Biomimetic individual pitch control for load alleviation

Biomimetic individual pitch control for load alleviation

M Coquelet[1,2], L Bricteux[2], M Lejeune[1] and P Chatelain[1]

[1] Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, 1348 Louvain-la-Neuve, Belgium.

[2] Fluids-Machines Department, Université de Mons, 7000 Mons, Belgium.

E-mail: marion.coquelet@uclouvain.be

Abstract. Individual pitch control has shown great capability of alleviating the oscillating loads experienced by wind turbine blades due to wind shear, atmospheric turbulence, yaw misalignement or wake impingement. This work presents a bio-inspired structure for individual pitch control where neural oscillators produce basic rhythmic patterns of the pitch angles, while a deep neural network modulates them according to the environmental conditions. This mimics, respectively, the central patterns generators present in the spinal chord of animals and their cortex. The mimicry further applies to the neural network as it is trained with reinforcement learning, a method inspired by the trial and error way of animal learning. Large eddy simulations of the reference NREL 5MW wind turbine using this biomimetic controller show that the neural network learns how to reduce fatigue loads by producing smooth pitching commands.

1. Introduction

Most modern wind turbines are capable of varying both their rotation speed and pitch angles. To do so, they rely on two control systems working independently: a generator-torque controller, maximizing the power captured below the rated operating point, and a collective blade pitch controller (CPC), regulating the rotor speed above the operating point, as described by Jonkman [1]. In addition to this, an increasing number of turbines is being equipped with a third control loop: an individual pitch controller. Indeed, wind turbine blades experience large once-per-revolution loads oscillations due to the varying velocity they see over the course of their rotation. Individual Pitch Control (IPC) has proved effective in reducing these fatigue loads relying on cyclic oscillations of the pitch angles. While the Coleman transform-based IPC (CT-IPC) is widely used [2], we aim at showing that a bio-inspired IPC structure (BI-IPC) could offer more adaptability as well as smoother commands for the pitch angles. We propose an IPC structure inspired by animal locomotion, as it entails rhythmic motion and is able to adapt to environmental conditions. Such agility and adaptability are enabled by a hierarchical structure separating low-level tasks from high-level ones. On the one hand, we use neural oscillators acting as the central pattern generators (CPG) present in animals' spinal chord to generate the low-level rhythmic patterns. On the other hand, we modulate those patterns with high-level signals obtained through a deep neural network (DNN), similarly to the way the brain of an animal would control the gait. After training the DNN with reinforcement learning (RL), the controller is deployed within Large Eddy Simulations (LES) of the reference NREL 5MW wind turbine [1]. The latter are performed by means of an in-house Vortex Particle-Mesh method (VPM) in which the blades are modelled by immersed lifting lines [3]. The performances of the BI-IPC are assessed against those of the CT-IPC.

The paper is organized as follows. Section 2 recalls the principle of the CT-IPC and describes the BI-IPC along with its training process. Results are presented in section 3, first by validating the trained BI-IPC in the case of the turbine subject to an unsteady sheared flow, second by assessing its performances thanks to LES of a sheared and turbulent flow. Conclusions are eventually drawn in section 4.

2. Methodology

We first recall the state-of-the-art individual pitch controller, which is based on the Coleman transform and was proposed by Bossanyi [2]. We then present the aforementioned biomimetic structure, based on reinforcement learning, and we finally outline the learning process.

2.1. Coleman transform-based individual pitch control

The principle of the CT-IPC is to project the loads expressed in the rotating coordinate frame of the blades on the fixed coordinate frame of the rotor. According to the azimuthal position $\theta(t)$ of the first blade, defined from the upward vertical position, the flapwise bending moments of the blades are projected on a mean bending moment $\overline{M}(t)$, a tilting one $M_{\text{tilt}}\ (t)$ and a yawing one $M_{\text{yaw}}\ (t)$ using the inverse Coleman transform (Eq. 1). The target values $M_{\text{tilt,target}}$ and $M_{\text{yaw,target}}$ are time-invariant and equal to zero. The tilt and yaw angles $\beta_{\text{tilt}}(t)$ and $\beta_{\text{yaw}}(t)$ are the control variables in the fixed frame. They are projected on the blades rotating coordinate frame by means of the Coleman transform (Eq. 2), thus giving the expression of the individual pitch angles $\beta_{1,2,3}$. The mean pitch $\overline{\beta}(t)$ is the one computed from the collective pitch controller.

$$\begin{bmatrix} \overline{M}(t) \\ M_{\text{tilt}}\ (t) \\ M_{\text{yaw}}\ (t) \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3}\cos\theta(t) & \frac{2}{3}\cos\left(\theta(t)-\frac{2\pi}{3}\right) & \frac{2}{3}\cos\left(\theta(t)+\frac{2\pi}{3}\right) \\ \frac{2}{3}\sin\theta(t) & \frac{2}{3}\sin\left(\theta(t)-\frac{2\pi}{3}\right) & \frac{2}{3}\sin\left(\theta(t)+\frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} M_1(t) \\ M_2(t) \\ M_3(t) \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \beta_1(t) \\ \beta_2(t) \\ \beta_3(t) \end{bmatrix} = \begin{bmatrix} 1 & \cos\theta(t) & \sin\theta(t) \\ 1 & \cos\left(\theta(t)-\frac{2\pi}{3}\right) & \sin\left(\theta(t)-\frac{2\pi}{3}\right) \\ 1 & \cos\left(\theta(t)+\frac{2\pi}{3}\right) & \sin\left(\theta(t)+\frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} \overline{\beta}(t) \\ \beta_{\text{tilt}}(t) \\ \beta_{\text{yaw}}(t) \end{bmatrix} \tag{2}$$

2.2. Biomimetic individual pitch control

As mentioned previously, animal locomotion is made possible under the assumption of dividing low level tasks from high level ones. For this reason, we decided not to consider each pitch angle following periodic oscillations individually, but rather opt for coupled oscillators to produce the pitching angles. Our control scheme takes the form of a three-block architecture, as shown in Fig. 1, where neural oscillators generate the rhythmic patterns that are modulated by a neural network. The latter is trained thanks to reinforcement learning according to the flow conditions.

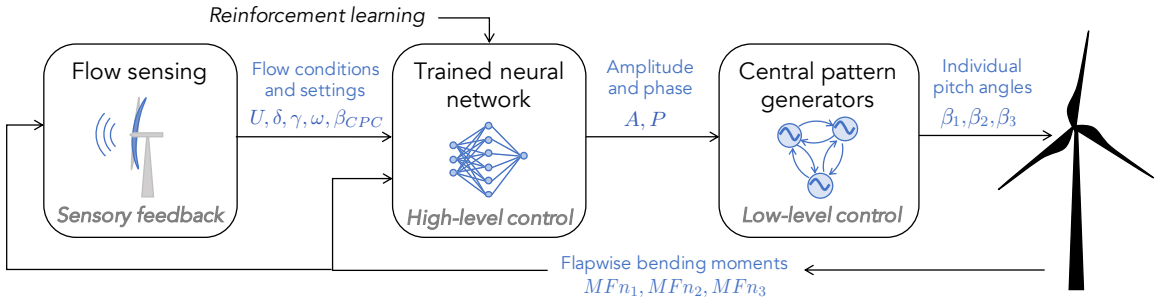


Figure 1. Biomimetic individual pitch controller.

2.2.1. Flow sensing The first block of the control scheme is the flow sensing module, which consists in using the wind turbine blades as sensors of the flow. It relies on an Extended Kalman Filter to estimate a linear approximation of the instantaneous velocity field upstream of the rotor:

$$V(x = x_{rotor}, y, z) = U + \gamma y + \delta z, \tag{3}$$

where x, y and z and respectively the streamwise, vertical and transverse directions. The estimation is made from the measurement of the flapwise bending moments experienced by each blade, $MF_{n_{1,2,3}}$, in the fashion of [4]. To do so, the filter is provided with a model of the physical system, namely Blade Element Momentum theory (BEM), that expresses the bending moments as a function of the velocity field. From a first guess of the velocity field parameters U, γ, δ, the filter uses the BEM to compute the bending moments $MF^*_{n_{1,2,3}}$ that should be affecting the blades, given the knowledge of the operating settings (the blade azimuth θ, the blade pitch β and the rotation speed ω). These estimated bending moments, $MF^*_{n_{1,2,3}}$, are then compared to the measured ones, $MF_{n_{1,2,3}}$, and the estimation of the upstream velocity field is updated accordingly.

2.2.2. Central pattern generators The pitch angles are computed by central pattern generators in the last block of the control scheme. CPGs are classically modelled by means of neural oscillators and are used to produce coordinated patterns. In this work, the CPGs take the form of a system of three oscillators ($i = 1, 2, 3$), inspired by Crepsi [5], and implemented as follows:

$$\ddot{p} = k_p \left(\frac{k_p}{4} (P - p) - \dot{p} \right) \tag{4}$$

$$\ddot{a} = k_a \left(\frac{k_a}{4} (A - a) - \dot{a} \right) \tag{5}$$

$$\ddot{x} = k_x \left(\frac{k_x}{4} (X - x) - \dot{x} \right) \tag{6}$$

$$\beta_i = x + a \cos(\theta_i + p) \triangleq x + a \cos(\varphi_i). \tag{7}$$

The patterns of the individual pitch angles $\beta_{1,2,3}$ are computed from three internal variables: a and x are respectively the amplitude and offset of the oscillations, p stands for the phase shift between the azimuthal position of the blade θ_i and the phase of the oscillator φ_i (Fig. 2).

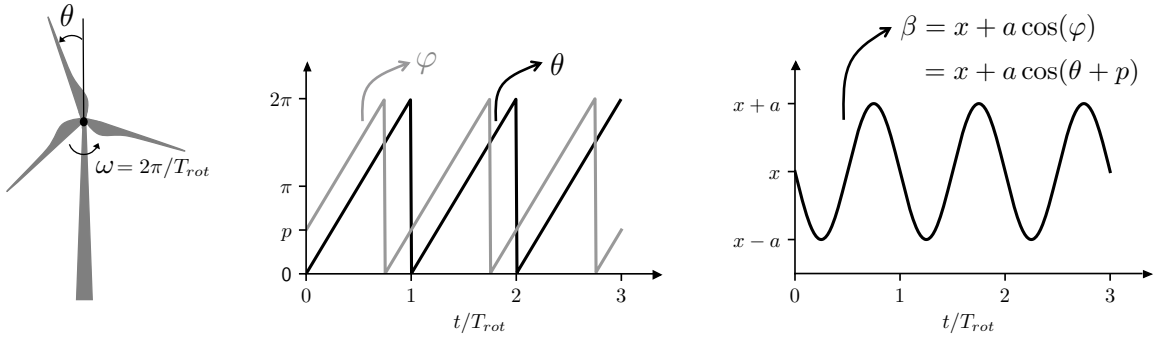


Figure 2. Central pattern generators: parameters definition.

Equations 4 to 6 are those of second order systems whose responsiveness is characterized by the positive gains k_p, k_a and k_x. The internal variables smoothly converge to the target values

P, A and X. The target value of the offset X is the collective pitch angle β_{CPC}, determined by the aforementioned variable-speed, variable-pitch baseline controller.

In the case of an inflow with shear only, one could expect that $P = 0 \Rightarrow \varphi = \theta$, so that the oscillator is phased with the blade azimuth. Nonetheless, it is not trivial to find the value of P when gusts are present in the flow or when a wake impinges on the rotor and neither is it to find the optimal amplitude A of the pitch oscillations. Both the target amplitude A and phase shift P of the oscillations are provided from a trained neural network, which helps close the gap between the CPG control variables and the environmental conditions.

2.2.3. Neural network and reinforcement learning The control block between the sensing module and the central pattern generators consists in a deep neural network. It constantly redefines the pitching targets according to the environmental conditions. It can be seen as an optimal controller that collects information on the flow and modulates the rhythms accordingly. We use reinforcement learning to train the neural network, as it is the branch of machine learning that best applies to control problems.

RL refers to how agents interacting with their environments learn the best way to behave, depending on the state they are in, so as to maximize a cumulative reward. It is thus a matter of finding the most appropriate mapping, which is called a policy $\pi\left(\mathbf{a}_t|\mathbf{s}_t\right)$, between the state $\mathbf{s}_t$ of the agent and the action $\mathbf{a}_t$ to take, given a certain objective function $J(\pi)$ [6].

In this paper, we consider the wind turbine as the agent, the environment being the flow surrounding it, and the policy is expressed by the deep neural network. The role of the DNN is to learn the action $\mathbf{a}_t = [A, P]$, namely the amplitude and the phase shift of the CPGs oscillations that most reduce the fatigue loads on the blades given the flow conditions determined by the sensing module. The action is updated at every turbine rotation based on the state of the system, described by

$$\mathbf{s}_t = \left[U, \gamma, \delta, \omega, \beta_{CPC}\right], \tag{8}$$

with U, γ and δ respectively the mean velocity, vertical shear coefficient and horizontal shear coefficient of the flow impinging on the rotor, ω the rotation speed of the turbine and β_{CPC} the collective pitch angle command. The reward associated to the action is computed as

$$r\left(\mathbf{s}_t, \mathbf{a}_t\right) = \left(\max_i MF_{n_i} - \min_i MF_{n_i}\right)^{-2}. \tag{9}$$

The network is trained using an off-policy model-free reinforcement learning algorithm called Soft Actor Critic (SAC) and presented by Haarnoja [7]. SAC is part of maximum entropy reinforcement learning methods, which consider an entropy augmented objective

$$J(\pi) = \mathbb{E}_\pi\left[\sum_t r\left(\mathbf{s}_t, \mathbf{a}_t\right) - \alpha \log\left(\pi\left(\mathbf{a}_t|\mathbf{s}_t\right)\right)\right], \tag{10}$$

with α 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. 10), and exploration of new ones, through entropy maximization (second term of Eq. 10). SAC shows great sample efficiency, meaning that it makes the most of each learning episode so as to reduce the number of episodes to learn a policy. It is also little sensitive to hyperparameters, which avoids massive parameters tuning. The aforementioned characteristics make SAC one of the most efficient algorithms available these days [7].

We use the open-source SAC implementation of Stable Baselines [8], a fork of Open AI baselines. The policy DNN is a fully-connected multilayer perceptron built from TensorFlow [9]. The input layer consists of 5 neurons (one for each component of the state), while the output one comprises 2 neurons (one for each component of the action), with tanh as activation function.

2.3. Learning process and controller performances assessment
Using Reinforcement Learning implies going through two steps. The first one consists in training the wind turbine so that it learns the policy. As this requires numerous learning episodes, simple models of both the flow and the turbine are needed for the sake of maintaining affordable computational costs. As far as the flow is concerned, we wish to reproduce the shear plane variation that a turbine could see when gusts or wakes impinge on its rotor. To do so, we use Porte-Agel's model for the wake deficit [10], accounting for meandering of the wake center through a sine motion. The wake deficit is superimposed on an exponential law to represent the shear present in the atmospheric boundary layer. Regarding the turbine model, a fast yet accurate way of computing loads acting on blades is the Blade Element Momentum Theory. We make use of a modified BEM accounting for the effects of shear [11]. This basic model of the physics allows for an efficient learning from the system with low computational and data storage costs. A learning step consists in a rotation of the turbine, while a learning episode contains 30 steps. We used early stopping [12] as a simple yet effective regularization method to avoid overfitting and ensure generalization.

The second step consists in testing the trained policy, which is no longer learning but is only exploiting what is has learned. This is done, in a first step, by using a BEM to compute residual loads. However, assessing the controller performances necessitates great accuracy in the computation of the flow physics and in the loads evaluation. The biomimetic controller is thus deployed within Large Eddy Simulations of the reference NREL 5MW wind turbine. These simulations are performed by means of a Vortex Particle-Mesh method (VPM) in which the blades are modelled by immersed lifting lines [3], coupled to the multi-body-system solver Robotran in charge of the dynamics of the turbine [13].

3. Results
The trained control scheme is first studied with a BEM in the simplified case of varying shear directions and intensities, providing a clear insight into the actions taken by the BI-IPC regarding the flow conditions it senses. The performances of the controller are further assessed in realistic wind conditions by analyzing results of large eddy simulations of turbulent flows. More precisely, the temporal and spectral characteristics of the blade loads as well as the damage equivalent loads are computed and compared for the cases of CPC, CT-IPC and BI-IPC.

3.1. BEM validation of the trained BI-IPC subject to unsteady sheared flows
We first test the policy with a BEM in the case of a sheared inflow with a meandering wake impinging on the turbine. This is seen by the turbine as a shear plane whose direction and intensity are changing in time. Figure 3 shows how the turbine perceives the flow it is subjected to as well as the actions that its neural network takes. The sensing module feels the passing of the wake, as the horizontal plane coefficient δ evolves with time. It also senses the presence of the constant vertical shear that is imposed, with a constant non-zero γ plane coefficient (Fig. 3 (a)). The wake impingement is also detected by the estimator, a reduction of velocity is indeed correlated with the wake passage (Fig. 3 (b)). The actions taken by the neural network show it has acquired the ability to shift the phase of the CPGs when the orientation of the upstream flow shear plane evolves (Fig. 3 (c)). It has also gained the capability of slightly adjusting the pitching amplitude regarding the overall shear perceived by the blades over the course of their rotation (Fig. 3 (d)).

The resulting pitching patterns and flapwise bending moments are shown in Fig. 4. They are compared to the use of collective pitch control only, showing that BI-IPC offers a significant reduction of the fatigue loads. While this case study is far from realistic wind conditions, it enables to outline the impact of the flow conditions on the actions taken by the neural network and to highlight the resulting pitching patterns.

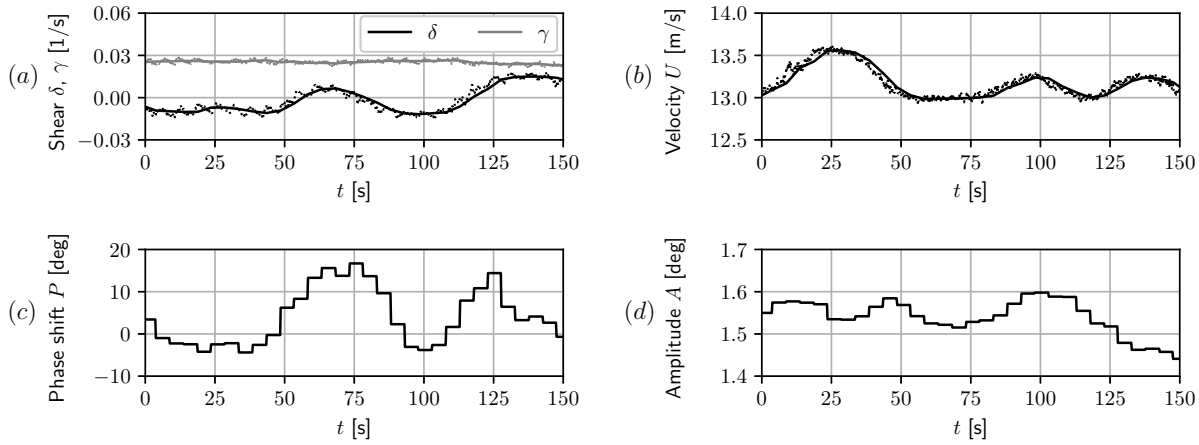


Figure 3. BEM validation of the BI-IPC: Estimated shear planes (a) and velocity U (b), instantaneous value (dots) and rotation-averaged one (solid line). Phase shift P (c) and amplitude A (d) determined by the trained neural network.

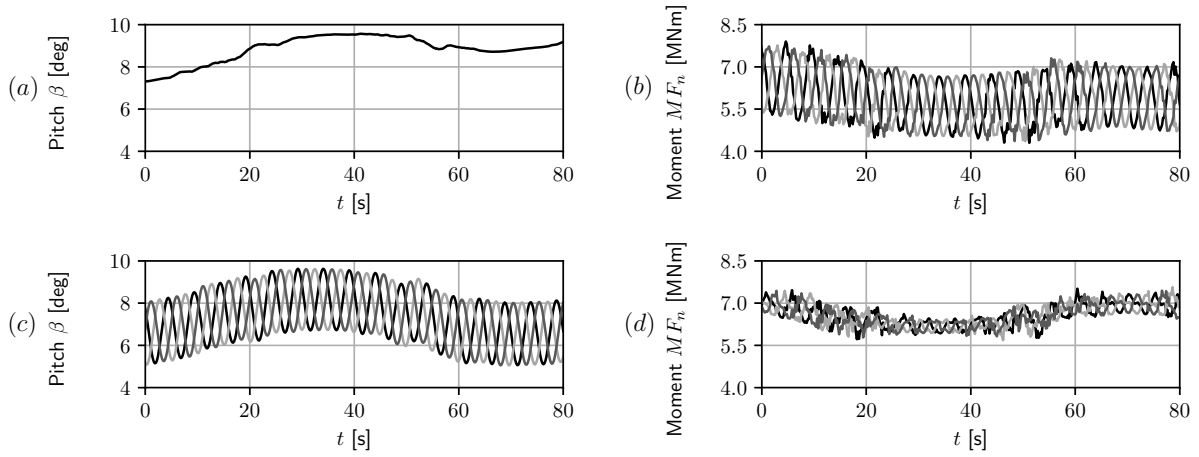


Figure 4. BEM validation of the BI-IPC and comparison with CPC: Pitching patterns and resulting flapwise bending moments in the case of collective pitch control, (a) and (b), and biomimetic pitch control, (c) and (d).

3.2. LES assessment of the BI-IPC in sheared and turbulent flows

This section offers a comparison between large eddy simulations of CPC, CT-IPC and BI-IPC in the case of a sheared inflow characterized by a hub velocity of 15 m/s and a turbulence intensity of 6%. The synthetic turbulence is generated by using Mann's algorithm [14].

A qualitative analysis shows that the pitch angles evolution obtained by both IPC methods present similar patterns (Fig. 5). This comes from the fact that the CT-IPC solves the load alleviation problem by projecting bending moments on a plane around the rotor while the BI-IPC does so with the upstream velocity field, the two being inextricably linked. Yet, it must be outlined that the BI-IPC commands smoother variations of the pitch angles, being less demanding for the pitch actuators. Even though the BI-IPC significantly reduces fatigue loads (Fig. 6), both the temporal and spectral analyses show that the CT-IPC further cuts flapwise bending moments oscillations at the frequency of the turbine rotation.

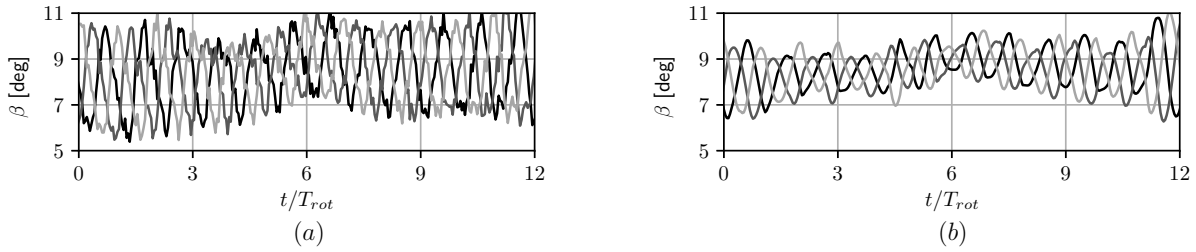
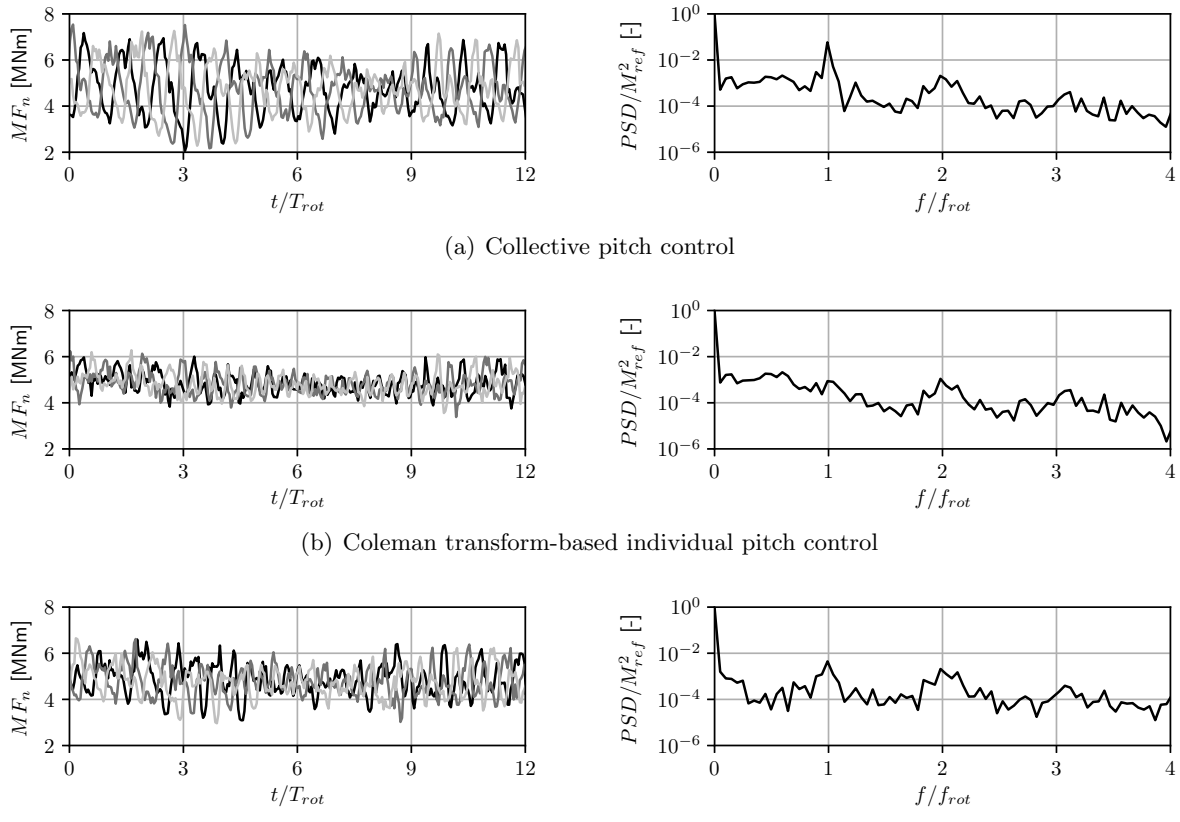


Figure 5. LES assessment of the BI-IPC: Pitch angles β using Coleman transform-based individual pitch control (a) and biomimetic individual pitch control (b), one color per blade.



(a) Collective pitch control

(b) Coleman transform-based individual pitch control

(c) Biomimetic individual pitch control

Figure 6. LES assessment of the BI-IPC: Flapwise bending moments MF_n (one color per blade) and their power spectral density PSD (average over the three blades) with collective pitch control only (a), with Coleman transform-based individual pitch control (b) and with biomimetic individual pitch control (c)

A more quantitative analysis of the impact of IPC on fatigue loads is put forward by computing damage equivalent bending moments [15]. The damage D perceived by a blade submitted to a varying bending moment $MF_n(t)$ over a finite time period is given by Palmgren-Miner's rule

$$D = \sum_{i=0}^{n_c} \frac{n_i}{N_{F,i}}, \tag{11}$$

where n_i in the number of cycles according to the rainflow algorithm and $N_{F,i}$ is the number of cycles to failure given a bending moment oscillating with an amplitude $M_{a,i}$. Using M-N curve and assuming that the bending moment is proportional to the stress, we obtain that $N_{F,i} = kM_{a,i}^{-m}$, with k and m being material properties. The equivalent bending moment M_{eq} is the moment, alternating at a chosen frequency f_{eq}, that generates the same damage over the time period considered than the measured time-varying moment. It is expressed as

$$M_{eq} = \left(\sum_{i=0}^{n_c} \frac{n_i M_{a,i}^m}{N_{eq}} \right)^{1/m}, \tag{12}$$

where we choose $m = 10$ for glass fiber and the number of equivalent cycles N_{eq} so that the equivalent loading frequency is the turbine rotation frequency.

Table 1 gives the value of the equivalent alternating moments for the different pitching strategies. BI-IPC reduces the fatigue damage by 44%, while CT-IPC does so by 61%. A fair quantitative comparison would also include the fatigue of the pitching actuators.

Table 1. Damage equivalent bending moments depending on the chosen pitching strategy

Method	M_{eq} [MNm]
CPC	2.54
CT-IPC	0.99
BI-IPC	1.40

3.2.1. Results discussion The linear approximation of the upstream velocity field proposed in this first-stage study is quite reductive, yet it demonstrates the learning capability of the system to alleviate fatigue blade loads. If provided with more information such as the local velocity experienced by the blades, the neural network could control the amplitude and phase offset of pattern generators oscillating at higher harmonics of the rotation frequency of the turbine. This would lead to more flexibility in terms of pitching patterns and, therefore, to further load reduction, while the intrinsically gentle pitching commands of the BI-IPC would be maintained. In brief, it would be a great compromise between the optimal load reduction of the CT-IPC and the smooth commands of the BI-IPC.

4. Conclusions

We present an individual pitch controller inspired by animal locomotion. The pitch angles are generated by neural oscillators that are modulated by a neural network. The former is trained with reinforcement learning based on the upstream flow conditions. After being trained with simple wake models and blade element momentum theory, the controller is 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. The latter performs better at alleviating loads, but produces jerkier pitching commands.

The range of actions made accessible to the BI-IPC is still quite limited in what is, to the authors' knowledge, the first study of reinforcement learned IPC. Nevertheless, the controller manages to significantly reduce fatigue loads, while providing smooth pitching commands. This work thus demonstrates that a wind turbine pursuing a certain objective can learn how to behave in the flow it is subjected to. BI-IPC should outperform CT-IPC if oscillators with

higher harmonics of the turbine rotation frequency are added to the scheme. Further work will then consist in providing the neural network with local information on the velocity field and not the global linear approximation anymore. Such improvement can clearly be envisioned thanks to the flexibility of the BI-IPC and should benefit from its smooth generation of commands.

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). 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.

References

[1] Jonkman J, Butterfield S, Musial W and Scott G 2009 *National Renewable Energy Laboratory (NREL) Report*
[2] Bossanyi E 2003 *Wind Energy* **6** 119–128
[3] Chatelain P, Backaert S, Winckelmans G and Kern S 2013 *Flow, Turbulence and Combustion* **91** 587–605
[4] Bottasso C, Cacciola S and Schreiber J 2017 *Renewable Energy* **116** 155–168
[5] Crespi A, Lachat D, Pasquier A and Ijspeert A 2008 *Autonomous Robots* **25** 3–13
[6] Sutton R and Barto A 2018 *Reinforcement Learning: An Introduction* 2nd ed (The MIT Press)
[7] Haarnoja T, Zhou A, Abbeel P and Levine S 2018 Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor *Int. Conf. on Machine Learning* vol 80 pp 1856–1865
[8] Raffin A *et al.* 2018 Stable Baselines: Improved implementations of reinforcement learning algorithms based on OpenAI Baselines `https://github.com/hill-a/stable-baselines`
[9] Abadi M *et al.* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems `http://tensorflow.org/`
[10] Bastankhah M and Porté-Agel F 2014 *Renewable Energy* **70** 116–123
[11] Madsen H A, Riziotis V, Zahle F, Hansen M, Snel H, Grasso F, Larsen T, Politis E and Rasmussen F 2012 *Wind Energy* **15** 63–81
[12] Bishop C 2007 *Pattern Recognition and Machine Learning (Information Science and Statistics)* 1st ed (Springer)
[13] Docquier N, Poncelet A and Fisette P 2013 *Mechanical Sciences* **4** 199–219
[14] Mann J 1998 *Probabilistic Engineering Mechanics* **13** 269–282
[15] Blasques J and Natarajan A 2013 Mean load effects on the fatigue life of offshore wind turbine monopile foundations *Computational Methods in Marine Engineering V - Proc. of the 5th Int. Conf. on Computational Methods in Marine Engineering, MARINE 2013* (International Center for Numerical Methods in Engineering) pp 818–829