CONTROL OF A WIND-TURBINE VIA MACHINE LEARNING TECHNIQUES

L. Schena[1], E. Gillyns[1], W. Munters[1], S. Buckingham[1], M. A. Mendez[1]

[1] von Karman Institute for Fluid Dynamics
Waterloose Steenweg 72, 1640 Sint-Genesius-Rode, Belgium
e-mail: lorenzo.schena@vki.ac.be

Key words: Wind-turbine control, Reinforcement Learning, Loads mitigation

Abstract. This article presents two *model-free* controllers for wind-turbine torque and pitch control. These controllers are based on reinforcement learning (RL) and Bayesian optimization (BO) and do not rely on any mathematical model of the wind-turbine dynamics, in contrast to classical approaches designed on linearized models. The model-free controllers were benchmarked against a proportional-integral-derivative (PID) regulator in a numerical environment using Blade Element Momentum theory for computing the aerodynamic torque and the blade loads. The results showed that the model-free approaches could increase power harvesting while reducing wind turbine loads.

1 INTRODUCTION

The control of wind turbines is a grand challenge in wind energy [1], and it is crucial to its viability [2]. Currently, most wind turbines operate in variable-speed or variable-pitch (VS-VP) conditions [3, 4] depending on the wind speed. Figure 1 shows the ideal power curve for a reference NREL 5-MW [5] wind turbine considered in this study. This is designed for rated power of $P_R = 5$ [MW] at a rated wind speed of $u_R = 11.4$ [m s^{-1}]. A standard controller [6] acts on the generator torque at a wind speed $u_\infty < u_R$, holding the blades fine-pitched to the optimal pitch angle, and on the pitch angle at $u_\infty > u_R$, keeping the nominal torque. This aims to maximize the power production in regions 2 and 2.5 and limit the loads in regions 2.5 and 3.

This work focuses on pitch and torque control in region 2.5, which is particularly critical in balancing the conflicting objectives of power maximization and load mitigation. Gusts or lulls in this region might cause undesired switches from pitch to torque control, leading to suboptimal power production or excessive loads. Standard controllers based on PI (Proportional and Integral) formulations might offer enough flexibility in the control law to handle these conditions optimally [7, 8]. These controllers rely on linearized models of the wind turbine dynamics, but no linearization can handle the switch between the two operating conditions.

In this work, we investigate the application of machine learning algorithms that learn directly from data without relying on any model of the underlying dynamics, and we benchmark their performances against classical PID controllers. Section 2 overviews the simulation environment and the algorithms employed. Next, Sect. 3 details the test cases on which the controllers are compared. Sect. 4 discusses the results and Sect. 5 outlines the conclusions and future work.
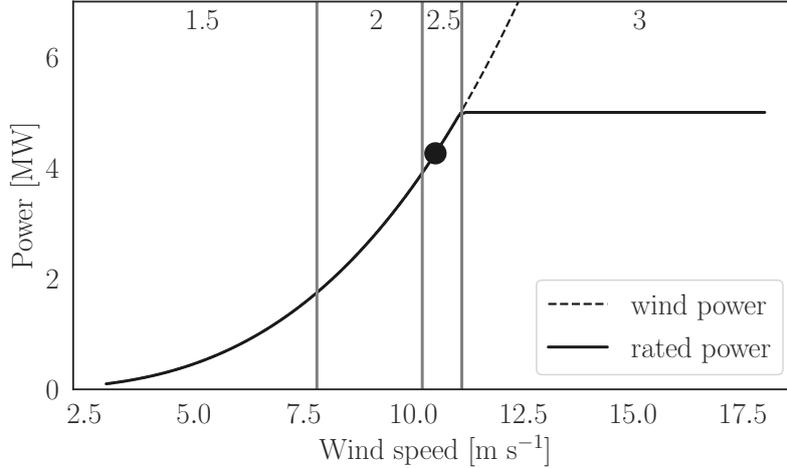
Figure 1: Steady power response as a function of wind-speed for the NREL 5-MW [5] considered in this work. The round markers indicate the investigated conditions.

2 METHODOLOGY

2.1 Simulation environment

The numerical environment considered in this work relies on a first-order model of the wind-turbine dynamics:

$$\dot{\omega} = \frac{N_g}{J}\left(\tau_a - N_g \tau_g \eta_{gb}\right), \tag{1}$$

where $\dot{\omega}$ [rad s^{-2}] is the rotor acceleration, N_g [-] the efficiency of the generator, J is the rotor inertia [kg m^{-2}], η_{gb}[-] is the gearbox ratio, and τ_a and τ_g are the aerodynamic and generator torque respectively. In a first approximation, the aerodynamic torque is a function of the tip speed ratio $\lambda = \omega R/u_\infty$ and the blade pitch angle β [deg]:

$$\tau_a = \frac{1}{2}\rho\pi R^2 \frac{C_p(\lambda,\beta)}{\omega}u_\infty^3 = \frac{1}{2}\rho\pi R^2 C_Q(\lambda,\beta)u_\infty^3, \tag{2}$$

where $C_p(\lambda,\beta)$ is the power coefficient and $C_Q = C_Q(\lambda,\beta)$ is the torque coefficient.

Equation (1) and (2) provide the simplest model for the wind turbine dynamics under the assumption of rigid blades and uniform wind over the rotor's swept area. Although these assumptions are rather restrictive (especially for large turbines), this model is commonly used by control engineers in wind applications [9, 10] and is thus considered appropriate for a first comparative analysis of model-free versus classic controllers.

In order to integrate (1) and (2), the power coefficient was computed using the Blade Element Momentum (BEM) formulation using the WISDEM CCBlade [14] `Python` package, whereas the loads have been computed using AeroDyn [15]. This formulation couples the momentum equation with the blade element theory to compute the forces per unit length acting at different nodes (see Fig. 2a) along each blade.
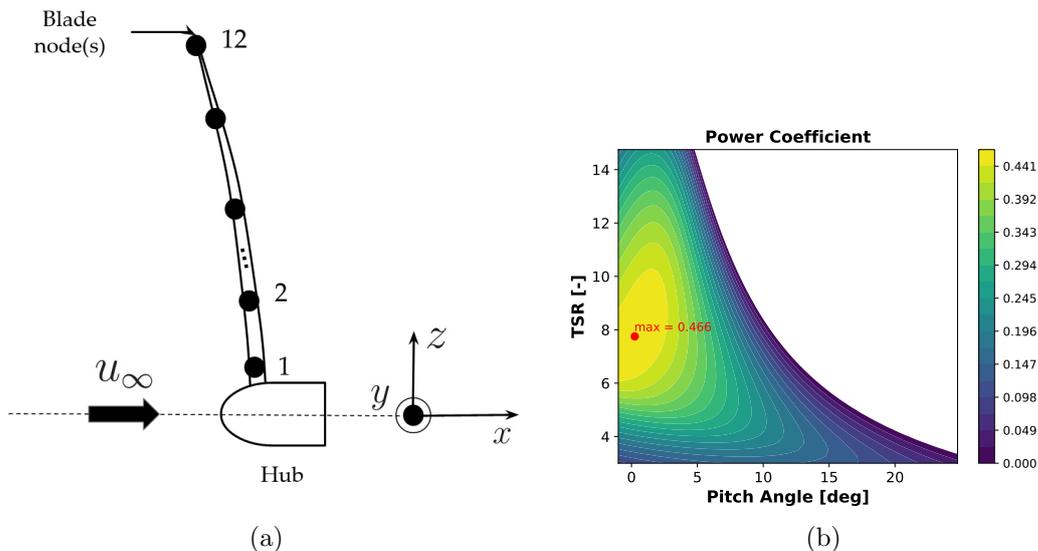
Figure 2: (a) Schematic of the blade, approximated with 12 twelve nodes and reference frame used in the BEM simulation for the test cases considered in this work. b) Contour of the resulting C_p coefficient as a function of λ and β.

The wind velocity and the (instantaneous) rotational speed constitute the *state* $\mathbf{s} = [u_\infty \quad \omega]^T$ vector for the dynamical system to be controlled. In order to make the system physically realistic, the controller *action/actuation* is given in terms of rate-of-change for the pitch angle ($\dot{\beta}$) and torque ($\dot{\tau}$), that is $\mathbf{a} = [\dot{\beta} \quad \dot{\tau}]^T$. Therefore, within the time integration of (1), the update on the β and τ under the action of the controller is:

$$\beta(t + \Delta t) = \beta(t) + \dot{\beta}\Delta t \qquad (3)$$
$$\tau(t + \Delta t) = \tau(t) + \dot{\tau}\Delta t$$

where Δt is the time step in the integration. The rates are clipped in a range $\dot{\beta} \in [-\dot{\beta}_{max}, \dot{\beta}_{max}]$ [deg s^{-1}] and $\dot{\tau} \in [-\dot{\tau}_{max}, \dot{\tau}_{max}]$ [Nm s^{-1}] (see Table 1).

The goal of the optimal controller is to find the *policy/actuation* law $\pi : \mathcal{S} \to \mathcal{A}$, with $\mathcal{S} \subseteq \mathbb{R}^2$ the space of admissible states and $\mathcal{A} \subseteq \mathbb{R}^2$ the set of admissible actions, such that a *reward* function is maximized. The definition of this reward function depends on the control objectives and is described in Sect. 3. First, we briefly introduce the investigated control strategies in Sect. 2.2.

2.2 Algorithms

Algorithm 1 details the implementation of the control evaluation and training processes for the different controllers. For training purposes, the algorithm begins with random initialization of the parameters (line 2) and ends with an update of the parameters at each episode (line 14).

The performance evaluation (whether used for training or not) is carried out within *episodes*

Algorithm 1: Algorithmic Approach for the Optimal Control Law Derivation

1	**begin**
2	Initialize a possible control law $\mathbf{a} = \pi(\mathbf{s}; \mathbf{w})$ of the controller;
3	Initialize the inflow velocity u_∞ and add random noise, such that:
	$u_\infty = u_{\infty,*} + \mathcal{W}(\mu = 0, \sigma)$;
4	**for** n *in (1, $N_{episodes}$)* **do**
5	**for** t *in (1,T)* **do**
6	Initialize wind-turbine at initial conditions;
7	Feed the current state $\mathbf{S} = (u_{\infty,t}, \omega_t)$ to the controller;
8	Get action(s) $\mathbf{a} = (\beta_{t+1}, \tau_{load,t+1})$ following current policy π;
9	Evolve the wind-turbine state $\mathbf{s}_t \rightarrow \mathbf{s}_{t+1}$, evaluating the look-up-tables to get the aerodynamic coefficients C_p, C_Q;
10	**if** *AeroDyn coupling is active* **then**
11	Compute loads F_x, F_y, F_z (in the hub reference frame) acting on the wind-turbine blades, and their rate of change F_{xt}, F_{yt}, F_{zt} via BEM simulation;
12	**end**
13	Compute the reward r_t associated with the change of state $\mathbf{s}_t \rightarrow \mathbf{s}_{t+1}$ and current action $\mathbf{a}_t$;
14	Update policy weights $\mathbf{w}_k$ (for DDPG);
15	**end**
16	Update policy weights $\mathbf{w}_k$ (for BO);
17	**end**
18	**end**

of duration $T = 100$ [s]. This also defines the duration of the simulation. This time scale is deemed adequate because it corresponds to roughly twenty blade rotations in rated conditions. Running the simulations with a time-step of $\Delta t = 0.025$ [s] and considering that the controller interacts with the system at every iteration, one episode produces 4000 agent/environment interactions per episode.

The wind-flow field is generated such that the time average is $\bar{u}_\infty = 10.5$ [m s^{-1}] (hence the turbine operates in Region 2.5, cf. Fig. 1), via TurbSim [18]. This software constructs realistic wind data from its spectral characteristic. White noise with zero mean and standard deviation $\sigma = 0.1$[m s^{-1}] is also added to mimic measurement noise and avoid overfitting (line 3).

As suggested by [5], the velocity signal is low-pass filtered at one-quarter of the first edgewise natural frequency of the blade (corresponding to $f \approx 1$ Hz) to avoid the high-frequency excitation of these control systems. The corner frequency of the filter (that is the -3dB attenuation) is thus set to $f_c = 0.25$ Hz. The implemented filter is a 5-th order Butterworth filter with cut-off frequency $f_{co} = 1.571/(f_s/2)$ Hz, and $f_s = 40$ Hz is the sampling frequency.

Thus, at each timestep, the agent receives the current (filtered) state $\mathbf{s}$ and outputs an action $\mathbf{a}$ which makes the environment (plant) evolve to its next state: $\mathbf{s}_t \rightarrow \mathbf{s}_{t+1}$. An instantaneous reward signal $r(t)$ is produced at each state transition, and a cumulative reward signal $R = \sum_t r(t)$ is produced at the end of the episode. Both measure the controller performances and

are maximum when the control law is optimal. The reward function definition is discussed in 3. The 'black-box' optimization (Section 2.3) updates the control law at the end of the episodes (line 17) while the reinforcement learning algorithm (Section 2.4) updates at each episode.

Before introducing the model-free approaches, Sect. 2.2.1 briefly recalls the fundamentals of the PID controllers used as reference.

2.2.1 PI (or PID) controllers

This is a classic control approach and serves as a reference controller in this work. The general multi-input-multi output (MIMO) formulation for the problem described in Eq. (4)

$$\mathbf{a}(\mathbf{s}) = \begin{bmatrix} k_{P1} & 0 \\ 0 & k_{P2} \end{bmatrix} \mathbf{e}(\mathbf{s}(t)) + \begin{bmatrix} k_{I1} & 0 \\ 0 & k_{I2} \end{bmatrix} \int_0^t \mathbf{e}(\mathbf{s}(\tau))d\tau + \begin{bmatrix} k_{D1} & 0 \\ 0 & k_{D2} \end{bmatrix} \frac{d\mathbf{e}(\mathbf{s}(t))}{dt} , \qquad (4)$$

where $\mathbf{e} = \mathbf{s} - \mathbf{s}_{ref}$ is the error between the current state and the reference state while the coefficients k_P, k_I and k_D are the proportional, integral and derivative gains respectively. The diagonal form of the matrices involved makes the controllers for each action independent from the others but one could include more general formulations with off-diagonal terms.

Considering that in this work we do not control the first state (the wind speed), Eq. (4) becomes a scalar equation ($k_{P1} = k_{I1} = k_{D1} = 0$). The remaining coefficients can be tuned from data or from dynamic considerations of the system at hand. In this work, we consider the second approach and we used the Reference Open Source COntroller (ROSCO, [9]) for determining k_{P2} and k_{I2}. As it is common practice for utility-scale wind turbines, the coefficient k_{D2} is set to zero to limit the control sensitivity to measurement noise. The final controller is thus a PI controller.

The reader is referred to [9] for more details on how the coefficients are derived. Briefly, these are computed from a linearized model of the wind turbine and designed to ensure that the (linear) closed loop system has a prescribed transfer function.

2.3 Bayesian Optimization (BO)

The BO is a classic surrogate based approach to minimize a 'black-box' function [11]. As done in [12], the training of a model-free controller is converted into a black-box optimization problem by prescribing the parametric form of the control law. In this work, we consider a linear law $\mathbf{a} = \mathbf{W}\mathbf{s}$ with $\mathbf{W} \in \mathbb{R}^{2\times 2}$ collecting the four coefficients to be identified. For a given choice of parameters, the environment can be simulated for one episode and the global cumulative reward $R(\mathbf{W})$ is the cost function that needs to be optimized.

The BO formulation used in this work is the classic combination of Gaussian Process Regression (GPR) for the surrogate model and Lower Confidence Bound (LCB) for the acquisition function. The GPR was carried out using a Matern kernel with $\nu = 1.5$ and length scale $l = 1.0$ (see [13] for more details).

2.4 Reinforcement Learning (RL)

Reinforcement Learning (RL) is the machine learning paradigm in which the agent (controller) learns how to interact with the environment (plant) to maximize a reward signal [16]. The conceptual architecture of the RL agent is the same as for the BO agent: this is simply a

different way of converting the problem of training a model-free controller into a black-box optimization. In modern RL approaches, based on Deep Neural Networks (also known as Deep RL), the policy $\mathbf{a} = \pi(\mathbf{s}; \mathbf{w})$ is encoded in an Artificial Neural Network (ANN), and the training is performed using stochastic gradient ascent approaches. This allows for deriving a general (nonlinear) control law without the need to prescribe a specific form (like in the linear policy trained by the BO formulation).

In this work, an in-house customized implementation of the Deep Deterministic Policy Gradient (DDPG, [17]) is employed (see also [12]). Two ANNs are employed, one to map states to actions, the *actor* and another one that approximates the long-term consequences of the applied actions - or *value function*-, the *critic*. The former consists in a 32×32 network plus a concatenation of the input layer to allow linear control parametrizations, whereas the latter concatenates two networks. The first, from the action taken by the agent composed as 2×64. The states are elaborated in two layers of size 2×32×32. These are concatenated and expanded by means of two layers with 32×32×1 , neurons, from which the output is the value estimated. Both the network architecture and the learning approach are the same as in [12], with 2 neurons in the first (inputs) and the last (actuation) layers. The total number of parameters to be trained is 6750.

Finally, it is noted that, while the problem has been posed as a minimization problem for the BO optimization and a maximization one for the RL approach, following the relative literature practice, the two are de facto identical, as the fitness functions have been modified of sign accordingly.

3   TEST CASES

All the test cases investigated in this work consider the reference wind turbine NREL-5 MW with all technical details available in open-source [5]. The relevant characteristics are reported listed in Tab. 1.

Rotor radius, R	[m]	63
Peak power coefficient, $C_{P,max}$	[-]	0.482
Tip-Speed ratio at peak power coefficient, λ_*	[-]	7.55
Rated mechanical power P_r	[MW]	5.296
Rated generator torque τ_r	[N · m]	43,093.55
Maximum generator torque τ_{max}	[N · m]	47,402.91
Maximum generator torque rate $\dot{\tau}_{max}$	[N · m s^{-1}]	15,000
Proportional gain K_P at minimum blade-pitch setting,	[s]	0.01882
Integral gain K_I at minimum blade-pitch setting,	[-]	0.0086
Minimum blade-pitch setting, β_{min}	[deg]	0
Maximum blade-pitch setting, β_{max}	[deg]	90
Maximum absolute blade-pitch rate, $\dot{\beta}_{max}$	[deg s^{-1}]	8

Table 1: NREL 5-MW wind-turbine main characteristics, [5]

Two control problems are considered. The first is that of tracking the optimal Tip-speed

Ratio (TSR, λ) regardless of the turbine loads. The second is that of balancing the power production with the wind blade loads. The settings for these two problems are described in Section 3.1 and 3.2 respectively.

3.1 TSR tracking on synthetic wind data (Test Case A)

This test case aims at maximizing the harvested power and the power coefficients regardless of the wind loads. Therefore, the instantaneous reward function is

$$r_1(t) = -\frac{|\lambda - \lambda_*|}{\lambda_*} - \frac{|P_g - P_{*,g}|}{P_{*,g}}, \tag{5}$$

were $\lambda(t)$ is the instantaneous TSR, $\lambda_* = 7.55$ is the TSR maximizing the power coefficient C_p at the optimal pitch angle $\beta_* = 0°$ (cfr. figure 2b), $P_g(t)$ is the instantaneous extracted power and P_r is the rated power.

It is worth recalling that the optimality of λ_* is valid in steady conditions but not dynamic ones. In real conditions, where C_p could be measured, the combination of the two terms could allow the controller to explore the action space departing from λ^* if the associated loss in the first term is out-weighted by a gain in the second.

3.2 Loads mitigation (Test Case B)

The goal in this test case is to minimize the overall loads met by the wind turbine. It is recalled that to compute the loads, the augmented framework embedding BEM simulations is employed, as presented in Sect. 2. The instantaneous reward in this second test case is

$$r_2(t) = \langle \tilde{P}_g \rangle_T - ||\tilde{F}||_1 \tag{6}$$

where the $\langle \tilde{P}_g \rangle = P_{g,t}/P_{rated}$ weights the extracted power $(P_{g,t})$ on the rated power (P_r), while $||\tilde{F}||_1$ is the l_1 norm of the aerodynamic load in the hub reference system, as illustrated in Fig. 2a. The ˜ denotes normalization of the forces with respect to the average values produced by a PI controller over an episode.

4 RESULTS

We analyze the performances of the model-free controllers over a training period of 40 episodes, corresponding to 4000 s of physical time. We here focus on the best results achieved at the end of the training and we postpone the analysis of the learning curves to the extended version of this work.

4.1 TSR tracking on synthetic wind data

The numerical results of Test case A (reward function in Eq. (5)) are listed in Tab. 2. The table collects the relative gain/increase in the power extracted, defined as $\Delta P_g = (\langle P \rangle_{PI} - \langle P \rangle_{ML})/\langle P \rangle_{PI}$, where $\langle P \rangle_{ML}$ represents the mean power extracted in the simulation time by the control parametrization at hand, and the relative TSR tracking error, defined as $\Delta \langle \lambda \rangle = (\langle \lambda \rangle_{PI} - \langle \lambda \rangle_{ML})/\langle \lambda \rangle_{PI}$. Both quantities are averaged within an episode.

Method	$\langle P_g \rangle$ [MW]	$\langle \lambda \rangle$ [-]	$\langle \omega \rangle$ [rad s^{-1}]	$\Delta P_g \%$	$\Delta \langle \lambda \rangle \%$
PI	4.05	7.28	1.22	-	-
Linear Controller (BO)	4.80	7.61	1.32	+18.1 %	- 70.21 %
RL (DDPG)	4.95	7.85	1.36	+ 22.22 %	+11.1 %

Table 2: Results testcase A. Method indicates the parametrization of the control-law, $\langle P_g \rangle$ [MW] is the generated power and ω [rad s^{-1}] is the mean rotational speed. The last two columns compare the performances with the baseline (PID).

The two machine learning methods increase greatly the power extraction with respect to the PI controller while maintaining the rotational speed of the turbine closer to its rated value $\omega_r = 1.33$ rad s^{-1}. Interestingly, the results shows that the second term in Eq. (5) is more important than the first.

Figs. 3a,3b, 3c and 3d show the system trajectory in the state plane $\mathbf{s} = (u_\infty, \omega)$ during an episode. For plotting purposes, only one every fifty time steps is included. The markers are colored by the controller actions at the corresponding states. The figures on the top show the pitch actuation (β), while the figures one the bottom show the torque actuation (τ).

Interestingly, both controllers opt for a pitch angle between $-1° \leq \beta \leq 1°$, even though they are not constrained to do so. This is the pitch angle corresponding to the highest C_p, for the λ met by the operative trajectories. Concerning the BO torque actuation, the controller seeks to slow down the wind turbine when the tip speed ratio is greater than the rated value, whereas it reduces its actuation in the lower-right part of the plot, where $\lambda \approx 5.25$, to speed the rotor up. The RL acts similarly, but it surprisingly acts less in the upper part of the trajectory (corresponding to a larger tip-speed ratio than the rated value). This behaviour, which might be due to poor state space sampling, will be further investigated in future works.

4.2 Load mitigation

The numerical results of Test case B (reward function in (6)) are listed in Tab. 3. Both model-free controllers significantly reduce the loads, albeit at the cost of a significant reduction in the extracted power. This result is due to the relative weights introduced in the reward function– one could change these to set different priorities for the two objectives. By repeating the training with different relative weights between the terms in (6), one could identify an approximation of the Pareto front in the action space [19].

Remarkably, the linear control driven by the BO achieves better performances in terms of load reduction for a nearly equal power reduction. Although a more in-depth analysis is needed to assess the learning performances of both agents, this result shows that the larger model capacity (and complexity) of the DDPG agent might not pay off over a more straightforward linear control law (at least within the 40 episodes considered for the training).

Finally, Figure 4 shows the loads in each node (averaged over an episode) per unit length. The node numbering increases from the hub to the tip (cf. Figure 2a). The load distribution in the absence of control is shown for comparison, and it is clear that this increase from left to right (i.e. at a larger distance from the hub). In the absence of control, the load re-partition is extremely non-uniform, with most of the loads acting on the tip while the PI controllers

(a) BO pitching policy

(b) RL (DDPG) pitching policy

(c) BO generator torque policy
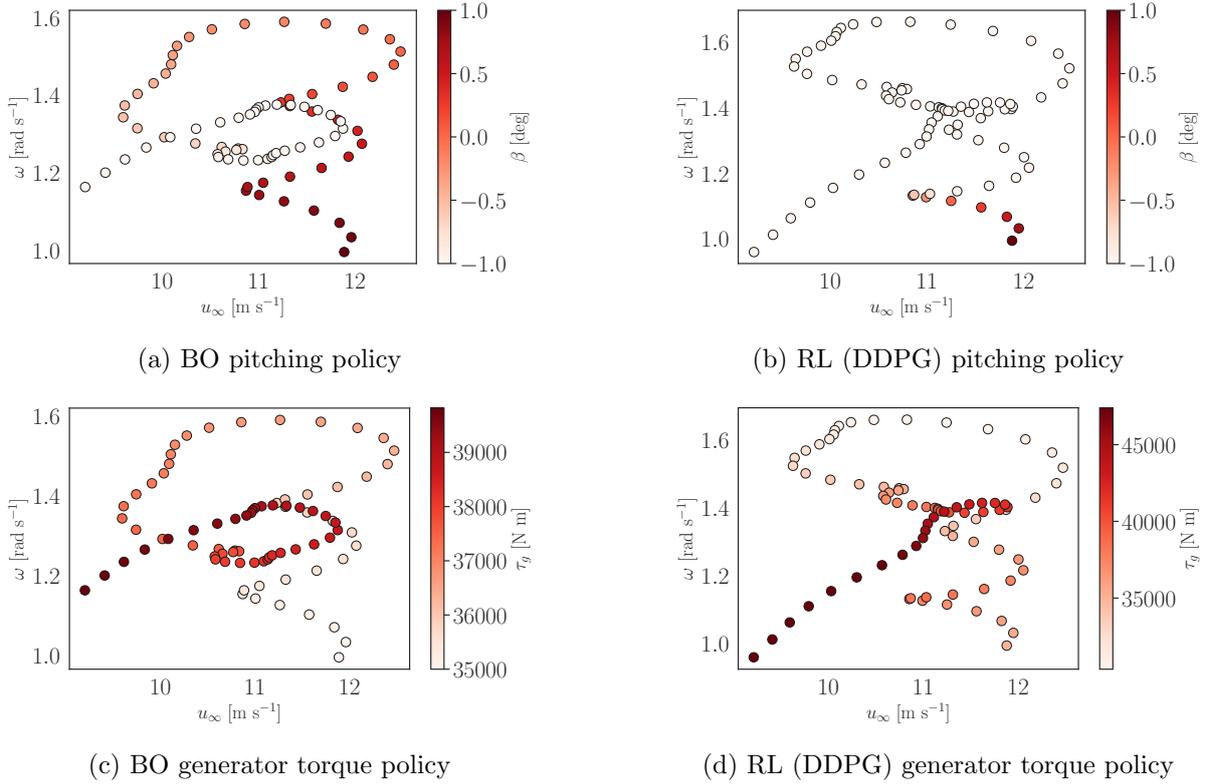
(d) RL (DDPG) generator torque policy

Figure 3: System trajectory and controller policy for the BO (on the left) and DDPG (on the right) algorithms. The first row refers to the pitch rate actuation while the second refers to the torque rate actuation.

Method	P_g [MW]	F_x [kN]	F_y [N]	F_z [N]	ΔF_x %	ΔF_y %	ΔF_z
PI	4.05	295.162	5.1E-02	3.0E-02	-	-	-
Linear Controller (BO)	2.8	81.82	4.8E-02	2.1E-02	-73%	-7%	-31%
RL (DDPG)	2.78	142.245	-2.8E-02	2.8E-02	-53.81 %	-44.38%	-8%

Table 3: Results Testcase B. Method indicates the parametrization of the control-law, $\langle P_g \rangle$ [MW] is the generated power, F_x, F_y and F_z are the forces met by the wind-turbine in the respective directions (in the hub frame of reference cfr. Fig. 2a), whereas the last three columns represent their variation against the baseline.

introduces a nearly triangular distribution.

On the other hand, the non-monotonic trend for the controlled conditions is due to a complex interplay between the time-varying accelerations produced by the controllers and the time-varying deflections of the blades. In the future, the reward function could be crafted to prescribe a specific load distribution and thus tailor the loadings to the structural design of the blades.
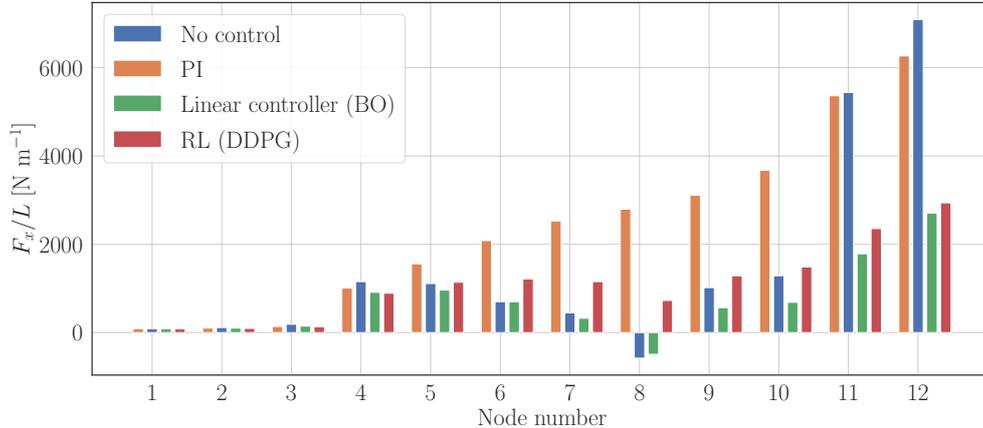
Figure 4: Loads per unit length on each blade node in case of no control (blue), baseline PI control (orange), linear BO controller (green) and nonlinear (DDPG) controller

5 CONCLUSIONS AND OUTLOOKS

This work presents two black-box machine learning strategies for the control of a wind turbine. These have been benchmarked against a classic PI controller in a numerical environment that reproduces the main features of the wind turbine power production and loading. The results show that these methods can outperform baseline (piecewise linearized) approaches and handle multiple objectives with minimal tuning. Because these controllers learn by trial and error and do not rely on (necessarily simplified) wind turbine models, they could offer new avenues for optimal wind energy extraction.

In the future, hybrid approaches will be considered as proposed in [12]. These could consist of two agents cooperating on two similar problems: (1) learning the control function and (2) identifying and tuning simplified models online. The controller could thus decide to operate in a model-based or a model-free approach and use one of the methods to improve the other. Both the proposed model-free and the hybrid methods will be implemented in a scaled wind turbine model in an experimental campaign in the L1 wind tunnel at the von Karman Institute.

Acknowledgements

REFERENCES

[1] Veers, P., Dykes, K., Lantz, E., Barth, S., Bottasso, C. L., Carlson, O., Clifton, A., Green, J., Green, P., Holttinen, H., Laird, D., Lehtomäki, V., Lundquist, J. K., Manwell, J., Marquis, M., Meneveau, C., Moriarty, P., Munduate, X., Muskulus, M., . . .

Wiser, R. (2019). Grand challenges in the science of wind energy. Science, 366(6464). https://doi.org/10.1126/science.aau2027

[2] van Kuik G A, et al. 2016 Long-term research challenges in wind energy - A research agenda by the European Academy of Wind Energy Wind Energy Science 1 1-39

[3] Ackermann, T. and S¨oder, L. (2002). An overview of wind energy-status 2002. Renewable and Sustainable Energy Reviews 6(1-2), 67–127.

[4] Gardner, P., Garrad, A., Jamieson, P., Snodin, H., and Tindal, A. (2003). Wind energy. The facts. Technical report, European Wind Energy Association (EWEA), Brussels, Belgium.

[5] Jonkman, J., Butterfield, S., Musial, W., Scott, G. (2009). Definition of a 5-MW reference wind turbine for offshore system development. Contract, February, 1–75.

[6] Fernando D. Bianchi, Ricardo J. Mantz, Hernán Battista. Wind Turbine Control Systems, Principles, Modelling and Gain Scheduling Design. Springer London (2007)

[7] Lio, W. H., Rossiter, J. A., Jones, B. L. (2014). A review on applications of model predictive control to wind turbines. 2014 UKACC International Conference on Control, CONTROL 2014 - Proceedings, July, 673–678. https://doi.org/10.1109/CONTROL.2014.6915220

[8] Laks, J. H., Pao, L. Y., Wright, A. D. (2009). Control of wind turbines: Past, present, and future. Proceedings of the American Control Conference, July, 2096–2103. https://doi.org/10.1109/ACC.2009.5160590

[9] Abbas, N. J., Zalkind, D. S., Pao, L., Wright, A. (2022). A reference open-source controller for fixed and floating offshore wind turbines. Wind Energy Science, 7(1), 53–73.

[10] Johnson, K. E., Fingersh, L. J., Balas, M. J., Pao, L. Y. (2004). Methods for increasing region 2 power capture on a variable speed hawt. Collection of ASME Wind Energy Symposium Technical Papers AIAA Aerospace Sciences Meeting and Exhibit, January 2014, 103–113.

[11] Archetti, Francesco and Candelieri, Antonio. Bayesian optimization and data science. Springer (2019).

[12] Pino, F., Schena, L., Rabault, J., Kuhnle, A., Mendez, M. (2022). Comparative analysis of machine learning methods for active flow control. arXiv preprint arXiv:2202.11664.

[13] Rasmussen, Carl Edward. Gaussian processes in machine learning. Summer school on machine learning. Springer (2003).

[14] Dykes, Katherine L and Damiani, Rick R and Graf, Peter A and Scott, George N and King, Ryan N and Guo, Yi and Quick, Julian and Sethuraman, Latha and Veers, Paul S and Ning, Andrew (2018). Wind Turbine Optimization with WISDEM.

[15] Jonkman, J. M., Hayman, G. J., Jonkman, B. J., Damiani, R. R. (2015). AeroDyn v15 User's Guide and Theory Manual. Renewable Energy, March, 46.

[16] Sutton, Richard S. and Barto, Andrew. Reinforcement Learning: An Introduction (2018). A Bradford Book

[17] Lillicrap, Timothy P. and Hunt, Jonathan J. and Pritzel, Alexander and Heess, Nicolas and Erez, Tom and Tassa, Yuval and Silver, David and Wierstra, Daan. Continuous control with deep reinforcement learning (2016). 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings.

[18] Jonkman, B. J., M. L. Buhl, J. (2006). NWTC Information Portal (TurbSim). NREL/TP-500-39797, 1–32.

[19] Costa, N. R., Lourenço, J. A., "Exploring Pareto Frontiers in the Response Surface Methodology", in G.-C. Yang, S.-I. Ao, L. Gelman, eds., Transactions on Engineering Technologies: World Congress on Engineering 2014 (Berlin/Heidelberg: Springer, 2015), pp. 399–412