



HAL
open science

Behavioral cloning for fatigue-oriented wind turbine optimal predictive control

David Collet, Mazen Alamir, Domenico Di Domenico, Guillaume Sabiron

► **To cite this version:**

David Collet, Mazen Alamir, Domenico Di Domenico, Guillaume Sabiron. Behavioral cloning for fatigue-oriented wind turbine optimal predictive control. IFAC-PapersOnLine, 2021, 54 (9), pp.590 - 595. 10.1016/j.ifacol.2021.06.121 . hal-03389034

HAL Id: hal-03389034

<https://ifp.hal.science/hal-03389034>

Submitted on 20 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Behavioral cloning for fatigue-oriented wind turbine optimal predictive control

David Collet^{*,**} Mazen Alamir^{**} Domenico Di Domenico^{*}
Guillaume Sabiron^{*}

^{*} IFPEN, Solaize, France (e-mail: david.collet@ifpen.fr,
domenico.didomenico@ifpen.fr, guillaume.sabiron@ifpen.fr).

^{**} GIPSA-Lab, CNRS, and University of Grenoble Alpes, Saint-Martin
d'Hères, France (e-mail: mazen.alamir@grenoble-inp.fr)

Abstract: As wind energy production keeps on increasing, it is necessary to optimize the fatigue/performance trade-off of operating wind turbines. However, considering fatigue directly in optimal control needs to be carefully done because its faithful model does not necessarily fit standard forms commonly required by general purpose solvers. It was recently shown that a signal variance and its induced fatigue can be statistically related. This suggests that a fatigue-related cost function can be expressed as a non-quadratic cost function, and used in an open-loop optimal control problem. However, it was also shown that such cost function becomes very efficient only for relatively long prediction horizons. This makes the on-line solution of the underlying open-loop optimal control problem computationally demanding and undermines the real-time applicability of associated MPC schemes. In this paper, a behavioral learning solution is proposed to imitate the optimal controller obtained through open-loop, long prediction horizon-based optimization. The behavioral cloning controller is compared to a finely tuned quadratic MPC regarding its ability to reduce fatigue. Preliminary results show that the proposed solution enables significant fatigue reduction on a broad range of realistic disturbances while being real-time implementable.

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Behavioral cloning, Optimal control, Fatigue, Wind turbines, Individual Pitch Control

1. INTRODUCTION

Global wind energy capacity has been growing exponentially for the last decade, increasing from 94 GW in 2007 to 591 GW in 2018, see (GWEC, 2016). World governments have set as objective during COP21 to maintain CO₂ emissions below 5.4×10^{12} kg/year. In order to achieve this goal, the wind energy industry is expected to expand even further (GWEC, 2016). This energetic transition requires a large economic investment and it is thus necessary to optimize Horizontal Axis Wind Turbine (HAWT) operation and maintenance cost.

Control of HAWT blade pitch angle can contribute to addressing this challenge. The main objectives of HAWT blade pitch control are to regulate output power and rotor speed while reducing mechanical fatigue. In the early works on blade pitch control, wind was assumed to be uniformly distributed over the rotor area and all the blades were pitched to the same angle. This technique is called Collective Pitch Control (CPC).

With recent increase in rotor diameter, this assumption is questionable. Aerodynamic forces on the blades fluctuate with the azimuth angle while the blade pitch angle remains constant, (Hansen, 2015). Therefore, by varying each blade pitch angle individually depending on its azimuth, blades fatigue loads can be alleviated. This technique is called Individual Pitch Control (IPC) (Bossanyi, 2003).

IPC is usually divided in two stages:

- (1) The CPC stage whose objective is to regulate the rotor rotational speed and power, while alleviating the fatigue loads on the tower.
- (2) The IPC stage that gives a differential pitch angle to each blade that adds to the collective pitch angle, in order to reduce the unbalanced loads on the rotor, contributing to the rotating components damage (i.e. blades, rotor bearing, blade bearings)

The goal of the IPC stage is essentially to optimize the trade-off between the fatigue damage of various components, in order to minimize the HAWT operating cost.

Optimal control is a natural choice to achieve such an objective. However, expressing this objective as an economic cost function for optimal control is a challenging task due to difficulties of fatigue damage modelling (Hammerum et al., 2007; Barradas-Berglind and Wisniewski, 2016). Fatigue damage is commonly quantified using the Palmgren-Miner fatigue theory that expresses it as a sum of damages caused by hysteresis load cycles (Palmgren, 1924). These cycles are counted using a RainFlow Counting (RFC) algorithm (Downing and Socie, 1982) which cannot be expressed as a simple algebraic function. Consequently, the way fatigue can be expressed as an explicit mathematical expression of the signals of interest remains an open topic

(Knudsen et al., 2015).

In (Collet et al., 2020a), a data-driven surrogate model relating quadratic norms (variance) and fatigue of given signals is used in order to approximate a fatigue trade-off objective cost function. This allows to obtain a global differentiable non-quadratic convex cost function, which is used to optimize the open-loop system input trajectory regarding an economic fatigue trade-off. In (Collet et al., 2020b), the sensitivity of this open-loop optimization in fatigue and CPU time is studied, and compared with an optimization using a finely tuned quadratic cost function. The latter paper shows that using this cost function in an open-loop optimization allows to obtain significant fatigue reduction, for long prediction horizons, greater than 10 to 100 seconds. However, the CPU time needed for such long prediction horizons, prevents the use of this optimization in Model Predictive Control (MPC).

Behavioral cloning (Pomerleau, 1991) is an area of machine learning that consists in imitating the behavior of an expert. The expert provides demonstrations of the optimal behavior, and from the data obtained, a machine learning regressor learns the policy that the controller must follow. In this paper, the expert is the open-loop optimization based on fatigue-oriented non quadratic cost function. In order to enhance the advantages of such a formulation in terms of fatigue reduction, without going through time consuming computations, a behavioral cloning solution is presented in order to imitate a controller which in closed loop would produce the open loop optimization optimum behavior. This behavioral cloning controller is then implemented in closed-loop in order to provide good approximate solutions to the minimization of the non-quadratic cost function on long prediction horizons.

The paper is organized as follows. In Section 2, the novel fatigue-oriented non quadratic cost function is introduced. In Section 3, the application framework is depicted along with the parametrization of a quadratic MPC to be tuned for comparison. Section 4 discusses the behavioral cloning concept and introduces the regressor that is used in the sequel. Afterwards, preliminary results on the comparison of closed-loop performances under the behavioral cloned controller on one hand and under the finely tuned quadratic MPC are depicted in Section 5. Eventually, a conclusion along with an outlook on the undergoing work are given in Section 6.

2. A FATIGUE-ORIENTED NON QUADRATIC COST FUNCTION

In this section, a fatigue-trade-off oriented cost function is first defined. Then the relationship between variance and fatigue is presented. Eventually the fatigue-oriented non quadratic cost function is derived.

2.1 Fatigue trade-off cost function

The HAWT IPC optimal control problem consists in optimizing the fatigue damage trade-off between various HAWT components. One way to express this objective through an economic cost function is to penalize each

component fatigue damage by its corresponding economic costs:

$$J_{\text{fat}}(\mathbf{y}) = \sum_{i=1}^n \pi_i D(\mathbf{y}_i) \quad (1)$$

where J_{fat} is the total economic cost of fatigue, \mathbf{y} is the system outputs trajectory, n is the number of HAWT components considered, $D(\cdot)$ is a function that yields the fatigue damage from a stress history, π_i and \mathbf{y}_i are the i^{th} component economic cost and output trajectory respectively.

2.2 Relationship between variance and fatigue

Using quadratic cost functions as objective in the formulation of optimization problems is usually desirable, as it is differentiable, convex, and makes the optimization relatively simple. Moreover, quadratic cost functions are generally suitable to represent energy cost in many areas.

In (Collet et al., 2020a), the relationship between variance, denoted by Var and fatigue damage D of stress histories is investigated. The variance is computed using its discrete expression:

$$\text{Var}(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^N z(t_k)^2 - \left(\frac{1}{N} \sum_{i=1}^N z(t_k) \right)^2 \quad (2)$$

where z is a signal whose trajectory on the time interval under consideration is denoted by \mathbf{z} , N is the discrete prediction horizon length and t_k is the k^{th} time instant. The ultimate load, denoted by L^{ult} , which is the highest load that the component can bear has a strong influence on the resulting damage. The Wöhler coefficients considered, denoted by m , are 4 and 10, corresponding to steel and glass fiber respectively. A linear regression is performed between variance and fatigue damage D time $L^{\text{ult}m}$, allowing to have the following estimation of fatigue damage for a load history \mathcal{H}_i :

$$L^{\text{ult}m} D(\mathcal{H}_i) \simeq e^b \text{Var}(\mathcal{H}_i)^a \quad (3)$$

where a and b served as the fitting parameters in the Logarithmic scale.

2.3 The fatigue-oriented cost function

Based on the above discussion and according to (1), the fatigue-induced cost function J_{fat} can be approximated by:

$$\tilde{J}_{\text{fat}}(\mathbf{y}) = \sum_{i=1}^n \frac{\pi_i e^{b_i}}{L_i^{\text{ult}m_i}} (\text{Var}(y_i))^{a_i} \quad (4)$$

where a_i and b_i are the regression's coefficients mentioned above and related to the output trajectory y_i . Note that, minimizing the variance of one signal is equivalent to minimizing its damage. However, as the relationship is nonlinear, minimizing a linear combination of variances is not equivalent to minimizing the linear combination of damages J_{fat} , defined in (1). Moreover, the cost function \tilde{J}_{fat} allows to approximate J_{fat} and better approach the minimum of J_{fat} in an optimization. The cost function \tilde{J}_{fat} is particularly interesting for optimal control problems because it is a convex differentiable algebraic function, as the variance is a convex function and if $a_i \geq 1 \forall i$. Therefore, there is no loss of convexity nor differentiability. Moreover the resulting problem can be solved directly using standard nonlinear programming solvers.

3. APPLICATION TO MPC DESIGN

This section first presents the system under interest, then the parametrization of a quadratic MPC is described. Eventually, the tuning procedure of the quadratic cost function is detailed. The reason for which we are interested in comparison with quadratic cost is that such costs are used in the majority of related works if not all.

3.1 The system dynamics

A HAWT is a multi-body system, disturbed by the wind, whose different bodies vibrations are coupled. It is a multi-inputs/multi-outputs system, that can be simulated using high fidelity nonlinear aero-elastic HAWT simulators, such as FAST (Jonkman et al., 2009). It can be noted that FAST allows to linearize the nonlinear equations governing the HAWT behavior for control purposes. In this study, the focus is set on the IPC stage of a HAWT blade pitch controller. As mentioned previously, the IPC stage objective is to reduce the fatigue of multiple HAWT components by giving differential pitch angles on each blade, to be added to the collective pitch angle. These differential pitch angles, which depend on azimuth angle denoted by ψ , can be obtained from a non-rotating reference frame by using the Coleman (Coleman and Feingold, 1957) or Multi-Blade Coordinate (MBC) transform (Bir, 2008):

$$\begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \\ \delta\theta_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\psi) & \sin(\psi) \\ \cos(\psi + \frac{2\pi}{3}) & \sin(\psi + \frac{2\pi}{3}) \\ \cos(\psi + \frac{4\pi}{3}) & \sin(\psi + \frac{4\pi}{3}) \end{bmatrix}}_{T(\psi)} \underbrace{\begin{bmatrix} \theta_{yaw} \\ \theta_{tilt} \end{bmatrix}}_{\Theta} \quad (5)$$

where $\delta\theta_i$ is the differential blade pitch angle of the i^{th} blade. θ_{yaw} and θ_{tilt} are respectively the yawing and tilting pitch angles.

FAST and its MBC module allows to obtain a linearization of the dynamic system, relating the hub-height wind velocity, denoted by v and, the yawing and tilting pitch angles, to the yawing and tilting out-of-plane blade root bending moments, denoted respectively by M_{yaw} and M_{tilt} . The model is linearized around an operating point, defined by its collective blade pitch angle θ_{col} , steady wind velocity v_0 and, yawing and tilting steady out-of-plane blade root bending moments, denoted respectively by M_{yaw}^0 and M_{tilt}^0 . The linearized model is expressed as follows:

$$\begin{aligned} \dot{x} &= Ax + B\Theta + B_d\delta v \\ \delta M &= Cx + D\Theta + D_d\delta v \end{aligned} \quad (6)$$

where $\delta M = [M_{yaw} - M_{yaw}^0, M_{tilt} - M_{tilt}^0]^T$ are the differential yawing and tilting out-of-plane blade root bending moments, $\Theta = [\theta_{yaw}, \theta_{tilt}]^T$, x is the state of the system, and $\delta v = v - v_0$ is the differential wind speed.

It is also considered that blade pitch actuators have the same first order dynamics, expressed as follows:

$$\tau\delta\dot{\theta}_i + \delta\theta_i = \delta\theta_i^{\text{sp}} \quad \forall i \in \{1, 2, 3\} \quad (7)$$

where Θ_i^{sp} is the setpoint angle for the i^{th} actuator and τ is the actuator characteristic time. Then the MBC transform must also be applied to (7), in order to extend the system expressed in (6). The actuators dynamic equation in the Coleman transform yields:

$$\dot{\Theta} = \Gamma\Theta + \Theta^{\text{sp}} \quad (8)$$

where $\Theta^{\text{sp}} = [\theta_{yaw}^{\text{sp}}, \theta_{tilt}^{\text{sp}}]^T$ and $\Gamma = \begin{bmatrix} -\frac{1}{\tau} & -\omega \\ \omega & -\frac{1}{\tau} \end{bmatrix}$ with ω the rotor rotational velocity. Then considering the state $\tilde{x} = [x^T, \Theta^T]^T$, it is possible to merge the systems (6) and (8) to get:

$$\begin{aligned} \dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{B}\Theta^{\text{sp}} + \tilde{B}_d\delta v \\ \tilde{y} &= \tilde{C}\tilde{x} + \tilde{D}\Theta^{\text{sp}} + \tilde{D}_d\delta v \end{aligned} \quad (9)$$

where $\tilde{y} = [\delta M^T, \Theta^T]^T$.

3.2 Quadratic MPC parametrization

The unconstrained quadratic MPC uses an open-loop optimization of the input trajectory on a prediction horizon of N_{MPC} time steps, whose first control action is fed to the system at each update of the MPC. The behavior of the closed-loop system controlled by the application of MPC depends on the ability to solve the open-loop optimization problem in real-time. The parameters of the optimization are defined in this section. In this paper, the updating period of the MPC is considered equal to the sampling period of the system. The quadratic cost function used in the MPC open-loop optimization, denoted by J_{quad} , is expressed as follows:

$$J_{\text{quad}}(\Theta_{\text{MPC}}^{\text{sp}} | \mathbf{v}_{\text{MPC}}, \tilde{x}_0) = \sum_{k=1}^{N_{\text{MPC}}} \tilde{y}_k(\Theta_{\text{MPC}}^{\text{sp}}, \mathbf{v}_{\text{MPC}}, \tilde{x}_0)^T Q \tilde{y}_k(\Theta_{\text{MPC}}^{\text{sp}}, \mathbf{v}_{\text{MPC}}, \tilde{x}_0) + [\Theta_{\text{MPC}}^{\text{sp}}]_k^T R [\Theta_{\text{MPC}}^{\text{sp}}]_k \quad (10)$$

where $\tilde{y}_k(\Theta_{\text{MPC}}^{\text{sp}}, \mathbf{v}_{\text{MPC}}, \tilde{x}_0)$, $\Theta_{\text{MPC}}^{\text{sp}}$ and \mathbf{v}_{MPC} are respectively the output vector at instant k , inputs and disturbance trajectories over the MPC horizon prediction. The index k represents the k^{th} time instant and \tilde{x}_0 is the initial state of the MPC open-loop optimization. $Q \in \mathbb{R}^n$ and $R \in \mathbb{R}^p$ are semi-definite and definite positive matrices respectively. p and n are respectively the system number of inputs and outputs. It should be noticed that $\tilde{\mathbf{y}}_{\text{MPC}}$ is therefore a function of $\Theta_{\text{MPC}}^{\text{sp}}$, \mathbf{v}_{MPC} and \tilde{x}_0 , driven by the dynamic system (9).

The unconstrained open-loop optimization problem is expressed as follows:

$$\min_{\Theta_{\text{MPC}}^{\text{sp}}} J_{\text{quad}}(\Theta_{\text{MPC}}^{\text{sp}} | \mathbf{v}_{\text{MPC}}, \tilde{x}_0) \quad (11)$$

for given \mathbf{v}_{MPC} and \tilde{x}_0 .

In order to design a controller that minimizes J_{fat} , it is required to tune the matrices Q and R . These matrices involve respectively $\frac{p(p+1)}{2}$ and $\frac{n(n+1)}{2}$ variables. For the system (9), this corresponds to 13 variables to tune. This can be computationally expensive given the computational burden involved in each function's evaluation.

Fortunately, in the specific case of a three bladed wind turbine rotor, each blade is independent from each other. Therefore only the diagonal terms of the matrices need to be considered. Moreover, it can also be assumed that as all the blades and blade pitch actuators are identical, the same weights should be used for each blade pitch actuator and its corresponding blade root bending moments.

Therefore, for the system presented in (9), the Q and R weight matrices can be parametrized as follows:

$$Q(\rho) = \begin{pmatrix} \rho I_2 & 0 \\ 0 & I_2 \end{pmatrix}, \quad R = \varepsilon I_2 \quad (12)$$

where $\varepsilon = \min(\rho, 1) \times 10^{-3} \ll \min(\rho, 1)$ is constant, such that the variations of blade pitch angles set-points do not impact the fatigue trade-off.

3.3 Parametrized quadratic MPC tuning procedure

Using the above simplifications, the parametrized cost depends only on the parameter ρ . The resulting MPC can be run in closed-loop with the system (9), for given initial state \tilde{x}_0 and disturbance trajectory, denoted by \mathbf{v} . This closed-loop simulation yields an output trajectory, which can be fed to $J_{\text{fat}}(\cdot)$ in order to estimate its fatigue cost. Therefore, each triplet $(\rho, \tilde{x}_0, \mathbf{v})$ corresponds to a fatigue cost, denoted by $J_{\text{fat}}(\rho, \tilde{x}_0, \mathbf{v})$. The tuning procedure consists in finding the parameter ρ that minimizes the expectation of the fatigue cost for a variety of \tilde{x}_0 and \mathbf{v} :

$$\min_{\rho} \mathbb{E}(J_{\text{fat}}(\rho, \tilde{x}_0, \mathbf{v})) \quad (13)$$

where \tilde{x}_0 and \mathbf{v} are drawn from a given relevant distribution to be carefully chosen.

4. BEHAVIORAL CLONING

Behavioral cloning consists in fitting a matching function (via machine learning tool for instance) that imitates an expert behavior. The features input in the function are the system state at a given time instant t_{k-1} , and the target values are the inputs to be fed into the system at the next time instant t_k . In this section, the expert is first introduced, then the data used for the machine learning fit are presented and eventually the regressor used for the fitting is depicted.

4.1 The expert

The expert, whose behavior must be imitated, is the result of the following open-loop optimization problem using the non quadratic cost function defined in (4):

$$P(\tilde{x}_0, \mathbf{v}) : \min_{\Theta}^{sp} \tilde{J}_{\text{fat}}(\tilde{\mathbf{y}}(\tilde{x}_0, \mathbf{v}, \Theta^{sp})) \quad (14)$$

for given \tilde{x}_0 and \mathbf{v} . This optimization yields optimal inputs and outputs trajectories, denoted respectively by Θ^{sp*} and $\tilde{\mathbf{y}}^*$, for given pair of disturbance trajectory \mathbf{v} and initial state \tilde{x}_0 .

4.2 Input features and target values spaces

Let us denote by X_{k-1} the following stored data that is available at instant t_{k-1} :

$$X_{k-1} = [\delta M_{k-M_x}^{*T}, \dots, \delta M_{k-1}^{*T}, \Theta_{k-M_x}^{sp*T}, \dots, \Theta_{k-1}^{sp*T}, v_{k-M_x}^T, \dots, v_{k-1}^T]^T \quad (15)$$

where M_x is the number of previous time steps used for the state estimation.

It should be noticed that only the outputs provided by the system (6), which are assumed to be measurable, are

considered as outputs for state estimation. In order to have the same amount of information as the MPC does, the information of the disturbance on the N_{MPC} time steps is given, denoted by V_k , such that:

$$V_k = [v_k, \dots, v_{k+N_{\text{MPC}}}] \quad (16)$$

The target values Y_k are thus k^{th} action in the optimal sequence of inputs $\Theta^{sp*} := [\Theta_0^{sp*}, \Theta_1^{sp*}, \dots, \Theta_N^{sp*}]$, that is to say:

$$Y_k = \Theta_k^{sp*} \quad (17)$$

4.3 The regressor

The regressor must fit a function f relating (X_{k-1}, V_k) to Y_k . After data generation, N_d triplets (X_{k-1}, V_k, Y_k) are obtained. The dataset \mathcal{D} is defined as:

$$\mathcal{D} = \left\{ (X^{(i)}, V^{(i)}, Y^{(i)}) \mid \forall i \in [1, \dots, N_d] \right\} \quad (18)$$

The regression is split in two steps: first a linear regression is fitted on the data, whose resulting model is denoted by f_L , then a nonlinear regression, whose resulting model is denoted by f_{NL} is fitted on the residual, denoted by ε , such that:

$$\varepsilon^{(i)} = Y^{(i)} - f_L(X^{(i)}, V^{(i)}) \quad \forall i \in [1, \dots, N_d] \quad (19)$$

Many nonlinear regressors have been tested, but due to the lack of space, only the regressor showing the best result is presented. This regressor is the gradient boosting decision trees, (Friedman et al., 2001). Gradient boosting decision trees is an ensemble method where each individual decision tree is called a weak learner. The weak learners are fitted successively on the residual of the previous predictions. Let Ξ_j be the vector of parameters of the j^{th} weak learner. Ξ_j is the result of the following optimization:

$$\min_{\Xi_j} \sum_{i=1}^{N_d} \|\varepsilon_{j-1}^{(i)} - \sum_{s=1}^j g(X^{(i)}, V^{(i)}, \Xi_s)\|_2^2 \quad (20)$$

where $g(\cdot)$ is a function that yields the decision tree output, taking as inputs the features and the parameters of the decision trees. Ξ_s for $s < j$ are fitted previously, beginning with Ξ_1 . $\varepsilon_{j-1}^{(i)}$ is the $j-1^{\text{th}}$ residual defined as follows:

$$\varepsilon_{j-1}^{(i)} = Y^{(i)} - \sum_{s=1}^{j-1} g(X^{(i)}, V^{(i)}, \Xi_s) \quad (21)$$

and $\varepsilon_0^{(i)} = \varepsilon^{(i)} \quad \forall i \in [1, \dots, N_d]$. The final prediction for N_r weak learners is the sum of the weak learners results:

$$f_{\text{NL}}(X^{(i)}, V^{(i)}) = \sum_{j=1}^{N_r} g(X^{(i)}, V^{(i)}, \Xi_j) \quad (22)$$

The parameters of the nonlinear regressor are thus the maximum depth of the decision trees and the number of weak learners considered. To prevent overfitting, an early stopping feature is implemented, such that the model stops adding weak learners when the cross validation scores stop increasing.

Eventually, for closed-loop implementation, the state is obtained by stacking the outputs and inputs values of the M_x past time steps into X_{k-1} , and the information on the future disturbance is assumed to be exact. In real life, this information could be estimated using LiDAR measurements, however with uncertainties. The two main

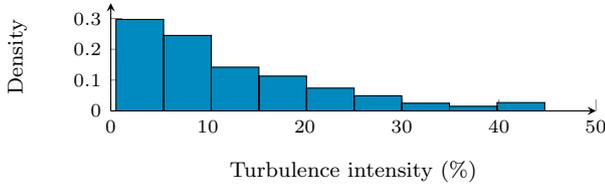


Fig. 1. Histogram of the turbulence intensities used to generate the stress histories and disturbances.

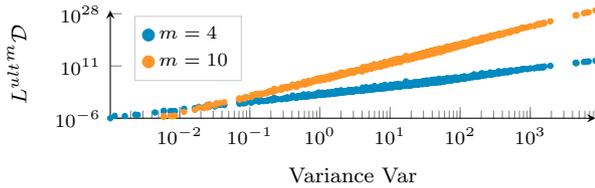


Fig. 2. Relation between variance Var and damage \mathcal{D} in the logarithmic space, for two values of the Wöhler coefficient.

drawbacks of this method are that there is no proof of stability of the controller for a given regressor and it can only be efficient in its learning space. However in real life, it could be possible to add a safety feature allowing to detect when the behavioral cloning is out of its learning space, in order to switch on a safe controller ensuring stability.

5. RESULTS

For both open-loop optimizations and closed-loop simulations, 200 seconds long winds with Kaimal spectrum are generated with TurbSim (see (Burton et al., 2011) and (Kelley and Jonkman, 2013)). All winds have an average speed of $12\text{m}\cdot\text{s}^{-1}$, and between winds the turbulence intensity is varied according to the distribution shown in Fig. 1. Concerning the initial state, \tilde{x}_0 is always taken to be the origin.

5.1 Variance and fatigue damage regression

The stress histories considered for variance and fatigue estimations are the disturbance generated for open-loop optimizations and closed-loop simulations. This set of stress histories is chosen because the dynamics of the disturbances are slower than the one of the system. Therefore its outputs will be strongly influenced by the disturbance spectrum. In Fig. 2, for each stress history \mathcal{H}_i , a couple $(\text{Var}(\mathcal{H}_i), \mathcal{D}(\mathcal{H}_i))$ is obtained and plotted in the Logarithmic space. It can be seen that there is a linear relationship between the logarithms of Var and \mathcal{D} . Using a linear regression, pairs of a linear coefficient and intercept (a, b) are obtained. This pairs are equal to $(2.19, 5.37)$ and $(4.86, 12.6)$ for Wöhler coefficients of 4 and 10 respectively.

5.2 Parametrized MPC

Recall that the control objective is to optimize a trade-off between the fatigue of the system outputs, corresponding to the rotor bearing, shaft and blade pitch actuators. The parameters used for fatigue estimation, used in J_{fat} and \tilde{J}_{fat} , defined in (1) and (4) respectively, are carefully

chosen. In order to yield a realistic fatigue trade-off and are summarized in Table 1.

Table 1. Summary of the parameters used for the fatigue estimation.

| Output | L_i^{ult} | π_i | m_i |
|--------|-----------------------|---------|-------|
| y_1 | 3×10^3 | 10^3 | 10 |
| y_2 | 3×10^3 | 10^3 | 10 |
| y_3 | 6.98×10^{-1} | 1 | 4 |
| y_4 | 6.98×10^{-1} | 1 | 4 |

Following the results of (Collet et al., 2020b), the sampling time considered for the optimization is 0.1 second and $N_{\text{MPC}} = 20$. In order to estimate the expectation of the fatigue cost, 100 winds with Kaimal spectrum are sampled. The value of ρ obtained is 5×10^{-2} .

5.3 Behavioral cloning regression

For data generation, 1000 winds are sampled for open-loop optimizations and among each resulting time series, 300 time instants are drawn to stack X_{k-1} , V_k and Y_k . The data coming from the 750 first winds, is used to train and test the regressors, while the data coming from the 250 last winds is used for the closed-loop validation. For the gradient boosting, 3 and 4 decision trees of maximum depth 6 are used, for the input 1 and 2 respectively. This leads to total numbers of 371 and 474 leaves for a total number of 169086 samples.

5.4 closed-loop validation

In this paper, the final goal is not to fit a regressor that predicts perfectly the values of the inputs from the current states, but to reduce efficiently the fatigue of closed loop simulation using the behavioral cloning controller. To assess the ability of the behavioral cloning to optimize the fatigue cost, the system controlled by the behavioral cloning is thus compared to the one controlled by the finely tuned parametrized Quadratic MPC. In Fig. 3, the times series of the system outputs are plotted for the fatigue-oriented open-loop optimization and, the system controlled by the behavioral cloning controller and the finely tuned Quadratic MPC. It is interesting to note that the time to predict the future input with the behavioral cloning regressor takes less than 0.01 second, which makes the behavioral cloning controller real-time implementable. It should be noticed that the systems controlled by the behavioral cloning and the open-loop optimization stay quite close from each other during the whole simulation. Regarding the finely-tuned Quadratic MPC, it can be seen that for the 2nd and 3rd outputs, the difference from the open-loop optimization and the behavioral cloning is quite significant.

A justice of the peace for controller performances is the fatigue cost on closed-loop simulation. Therefore, let be $J_{\text{fat}}^{\text{REG}}$ the fatigue cost of the closed-loop system controlled by the behavioral cloning and $J_{\text{fat}}^{\text{QMPC}}$ the cost of the one controlled by finely tuned Quadratic MPC. In Fig. 4 is plotted the scatter plot of $J_{\text{fat}}^{\text{QMPC}}$ and $J_{\text{fat}}^{\text{REG}}/J_{\text{fat}}^{\text{QMPC}}$ and in Fig. 5 is plotted the cumulative probability distribution of $J_{\text{fat}}^{\text{REG}}/J_{\text{fat}}^{\text{QMPC}}$. It can be seen that the global trend

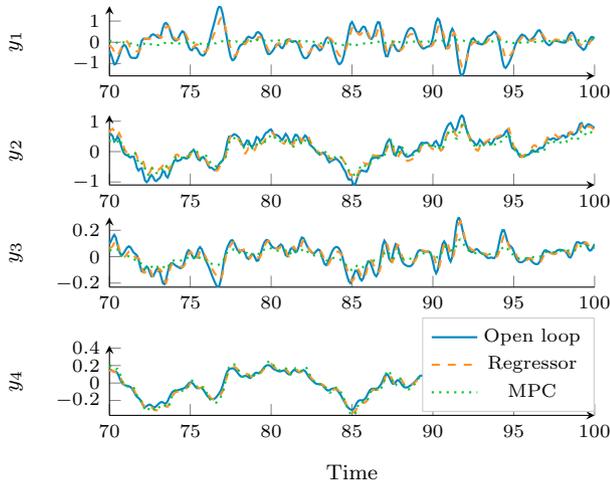


Fig. 3. Time series of the system outputs for the open-loop optimization with the non-quadratic cost function, the behavioral cloning and the finely tuned quadratic MPC.

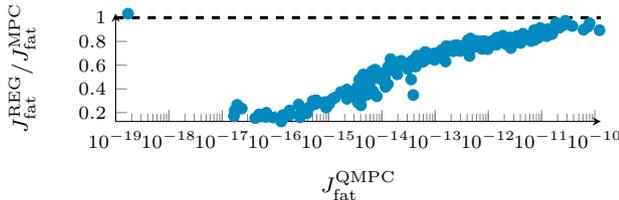


Fig. 4. Scatter plot of J_{fat}^{QMPC} and $J_{fat}^{REG}/J_{fat}^{QMPC}$.

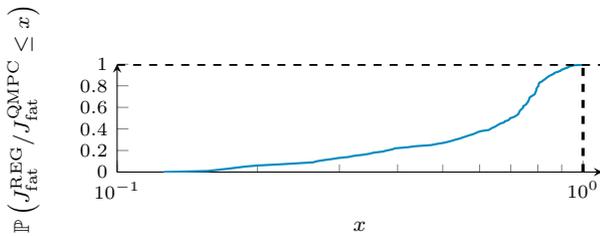


Fig. 5. Cumulative density of the behavioral cloning fatigue cost relative to the Quadratic MPC fatigue cost.

is that the behavioral cloning yields lower fatigue than the Quadratic MPC as $J_{fat}^{REG}/J_{fat}^{QMPC} \leq 1$ in more than 99% of the cases. It should be outlined that on average, $J_{fat}^{REG}/J_{fat}^{QMPC}$ equals 0.62, while its median is 0.70. Moreover, the ratio of the averages of J_{fat}^{REG} and J_{fat}^{QMPC} , which corresponds to the fatigue reduction expectation for this wind distribution is 0.89.

It is important to note that those results depend a lot on the wind distribution considered, and that perfect knowledge of the future disturbance and system are assumed, which is also true for the Quadratic MPC.

6. CONCLUSION AND PERSPECTIVES

This paper presents a behavioral cloning solution using the results of a fatigue-oriented open-loop optimization, for real-time implementation as a closed-loop controller. The closed-loop system controlled by the behavioral cloning is compared to a finely tuned parametrized quadratic MPC on their ability to reduce wind turbine fatigue cost. Even

though the parametrization of the MPC limits its potential and there are few assumptions such as perfect knowledge of the system and future disturbances, the behavioral cloning solution and the fatigue oriented non quadratic cost function show a great potential in fatigue reduction. Therefore, this is encouraging to pursue the study and try to reach the limits of this cost function. Future studies will be about its sensitivity to model and disturbances uncertainties. Eventually this kind of controller should be implemented on a nonlinear HAWT simulator such as FAST.

REFERENCES

- Barradas-Berglind, J. and Wisniewski, R. (2016). Representation of fatigue for wind turbine control. *Wind Energy*, 19, 2189–2203.
- Bir, G. (2008). Multi-blade coordinate transformation and its application to wind turbine analysis. In *Proc. 46th AIAA aerospace sciences meeting and exhibit*, 1300.
- Bossanyi, E. (2003). Individual blade pitch control for load reduction. *Wind Energy*, 6, 119–128.
- Burton, T., Jenkins, N., Sharpe, D., and Bossanyi, E. (2011). *Wind energy handbook*. John Wiley & Sons.
- Coleman, R.P. and Feingold, A.M. (1957). Theory of self-excited mechanical oscillations of helicopter rotors with hinged blades. *National Advisory Committee for Aeronautics*.
- Collet, D., Alamir, M., Di Domenico, D., and Sabiron, G. (2020a). A fatigue-oriented cost function for optimal individual pitch control of wind turbines. Unpublished, submitted to IFAC 2020.
- Collet, D., Alamir, M., Di Domenico, D., and Sabiron, G. (2020b). Non quadratic smooth model of fatigue for optimal fatigue-oriented individual pitch control. Unpublished, submitted to TORQUE 2020.
- Downing, S.D. and Socie, D. (1982). Simple rainflow counting algorithms. *International journal of fatigue*, 4, 31–40.
- Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA.
- GWEC (2016). Global wind energy outlook. Technical report, Global Wind Energy Council.
- Hammerum, K., Brath, P., and Poulsen, N.K. (2007). A fatigue approach to wind turbine control. *Journal of Physics: Conference Series*, 75, 012081.
- Hansen, M.O. (2015). *Aerodynamics of wind turbines*. Routledge.
- Jonkman, J., Butterfield, S., Musial, W., and Scott, G. (2009). Definition of a 5-MW reference wind turbine for offshore system development. Technical report, National Renewable Energy Laboratory.
- Kelley, N. and Jonkman, B. (2013). NWTC computer-aided engineering tools (Turbsim). *Last Modified*, 442.
- Knudsen, T., Bak, T., and Svenstrup, M. (2015). Survey of wind farm control—power and fatigue optimization. *Wind Energy*, 18, 1333–1351.
- Palmgren, A. (1924). Die lebensdauer von kugellagern. *Zeitschrift des Vereines Duetsher Ingenieure*, 68, 339.
- Pomerleau, D.A. (1991). Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1), 88–97.