Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

A reinforcement-learning approach for individual pitch control

Marion Coquelet[1,2] | **Laurent Bricteux**[2] | **Maud Moens**[1] | **Philippe Chatelain**[1]

[1]Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, Louvain-la-Neuve, Belgium
[2]Fluids-Machines Department, Université de Mons, Mons, Belgium

Correspondence
Marion Coquelet, Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, Louvain-la-Neuve, Belgium.
Email: marion.coquelet@uclouvain.be

Summary

Individual pitch control has shown great capability of alleviating the oscillating loads experienced by wind turbine blades due to wind shear, atmospheric turbulence, yaw misalignment or wake impingement. This work presents a novel controller structure that relies on the separation of low-level control tasks and high-level ones. It is based on a neural network that modulates basic periodic pitch angle signals. This neural network is trained with reinforcement learning, a trial and error way of acquiring skills, in a low-fidelity environment exempt from turbulence. The trained controller is further deployed in large eddy simulations to assess its performances in turbulent and waked flows. Results show that the method enables the neural network to learn how to reduce fatigue loads and to exploit that knowledge to complex turbulent flows. When compared to a state-of-the-art individual pitch controller, the one introduced here presents similar load alleviation capacities at reasonable turbulence intensity levels, while displaying very smooth pitching commands by nature.

KEYWORDS:
Individual pitch control, load alleviation, reinforcement learning, large eddy simulation

1 | INTRODUCTION

Increasing the size of wind turbines is one path to achieve the wind-sourced electrical production targets. Wind turbine producers are following this path and announcing turbines with rated powers of 15 MW by 2025 and rotor diameters exceeding 200 m[1]. Such sizes lead to ever greater challenges in the domain of structure design and load alleviation, as lighter and thus more flexible structural components show more sensitivity toward fatigue loading[2]. Though the widespread variable-speed variable-collective-pitch controllers consider a uniform wind distribution across the rotor plane[3], such gigantic swept surfaces demand to rethink the problem. Indeed, if all blades are pitched at the same angle, the spatial variations of wind speed generate large oscillating loads on the blades over the course of the rotation. Individually controlling the pitch of each blade has repeatedly proven its potential in reducing the rotor-unbalanced loads arising from wind shear, tower shadow, yaw misalignment and atmospheric turbulent structures[4]. A growing body of research has thus emerged in the wind energy community, seeking the best approach to design these individual pitch controllers (IPC)[5].

Most of the IPC research[5] that has been conducted over the years relies on the generic structure depicted in Fig. 1, which is based on two state spaces, the wind turbine space and the control space, linked by a transformation. The blade-root flapwise bending moments $MFn_{1,2,3}$ are oscillating by nature, mainly due to the presence of shear and turbulence, and it is thus expected from the pitching targets $\beta_{1,2,3}$ to react to those measured loads in an oscillatory manner. A transformation is therefore used to go from the wind turbine space, where the pitching targets $\beta_{1,2,3}$ are oscillating, to the controller space, where targeted states $\mathbf{x_{ref}}$ are time-invariant. In practice, the bending moments $MFn_{1,2,3}$ are transformed into a state $\mathbf{x}$ defined in the control space. The load regulation problem is solved in the control space by determining the command $\mathbf{u}$. Various types of controllers, from simple PIDs to optimal controllers or even robust controllers can be implemented. The control space command is eventually transformed back to the turbine space, leading to a pitch angle command β_i for each blade.

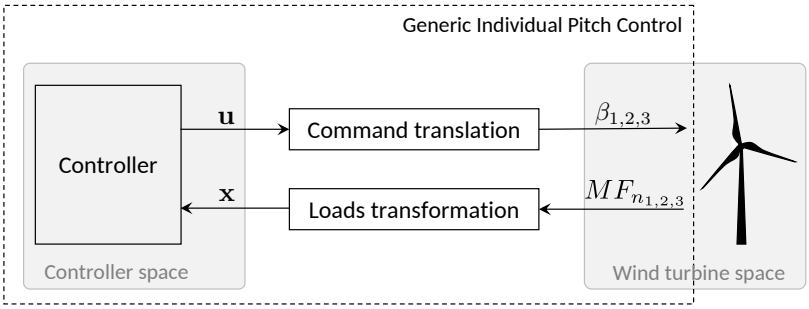


FIGURE 1 Generic representation of IPC controllers.

Three main types of transformation have been reported in the literature. First, many IPC studies have focused on the Coleman [6] transformation, which emerged from the field of helicopter rotor control. The Coleman transformation consists in projecting the loads expressed in the frame of the rotating blades on the fixed coordinate frame of the rotor disc. First investigations of individual pitch control using this transformation were reported in Bossanyi[4] and Caselitz[7], with a first controller design method proposed in van Engelen[8]. Another coordinate transformation was reported by Zhang[9], who uses the Clarke[10] transformation, widely used for electrical machines. The flapwise bending moments are mapped onto a set of orthogonal axes associated to the rotating rotor. Eventually, Leithead and co-workers[11,12,13] extensively studied the case of no transformation of the loads, which, following the aforementioned representation, could mathematically translate to an identity transform. This type of formulation is often referred to as individual blade control instead of individual pitch control.

As the Coleman transformation is the most widely used[5], we focus on this approach to show the multiple types of controllers that have been implemented to alleviate fatigue loads on wind turbines. First, a wide variety of what could roughly be described as feedback controllers have been tested. The original controller presented in Bossanyi[4] and still used in many applications for its simplicity of implementation consists in two separate proportional-integral (PI) loops, one for the tilt axis and one for the yaw axis. This controller architecture then assumes that the tilt and yaw dynamics of the rotor are decoupled, which was shown to be incorrect by Selvam[14] and Geyler[15]. The existence of this dynamic tilt-yaw coupling motivated the investigation of multi-variable control strategies and optimal control was therefore a good candidate. Among the numerous studies, one can highlight the implementation of a Linear Quadratic Regulator (LQR) in Stol[16], where both the collective pitch and individual pitch targets are dealt with in the optimization problem. The reference IPC study[4] also presented a Linear Quadratic Gaussian (LQG) regulator, with a Kalman filter as state estimator. Studies like that of Mirzaei[17] also investigated Model Predictive Control (MPC). A well-known problem with optimal controllers is that they offer no guaranteed stability margins[18], as they do not take into account uncertainty in the model of the system, here the turbine, used for the optimization. This is what motivated the use of robust control approaches, that account for parametric and dynamic uncertainties. Among the possible methods, $\mathcal{H}_\infty$ loop-shaping has been widely used, see examples in Lu[3] and Geyler[15]. It is indeed well adapted to IPC as the control design is performed in the frequency domain. The aforementioned controllers represent the common basis to many IPC controllers, yet additional control loops are sometimes added. On the one hand, feedforward has been investigated. The control approach presented in Selvam[14] consists in an optimal LQG controller working along with a feedforward disturbance rejection controller based on estimated wind speed signals, while Dunne[19] considered the state-of-the-art PI supplemented by a feedforward loop providing the LIDAR measurement of incoming wind speed. On the other hand, repetitive control has also been reported due to its ability to deal with periodic signals and track periodic references[20]. Kallen[21] takes advantage of that and uses the information perceived on a blade to anticipatively pass it on to the next blade. Friis[22] used repetitive model predictive control by including a repetitive wind disturbance in the MPC prediction.

This review shows that the field of feedback control, completed with add-ons like repetitive control and feedforward control, has been thoroughly studied in the literature. Nonetheless, recent years have seen Machine Learning (ML) become a new fundamental way of problem solving in a growing number of domains. ML is commonly divided into (1) supervised learning, dealing with classification and regression tasks; (2) unsupervised learning, addressing clustering and dimensionality reduction topics, and (3) reinforcement learning, focusing on the determination of the optimal actions that an agent should take to achieve a given goal[23]. First studies of ML in the field of fluid dynamics are reported in Koumoutsakos[23]. Reinforcement Learning (RL)[24] is typically the ML paradigm used for control applications and it has been applied for wind power control. Tomin[25] used RL for optimal generator torque and collective pitch control, while Sierra-García[26] focused on the collective pitch controller for above-rated wind speeds. RL investigations were also carried for wind farm power maximization using yaw control: Saénz-Aguirre[27] reported results on

optimally aligning a wind turbine with the upcoming wind, where as Stanfel[28] presented distributed yaw steering for wind farm energy capture maximization. When it comes to load alleviation, reinforcement learning has not been reported so far, though supervised learning has been used as an add-on to classical control theory paradigms. One can mention the data-driven controller selection based on wind conditions clustering proposed by Collet[29] or the Bayesian Optimization ML approach of Mulders[30] for the efficient tuning of Coleman transform-based IPC.

This paper thus aims at investigated the IPC problem as a whole under the eye of RL. This would offer IPC the opportunity to benefit from the ability of RL to learn non-linear problems or high-dimensionality ones, but also to be independent from predetermined control paradigms. A subsidiary goal therefore consists in providing a framework that handles more information and more flexibly than the usual ones. The envisioned paradigm should enable an improved management of the aerodynamic loads that will be encountered in future large rotors. To do so, this work builds on previous investigations by the authors[31] to go further in terms of controller architecture and performance assessment. RL is sometimes referred to as a bio-inspired control mechanism, as it mimics an animal's way of learning by experience[23]. We took the option to go further in terms of bio-inspiration and present, in section 2, a control framework that mimics animal locomotion, as it entails rhythmic motion and is able to adapt to environmental conditions, just as IPC should. The control policy is trained in a low-fidelity environment using blade element momentum theory and a synthetic flow generator, yet it is further tested in a higher fidelity environment using Large Eddy Simulations (section 3). This goes a step further in terms of fidelity from what is seen in the literature: in the aforementioned RL studies[25,26,27,28], the intelligent controllers are trained and tested in low-fidelity environments. Conclusions are drawn in section 4.

2 | METHODOLOGY

We first present the structure developed to address the IPC challenge in 2.1. The controller training through reinforcement learning is then discussed in 2.2. The reference controllers against which the reinforcement-learned IPC is compared are eventually presented in 2.3.

2.1 | Controller structure

As mentioned in section 1, the individual pitch controller structure is somewhat inspired by animal locomotion. Animal locomotion is enabled by a hierarchical structure separating central pattern generators present in the spinal chord and generating the low-level rhythmic patterns from the brain controlling high-level gaits based on the perception of the environment[32]. A similar approach can be used for the IPC problem due to its important once-per-revolution (1P) periodicity. Our control scheme is made of three blocks: a sensing module to provide information on the turbine state, a neural network reacting to it with high-level signals and oscillators generating low-level periodic signals. The generic IPC of Fig. 1 actually fits within this paradigm: the load transformation is the sensing module, the controller corresponds to the trained neural network and the command translators are the oscillators.

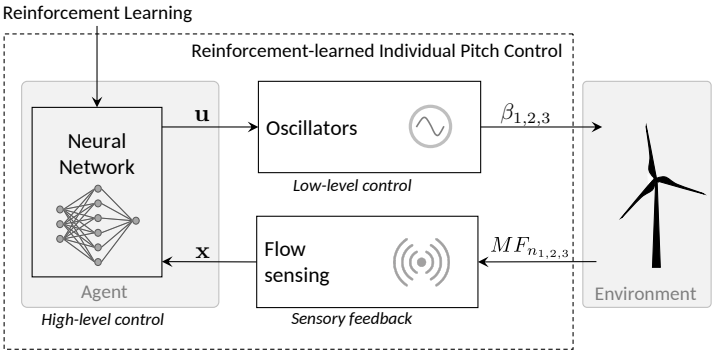


FIGURE 2 Reinforcement-learned individual pitch controller.

2.1.1 | Sensing module: load transformation

We present a novel load transformation approach that is based on the decomposition of the flapwise load into two contributions: the load resulting from the mean flow over the rotor and the azimuthal load oscillation due to the non uniformities in the flow field impacting it. Note that throughout this discussion, temporal variables will be defined in the discrete time formalism: the index k corresponds to the value of a variable at time $t = k\,\Delta t$, where Δt is the control time step. We thus introduce $\epsilon_{b,k}$, the blade-effective load imbalance of blade b at time k, defined as

$$\epsilon_{b,k} = MFn_{b,k} - \overline{M}^f_k\,. \tag{1}$$

$MFn_{b,k}$ is the blade root flapwise bending moment and $\overline{M}^f_k$ is the filtered value of

$$\overline{M}_k = \frac{1}{n_B}\sum_{b=1}^{n_B} MFn_{b,k}, \tag{2}$$

with n_B is the number of blades. As we consider a three-bladed turbine, the mean flapwise load $\overline{M}_k$ shows a significant three-times per revolution (3P) component[4], here we use a notch filter rejecting that 3P frequency. The digital filter transfer function is expressed in the z-domain[33] as

$$G(z) = \frac{\overline{M}^f(z)}{\overline{M}(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \tag{3}$$

leading to the discrete time domain expression of $\overline{M}^f_k$ when using the inverse z-transform[33]:

$$\overline{M}^f_k = b_0\overline{M}_k + b_1\overline{M}_{k-1} + b_2\overline{M}_{k-2} - a_1\overline{M}^f_{k-1} - a_2\overline{M}^f_{k-2}. \tag{4}$$

The coefficients, recovered from the bilinear transformation of a second-order analog notch filter[33], write

$$\{b_0, b_1, b_2\} = \{(1+\gamma)/2, -(1+\gamma)\cos(\omega_0), (1+\gamma)/2\}, \tag{5}$$

$$\{a_1, a_2\} = \{(1+\gamma)\cos(\omega_0), \gamma\}, \tag{6}$$

with $\omega_0 = 2\pi\, f_c\, \Delta t$ (f_c is the cutting frequency and $1/\Delta t$ is the sampling frequency) and $\gamma = \frac{1-\tan(\mathrm{BW}/2)}{1+\tan(\mathrm{BW}/2)}$ (BW is the the -3dB bandwidth)[33].

The blade-effective load oscillation $\epsilon_{b,k}$, as the name suggests, is a quantity associated to the blade and is thus observed at the blade azimuthal position $\theta_{b,k}$. This quantity, which is expressed in a rotating frame, can be translated into a fixed-frame quantity. To do so, we consider the disk swept by the blades and decompose it into n_S sectors of equal size $\Delta\theta = 2\pi/n_S$. We define the sector-effective load oscillation $\epsilon_{s,k}$ as the mean value of $\epsilon_{b,k}$ seen during the passage of any blade b in sector s. To translate from the blade-effective to the sector-effective load oscillation, we make use of two additional variables: $s_{b,k}$, the sector through which blade b is passing at time k, and N_b, the number of time steps that blade b has already spent in its current sector. In practice, sector-related information is updated every time a blade leaves a sector following the procedure described in algorithm 1. The chosen number of sectors in this study in $n_S = 24$, as it offers a fair azimuthal discretisation of the rotor.

Algorithm 1 Algorithm to translate from $\epsilon_{b,k}$ to $\epsilon_{s,k}$

$\overline{M}^f_0 = \overline{M}_0 \quad , \overline{M}^f_1 = \overline{M}_1, \quad \epsilon_{s,1} = 0 \quad \forall\, s, \quad N_b = 1 \quad \forall\, b, \quad s_{b,1} = \text{integer}\left(\theta_{b,1}/\Delta\theta\right) \quad \forall\, b$
while $k \geq 2$ and controller is active **do**
  $\epsilon_{s,k} \leftarrow \epsilon_{s,k-1} \quad \forall\, s$
  $\overline{M}_k = 1/n_B \sum_{b=1}^{n_B} MFn_{b,k}$
  $\overline{M}^f_k = b_0\overline{M}_k + b_1\overline{M}_{k-1} + b_2\overline{M}_{k-2} - a_1\overline{M}^f_{k-1} - a_2\overline{M}^f_{k-2}$
  for b in n_B **do**
    $\epsilon_{b,k} = MFn_{b,k} - \overline{M}^f_k$
    $s_{b,k} = \text{integer}\left(\theta_{b,k}/\Delta\theta\right)$
    if $s_{b,k} \neq s_{b,k-1}$ **then**
      $s^* = s_{b,k-1}$
      $\epsilon_{s^*,k} \leftarrow \frac{1}{N_b}\sum_{\kappa=k-N_b}^{k-1} \epsilon_{b,\kappa}$
      $N_b \leftarrow 1$
    else
      $N_b \leftarrow N_b + 1$
    end if
  end for
end while

The presented sensing module stores the loads temporally within sectors to provide an azimuthal representation of the rotor load imbalances that is updated every time a blade leaves a sector. It is thus recorded information that is used to create a control trajectory, as it is discussed hereunder. This is slightly different from the the Coleman transform, which uses measured loads and directly transforms them using a matrix multiplication. Some temporal relaxation, reduced as the number of sectors increases, therefore exists in this sensing module, but it allows to retain more spatial information than the Coleman transform. Indeed, the latter transforms the loads into a yawing moment M_{yaw} and a tilting one M_{tilt}. Mathematically, this amounts to representing the load imbalances as a 2D linear function. Seen from a sector perspective, this is equivalent to having 3 sectors only. Though this is very well suited to represent shear for example, it is quite limited if one aims at capturing the load imbalances resulting from complex flows, such as the passage of a gust or a lull, the partial impingement of a wake or the kidney shape of the wake of a yawed turbine. For all these complex cases, using a transformation such as the one proposed here allows to have a less constrained azimuthal representation of the oscillating load. This is even more relevant for larger turbines as the size of the structures present in the wind approaches that of their rotor and larger turbines are specifically the ones that should use IPC due to the increased slenderness and flexibility of their blades.

2.1.2 | Neural network: controller

The role of the controller is to bring the ϵ_s of every sector s to zero. As the IPC problem has a strong 1P periodicity, we rely on low-level oscillators that are formally described hereafter to regulate ϵ_s. The controller must therefore provide high-level control target, namely the amplitude A and phase P that will modulate the low-level oscillations. It is a an artificial neural network (NN) that is in charge of mapping the input vector $\mathbf{x}$ (mostly defined as the azimuthally oscillating load) and the output one $\mathbf{u} = [A, P]$. There are various types of NN; here a standard fully-connected multi-layer perceptron (MLP) is chosen [34]. The control problem in itself consists in finding the weights and biases of the NN that optimally alleviate fatigue loads. This is done by Reinforcement Learning and is developed in section 2.2.

2.1.3 | Oscillators: command translation

What generates the periodic movement of animal motion is central pattern generators [35]. They are neural circuits that produce rhythmic motor patterns. As mentioned before, we rely on mathematical oscillators to produce the pitching patterns, as they mimic the low-level locomotion signals generated by central pattern generators. For each blade, the individual pitch angle β_b is computed as the sum of the collective pitch demand β_{CPC} and the differential pitch demand $\Delta\beta_b$. The former is determined by the general variable-speed variable-pitch controller, while the latter is computed as

$$\Delta\beta_b = a\cos\left(\theta_b + p\right) \triangleq a\cos\left(\varphi_b\right). \tag{7}$$

We produce smooth pitching patterns by damping the controller outputs $\mathbf{u} = [A, P]$ into the signals $[a, p]$, where a and p are respectively the actual amplitude of oscillations and phase shift between the azimuthal position of the blade θ_b and the phase of the oscillator φ_b. To this end, we cast a and p as the state variables of second-order systems having respectively A and P as inputs. The dynamics of a and p then obey

$$\frac{1}{\omega_n^2}\ddot{a} + \frac{2\zeta}{\omega_n}\dot{a} + a = A \qquad \text{and} \qquad \frac{1}{\omega_n^2}\ddot{p} + \frac{2\zeta}{\omega_n}\dot{p} + p = P\,. \tag{8}$$

The natural angular frequency ω_n and the damping ratio ζ are determined based on the desired time response of the systems in section 2.2.2.

2.1.4 | Discussion

In summary, the control methodology presented here can be synthesized as a NN (controller) defining high-order pitching targets $\mathbf{u}$ depending on the azimuthally-varying environmental conditions sensed through the blade loads (load transformation) and modulating low-level rhythms accordingly (command translation). Several advantages arise from this IPC formulation. First, the expression of the azimuthally oscillating load offers a good knowledge of the flow impinging on the rotor. It is not limited to what is instantly sensed by the blades, but presents some memory effect and each blade contributes to the collective knowledge of the rotor loads imbalance that needs to be reduced. Information sensed by a blade can be directly leveraged by the controller, which updates its value of A and P and therefore adapts the pitching trajectory to upcoming loading events. This transformation maps the rotating blade loads to fixed loads and could also be of use in other contexts. Second, the specific use of oscillators offers two clear advantages. On the first hand, it ensures very smooth pitching command, as will be shown in the results section. On the other hand, it translates non rhythmic commands (provided by the policy NN) to rhythmic ones. This allows to maintain a reasonable level of complexity for the policy by bringing prior knowledge into the learning process, in the fashion of the policies modulating trajectory generators proposed Iscen [36].

2.2 | Learning methodology

As mentioned before, the NN controller is trained, i.e. its weights and biases are determined, using Reinforcement Learning, the branch of Machine Learning that focuses on control applications. The concept of RL consists in an agent interacting with its environment to learn the best way to

behave as to achieve a goal. It is therefore a bio-inspired way of tackling control problems, as it is not about determining control laws but learning from interaction in a goal-directed manner [24]. It is formalized as finding the most appropriate policy $\pi(\mathbf{a}|\mathbf{s})$, which is a mapping between the state $\mathbf{s}$ of the agent and the action $\mathbf{a}$ to take, given a certain objective function $J(\pi)$ to maximize [24]. In this section, we first briefly present the fundamentals of RL and formalise the IPC problem accordingly. We then present the learning environment and the policy performance evaluation. We eventually present the testing environment and process.

2.2.1 | Reinforcement Learning fundamentals

From the RL perspective, the wind turbine controller is the agent, the environment is the wind turbine itself and the flow surrounding it, and the policy is expressed by the neural network. The role of the NN is to learn the action $\mathbf{a}_l$ that most reduces the fatigue loads on the blades given the loads determined by the sensing module. A new action is taken every learning step l based on the state of the system $\mathbf{s}_l$.

Two important time intervals are defined in RL: the learning step and the learning episode [24]. At each step, the agent receives some representation of the environment through the state $\mathbf{s}_l$, based on what it picks an action $\mathbf{a}_l$. The agent interacts with the environment given this action, receives a numerical reward $r_{l+1} = r(\mathbf{s}_l, \mathbf{a}_l)$ and ends up in a new state $\mathbf{s}_{l+1}$. A learning episode is a succession of learning steps. When it comes to continuous control tasks as it is the case here, an episode does not end when a terminal state is reached but after a chosen number of learning steps. The precise goal of RL is to maximize the objective function $J(\pi)$ built on the cumulated rewards obtained during an episode. The backpropagation, in charge of the update of the weights and biases of the NN, is performed at the end of an episode. In this case, the network is trained using an off-policy model-free Reinforcement Learning algorithm called Soft Actor Critic (SAC) [37]. SAC is part of maximum entropy Reinforcement Learning methods, which consider an entropy-augmented objective function

$$J(\pi) = \mathbb{E}_\pi \left[\sum_l r(\mathbf{s}_l, \mathbf{a}_l) - \xi \log(\pi(\mathbf{a}_l|\mathbf{s}_l)) \right], \tag{9}$$

with ξ the non-negative entropy parameter. In such framework, the objective $J(\pi)$ can be viewed as a way to insure the trade-off between exploitation of proven actions, through return maximization (first term of Eq. 9), and exploration of new ones, through entropy maximization (second term of Eq. 9) [37]. SAC shows great sample efficiency, meaning that it makes the most of each learning episode so as to reduce the number of episodes needed to learn a policy. It also shows little sensitivity to hyperparameters, which avoids massive parameters tuning. The aforementioned characteristics make SAC one of the most efficient algorithms available these days [37]. We use here the open-source SAC implementation of Stable Baselines [38], a fork of Open AI baselines. The policy NN is a fully-connected multilayer perceptron built from TensorFlow [39].

2.2.2 | IPC problem definition in the RL formalism

In the present context, the action is defined as $\mathbf{a}_l = [A, P]$, namely the amplitude and the phase shift of the oscillators. It corresponds to $\mathbf{u}$ in the controller framework from Fig. 2. The output layer of the NN thus comprises two neurons. Regarding the state definition, the key information to be provided consists in the azimuthally oscillating load $\epsilon_s(l)$. However, it does not unequivocally define the state and it should therefore be supplemented with the effective amplitude a and phase p of the pitch oscillations, as well as the rotation speed ω of the turbine and the collective pitch angle command β_{CPC}. Eventually, the learning state is defined as $\mathbf{s} = [\epsilon_1, ..., \epsilon_{n_S}, \omega, \beta_{CPC}, a, p]$[1] and corresponds to the input layer of the neural network, i.e. to the vector $\mathbf{x}$ when referring to Fig. 1. The NN input layer is thus made of $n_S + 4$ neurons. Three hidden layers, comprising 128 neurons each, are present between the input and output layers. All neurons are perceptrons relying on an activation function $y = f_a(s)$ applied to the weighted sum s of their inputs and bias to generate their output y. The activation function is the Rectified Linear Unit or ReLU for short ($y = \max(0, s)$) for the input and inner layers and hyperbolic tangent ($y = \tanh(s)$) for the output layer to enable negative outputs of the NN.

The reward associated to each action is computed as

$$r(\mathbf{s}_l, \mathbf{a}_l) = k_1 \exp\left(-k_2 \frac{1}{n_S} \sum_{s=1}^{n_S} |\epsilon_{s,l}| \right), \tag{10}$$

where k_1 [-] and k_2 [Nm]$^{-1}$ are two coefficients. We here consider the 1-norm of $\epsilon_{s,l}$, yet higher order norms could be envisioned to further penalize high loads imbalances. During the training process, an action is taken for every rotation of the turbine, i.e. the learning step index l corresponds to the instant $t = l\,T_{rot}$, while a learning episode corresponds to 20 steps or rotations. It is important to highlight that two control time scales are thus at stake when learning. On the one hand, the variable-speed variable-pitch commands (β_{CPC} and the generator torque Q_{gen}) and the internal variables a and p of the oscillators are updated every control time step k having a duration $\Delta t = \mathcal{O}(T_{rot}/100)$. On the other hand, the target amplitude A and phase P, which are the inputs to second-order systems describing the evolution of a and p, are updated every learning

[1] In the previous investigations by the authors [31], the state vector did not include the sector-effective load oscillation ϵ, but rather a sector-effective infinite upstream velocity. The latter was estimated from the blade loads in the fashion of [40]. The direct use of blade loads at a sector level was expected to reduce the uncertainties and delays associated to estimations.

step l of duration T_{rot}. Indeed, during the training process, these actions should be maintained during a rotation period to enable the observation of their effect on the loads and compute a meaningful reward.

The natural angular frequency ω_n and the damping ratio ζ of the second-order systems ruling the evolution of a and p (Eq. 8) are determined from the desired time response of these systems. The picked actions A and P are seen, from a control point of view, as step function input. For stability concerns, we desire to strictly avoid oscillations in the step response and therefore opt for critically-damped systems with $\zeta = 1$. The value of ω_n is chosen so as to obtain a specific settling time t_s (time after which the system has reached 99.9% of the step value) $\mathcal{O}(T_{rot}/4)$, i.e. the step value should be reached within a quarter of rotation. We recall that the step response of a critically-damped system is given by $y(t) = Y(1-\exp(-\omega_n t)(1+\omega_n t))$, where y stands for a or p and Y stands for A or P [41]. Solving the step response equation with $y(t_s) = 0.999Y$ leads to $\omega_n t_s = 9.2$. The natural angular frequency is thus chosen as $\omega_n = \frac{4\times 9.2}{T_{rot}} = \frac{4\times 9.2}{2\pi}\omega_{rot}$, where ω_{rot} is the nominal turbine rotation speed.

2.2.3 | Learning environment

As RL consists in learning by interacting with the environment, it is necessary to provide the environment that will react to the actions tried during the learning process. Training a model requires numerous learning episodes, so simple models of both the flow and the turbine are needed for the sake of computational affordability. First, the flow generator is low-fidelity and turbulent effects are not considered. The key wind characteristic that generates fatigue on the blades is the shear of the atmospheric boundary layer. The latter is modelled with the following exponential law

$$\frac{U(y)}{U_{hub}} = \left(\frac{y}{H_{hub}}\right)^{\alpha}, \tag{11}$$

where y is the vertical elevation from the ground, H_{hub} and U_{hub} are the hub height and velocity respectively and α is the shear coefficient. A reference velocity is randomly chosen in the working range of the wind turbine every 500 steps and U_{hub} periodically oscillates around that reference velocity. The same process is applied to the shear coefficient α. In order to reproduce the wake impingement effect, Porte-Agel's model for the wake deficit [42] is used. The wake deficit is superimposed onto the shear law and its intensity varies with U_{hub}. The position of the turbine from which the wake originates is randomly chosen every 500 steps. The environment is made more dynamic by taking into account pseudo-meandering of the wake deficit through a horizontal sine motion of the wake center (z_{WCL}). The amplitude and frequency of that pseudo-meandering are based on the statistical values obtained from the stochastic motion of real meandering [43]. The synthetic flow generated thus consists in a sheared flow, with varying shear intensity and mean velocity, supplemented by a periodically meandering wake whose origin can vary. As far as the turbine is concerned, we make use of a modified Blade Element Momentum theory (BEM) that accounts for the effects of shear and individual pitch angles [44]. This basic model of the physics enables an efficient learning by the system with reasonable computational and data storage costs.

2.2.4 | Policy evaluation using damage equivalent loads

Policy evaluation or analysis of the cumulated reward is commonly used to monitor the learning evolution. In addition to it, we also used a more physical criterion, easing the performance comparison with other control schemes. The resulting load alleviation capacities of the trained network is quantified in terms of fatigue reduction. A quantitative fatigue indicator is the damage equivalent load (DEL) [45] and is built as follows. The damage D perceived by a component submitted to a varying moment $M(t)$ over a finite time period T is given by Palmgren-Miner's rule

$$D = \sum_{i=0}^{n} \frac{n_{c,i}}{N_{F,i}}, \tag{12}$$

where n is the number of range stress classes, $n_{c,i}$ is the number of cycles for each range stress class according to the rainflow algorithm and $N_{F,i}$ is the number of cycles to failure for each range stress class. The damage equivalent load is the sinusoidal load $M_{a,eq}$ at frequency f_{eq} and thus showing $n_{c,eq} = f_{eq}\,T$ cycles over the period T whose damage $D = \frac{n_{c,eq}}{N_{F,eq}}$ would be equivalent to that of the signal of interest. Using the S-N relation (also known as the Wohler's curve) for the particular material and assuming that the moment is proportional to the stress, we obtain that $N_{F,i} = kM_{a,i}^{-m}$ and $N_{F,eq} = kM_{a,eq}^{-m}$, with k and m being material properties. Eventually, the DEL is expressed as

$$DEL_M = M_{a,eq} = \left(\sum_{i=0}^{n} \frac{n_{c,i} M_{a,i}^{m}}{n_{c,eq}}\right)^{1/m}, \tag{13}$$

where $N_{F,eq}$ is the number of equivalent cycles. We choose it such that the equivalent loading frequency is the turbine rotation frequency. The m exponent depends on the material and thus on the studied component.

During the learning process, intermediate validation helps evaluate the neural network and verify whether its performances start degrading at some point, i.e. if it starts overfitting. The policy is therefore evaluated every 500 learning steps on 10 reference inflow cases using the same BEM-coupled synthetic flow generator as the one used for training. We eventually keep the intermediate policy that offers the best results in terms of blade flapwise DEL. This process of permanent evaluation and best policy selection is a regularization method called early stopping [46].

2.2.5 | Real conditions testing

Once the best policy NN is selected thanks to early stopping, the biomimetic IPC framework is deployed within Large Eddy Simulations (LES). Indeed, properly assessing the controller performances requires a higher fidelity representation of the flow physics and the loads generation. The LES are performed by means of a Vortex Particle-Mesh method (VPM)[47] in which the blades are modelled by immersed lifting lines[48], coupled to the multi-body-system solver ROBOTRAN in charge of the dynamics of the turbine[49]. Turbulence is injected at the inflow using Mann boxes[50]. The analytical exponential shear law from Eq. 11 is translated into vorticity so as to account for wind shear and comply with the vorticity formulation of the Navier-Stokes equations.

The RL-IPC control time scales are slightly adapted when the training is over, as all control-related variables (β_{CPC}, Q_{gen}, A, P, a and p) are updated every control time step k of duration $\Delta t = \mathcal{O}(T_{rot}/100)$. It is indeed not relevant any more, when deploying the controller, to update the actions only once per rotation as during the training process, as it would lead to a lack of real time response to the flow sensing. This means that, every time a blade leaves a sector the updated sector-effective load oscillations are utilized for controller action.

2.3 | Reference controllers for comparison

Two reference controllers will be used for comparison: one without IPC and one with a standard IPC scheme. The reference controller without IPC, which will further be referred to as CPC, for Collective Pitch Controller, is a classical implementation of a variable-speed, variable-pitch controller[51]. It relies on a generator-torque controller, maximizing the power captured below the rated wind speed, and a collective blade pitch controller, maintaining nominal power production above the rated wind speed.

The chosen reference IPC controller is based on the Coleman transform and will further be referred to as CT-IPC. It applies to three-bladed wind turbines and consists in projecting the blade loads on the fixed coordinate frame of the rotor disc as follows

$$\begin{bmatrix} M_{\text{tilt},k} \\ M_{\text{yaw},k} \end{bmatrix} = \begin{bmatrix} \frac{2}{3}\cos\theta_k & \frac{2}{3}\cos\left(\theta_k - \frac{2\pi}{3}\right) & \frac{2}{3}\cos\left(\theta_k + \frac{2\pi}{3}\right) \\ \frac{2}{3}\sin\theta_k & \frac{2}{3}\sin\left(\theta_k - \frac{2\pi}{3}\right) & \frac{2}{3}\sin\left(\theta_k + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} MF_{n_1,k} \\ MF_{n_2,k} \\ MF_{n_3,k} \end{bmatrix}. \tag{14}$$

Given the azimuthal position θ of the first blade, defined from the vertical upward direction, the flapwise bending moments are mapped to a tilting moment M_{tilt} and a yawing moment M_{yaw}. As commonly done in the literature[4,52], the 3P component of these moments is eliminated thanks to a notch filter. The control state, as defined in Fig. 1, writes $\mathbf{x} = \left[M^f_{\text{tilt}}, M^f_{\text{yaw}}\right]$. A proportional integral controller is chosen to bring the tilt and yaw moment to zero ($\mathbf{x_{ref}} = [0, 0]$)[4]. The control-space command vector consists in the tilt and yaw pitch angles $\mathbf{u} = [\beta_{\text{tilt}}, \beta_{\text{yaw}}]$. The command vector is projected back on the blades rotating coordinate frame by means of the inverse Coleman transform

$$\begin{bmatrix} \Delta\beta_{1,k} \\ \Delta\beta_{2,k} \\ \Delta\beta_{3,k} \end{bmatrix} = \begin{bmatrix} \cos\theta_k & \sin\theta_k \\ \cos\left(\theta_k - \frac{2\pi}{3}\right) & \sin\left(\theta_k - \frac{2\pi}{3}\right) \\ \cos\left(\theta_k + \frac{2\pi}{3}\right) & \sin\left(\theta_k + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} \beta_{\text{tilt},k} \\ \beta_{\text{yaw},k} \end{bmatrix}, \tag{15}$$

thus giving the expression of the differential individual pitch angles $\Delta\beta_{1,2,3}$ to be added to the collective pitch angle β_{CPC}[4] to obtain the individual pitch angles $\beta_{1,2,3}$.

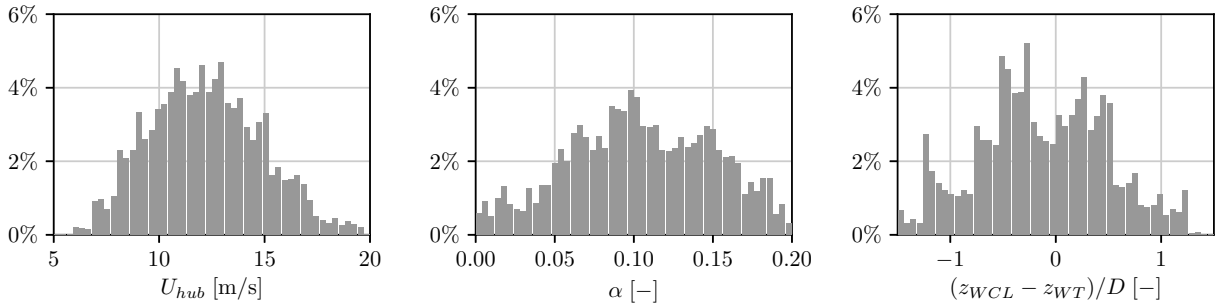
3 | RESULTS

This section first presents the training results in terms of environment sampling and learning convergence. The performances of RL-IPC are then verified for a single wind turbine in turbulent conditions using LES. Eventually, RL-IPC is tested in more complex cases resulting from a pair of in-line wind turbines. The simulated turbine is the NREL-5MW reference wind turbine[51]. The response of the pitch actuator from demanded to achieved pitch angular position is decisive in terms of load alleviation. The pitch actuator is therefore modelled as a second-order transfer function as in Bossanyi[4] with a natural frequency of 2 Hz and a damping factor of 0.8.

3.1 | Training

3.1.1 | Environment sampling and exploration

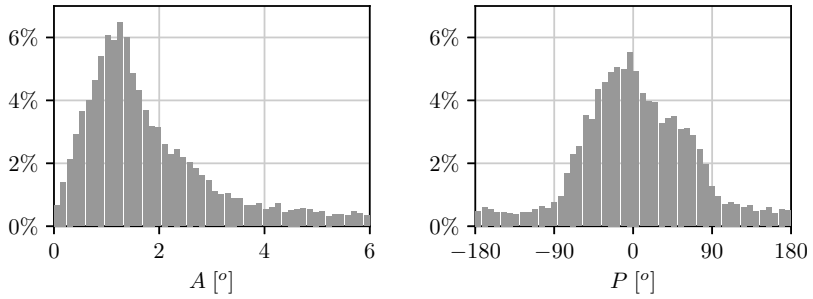
The low-fidelity environment used for training relies on varying three parameters that define the waked flow impinging on a downstream wind turbine: the velocity at hub height, the shear coefficient and the lateral distance between the wake center line, z_{WCL}, and the downstream wind turbine center, z_{WT}. The bounds of the accepted values are presented in Fig. 3, along with the percentage of occurrence of the values during the learning process. It shows that the flow generator achieves a wide sweeping of the environment.



(a) Velocity at hub height (b) Shear coefficient (c) Wake center line (WCL) position

FIGURE 3 Sampling of the low-fidelity environment during training: percentage of occurrence of each flow parameter.

Fig. 4 displays the distribution of the actions taken during the training process, those show a good exploration-exploitation trade-off. Indeed, as the shear is the dominant effect on the rotor load imbalance, one could expect to have the phase of the pitch oscillations centered on zero, which means maximum pitch at the uppermost blade position and minimum pitch at the lowest one. It is positive to see that the agent mostly picks phases around zero but still explores other possibilities. Also, the amplitude of the pitch oscillations are mainly chosen between 0^o and 2^o, yet higher amplitudes are explored.

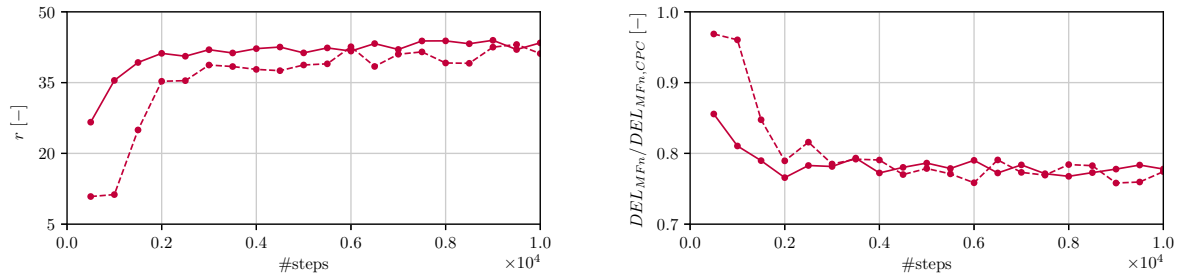


(a) Pitch oscillations amplitude (b) Pitch oscillations phase

FIGURE 4 Exploration of the action space: percentage of time an action was taken.

3.1.2 | Learning convergence

In order to assess the learning convergence, 20 instances of the learning process are launched using these hyperparameters. For each of them, the neural network performances are evaluated on 10 reference inflow cases while the learning process is running, as mentioned in section 2.2. More precisely, the NN is evaluated every 500 learning steps, through the mean reward (r) received during these 500 steps and the damage equivalent flapwise bending moment (DEL_{MFn}) associated to that time window. The mean reward is an intrinsic learning quantity, while the damage is a post-processed quantity that is physically relevant. At every evaluation, both quantities are averaged over the 10 inflows and over the 20 learning instances. The results are reported in Fig. 5 with and without the use of layer normalization. Layer normalization is a technique inherited from batch normalization, which is known to avoid overfitting, increase generalization and also reduce the training time[34]. Fig. 5 indeed shows that using layer normalization accelerates the reward convergence but also helps converging to a slightly higher value. This impact is not as obvious in terms of the post-process DEL_{MFn}. From all the intermediate policies that have been evaluated, we eventually keep the layer-normalized one that presents the lowest DEL_{MFn}, in the fashion of early stopping[46], and deploy it in the high-fidelity simulations.



(a) Mean step reward r

(b) Mean DEL_{MFn}

FIGURE 5 Mean evolution of the step reward r and of the DEL_{MFn} normalized by the CPC DEL_{MFn} over the course of learning, with (—●—) and without (- -●- -) layer normalization.

3.2 | Calibration and validation of the reference IPC controller

For the sake of reproducibility, we here document the calibration of the implemented Coleman transform-based individual pitch controller and its validation against literature results. The notch filter for the tilting and yawing moments relies on the same expression as Eq. 3, with $\gamma = 0.6$. The controller gains are tuned so as to maintain reasonable pitch demands and to match load reductions reported in the literature, leading to $K_P = 6 \times 10^{-9}$ rad/Nm and $K_I = 1 \times 10^{-9}$ rad/(Nm s). The controller as such yielded a reduction of 19% of flapwise DEL for the NREL 5MW at 15 m/s under a turbulence intensity level of 19%. Chen[53] reported 16% fatigue load reduction in the same conditions and for the same turbine, while Bossanyi[52] reported 15% flapwise DEL reduction for a 2 MW wind turbine operated at 12 m/s (above-rated wind speed) under a turbulence intensity level of 19% too. Our results are, if a bit optimistic, close to the performances reported in the literature, the implemented controller can thus reasonably be used as reference IPC controller.

3.3 | Testing with LES of a single wind turbine

The purpose here is to investigate the ability of RL-IPC to exploit the knowledge acquired in the low-fidelity non turbulent environment to realistic wind conditions. The LES set-up is shown in Fig. 6. As IPC is typically used in region 3 of control, i.e. at above-rated wind speeds[54], we consider two mean wind speeds: one in region 3 ($U_{hub} = 15$ m/s) and the other one at the end of region 2 ($U_{hub} = 10$ m/s). Indeed, the wind speed can temporarily exceed the rated one due to turbulence and the turbine is therefore operating in region 3. For the sake of investigation, we maintain IPC active even in region 2, though this is subject to discussion in the literature[55].

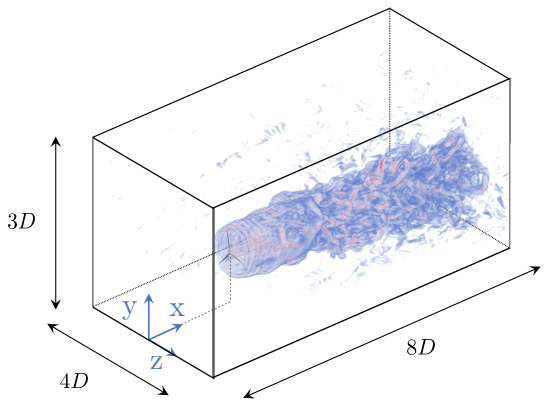
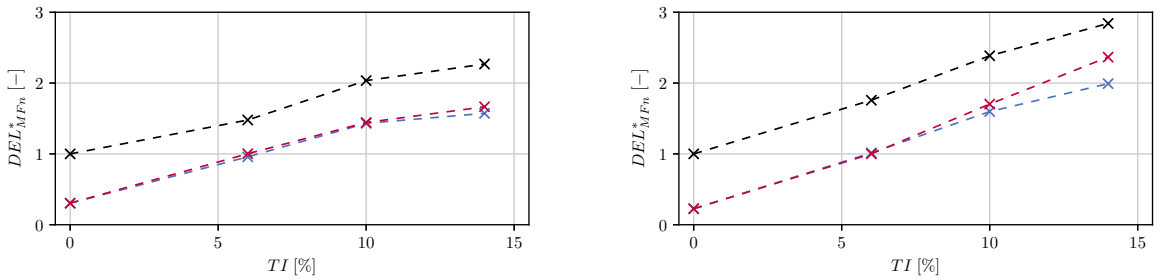


FIGURE 6 Numerical set-up: $8\mathrm{D} \times 3\mathrm{D} \times 4\mathrm{D}$ domain, with the turbine located $2\mathrm{D}$ downstream the inlet at hub height $H_{hub} = 90$ m. The mesh is isotropic with a mesh size h such that $D/h = 32$, yielding $256 \times 96 \times 128 = 3.1 \times 10^6$ points. Boundary conditions are inflow-outflow in the streamwise direction, no-through flow in the vertical direction and periodic in the spanwise direction.

We quantify the performances of RL-IPC under four turbulence intensity levels (TI): 0%, 6%, 10%, 14%. The shear exponent α defined in Eq. 11 is set to 0.2 as recommended by the IEC standards [56]. As it is common for fatigue computations, statistics are computed over 10-minute flow samples (equivalent to $48D/U$ for $U_{hub} = 10$ m/s and to $72D/U$ for $U_{hub} = 15$ m/s), which corresponds to the period of the Mann boxes used for the injection of turbulence.

3.3.1 | Fatigue

Figure 7 shows the flapwise DEL at the two considered wind speeds for the different TIs. Comparison is made between CPC, CT-IPC and RL-IPC. The coefficient m used for the computation of DEL in Eq. 13 is set to 10 as blades are supposed to be made of glass fiber [51,4]. First, comparing RL-IPC and CT-IPC shows that they yield similar load reductions at low and moderate TIs. This means that the RL-IPC formulation is robust enough for the NN to be able to generalize what it has learned on sheared inflows to cases with turbulence on top of it. While the generalization holds for high TI (i.e. 14%), it is not good enough to compete with CT-IPC. Second, the fatigue loads evolution with respect to the turbulence intensity and the impact that IPC can have, whether RL or CT, calls for a discussion. The DEL increases almost linearly with the TI and IPC cuts the same absolute part of the DEL, whatever the TI (about 0.8 MNm for $U_{hub} = 10$ m/s and 1 MNm for $U_{hub} = 15$ m/s). This is in agreement with the results presented in Bergami [57], where the load variations are split into deterministic and stochastic contributions. The deterministic part of the DEL, mainly generated by the atmospheric shear, is the one that is efficiently tackled by feedback controllers like CT-IPC and RL-IPC. When the TI increases, the stochastic part of the load variations, and hence of the DEL, increases, yet both IPCs have a hard time dealing with these additional fatigue loads. Several contributions in the literature have addressed the challenge of alleviating non-deterministic loads, like Kallen et al. [21] and Bottasso et al. [58], by explicitly designing the controller with two layers: one in charge of the periodical loads and the other focusing on the stochastic ones typically resulting from turbulence. Though the RL-IPC controller does not have such a two-layer architecture, it can detect gusts through the sensing module and directly react to them by adapting the pitching amplitude and phase of the blades. The decreasing performances at high turbulence levels show the limits of this ability to adapt quickly to stochastic structures. We identify three possible causes for this: (1) the sensing module does not react fast enough, (2) the NN was trained without turbulence and can only extend to moderate TI levels, (3) the action space is too constrained as it is limited to one common amplitude and phase for all three blades.



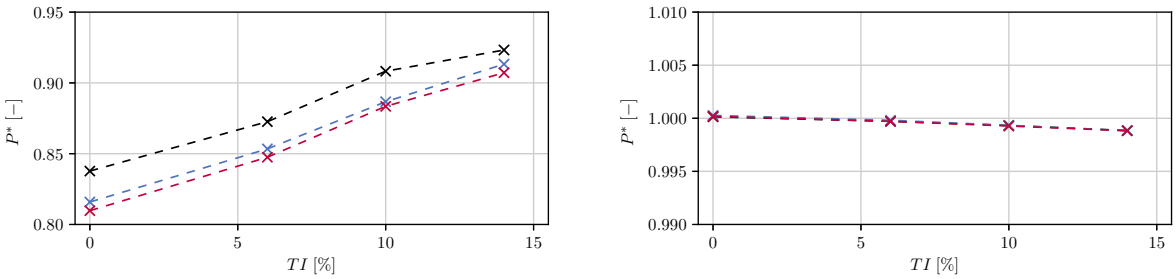
(a) $U_{hub} = 10$ m/s (b) $U_{hub} = 15$ m/s

FIGURE 7 Flapwise DEL normalized by the CPC DEL_{MFn} at $TI = 0\%$: CPC (– ×–), CT-IPC (– ×–), RL-IPC (– ×–).

3.3.2 | Power production

The impact of IPC on power production is presented in Fig. 8. At under-rated wind speed (here $U_{hub} = 10$ m/s, Fig. 8(a)), using IPC is slightly penalizing in terms of power production, with a few percent loss. At above-rated wind speed (here $U_{hub} = 15$ m/s, Fig. 8(b)), IPC has no significant effect on the power production. Indeed, IPC changes the operating conditions of the airfoils to reduce the flapwise loads oscillations. This implies a much reduced loading in the top part of the rotor and a highly increased one on in its bottom part. This does not necessarily translate in proportional reduction for edgewise moments. Hence, the reduction of edgewise bending moment at the top is not totally compensated by its increase at the bottom. In the above-rated region, the controller compensates this small loss of aerodynamic performance by slightly reducing the collective pitch component (see Fig. 9(b)). This can unfortunately not be done at under-rated wind speeds as, when the turbine is operated at CPC, the collective pitch is already at its minimal value of zero. This explains the small power losses in the under-rated region, which were also observed in [54], both numerically and experimentally. Also, at under-rated wind speeds, the power production of the turbine somewhat increases with the TI, getting

closer to the nominal power production. The detrimental effect of IPC on power production is therefore lower when TI increases as shown in Fig. 8(a). RL-IPC degrades the power production by about 1% more than CT-IPC does.



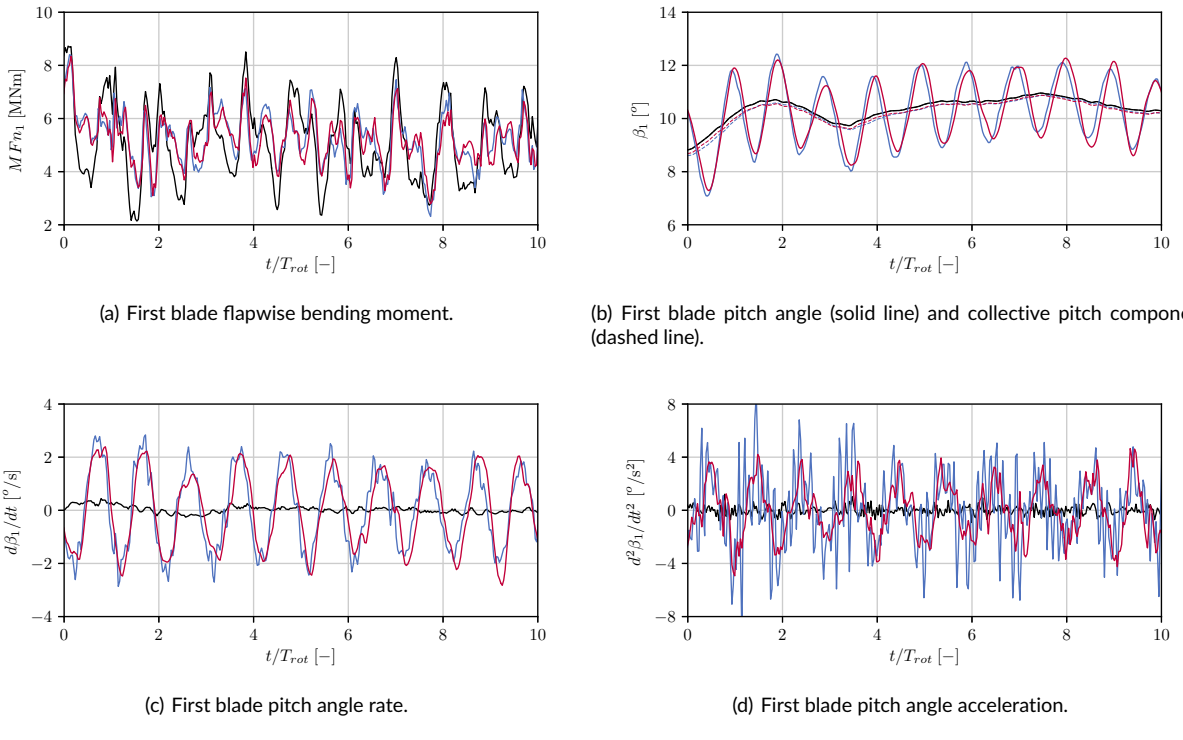
(a) $U_{hub} = 10$ m/s

(b) $U_{hub} = 15$ m/s

FIGURE 8 Power production normalized by the wind turbine rated power: CT-IPC (- ×-), RL-IPC (- ×-).

3.3.3 | Blade loads and pitch signals

Further analysis on the loads is performed for the case at $U_{hub} = 15$ m/s and $\mathrm{TI} = 10\%$. First, the temporal evolution of the flapwise bending moment on the first blade (MFn_1) as well as its pitch angle position (β_1), rate ($d\beta_1/dt$) and acceleration ($d^2\beta_1/dt^2$) are shown in Fig. 9 for the three controllers. While RL-IPC and CT-IPC produce pitch oscillations with similar phases and amplitudes, they result in quite different rates and accelerations. This shows that, as expected, the RL-IPC framework intrinsically leads to reduced pitching rates and accelerations. This is not negligible as it is known among IPC researchers that a too demanding pitch activity may lead to an acceleration in wear and tear of the pitch actuators.



(a) First blade flapwise bending moment.

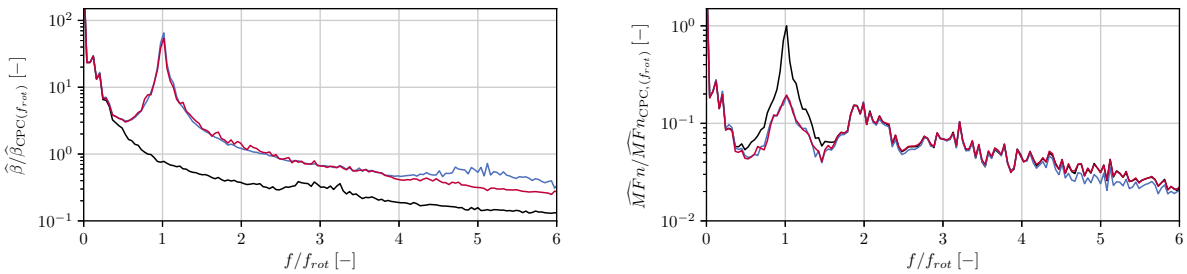
(b) First blade pitch angle (solid line) and collective pitch component (dashed line).

(c) First blade pitch angle rate.

(d) First blade pitch angle acceleration.

FIGURE 9 Temporal evolutions of loads and pitch angle value, rate and acceleration for the case $U_{hub} = 15$ m/s and $\mathrm{TI} = 10\%$: CPC (——), CT-IPC (——), RL-IPC (——). Time is made dimensionless with respect to the turbine rotation period T_{rot}.

Signals are also studied from the frequency domain perspective in Fig. 10. It is interesting to notice that the frequency content of the loads reflects the pitch angles frequency content. IPC mainly acts on the 1P frequency and the loads spectrum is close to unchanged from the CPC case after 1.5P.



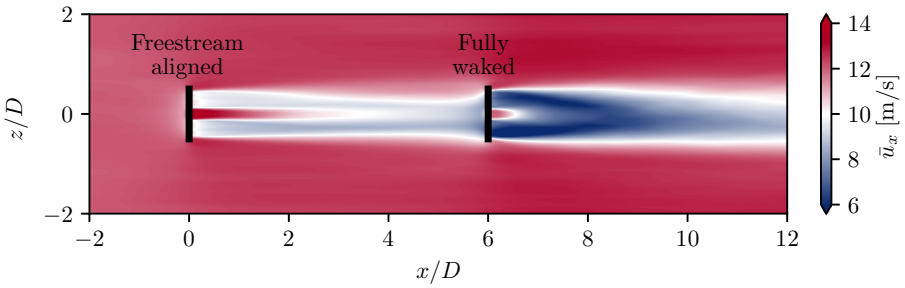
(a) Pitch angle.

(b) Flapwise bending moment.

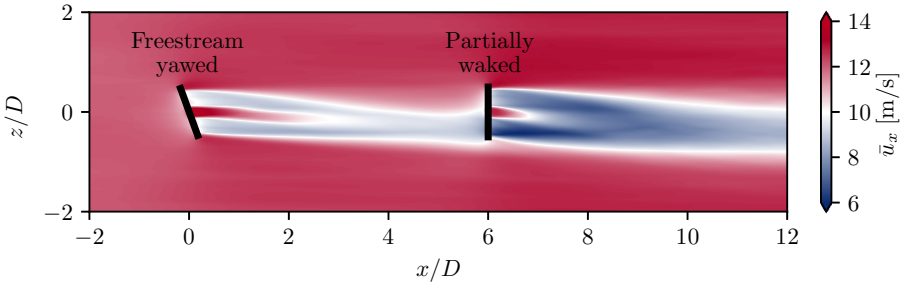
FIGURE 10 Frequency analysis of pitch and loads for the case $U_{hub} = 15$ m/s and $\mathrm{TI} = 10\%$: CPC (———), CT-IPC (———), RL-IPC (———). Signals are averaged over the three blades and made dimensionless with the amplitude of the FFT for the CPC case at the rotation frequency f_{rot}.

3.4 | Testing with LES of a pair of in-line wind turbines

Let us now assess RL-IPC in waked conditions. To this end, we consider a pair of in-line wind turbines for which the wind conditions are chosen as follows. First, the shear coefficient is high enough to generate large 1P oscillations on the turbine blades and the turbine operates at above-rated wind speed. Second, we wish to have a significant impact of the upstream turbine wake on the downstream turbine.



(a) First configuration: both turbines are aligned with the flow.



(b) Second configuration: the upstream turbine is misaligned as to steer its wake from the downstream one.

FIGURE 11 Time-averaged streamwise velocity field for the two configurations of a pair of in-line turbines. The domain is $14\mathrm{D} \times 3\mathrm{D} \times 4\mathrm{D}$, with the turbines located $2\mathrm{D}$ and $8\mathrm{D}$ away from the inlet respectively. The resolution is such that $D/h = 32$, leading to a total of 5.5×10^6 particles.

This implies (1) a strong wake deficit and hence the high thrust coefficient encountered at the lower wind speeds of region 3, and (2) limited wake recovery met in moderate TI, leading to small wake meandering and remixing. We therefore set $U_{hub} = 12$ m/s, TI = 6% and $\alpha = 0.2$, these conditions are close to the IEC standards for offshore wind turbines [56].

We study two configurations of a pair of turbines spaced by $6D$ in the streamwise direction and aligned in the transverse direction. In the first configuration, the two turbines are aligned with the wind direction. In the second one, the upstream turbine is statically yawed, with a yaw angle of 20^o, in order to steer its wake from the downstream turbine for power maximization. Fig. 11 shows the temporally-averaged streamwise velocity field of the two configurations; those give rise to four turbine-specific situations: (1) the reference case of a first-row aligned turbine, (2) a first-row yawed wind turbine, (3) a totally waked wind turbine subjected to the meandering phenomenon and (4) a wind turbine facing a partially-impinging and meandering wake. Henceforth, these four situations will respectively be referred to as freestream aligned, freestream yawed, fully waked and partially waked, as in Fig. 11. Here again, statistics are computed over 10-minute flow samples corresponding to the Mann flow-through time. The same pitching strategy is always applied to the two in-line turbines, whether it is CPC, CT-IPC or RL-IPC.

3.4.1 | Power production

The power production of the CPC-controlled turbines exhibits an expected behaviour (see Fig. 12). Both upstream turbines produce the nominal power as the velocity component normal to the rotor is superior to the nominal wind speed whether the turbine is yawed or not. In both configurations, the downstream turbine produces less power, as it undergoes the wake deficit of its upstream counter part. However, the power is less reduced when the upstream turbine is yawed, as only part of the wake impinges on the rotor. In terms of power losses due to IPC, the same remarks as in the previous section can be made: no impact for the upstream turbines as they operate at above-rated wind speed and slight reduction for the downstream ones as they operate at under-rated wind speeds due to the wake impingement.

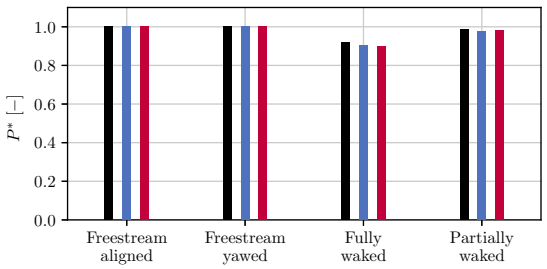


FIGURE 12 Power production normalized by the wind turbine rated power, for the cases $U_{hub} = 12$ m/s and $TI = 6\%$ for the four turbines: CPC (■), CT-IPC (■), RL-IPC (■).

3.4.2 | Blade loads

The flapwise DELs are reported in Fig.13. The freestream aligned turbine follows the same trend as the results presented for the isolated turbines. The freestream yawed turbine yields two comments. On the one hand, even when CPC only is applied, the chosen yaw misalignment reduces the DEL. This result is also obtained in [54]: it depends on the combination of the direction of yaw rotation and the direction of blade rotation and finds an explanation in velocity triangles. Indeed, the vertical gradient of velocity due to shear is partially compensated by the upstream velocity component parallel to the yawed rotor, resulting is smaller oscillations of the angle of attack, and thus of the loads, over one rotation. Yawing the turbine in the other direction would have led to a DEL increase. On the other hand, RL-IPC demonstrates its ability to alleviate fatigue loads on yawed turbines, even though not including yaw considerations in its formulation nor having learnt on similar cases. It then shows that the RL-IPC formulation can natively deal with yaw, reducing loads as CT-IPC would do [54]. Regarding the two waked turbines, two comments arise as well. First, the DEL are increased compared to the freestream turbines due to the meandering wake and even further in the partially waked case as the left-right rotor load imbalance is higher. Second, in both the fully and the partially waked cases, the two IPCs do reduce the fatigue loads but to a lesser extent than for the freestream machines. This is most likely due to the complexity of the flows impinging on the rotor.

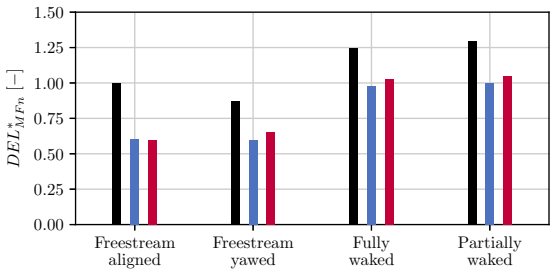
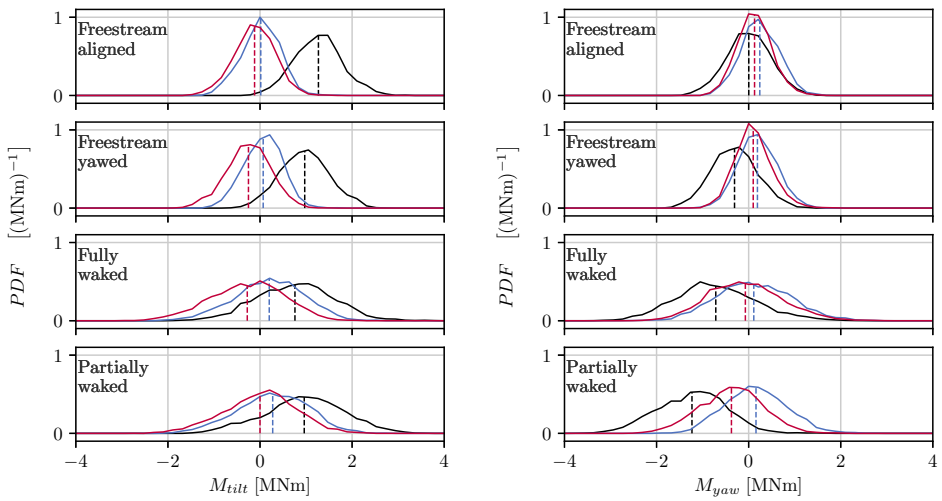


FIGURE 13 DEL_{MFn}, made dimensionless by the DEL_{MFn} of the freestream aligned turbine controlled with CPC, for the cases $U_{hub} = 12$ m/s and $TI = 6\%$ for the four turbines: CPC ( ), CT-IPC ( ), RL-IPC ( ).

3.4.3 | Rotor loads

Rotor loads, namely tilt and yaw moments M_{tilt} and M_{yaw} as defined by the Coleman transform in Eq. 14, can also be analysed. Indeed, CT-IPC is designed to bring their filtered values to zero, which results in reduced blade loads oscillations. The purpose here is to show the effective action of CT-IPC on these rotor loads, but also to see how RL-IPC acts on them, as they are never explicitly handled in this novel IPC formulation. The probability density functions (PDF) of these loads are depicted in the Fig. 14 and all present a Gaussian shape. This comes from the simulation set-up: we impose a mean velocity and Mann boxes generate perturbations around it to account for turbulence. The resulting loads therefore oscillate around the load corresponding to the mean upstream velocity U_{hub}.



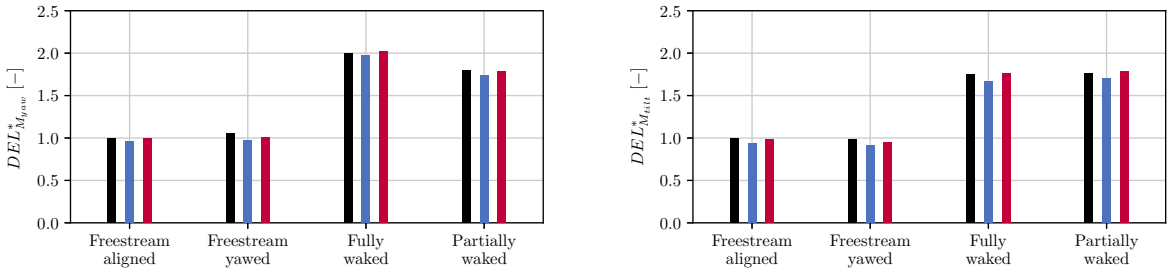
(a) Tilting moment. (b) Yawing moment.

FIGURE 14 Probability density function (PDF) of rotor loads for the four turbines at $U_{hub} = 12$ m/s and $TI = 6\%$: CPC ( ), CT-IPC ( ), RL-IPC ( ), dashed lines denote the mean value.

Let us first focus on the CPC cases to better understand the physics at stake. When it comes to the freestream turbines (aligned and yawed), the turbulent gusts or lulls impinging on the rotor temporarily create left-right and up-down imbalances. The latter are superimposed to the shear-generated up-down imbalance, leading to a positive mean tilting moment. For the aligned turbine, the mean left-right imbalance and thus the mean yawing moment are close to zero, while it is slightly negative for the yawed turbine due to its misalignment. Regarding the waked turbines, PDFs behave like wider Gaussians. This is consistent with the high turbulence level of the wake and the wake meandering, both resulting in a higher loads variability. For the tilting moment in particular, note that the shear-generated positive mean value is still visible, while for the yawing moment, a negative mean value is observed.

Let us now consider the impact of both IPCs on the rotor loads. Two major trends stand out: both CT-IPC and RL-IPC tend (1) to bring the mean value of the rotor moments closer to zero and (2) to reduce the variance of these Gaussian-like PDFs. Only two exceptions to this trend are notable. First, the mean yawing moment of the freestream aligned turbine moves slightly away from zero when IPCs are used. Second, the variance reduction is more noticeable on freestream turbines than on waked ones. It is indeed more challenging for the IPC controllers to cancel the rotor loads at every instant due to the complexity and unsteadiness of the impinging wakes.

To provide a quantitative fatigue indicator, DELs are also computed for rotor loads. As the rotor loads end up impacting the rotor shaft, which is made of steel [59], the value of m in Eq. 13 is set to 4 [60]. The DEL associated to each case is provided on Fig. 15. It shows that CT-IPC tends to slightly reduce the rotor loads fatigue, while RL-IPC is more neutral and does not have a real impact on it.



(a) Yawing DEL

(b) Tilting DEL

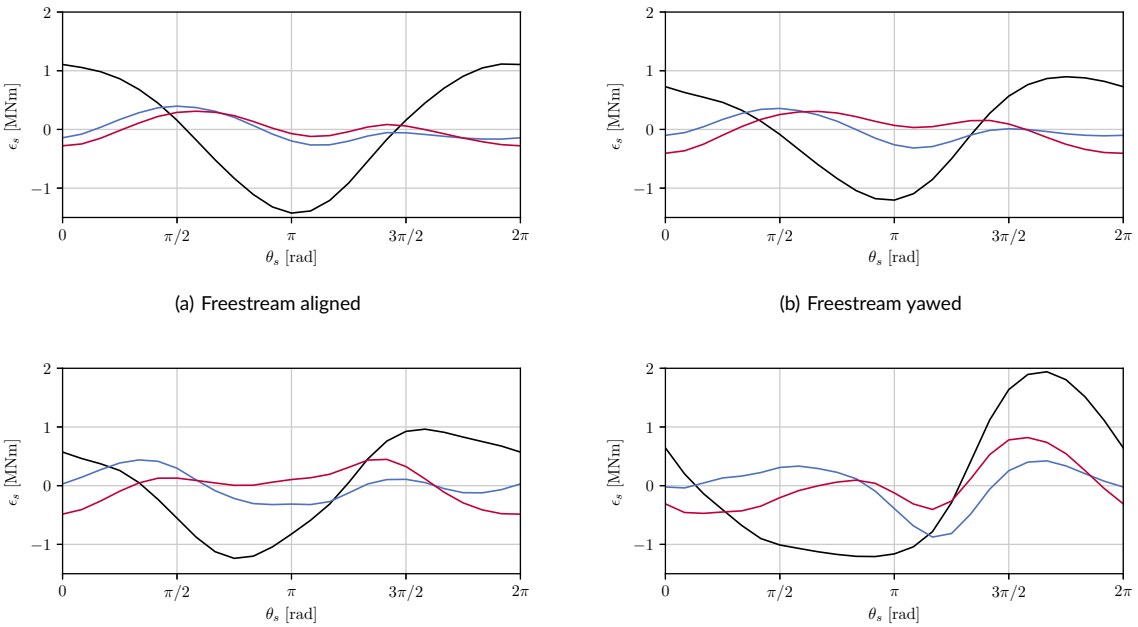
FIGURE 15 DEL, made dimensionless by the DEL of the freestream aligned turbine controlled with CPC, for the cases $U_{hub} = 12$ m/s and $TI = 6\%$ for the four turbines: CPC ( ), CT-IPC ( ), RL-IPC ( ).

3.4.4 | Azimuthal loads

Attention has been paid to the rotor loads, as they are the effective control variables of CT-IPC. Their RL-IPC counterpart is the sector-effective flapwise load oscillation defined in section 2.1.1. Its time-averaged value is presented in Fig. 16 for the four turbines and the three controllers.
We first notice the sine-like shape for the freestream aligned turbine with CPC (Fig. 16(a)) and the asymmetrical flattening impact of both IPCs on that curve. It was mentioned earlier that yawing the machine with a positively defined yaw angle helps reduce fatigue loads. Indeed, it can be seen in Fig. 16(b) that not only does the yaw deform the sine-like shape, but it also reduces the amplitude of oscillation. The negative mean yawing moment of the fully waked turbine observed in the CPC case finds an explanation in Fig. 16(c), with a little lateral shift of the sine-like profile yielding a left-right imbalance (minimum around $3\pi/4$ and maximum around $3\pi/2$, against π and 2π for the freestream aligned turbine). Finally, Fig. 16(d) highlights the huge load imbalance appearing in the partially waked case when no IPC is active. The right side of the rotor ($\theta \in [0, 5\pi/4]$) is completely under-loaded due to the wake impingement, while the left side of the rotor ($\theta \in [5\pi/4, 2\pi]$) is over-loaded by the almost-free impinging flow. Indeed, the velocity in the wake region makes the turbine operate in region 2, where reducing the velocity reduces the thrust [51].
In all cases, the positive impact of both RL-IPC and CT-IPC is quite clear in terms of oscillations reduction. However, it is noticeable that some limits are present. Even in the simple case of the freestream aligned turbine, none of the IPCs manage to totally flatten the time-averaged sector-effective flapwise load oscillation. This is coherent with Fig. 7, where even when in a non turbulent flow, IPCs do not manage to take the DEL to zero, because a residual load oscillation exists.

3.5 | Computational costs and perspectives on more realistic wind data for training

This last section discusses the computational costs associated to the method. All computations are performed on SkyLake 2.3 GHz CPUs. We recall that the whole learning process consists in three main parts: first training the NN in the low-fidelity environment, second verifying the learning convergence, also in the low-fidelity environment, and third testing the RL-IPC controller in large eddy simulations. The first part comprises the training of 20 NN-policies over 10^4 learning steps, i.e. 10^4 turbine rotations. This represents about 15h of wind for each NN and requires a computational time of 20h on a single CPU. The overall training process thus requires 400 CPUh for the simulation of 15 days of wind. Second, the validation process assesses the learning converge of these 20 NN. The latter are sampled 20 times over the course of their training



(a) Freestream aligned

(b) Freestream yawed

(c) Fully waked

(d) Partially waked

FIGURE 16 Time-averaged sector-effective flapwise load oscillation for the cases $U_{hub} = 12$ m/s and $TI = 6\%$ for the four turbines: CPC ( ), CT-IPC ( ), RL-IPC ( ).

(i.e. every 500 learning steps) and for each of these samples, each NN is tested on 10 reference inflow cases lasting 10 min. For every of the 20 trained policies, 20×10×10 min = 30h of wind are thus simulated, with a computational time of 20h on a single CPU. The total validation process therefore requires 400 CPUh for the simulation of 30 days of wind. Third, the LES testing of the best NN-policy gathers 10 cases (8 for the single turbine part and 2 for the two-turbine part), each simulating 10 min of wind and run in parallel on 32 CPUs for 3 h. Hence, the total cost of the LES testing is 1000 CPUh for the simulation of 1h40 of wind.

This reflection on computational costs legitimates the choice of training the NN in a low-fidelity environment before being able to bring the knowledge to turbulent flows. Even though it would be interesting to train the NN in realistic turbulent flow fields, it is quite clear that the cost of the LES used in this work is prohibitive. However, multiple compromises can be envisioned for the training in a higher-fidelity environment such as (1) generating synthetic turbulence data bases, typically using the Mann algorithm [50] or atmospheric boundary layer precursor simulations [61], to train the NN in turbulent wind at affordable cost; (2) using dynamic wake models like the one proposed in Lejeune [62] to produce much more realistic meandering waked flow conditions, (3) using LES at coarser spatio-temporal resolutions, which implies using disk approaches to model the rotor and thus handling IPC within an actuator disk framework, as proposed in Moens [63], or (4) use the low-fidelity environment to first train the NN and then refine the training in a LES framework. The best option should be chosen based on some cost-benefit analysis, which leaves a variety of perspectives open for the present methodology.

4 | CONCLUSIONS

We present a novel individual pitch control architecture that is inspired by animal locomotion as it separates low-level control tasks from high-level ones. The pitch angles are generated by oscillators (low-level) that are modulated by a neural network (high-level). The former is trained in a bio-inspired trial-and-error manner with Reinforcement Learning based on the upstream flow conditions. The latter are sensed through a novel load transformation strategy which preserves more information about the unsteady loads and puts it in a usable form for more advanced pitch control schemes. The learning environment comprises a synthetic model of sheared and waked inflows and the blade element momentum theory for the loads computation. The best neural network obtained throughout the learning is then deployed within large eddy simulations of the NREL 5MW to assess its performances and compare them with the state-of-the-art Coleman transform-based IPC. Multiple cases are envisioned: freestream turbines, either aligned or yaw, with TIs going up to 14% but also fully waked and partially waked turbine.

The major result of this work is that the RL-IPC is capable of achieving significant load alleviation whatever the wind conditions. It competes with the CT-IPC performances and does so generating smooth pitching oscillations. A remarkable outcome concerns the formulation of the problem, which is robust enough for the NN to generalize the knowledge it has acquired on very simple synthetic flows. Indeed, while turbulence was never part of the training, load alleviation is effective up to TIs above 10% but also in the case of total and partial wake impingement. The formulation does also natively apply to misaligned turbines and no specific training needs to be done for that configuration.

Another fundamental outcome of this work is that the proposed method opens a lot of promising perspectives for improvement. So far, attention has been paid to the development of the framework and the load transformation, which provides more complete information than the Coleman transform. The efforts should now focus on two main directions: enlarging the action space and making the training environment more realistic. Discussing the action space first, we recall that it is currently of dimension two and that extending it seems necessary to fully exploit the rich information available on the loads. In that perspective, the proposed framework is very advantageous, because it makes it pretty straightforward to add states and learn their optimal values. One could think of adding harmonics of the 1P frequency or including shape functions to slightly deform the sine-like shape of the oscillators outputs. This might raise concern in terms of pitch activity, but again, thanks to the flexibility of the framework, penalization on the pitch activity could directly be included in the training process through the expression of the reward. With an improved load sensing and an enlarged action space, the missing element is a realistic training environment. Multiple possibilities towards that purpose were proposed before, such as using wake models [62], synthetic [50] or precursor-generated [61] turbulence or disk-based LES [63].

To conclude on a more general level, this work has demonstrated that a wind turbine pursuing a certain objective can learn how to behave in the simple flow it is subjected to and exploit that knowledge when evolving in more complex flows. Including an extended action and a more realistic training environment to the framework could lead RL-IPC to outperform conventional IPC, both in terms of loads and pitch activity, especially in the case of flows like the meandering wakes encountered in wind farms.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement No. 725627) and from the Université de Mons under the 50/50 PhD funding program. This research benefited from computational resources made available on the Tier-1 supercomputer of the Fédération Wallonie-Bruxelles, infrastructure funded by the Walloon Region under the grant agreement No. 1117545. Computational resources were also provided by the Consortium des Équipements de Calcul Intensif, funded by the Fonds de la Recherche Scientifique de Belgique under Grant No. 2.5020.11 and by the Walloon Region.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available through the UCLouvain Dataverse https://dataverse.uclouvain.be/dataverse/wakeopcoll.

References

1. Systems VW. Vestas V236-15.0MW. https://www.vestas.com/en/products/offshore/V236-15MW; 2021.

2. McKenna R, Ostman v.d. Leye P, Fichtner W. Key challenges and prospects for large wind turbines. *Renewable and Sustainable Energy Reviews* 2016; 53: 1212-1221.

3. Lu Q, Bowyer R, Jones BL. Analysis and design of Coleman transform-based individual pitch controllers for wind-turbine load reduction. *Wind Energy* 2015; 18(8): 1451–1468.

4. Bossanyi E. Individual blade pitch control for load reduction. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 2003; 6(2): 119–128.

5. Lio WH, Jones BL, Lu Q, Rossiter JA. Fundamental performance similarities between individual pitch control strategies for wind turbines. *International Journal of Control* 2017; 90(1): 37–52.

6. Coleman R, Feingold A. Theory of self-excited mechanical oscillations of helicopter rotors with hinged blades. Tech. Rep. 1351, National Advisory Committe for Aeronautics (NACA); 1957.

7. Caselitz P, Kleinkauf W, Krüger T, Petschenka J, Reichardt M, Störzel K. Reduction of fatigue loads on wind energy converters by advanced control methods. *EWEC-CONFERENCE* 1997: 555–558.

8. Van Engelen T, Van Der Hooft E. Individual pitch control inventory. *Technical Report ECN-C-03-138* 2005.

9. Zhang D, Cross P, Ma X, Li W. Improved control of individual blade pitch for wind turbines. *Sensors and Actuators A: Physical* 2013; 198: 8–14.

10. Duesterhoeft W, Schulz MW, Clarke E. Determination of instantaneous currents and voltages by means of alpha, beta, and zero components. *Transactions of the American Institute of Electrical Engineers* 1951; 70(2): 1248–1255.

11. Leithead W, Neilson V, Dominguez S. Alleviation of unbalanced rotor loads by single blade controllers. *European Wind Energy Conference and Exhibition 2009* 2009.

12. Han Y, Leithead W. Comparison of Individual Pitch Control and Individual Blade Control for Wind Turbine Load Reduction. In: ; 2015.

13. González C, Leithead W, Han Y, Day J. Field tests of Individual Blade Control and its impact on the wind turbine components lifetime. In: ; 2016.

14. Selvam K, Kanev S, Wingerden vJW, Engelen vT, Verhaegen M. Feedback–feedforward individual pitch control for wind turbine load reduction. *International Journal of Robust and Nonlinear Control* 2009; 19(1): 72-91.

15. Geyler M, Caselitz P. Robust Multivariable Pitch Control Design for Load Reduction on Large Wind Turbines. *Fraunhofer IWES* 2008; 130.

16. Stol KA, Zhao W, Wright AD. Individual blade pitch control for the controls advanced research turbine (CART). 2006.

17. Mirzaei M, Soltani M, Poulsen NK, Niemann HH. An MPC approach to individual pitch control of wind turbines using uncertain LIDAR measurements. 2013: 490-495.

18. Doyle JC. Guaranteed margins for LQG regulators. *IEEE Transactions on automatic Control* 1978; 23(4): 756–757.

19. Dunne F, Pao LY, Wright AD, Jonkman B, Kelley N. Adding feedforward blade pitch control to standard feedback controllers for load mitigation in wind turbines. *Mechatronics* 2011; 21(4): 682–690.

20. Wang Y, Gao F, Doyle FJ. Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control* 2009; 19(10): 1589–1600.

21. Kallen T, Zierath J, Dickler S, Konrad T, Jassmann U, Abel D. Repetitive Individual Pitch Control for Load Alleviation at Variable Rotor Speed. *Journal of Physics: Conference Series* 2020; 1618: 022055.

22. Friis J, Nielsen E, Bonding J, Adegas FD, Stoustrup J, Odgaard PF. Repetitive model predictive approach to individual pitch control of wind turbines. In: IEEE; 2011.

23. Brunton SL, Noack BR, Koumoutsakos P. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics* 2020; 52: 477–508.

24. Sutton RS, Barto AG. *Reinforcement learning: An introduction*. MIT press . 2018.

25. Tomin N, Kurbatsky V, Guliyev H. Intelligent Control of a Wind Turbine based on Reinforcement Learning. *2019 16th Conference on Electrical Machines, Drives and Power Systems (ELMA)* 2019: 1-6.

26. Sierra-García JE, Santos M. Exploring Reward Strategies for Wind Turbine Pitch Control by Reinforcement Learning. *Applied Sciences* 2020; 10(21): 7462.

27. Saenz-Aguirre A, Zulueta E, Fernandez-Gamiz U, Lozano J, Lopez-Guede JM. Artificial neural network based reinforcement learning for wind turbine yaw control. *Energies* 2019; 12(3): 436.

28. Stanfel P, Johnson K, Bay CJ, King J. A Distributed Reinforcement Learning Yaw Control Approach for Wind Farm Energy Capture Maximization. In: IEEE. ; 2020: 4065–4070.

29. Collet D, Di Domenico D, Sabiron G, Alamir M. A data-driven approach for fatigue-based individual blade pitch controller selection from wind conditions. In: IEEE. ; 2019: 3500–3505.

30. Mulders S, Pamososuryo A, Wingerden vJW. Efficient tuning of Individual Pitch Control: A Bayesian Optimization Machine Learning approach. *Journal of Physics: Conference Series* 2020; 1618: 022039.

31. Coquelet M, Bricteux L, Lejeune M, Chatelain P. Biomimetic individual pitch control for load alleviation. *Journal of Physics: Conference Series* 2020; 1618: 022052.

32. Ijspeert AJ. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks* 2008; 21(4): 642–653.

33. Mitra SK, Kuo Y. *Digital signal processing: a computer-based approach*. McGraw-Hill New York. 3 ed. 2005.

34. Ba JL, Kiros JR, Hinton GE. Layer normalization. *arXiv preprint arXiv:1607.06450* 2016.

35. Dzeladini F, Ait-Bouziad N, Ijspeert A. CPG-based control of humanoid robot locomotion. *Humanoid Robotics: A Reference* 2019: 1–35.

36. Iscen A, Caluwaerts K, Tan J, et al. Policies modulating trajectory generators. In: PMLR. ; 2018: 916–926.

37. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In: . 80. ; 2018: 1856-1865.

38. Hill A, Raffin A, Ernestus M, et al. Stable Baselines. https://github.com/hill-a/stable-baselines; 2018.

39. Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Software available from tensorflow.org.

40. Bottasso C, Cacciola S, Schreiber J. Local wind speed estimation, with application to wake impingement detection. *Renewable Energy* 2018; 116: 155–168.

41. Bolton W. *Control systems*. Newnes . 2002.

42. Bastankhah M, Porté-Agel F. A new analytical model for wind-turbine wakes. *Renewable Energy* 2014; 70: 116–123.

43. Coudou N, Moens M, Marichal Y, Van Beeck J, Bricteux L, Chatelain P. Development of wake meandering detection algorithms and their application to large eddy simulations of an isolated wind turbine and a wind farm. In: . 1037. IOP Publishing. ; 2018: 072024.

44. Madsen H, Riziotis V, Zahle F, et al. Blade element momentum modeling of inflow with shear in comparison with advanced model results. *Wind Energy* 2012; 15(1): 63–81.

45. Blasques J, Natarajan A. Mean load effects on the fatigue life of offshore wind turbine monopile foundations. In: International Center for Numerical Methods in Engineering; 2013: 818-829.

46. Bishop C. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer. 1 ed. 2007.

47. Chatelain P, Duponcheel M, Caprace DG, Marichal Y, Winckelmans G. Vortex particle-mesh simulations of vertical axis wind turbine flows: from the airfoil performance to the very far wake. *Wind Energy Science* 2017; 2(1): 317–328.

48. Caprace DG, Chatelain P, Winckelmans G. Lifting line with various mollifications: theory and application to an elliptical wing. *AIAA Journal* 2019; 57(1): 17–28.

49. Docquier N, Poncelet A, Fisette P. ROBOTRAN: A powerful symbolic gnerator of multibody models. *Mechanical Sciences* 2013; 4: 199-219.

50. Mann J. Wind field simulation. *Probabilistic engineering mechanics* 1998; 13(4): 269–282.

51. Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-MW reference wind turbine for offshore system development. tech. rep., National Renewable Energy Lab.(NREL); 2009.

52. Bossanyi E. Further load reductions with individual pitch control. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 2005; 8(4): 481–485.

53. Chen ZJ, Stol KA. An assessment of the effectiveness of individual pitch control on upscaled wind turbines. *Journal of Physics: Conference Series* 2014; 524: 012045.

54. Wang C, Campagnolo F, Bottasso C. Does the use of load-reducing IPC on a wake-steering turbine affect wake behavior?. *Journal of Physics: Conference Series* 2020; 1618(2).

55. Chen ZJ, Stol K, Mace B. System identification and controller design for individual pitch and trailing edge flap control on upscaled wind turbines. *Wind Energy* 2015; 19: n/a-n/a.

56. Burton T, Sharpe D, Jenkins N, Bossanyi E. *Wind Energy Handbook* . 2002.

57. Bergami L, Gaunaa M. Analysis of aeroelastic loads and their contributions to fatigue damage. *Journal of Physics Conference Series* 2014; 555: 012007.

58. Bottasso CL, Croce A, Riboldi C, Nam Y. Multi-layer control architecture for the reduction of deterministic and non-deterministic loads on wind turbines. *Renewable Energy* 2013; 51: 159–169.

59. *Wind Energy Industry Manufacturing Supplier Handbook*. Global Wind Network . 2011.

60. *Guideline for the Certification of Wind Turbines*. Germanischer Lloyd . 2010.

61. Stevens RJ, Graham J, Meneveau C. A concurrent precursor inflow method for large eddy simulations and applications to finite length wind farms. *Renewable energy* 2014; 68: 46–50.

62. Lejeune M, Moens M, Coquelet M, Coudou N, Chatelain P. Data assimilation for the prediction of wake trajectories within wind farms. *Journal of Physics: Conference Series* 2020; 1618(6): 062055.

63. Moens M, Coquelet M, Trigaux F, Chatelain P. Handling Individual Pitch Control within an Actuator Disk framework: verification against the Actuator Line method and application to wake interaction problems. *Journal of Physics: Conference Series* 2022.