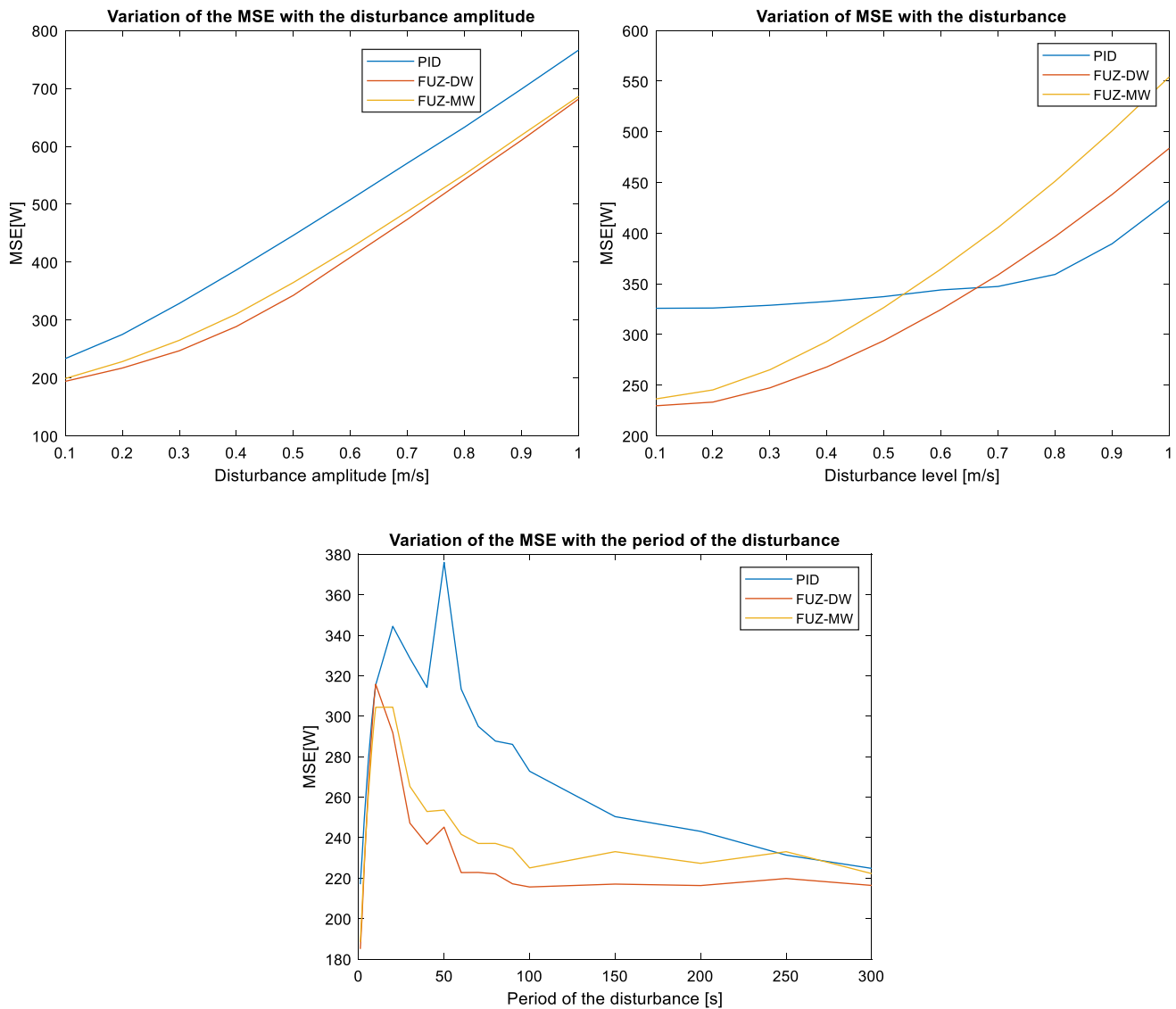


Fig. 9 Evolution of MSE with the amplitude of the disturbance A_d (top-left), with the disturbance level C_d (top-right), and with the period of the disturbance T_d (bottom)

period of the disturbance, though there are local minimums at 100 s, 200 s and 300 s.

5.3 Influence of the configuration of the deep learning neural networks

In this section, in order to evaluate the influence of the configuration of the deep neural networks in the training process, different experiments varying some parameters of the configuration have been carried out. Specifically, the influence of the number hidden units, the drop coefficient, the batch size, the number of LSTM networks, and the solver algorithm have been considered. In all these experiments, the wind profile is sinusoidal as in Fig. 7, and

the disturbance has an amplitude of 0.3 m/s, a gain of 0.3 m/s, and a period of 30 s.

5.3.1 Variation of the number of hidden units

In this experiment, the neural networks have two LSTM structures, the drop coefficient is 0.2, the batch size is 35, and the solver algorithm is “adam.” Figure 10 shows the training process for different number of hidden units. The y-axis represents the RMSE of the training. The color code of the legend indicates the number of hidden units.

In general, increasing the number of hidden units improves the training. In fact, it is possible to observe in Fig. 10 how the learning accelerates with the number of

Fig. 10 Training process for different number of hidden units, nH



hidden units and the RMSE is reduced. However, values bigger than 175 do not provide relevant improvements.

5.3.2 Variation of the dropout coefficient

In the next experiment, the neural networks have two LSTM structures with 200 hidden units, the batch size is 35, and the solver algorithm is “adam.” Figure 11 shows the training process for different drop coefficients. The y-axis represents the RMSE of the training. The color code of the legend indicates the drop coefficient.

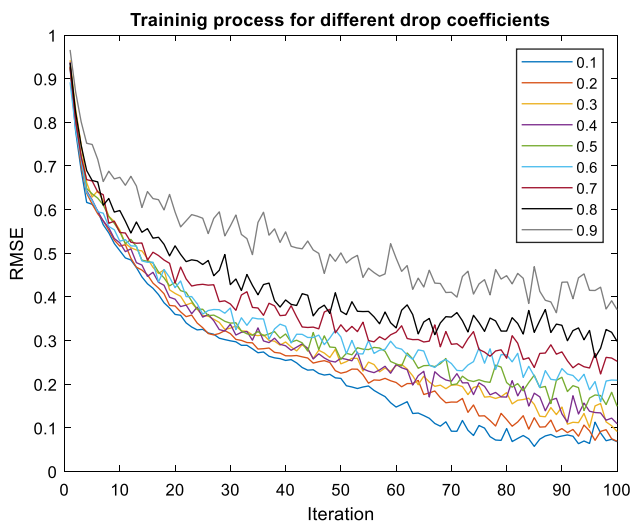


Fig. 11 Training process for different dropout coefficients, δ

The dropout layer randomly sets some input elements to zero, with a probability given by the dropout coefficient. This helps prevent the network from overfitting [35]. However, in our case we have set the number of training iterations to 100 and the overfit is not relevant. In fact, it is possible to see how larger drop coefficients produce bigger oscillations in the training and larger RMSE. The performance of the dropout coefficients with 0.1 and 0.2 value is comparable; thus, we have set the dropout coefficient to 0.2 in the rest of experiments.

5.3.3 Variation of the mini-batch size

The neural networks have two LSTM structures with 200 hidden units, the drop coefficient is 0.2, and the solver algorithm is “adam”. Figure 12 shows the training process for different batch sizes. The y-axis represents the RMSE of the training. The color code of the legend indicates the batch size.

The mini-batch is a subset of the training set that is used to evaluate the gradient of the loss function and to update the weights of the network. As it can be seen in Fig. 12, in our case the effect of the mini-batch size is not remarkable.

5.3.4 Variation of the number of LSTM structures

In this experiment, the number of LSTMs structures varies, but the total number of hidden units is kept equal to 400. In order to calculate the number of hidden units of each structure, the total number of hidden units is divided by the

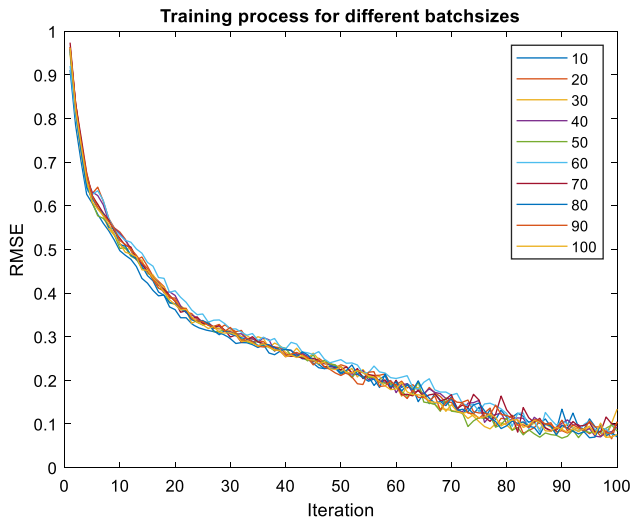


Fig. 12 Training process for different mini-batch sizes, b_s

number of LSTMs. The integer coefficient of the division is used as number of hidden units in all the LSTM structures. If the division is not an integer, the rest is assigned to the first LSTM structure. For example, for 7 LSTMs the distribution of number of hidden units between the LSTM structures is [58, 57, 57, 57, 57, 57, 57]. Therefore, all the structures have the same number of units except the first one that may have extra hidden units. On the other hand, the drop coefficient is 0.2, and the solver algorithm is “adam”. Figure 13 shows the training process for different batch sizes. The y-axis represents the RMSE of the training. The color code of the legend indicates the number of LSTM structures.

In general, increasing the number of LSTMs decelerates the training and increases the RMSE for large numbers. It may be explained as a dropout layer with the same

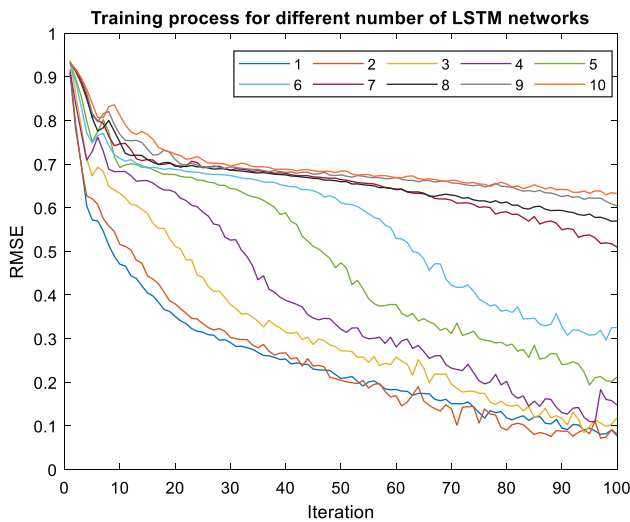


Fig. 13 Training process for different number of LSTM structures, n_L

coefficient is inserted between each two LSTM structures. This increases the effect of the dropout. As a slight improvement is observed with 2 LSTMs vs. 1 LSTM from iteration 60, the number of LSTM networks is set to 2.

5.3.5 Variation of the solver algorithm

Finally, the influence of the solver algorithm is evaluated. The configuration of the hybrid controller is 2 LSTMs structures of 200 hidden units, the dropout coefficient is 0.2, and the mini-batch size is 35. Three different solvers have been tested: stochastic gradient descent with momentum (SGDM); root mean square propagation (RMSprop); and adaptive moment estimation (ADAM). Figure 14 shows the training process for different solvers. The y-axis represents the RMSE of the training. The color code of the legend indicates the solver algorithm.

In this case, the “adam” solver produces the most regular training process and the smallest RMSE at iteration 100. On the other hand, “rmsprop” provides a faster training during the first iterations, but “adam” overpasses it around iteration 20. For these reasons, we have used the “adam” solver in the other experiments.

6 Conclusions and future works

Nowadays it may be necessary in many cases to combine intelligent and classical techniques that complement each other in order to address complex control problems; this combination is also commonly called hybridization. This is the case of the pitch control of a wind turbine here proposed.

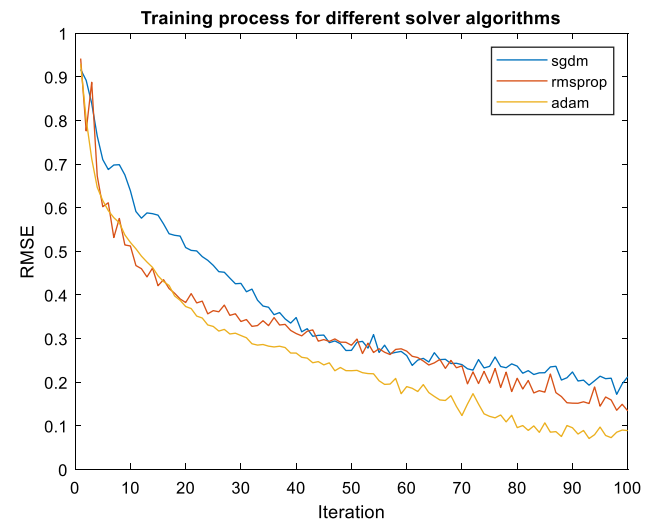


Fig. 14 Training process for different solver algorithms

In this work, a hybrid control consisting of a fuzzy logic controller and a deep learning module with two neural networks has been proposed. This intelligent control strategy is applied to control the blade angle of a wind turbine subjected to disturbances. These disturbances cause the wind measured by the anemometers to be different from the wind that reaches the blades and which is effectively transformed into mechanical energy.

As the response of the WT depends on the wind speed, it is a must to consider the wind information in the control strategy. However, as explained, this information can be inaccurate or uncertain due to the disturbances. Therefore, the use of intelligent techniques seems to be a good approach to face the imprecision of the wind measures.

Thus, deep learning neural networks are used for effective wind estimation and prediction. This effective wind is used as one of the fuzzy logic controller inputs. LSTM structures are used in the design of neural networks. The simulation results show an improvement in the control performance when using deep learning neural networks.

In addition, the influence of the deep learning configuration parameters on the training has been evaluated. The conclusions of this analysis can be summarized as follows. The batch size practically does not affect the training performance. Besides, in this application, it is more efficient to work with small dropout coefficients. The adam solver algorithm provides the best performance. It has been also shown how increasing the number of hidden units improves the training. However, increasing the number of LSTMs while keeping the total number of hidden units decelerates the training.

Among other possible future works, we may highlight the extension of the DLM to predict the vibrations in a floating wind turbine. It would be also desirable to implement the controller in a real turbine and to generalize it to the control of large-scale turbines.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was partially supported by the Spanish Ministry of Science, Innovation and Universities under MCI/AEI/FEDER Project Number RTI2018-094902-B-C21.

Data availability The findings of this study have been generated by the equations and parameters cited within the article.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate

if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Green Peace (2021) <https://es.greenpeace.org/es/trabajamos-en/cambio-climatico/carbon/> Last accessed on 2021/02/13
- Paris, Climate (2021) https://ec.europa.eu/clima/policies/international/negotiations/paris_en. Last accessed on 2021/02/13
- Our World in Data (2020) <https://ourworldindata.org/renewable-energy> Last accessed on 2021/02/13
- IRENA (2019) Future of wind: deployment, investment, technology, grid integration and socio-economic aspects (A Global Energy Transformation paper), International Renewable Energy Agency, Abu Dhabi. https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2019/Oct/IRENA_Future_of_wind_2019.pdf. Last accessed on 2021/02/13
- Shen YW, Yuan JR, Shen FF, Xu JZ, Li CK, Wang D (2019) Finite control set model predictive control for complex energy system with large-scale wind power. Complexity. <https://doi.org/10.1155/2019/4358958>
- Tomás-Rodríguez M, Santos M (2019) Modelling and control of floating offshore wind turbines. *Revista Iberoamericana de Automática e Informática Industrial* 16(4):381–390
- Sierra-García JE, Santos M (2020) Exploring reward strategies for wind turbine pitch control by reinforcement learning. *Appl Sci* 10(21):7462
- Sierra JE, Santos M (2018) Modelling engineering systems using analytical and neural techniques: hybridization. *Neurocomputing* 271:70–83
- Menoyo-Larrazabal J, Santos Peñas M (2016) Intelligent rudder control of an unmanned surface vessel. *Expert Syst Appl* 55:106–117
- Sierra-García JE, Santos M (2021) Switched learning adaptive neuro-control strategy. *Neurocomputing* 452:450–464
- Chavero-Navarrete E, Trejo-Perea M, Jáuregui-Correa JC, Carrillo-Serrano RV, Ronquillo-Lomeli G, Ríos-Moreno JG (2020) Hierarchical pitch control for small wind turbines based on fuzzy logic and anticipated wind speed measurement. *Appl Sci* 10(13):4592
- Moodi H, Bustan D (2019) Wind turbine control using TS systems with nonlinear consequent parts. *Energy* 172:922–931
- Ngo QV, Chai Y, Nguyen TT (2020) The fuzzy-PID based-pitch angle controller for small-scale wind turbine. *Int J Power Electron Drive Syst* 11(1):135
- Neugebauer M, Sołowiej P, Wesolowski M, Nalepa K, Hałacz J (2020). Fuzzy model of wind turbine control. In *renewable energy sources: engineering, technology, innovation* (pp. 541–550). Springer, Cham
- Iqbal A, Ying D, Saleem A, Hayat MA, Mehmood K (2020) Efficacious pitch angle control of variable-speed wind turbine using fuzzy based predictive controller. *Energy Rep* 6:423–427
- Sitharthan R, Karthikeyan M, Sundar DS, Rajasekaran S (2020) Adaptive hybrid intelligent MPPT controller to approximate effectual wind speed and optimal rotor speed of variable speed wind turbine. *ISA Trans* 96:479–489
- Sarkar MR, Julai S, Tong CW, Uddin M, Romlie MF, Shafiullah GM (2020) Hybrid pitch angle controller approaches for

- stable wind turbine power under variable wind speed. *Energies* 13(14):3622
18. Rubio PM, Quijano JF, López PZ et al (2019) Intelligent control for improving the efficiency of a hybrid semi-submersible platform with wind turbine and wave energy converters. *Revista Iberoamericana de Automática e Informática Industrial* 16(4):480–491
 19. Abdelbaky MA, Liu X, Jiang D (2020) Design and implementation of partial offline fuzzy model-predictive pitch controller for large-scale wind-turbines. *Renew Energy* 145:981–996
 20. Rehman S, Khan SA, Alhems LM (2020) A rule-based fuzzy logic methodology for multi-criteria selection of wind turbines. *Sustainability* 12(20):8467
 21. Wu Z, Wang X, Jiang B (2020) Fault diagnosis for wind turbines based on ReliefF and eXtreme gradient boosting. *Appl Sci* 10(9):3258
 22. Duan H, Lu M, Sun Y, Wang J, Wang C, Chen Z (2020) Fault diagnosis of PMSG wind power generation system based on LMD and MSE. *Complexity*, 2020
 23. Khan M, Liu T, Ullah F (2019) A new hybrid approach to forecast wind power for large scale wind turbine data using deep learning with TensorFlow framework and principal component analysis. *Energies* 12(12):2229
 24. Mujeeb S, Alghamdi TA, Ullah S, Fatima A, Javaid N, Saba T (2019) Exploiting deep learning for wind power forecasting based on big data analytics. *Appl Sci* 9(20):4417
 25. Fu J, Chu J, Guo P, Chen Z (2019) Condition monitoring of wind turbine gearbox bearing based on deep learning model. *IEEE Access* 7:57078–57087
 26. Li M, Wang S, Fang S, Zhao J (2020) Anomaly detection of wind turbines based on deep small-world neural network. *Appl Sci* 10(4):1243
 27. Jie W, Jingchun C, Lin Y, Wenliang W, Jian D (2020). Pitch control of wind turbine based on deep neural network. In IOP conference series: earth and environmental science (Vol. 619, No. 1). IOP Publishing, p. 012034
 28. Zhang Z, Zhang D, Qiu RC (2019) Deep reinforcement learning for power system applications: an overview. *CSEE J Power Energy Syst* 6:213–225
 29. Lin Z, Liu X (2020) Assessment of wind turbine aero-hydro-servo-elastic modelling on the effects of mooring line tension via deep learning. *Energies* 13(9):2264
 30. Mikati M, Santos M, Armenta C (2012) Modelado y simulación de un sistema conjunto de energía solar y eólica para analizar su dependencia de la red eléctrica. *Revista Iberoamericana de Automática e Informática industrial* 9(3):267–281
 31. Sierra-García JE, Santos M (2020) Performance analysis of a wind turbine pitch neurocontroller with unsupervised learning. *Complexity*, 2020
 32. Munk WH (1951). Origin and generation of waves. Scripps Institution of Oceanography La Jolla Calif
 33. Sierra-García JE, Santos M (2021) Improving wind turbine pitch control by effective wind neuro-estimators. *IEEE Access* 9:10413–10425
 34. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw* 18(5–6):602–610
 35. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15:1929–1958

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.