



# Non quadratic smooth model of fatigue for optimal fatigue-oriented individual pitch control.

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

## ► To cite this version:

David Collet, Mazen Alamir, Guillaume Sabiron, Domenico Di Domenico. Non quadratic smooth model of fatigue for optimal fatigue-oriented individual pitch control.. Journal of Physics: Conference Series, 2020, 1618, pp.022004. 10.1088/1742-6596/1618/2/022004 . hal-03010303

**HAL Id: hal-03010303**

**<https://ifp.hal.science/hal-03010303>**

Submitted on 17 Nov 2020

**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 4.0 International License

PAPER • OPEN ACCESS

## Non quadratic smooth model of fatigue for optimal fatigue-oriented individual pitch control

To cite this article: D Collet *et al* 2020 *J. Phys.: Conf. Ser.* **1618** 022004

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

# Non quadratic smooth model of fatigue for optimal fatigue-oriented individual pitch control

D Collet<sup>1,2</sup>, M Alamir<sup>2</sup>, D Di Domenico<sup>1</sup>, G Sabiron<sup>1</sup>

<sup>1</sup> IFPEN, Rond-point de l'échangeur de Solaize, Solaize, France

<sup>2</sup> GIPSA-Lab, CNRS, and University of Grenoble Alpes, 11 rue des Mathématiques, Saint-Martin-d'Hères, France

E-mail: david.collet@ifpen.fr, mazen.alamir@grenoble-inp.fr, domenico.didomenico@ifpen.fr, guillaume.sabiron@ifpen.fr

**Abstract.** Operation and maintenance cost of wind turbine is intimately related to fatigue damage. However, considering fatigue directly in an optimization needs to be carefully done because its faithful model does not fit standard forms. Recent results have shown that the fatigue damage of a signal estimated using fatigue theory is non-linearly related to its corresponding variance. Therefore, this relationship is used to design a convex cost function approximating fatigue. In this paper, the sensitivity of open-loop optimizations using this fatigue-oriented non-quadratic cost function to the time horizon considered is studied, in terms of both fatigue and computational cost. Moreover, in order to alleviate the undesirable effects of the computational cost, an efficient fixed-point iteration-based way to optimize this fatigue-oriented cost function is presented. Results show that the fatigue-oriented cost function allows to obtain sensitive fatigue reduction compared to optimizations using a family of classical quadratic cost functions, especially for long optimization horizons. It is eventually, shown that the prediction horizon length required for the non-quadratic criterion to be efficient in an MPC makes its use prohibitive in such framework. However, prospects of reaping the benefits of the fatigue-oriented optimization using imitation learning are given.

## 1. Introduction

The economic profit that an Horizontal Axis Wind Turbine (HAWT) owner makes is mainly driven by the difference between gains, e.g. energy sold, and losses, e.g. Operation and Maintenance (O&M) cost [1]. O&M cost can be approximated as a weighted sum of the prices of replacement and mechanical fatigue damage of given HAWT components [2], as fatigue damage is equal to 1 when a component is broken. This weighted sum is also called fatigue trade-off cost. Therefore, in order to maximize HAWT economic profit, it is important to optimize this fatigue trade-off. Dedicated control of HAWT blade pitch angle can contribute to this cost reduction.

The main objectives of HAWT blade pitch control are to regulate output power and rotor speed while reducing mechanical fatigue. In the early work on blade pitch control, wind was assumed to be uniformly distributed over the rotor area. Therefore, 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 becomes questionable. Aerodynamic forces on the blades fluctuate with the azimuth angle while the blade pitch angle remains constant [3]. 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) [4]. IPC is usually divided in two stages:

- (i) The CPC stage whose objective is to regulate the rotor rotational speed and power, while alleviating the fatigue loads on the tower.
- (ii) The IPC stage that gives a differential pitch angle on each blade, added 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)

Therefore, the goal of the IPC stage is essentially to optimize the fatigue damage trade-off. Optimal control is a good solution to achieve this objective. However, expressing this objective as an economic cost function for optimal control is a challenging task due to the fatigue damage estimation [5, 6]. Fatigue damage is quantified using the Palmgren-Miner fatigue theory, where it is expressed as a sum of damages caused by hysteresis load cycles [7]. These cycles are counted using a RainFlow Counting (RFC) algorithm [8] which cannot be expressed as a simple algebraic function. Consideration of fatigue as an explicit control objective remains an open topic [9]. To our best knowledge, it is possible to find in the literature five methods considering directly fatigue as an objective cost function for HAWT control. In [5], the spectral properties of the stress history are used to design an optimal feedback controller. In [2], a data driven surrogate model is designed to select the best controller among several candidates. In [10], a quadratic cost function parameters are varied to match an on-line estimated economic fatigue cost function. In [1], a sequential optimization is made where the cost function parameters are iteratively modified between gradient steps, based on RFC. Only the two last methods are used to design an MPC controller, as their approach is temporal. In [11], several aspects of the previous methods are bridged together. A data-driven surrogate model relating quadratic norms (variance) and fatigue of given signals is used in order to approximate the fatigue trade-off cost with a non-quadratic cost function. This approximated objective function is used to optimize the system input trajectory during an open-loop optimization.

This paper is organized as follows:

- In Section 2, the equations governing the rotor dynamics of an HAWT are presented.
- In Section 3 the parameterization of a quadratic cost function used as benchmark, which was firstly presented in [11], is depicted.
- In Section 4, the novel fatigue-oriented cost function presented in [11] is introduced.
- In Section 5, an effective optimization method for optimal control using the presented fatigue-oriented cost function is detailed, involving a fixed-point method and a decomposition of the optimization horizon.
- In Section 6, performance of an optimization using the fatigue-oriented cost function is compared to one using a parameterized quadratic cost function, along with suggestions suggestions for closed-loop implementation.
- In Section 7, a conclusion and an outlook on the undergoing work are given.

## 2. Presentation of the system: An HAWT rotor in MBC coordinates

This section presents the system under interest, which is a multi-body HAWT 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 [12]. 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 an 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 can be obtained from a non-rotating reference frame by using the Coleman [13] or Multi-Blade Coordinate (MBC) transform [14].

FAST and its MBC module allows to obtain a linearization of the dynamics around an operating point, relating the hub-height wind velocity, denoted by  $v$  and, the yawing and tilting pitch angles, denoted respectively by  $\theta_{yaw}$  and  $\theta_{tilt}$ , to the yawing and tilting out-of-plane blade root bending moments, denoted respectively by  $M_{yaw}$  and  $M_{tilt}$ . The operating point is defined by its steady wind velocity  $v_0$ , yawing and tilting steady out-of-plane blade root bending moments, denoted respectively by  $M_{yaw}^0$  and  $M_{tilt}^0$ . Let  $\delta M = [M_{yaw} - M_{yaw}^0, M_{tilt} - M_{tilt}^0]^T$  be the differential yawing and tilting out-of-plane blade root bending moments, moreover  $\Theta = [\theta_{yaw}, \theta_{tilt}]^T$ ,  $x \in \mathbb{R}^6$  is the state of the system corresponding to the blades first flapwise mode, and  $\delta v = v - v_0$  is the differential wind speed. By augmenting the system obtained from FAST linearization and MBC modules with the actuators dynamics modeled by a first order low-pass filter, similar to the one presented in [11], the following Linear Time Invariant (LTI) system can be obtained:

$$\dot{x} = Ax + Bu + B_d \delta v \quad (1a)$$

$$y = Cx + Du + D_d \delta v \quad (1b)$$

where  $y = [\delta M^T, \Theta^T]^T$ ,  $u = [\theta_{yaw}^{sp}, \theta_{tilt}^{sp}]^T$ ,  $\theta_{yaw}^{sp}$  and  $\theta_{tilt}^{sp}$  denote respectively the yawing and tilting blade pitch angles to be given to the actuators via the inverse MBC transform.  $A$ ,  $B$ ,  $B_d$ ,  $C$ ,  $D$  and  $D_d$  are matrices of appropriate dimensions allowing to model the dynamics of the system. It should be noticed that in this paper, it is assumed that the system is not restricted by any constraints whatsoever, or that it cannot reach its constraints in the operational range it is working on. This can be forced by appropriate choice of the weighting coefficient and checked a posteriori. In the sequel, in order to alleviate notations  $\delta v := v$ .

### 3. Parameterized quadratic cost function benchmark

Quadratic cost functions are widely used in the formulation of fatigue-oriented open-loop optimal control problems. This is the reason why such cost functions are used as a benchmark for comparison with other open-loop formulations. The tuning of a quadratic cost functions can be computationally expensive and might be sub-optimal. However, as far as the tuning of a quadratic structure is concerned, for the specific case of HAWT rotors, it is possible to use this knowledge in order to derive a low-dimensional parameterization of the quadratic cost function weights.

#### 3.1. Quadratic cost function parameterization

In [11], a parameterization of a quadratic cost function for the specific problem of HAWT fatigue alleviation with IPC is described. The quadratic cost function is denoted by  $J_{quad}$  and parameterized by  $\rho$ :

$$J_{quad}(\mathbf{u}|\mathbf{v}, x_0, \rho) = \sum_{k=1}^N y_k(\mathbf{u}, \mathbf{v}, x_0)^T Q(\rho) y_k(\mathbf{u}, \mathbf{v}, x_0) + u_k^T R(\rho) u_k \quad (2)$$

where  $y_k(\mathbf{u}, \mathbf{v}, x_0)$ ,  $\mathbf{u}$  and  $\mathbf{v}$  are respectively the output vector at instant  $k$ , inputs and disturbance trajectories over the optimization horizon. The index  $k$  represents the  $k^{\text{th}}$  time instant,  $x_0$  is the initial state of the open-loop optimization and  $N$  is the number of prediction steps considered in the optimization horizon.  $Q \in \mathbb{R}^4$  and  $R \in \mathbb{R}^2$  are respectively semi-definite and definite positive matrices, parameterized by  $\rho > 0$ , such that  $Q(\rho) = \text{diag}([\rho, \rho, 1, 1])$  and  $R(\rho) = \min(\rho, 1) \times 10^{-3} I_2$  where  $I_n$  is the identity matrix of dimension  $n$ . It should be noticed

that  $\mathbf{y}$  is therefore a function of  $\mathbf{u}$ ,  $\mathbf{v}$  and  $x_0$ , driven by the dynamic system (1). Moreover, this parameterization is designed such that the variations of blade pitch angles set-points do not impact the fatigue trade-off.

### 3.2. Parameterized quadratic cost function tuning procedure

The parameterized quadratic cost function is thus used to solve the following unconstrained open-loop optimal control problem:

$$\min_{\mathbf{u}} J_{\text{quad}}(\mathbf{u}|\mathbf{v}, x_0) \quad (3)$$

for given  $\mathbf{v}$  and  $x_0$ . This open-loop optimization yields an output trajectory, which can be fed to the fatigue cost function denoted by  $J_{\text{fat}}(\cdot)$  in order to estimate the output trajectory fatigue cost. Therefore, each triplet  $(\rho, x_0, \mathbf{v})$  corresponds to a fatigue cost, denoted by  $J_{\text{fat}}(\rho, x_0, \mathbf{v})$ . The tuning procedure consists in finding the parameter  $\rho$  that minimizes the fatigue cost for given  $x_0$  and  $\mathbf{v}$ :

$$\min_{\rho} J_{\text{fat}}(\rho, x_0, \mathbf{v}) \quad (4)$$

where  $x_0$  and  $\mathbf{v}$  are randomly drawn from a given relevant distribution to be carefully chosen.

## 4. The novel fatigue-oriented cost function

In [2], the fatigue cost function  $J_{\text{fat}}$  is defined to be a weighed sum of the damages and prices of replacements, denoted respectively by  $D_i$  and  $\pi_i$  for the  $i^{\text{th}}$  component, out of  $n$  in total:

$$J_{\text{fat}}(\mathbf{y}) = \sum_{i=1}^n \pi_i \mathcal{D}_i(\mathbf{y}_i) \quad (5)$$

where  $\mathbf{y}_i$  is the  $i^{\text{th}}$  output trajectory.

In [11], it was shown that it is possible to derive a data-driven relationship between a variance such as variance of a signal and fatigue damage, thanks to a linear regression between the logarithms of variance and fatigue damage of a given signal. This allows to approximate the fatigue damage of the  $i^{\text{th}}$  component considered in the fatigue cost estimation with a function denoted by  $\hat{\mathcal{D}}_i$ :

$$\hat{\mathcal{D}}_i(\mathbf{y}_i) = \frac{e^{b_i}}{(L_i^{\text{ult}})^{m_i}} \text{Var}(\mathbf{y}_i)^{a_i} \quad (6)$$

where  $L_i^{\text{ult}}$ ,  $m_i$ ,  $a_i$  and  $b_i$  are respectively the ultimate load, Wöhler exponent, linear coefficient and bias of the regression corresponding to the  $i^{\text{th}}$  component. It should be noticed that the ultimate load is the maximum load that the component can bear and the Wöhler exponent depends on the material (4 for steel and 10 for glass fiber). Therefore,  $J_{\text{fat}}$  can be approximated by the fatigue-oriented cost function, denoted by  $\tilde{J}_{\text{fat}}$ , for a given output trajectory  $\mathbf{y}$ :

$$\tilde{J}_{\text{fat}}(\mathbf{y}) = \sum_{i=1}^n \pi_i \frac{e^{b_i}}{(L_i^{\text{ult}})^{m_i}} (\text{Var}(\mathbf{y}_i))^{a_i} \quad (7)$$

Hence,  $\tilde{J}_{\text{fat}}$  allows to approximate  $J_{\text{fat}}$  with a differentiable cost function based on variance. This cost function can be efficiently implemented in an optimization, in order to better reduce  $J_{\text{fat}}$  in optimal control problems.

## 5. The fatigue-oriented optimization

The cost function  $\tilde{J}_{\text{fat}}$  is particularly interesting for optimal control problems because it is convex provided that  $a_i \in ]-\infty, 0] \cup ]1, +\infty[ \quad \forall i \in \{1, \dots, n\}$ , differentiable and expressed algebraically. Therefore, there is no loss of convexity nor differentiability in using it instead of standard quadratic function of time series. The resulting problem can be solved directly using standard NonLinear Programming (NLP) solvers:

$$\min_{\mathbf{u}} \quad \tilde{J}_{\text{fat}}(\mathbf{y}(x_0, \mathbf{u}, \mathbf{v})) \quad (8)$$

where  $x_0$  and  $\mathbf{v}$  are given. Moreover, as  $\tilde{J}_{\text{fat}}$  is based on variance, it is possible to efficiently solve the problem with sequential Quadratic Programming (QP) in a fixed-point problem, by approximating  $\tilde{J}_{\text{fat}}$  with a first order Taylor expansion. Besides, it is possible to decompose the horizon of the successive QP free of inequality constraints, in order to significantly reduce the computational time. In this section are presented the derivation of the fixed-point problem from the NLP and the decomposition of the horizon, which together significantly reduce the computational time needed for the resolution of the NLP.

### 5.1. Fixed-point problem formulation

It is a common approach to approximate NLP with QP which are easier to solve than NLP and convex, in order to efficiently approach a suboptimal solution of the NLP. The approximated fatigue cost function  $\tilde{J}_{\text{fat}}$  has the advantage to be based on a quadratic cost function which is variance. Therefore, a first order Taylor expansion of  $\tilde{J}_{\text{fat}}$ , defined by (7), around an output trajectory  $\mathbf{y}^{(\sigma)}$  is sufficient to make appear a quadratic form:

$$\tilde{J}_{\text{fat}}^{(\sigma)}(\mathbf{y}) = \tilde{J}_{\text{fat}}(\mathbf{y}^{(\sigma)}) + \underbrace{\sum_{i=1}^n \frac{\pi_i e^{b_i a_i}}{(L_i^{\text{ult}})^{m_i}} \left( \text{Var}(\mathbf{y}_i^{(\sigma)}) \right)^{a_i-1}}_{\alpha_i(\mathbf{y}^{(\sigma)})} \left( \text{Var}(\mathbf{y}_i) - \text{Var}(\mathbf{y}_i^{(\sigma)}) \right) \quad (9)$$

where  $\sigma$  is the iteration number of the fixed-point problem. Let  $Q(\mathbf{y}^{(\sigma)}) = \text{diag}([\alpha_1(\mathbf{y}^{(\sigma)}), \dots, \alpha_n(\mathbf{y}^{(\sigma)})])$  and  $R(\mathbf{y}^{(\sigma)}) = I_2 \min([1, \alpha_1(\mathbf{y}^{(\sigma)})]) \times 10^{-3}$  be the weighting matrices of an QP that approximates the solution of (8) at the  $\sigma^{\text{th}}$  iteration:

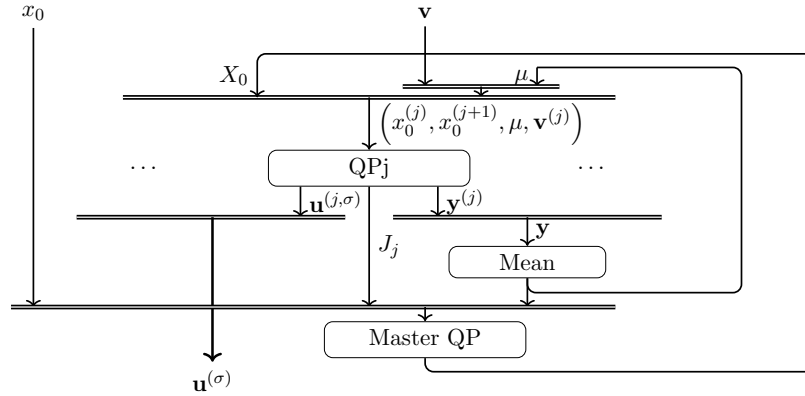
$$\mathcal{P}(x_0, \mathbf{v}, \mathbf{y}^{(\sigma)}) : \quad \min_{\mathbf{u}} \quad \tilde{J}_{\text{fat}}^{(\sigma)}(\mathbf{y}) = \sum_{k=1}^N (y_k(\mathbf{u}, \mathbf{v}, x_0) - \mu)^T Q(\mathbf{y}^{(\sigma)}) (y_k(\mathbf{u}, \mathbf{v}, x_0) - \mu) + \quad (10a)$$

$$u_k^T R(\mathbf{y}^{(\sigma)}) u_k$$

$$\text{s.t.} \quad \mu = \frac{1}{N} \sum_{k=1}^N y_k(\mathbf{u}, \mathbf{v}, x_0) \quad (10b)$$

where  $R$  avoids singularities in the optimization problem. It should be noticed that the optimization (10) gives an optimal solution  $\mathbf{u}^{(\sigma+1)}$ , which integrated with the system dynamic equation (1) for given  $x_0$  and  $\mathbf{v}$  yield an output trajectory  $\mathbf{y}^{(\sigma+1)}$ . This  $\mathbf{y}^{(\sigma+1)}$  gives a new optimal control problem  $\mathcal{P}(x_0, \mathbf{v}, \mathbf{y}^{(\sigma+1)})$ , defined by (10), resulting in a  $\mathbf{u}^{(\sigma+2)}$  different from  $\mathbf{u}^{(\sigma+1)}$ , so on and so forth. There is thus a fixed-point problem consisting in finding the right  $\mathbf{u}^{(\sigma)}$  and  $\mathbf{y}^{(\sigma)}$ . The following algorithm is proposed to solve this fixed-point problem:

- (i)  $\sigma \leftarrow 0$ , Give an initial guess for  $\mathbf{y}^{(0)}$  and  $\mathbf{u}^{(0)}$
- (ii) Compute the weighting matrices  $Q(\mathbf{y}^{(\sigma)})$  and  $R(\mathbf{y}^{(\sigma)})$
- (iii)  $\mathbf{u}^{(\sigma+1)} \leftarrow$  Solve the optimal control problem  $\mathcal{P}(x_0, \mathbf{v}, \mathbf{y}^{(\sigma)})$  defined by (10)
- (iv)  $\mathbf{y}^{(\sigma+1)} \leftarrow$  Integrate the dynamic equations, defined by (1), for  $\mathbf{u}^{(\sigma+1)}$ ,  $x_0$  and  $\mathbf{v}$



**Figure 1.** Schematization of the temporal decomposition scheme.

(v) If  $\mathbf{y}^{(\sigma)}$  and  $\mathbf{u}^{(\sigma)}$  have not converged, increment  $\sigma$  and go to step (ii), otherwise stop

This fixed-point method is used in the sequel instead of NLP solvers. However, for a high number of decision variables, the QP is computationally and time expensive because of the inversion of high dimensional matrices. That is the reason why a time decomposition is introduced in the next section in order to define a change in the decision variable's definition and significantly reduce the RAM and time needed for the QP resolution.

### 5.2. Optimization horizon decomposition scheme

The solution to the QP defined by (10) is a linear combination of  $x_0$  and  $\mathbf{v}$ , which requires a matrix inversion of dimension  $[N \times 2, N \times 2]$ . The longer the prediction horizon is, the more RAM and CPU time are required, moreover the QP must be solved at every iterations of the fixed-point method. Therefore, reducing significantly the time needed for the QP resolution can also significantly reduce the time to solve the fixed-point method and the NLP defined by (8).

The optimization horizon of an QP can be temporally decomposed as in [15, 16], allowing to significantly alleviate the computational cost. The temporal decomposition of an QP optimization horizon consists in decomposing the problem into a series of smaller QPs with shorter optimization horizons, constrained by their initial and final states, provided that the cost function on the interval is independent from the others. Then, the solution to (10) is recovered by solving another QP that manages the initial states of the smaller QPs. However, for the QP defined by (10), the cost function on the smaller intervals is not independent from the others, as  $\mu$ , which is the average of the outputs on the whole horizon has an influence on the cost function. Therefore,  $\mu$  is first considered as an exogenous parameter in the smaller QPs and retrieved later with the QP managing the initial states. A schematization of the optimization horizon decomposition is proposed in Figure 1. Let  $M$  be the number of intervals the horizon is decomposed into, which is also the number of smaller QPs. The QP on the  $j^{\text{th}}$  interval, for  $j \in \{1, \dots, M\}$ , at the  $\sigma^{\text{th}}$  iteration, denoted by QPj, is parameterized by the initial condition and disturbance on the interval, denoted respectively by  $x_0^{(j)}$  and  $\mathbf{v}^{(j)}$ , and the initial condition on the next interval  $x_0^{(j+1)}$ . All the QPj are using the same cost function parameterized by  $\mathbf{y}^{(\sigma)}$  and  $\mu$  which are considered as external parameters:

$$\min_{\mathbf{u}^{(j)}} J_j = \sum_{k=k_0(j)}^{k_f(j)} \left( y_k^{(j)} \left( \mathbf{u}^{(j)}, \mathbf{v}^{(j)}, x_0^{(j)} \right) - \mu \right)^T Q(\mathbf{y}^{(\sigma)}) \left( y_k^{(j)} \left( \mathbf{u}^{(j)}, \mathbf{v}^{(j)}, x_0^{(j)} \right) - \mu \right) + \quad (11a)$$

$$\text{s.t.} \quad x_0^{(j+1)} = x_f \left( \mathbf{u}^{(j)}, \mathbf{v}^{(j)}, x_0^{(j)} \right) \quad (11b)$$



where  $\mathbf{u}^{(j)}$  is the input trajectory on the  $j^{\text{th}}$  interval,  $u_k^{(j)}$  is the input value at the  $k^{\text{th}}$  time instant in  $\mathbf{u}^{(j)}$ ,  $k_0(j) = (j-1)\frac{N}{M} + 1$  and  $k_f(j) = j\frac{N}{M}$  are respectively the index of the first and last time instant of the  $j^{\text{th}}$  interval.  $y_k^{(j)}$  and  $x_f$  are affine functions giving respectively the output value at the  $k^{\text{th}}$  time instant and the  $j^{\text{th}}$  interval final state for given  $x_0$ ,  $\mathbf{v}^{(j)}$  and  $\mathbf{u}^{(j)}$ . It should be noticed that  $x_0^{(1)} = x_0$ , which is the initial state of the QP defined by (10). The solution to (11), denoted by  $\mathbf{u}^{(j,\sigma)}$ , is an affine function of  $x_0^{(j)}$ ,  $x_0^{(j+1)}$ ,  $\mathbf{v}^{(j)}$  and  $\mu$ , besides each resolution of the QPj needs a matrix inversion of dimension  $[\frac{N}{M} \times 2, \frac{N}{M} \times 2]$ . Therefore, the output trajectory, denoted by  $\mathbf{y}^{(j)}$ , which is a concatenation of the  $y_k^{(j)} \left( \mathbf{u}^{(j)}, \mathbf{v}^{(j)}, x_0^{(j)} \right) \quad \forall k \in \{k_0(j), \dots, k_f(j)\}$  is an affine function of  $x_0^{(j)}$ ,  $\mathbf{u}^{(j,\sigma)}$  and  $\mathbf{v}^{(j)}$ , moreover by composition it is an affine function of  $x_0^{(j)}$ ,  $x_0^{(j+1)}$ ,  $\mathbf{v}^{(j)}$  and  $\mu$ . Hence,  $J_j$  can be expressed as a quadratic function of  $x_0^{(j)}$ ,  $x_0^{(j+1)}$ ,  $\mathbf{v}^{(j)}$  and  $\mu$ . Let  $X_0 = [x_0^{(1)}, \dots, x_0^{(M+1)}]$  be the vector of the interval initial conditions. The cost function  $J_j$  can thus be expressed as quadratic cost function of  $X_0$  and  $\mu$ , given the disturbance trajectory  $\mathbf{v}$ . Therefore the cost function  $\tilde{J}_{\text{fat}}^{(\sigma)}(\mathbf{y})$  of the QP defined by (10) can be expressed as the sum of the  $J_j \quad \forall j \in \{1, \dots, M\}$ . Moreover,  $\mu$  is the average of  $\mathbf{y}$  which is a concatenation of the  $\mathbf{y}^{(j)} \quad \forall j \in \{1, \dots, M\}$ , which makes thus  $\mu$  an affine function, denoted by  $\mathcal{F}$ , of  $X_0$  and  $\mu$ , given  $x_0$  and  $\mathbf{v}$ . The master QP then manages the vector of initial conditions  $X_0$ , such that the concatenation of  $\mathbf{y}^{(j,\sigma)} \quad \forall j \in \{1, \dots, M\}$  matches the solution of (10) is the following:

$$\min_{X_0} \quad \tilde{J}_{\text{fat}}^{(\sigma)} = \sum_{j=1}^M J_j \left( X_0, \mu | \mathbf{v}^{(j)}, x_0 \right) \quad (12a)$$

$$\text{s.t.} \quad \mu = \mathcal{F} \left( X_0, \mu | \mathbf{v}^{(j)}, x_0 \right) \quad (12b)$$

where the implicit equation defined by (12b) is actually a linear equality constraint allowing to retrieve  $\mu$ . The solution of (12) is thus a linear combination of  $x_0$  and  $\mathbf{v}$ , requiring a matrix inversion of dimension  $[6 \times (M+1), 6 \times (M+1)]$  as the state dimension is 6. Eventually, the optimal input trajectory  $\mathbf{u}^{(\sigma)}$  can be recovered by concatenating the solutions of the QPj defined by (11), denoted by  $\mathbf{u}^{(j,\sigma)}$  with the  $x_0^{(j)}$  and  $x_0^{(j+1)}$  obtained from the solution of (12).

From the above discussion, it comes out that  $M$  determines the computation burden involved in the solution of (11) and (12). The computational complexity of the QP problem, for the considered system (1) without temporal decomposition is  $\mathcal{O}((2N)^2(2N+6+N))$ , while using the temporal decomposition it becomes  $\mathcal{O}\left(M6(M-1)^2(6+1) + \frac{(2N)^2}{M}(\frac{N}{M}(2+1) + 12+m)\right)$ . This means that if  $N$  takes values of 100, 1000 or 10000, the  $M$  giving the lowest computational complexity are respectively 5, 16 and 49, allowing to divide the computational complexity of the global QP problem by 12, 120 and 1200 respectively.

## 6. Results

In this section, an assessment regarding the relationship between variance and fatigue damage is first presented. Then the sensitivity of the optimal solution to horizon's length is analyzed in terms of both mechanical fatigue and CPU time. Furthermore, comparison with optimization using a finely tuned quadratic cost function is proposed. Finally, in a prospect of using the open-loop optimization with the fatigue-oriented cost function in closed-loop, a sensitivity analysis to the prediction horizon of a linear quadratic MPC in terms of fatigue is conducted. Statistics on each results are made using 1000 disturbances which are Gaussian noises filtered through a family of first order low pass filters. The filters are parameterized by their static gain and characteristic time, taking values in uniform distributions ranging respectively on  $]0, 10^4]$  and  $]0, 10]$ . In the sequel, all simulations are considering an initial state located at the origin.

### 6.1. Relationship between fatigue damage and variance

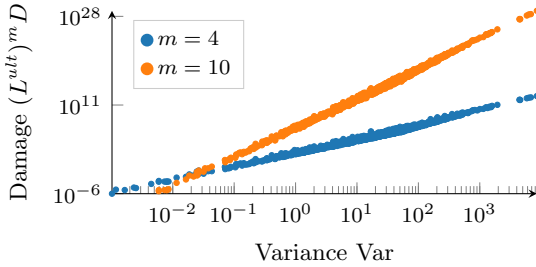
The experience described in Section 4 is conducted and the results are plotted in Figure 2. The relation can be effectively approximated by a linear regression in Logarithmic scale and yield couples of parameters  $(a, b)$  of value  $(2.6, 4.6)$  and  $(5.6, 15.6)$ , for Wöhler coefficients of 4 and 10 respectively.

### 6.2. Fatigue reduction and computing time-sensitivity analysis

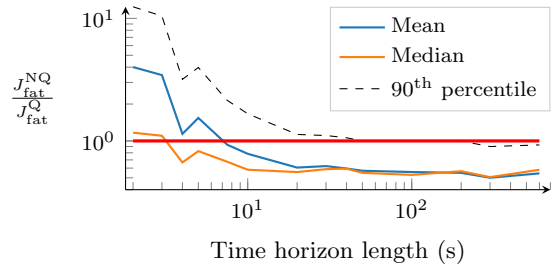
Concerning the sensitivity study of optimal control problem solutions using the parameterized quadratic and non-quadratic fatigue-oriented cost function to the horizon's length, a sampling time of 0.1 second is considered. For both optimizations, the optimization horizon's temporal decomposition scheme described above is implemented. The parameter  $M$  takes the values  $\{5, 5, 5, 5, 5, 10, 10, 10, 10, 10, 20, 20, 20, 20, 20, 20\}$  together with the horizon lengths  $\{2, 3, 4, 5, 7.5, 10, 20, 30, 40, 50, 100, 200, 300, 600\}$  seconds respectively.

The parameterized quadratic cost function, defined in Section 3, is optimized for a disturbance whose filter parameters are randomly drawn, in the same distribution that the one used for disturbance generation. In this presented case, the static gain and characteristic time drawn are 6950 and 2.86 seconds respectively. The optimization of  $\rho$  is conducted for each horizon considered. The fatigue cost of the 1000 optimizations solutions using the non-quadratic cost function, defined by (7), is estimated and denoted by  $J_{\text{fat}}^{\text{NQ}}$ . The fatigue cost of the optimization solutions using the finely tuned parameterized quadratic cost function, defined by (3) is also estimated and denoted by  $J_{\text{fat}}^{\text{Q}}$ . For the fatigue estimations, parameters are carefully chosen in order to yield a realistic fatigue trade-off. The values of  $L^{\text{ult}}$ ,  $\pi$  and  $m$  for the yawing and tilting blade root bending moments are  $3 \times 10^3$ ,  $10^3$ , 10 respectively, while for the yawing and tilting pitch angles they are respectively  $6.98 \times 10^{-1}$ , 1 and 4.

In Figure 3, the mean, the median and the 90<sup>th</sup> percentile of the ratio between the fatigue costs



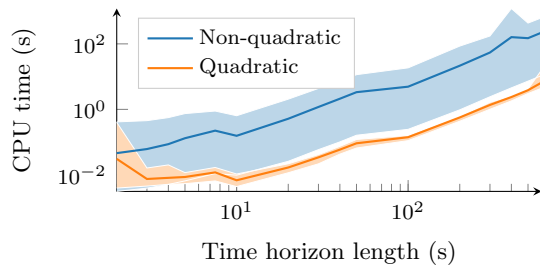
**Figure 2.** Plot of the relation between Var and  $(L^{\text{ult}})^m D$  in Logarithmic scale, for two Wöhler coefficients.



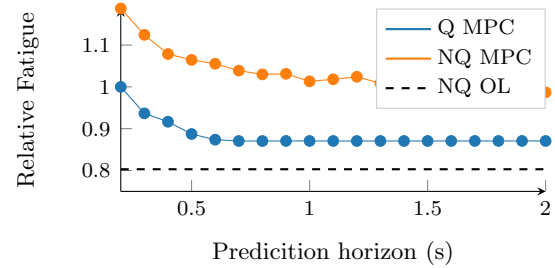
**Figure 3.** Plot of the influence of prediction horizon time length on the relative system fatigue cost between the non-quadratic and quadratic cost functions.

open-loop optimal trajectories using the non-quadratic and quadratic cost functions, defined by  $\frac{J_{\text{fat}}^{\text{NQ}}}{J_{\text{fat}}^{\text{Q}}}$  are shown. The thick red line represents the border between cases where non-quadratic optimization induces higher fatigue cost compared to the quadratic one (above), and cases where it induces lower fatigue costs (below). It can be seen that the longer the time horizon is, the smaller the relative fatigue. More precisely, the median relative fatigue cost is lower than 1 for time horizon length longer than 4 seconds, while for the mean and 90<sup>th</sup> percentile of the relative fatigue cost it is respectively 7.5 and 50 seconds. Moreover, beyond 20 seconds, the fatigue reduction ratio induced by the non quadratic cost is as high as 50%.

In Figure 4, the mean and the range of the CPU times needed for the 1000 quadratic and



**Figure 4.** Influence of time horizon length on the CPU time. Thick lines correspond to the mean and the shaded areas to the range of values.



**Figure 5.** Plot showing that the prediction horizon typically used in common MPC design is far too short to be relevant.

Method	$N$	Compilation time (s)	Resolution time (s)
NLP solver CasADi	100	17	0.21
	1000	5833	26.50
Fixed-point	100	None	3.20
	1000	None	2769.40
Fixed-point with decomposed horizon	100	None	0.10
	1000	None	2.04

**Table 1.** Summary of the compilation and solving time for the NLP solver, fixed-point and fixed-point method with a decomposed horizon.

non-quadratic open-loop optimizations are plotted as a function of the horizon time length, for optimization run on a desktop PC equipped with a 2.60GHz processor and 16Gb RAM. It can be seen that the optimization using the non-quadratic cost function requires a much longer CPU time than the quadratic one. This results was expected as the fixed-point method requires to run a quadratic optimization per iteration.

In Table 1 are summarized the CPU times needed to solve the optimization problem defined by (8) for  $N = 100$  and 1000, using the NLP solver *CasADi* [17] and the fixed-point method with and without a decomposed horizon. It can be seen that, compared to the NLP solver, the fixed-point method with a decomposed horizon is much less time consuming, especially for long optimization horizons. Moreover, the fixed-point method avoids the compilation time, which even though it is performed off-line, can be highly time consuming for long optimization horizons and mandatory for the NLP solver. It also emphasizes the efficiency of the horizon decomposition, which allows to decrease the CPU time from one to three order of magnitudes compared to the fixed-point method without decomposition. Even though, they might have possibilities to decompose the optimization horizon of the NLP as well, the optimization problem managing the initial states of the intervals might be very challenging to express for the NLP case, while the fixed-point method allows to consider only QPs and to relatively easily decompose the optimization horizon.

### 6.3. Prospects of closed-loop implementation

The non-quadratic criterion is used in an MPC and simulated in closed-loop for several prediction horizons ranging from 0.2 to 2 seconds. This MPC is compared to another MPC using the parameterized quadratic cost function on 100 closed-loop simulations of 200 seconds long. In Figure 5, the fatigue averaged on the 100 closed-loop simulations and normalized by the average fatigue of the quadratic MPC for a 0.2 seconds prediction horizon is plotted in function of the prediction horizon. It can be seen that for relatively short prediction horizon, compared to the ones used for the open-loop optimization, the MPC using the non-quadratic cost function

is less efficient in reducing fatigue than the quadratic MPC. This coincides with the previous observation, stating that the open-loop optimization using the non-quadratic criterion becomes efficient compared to the parameterized quadratic one for horizons longer than 4 to 50 seconds. This is due to the fact that considering fatigue on such short prediction horizons is very limited as very few hysteresis cycles can be counted, for the dynamics considered in this application and therefore the statistical relation observed between variance and damage might be weak. However, the CPU time needed for the open-loop optimizations using the non-quadratic cost function makes its use in an MPC prohibitive. Nevertheless, it might be possible to learn from the open-loop optimization using the non-quadratic criterion the behavior of a non-linear feedback controller, using information on the future disturbance on an acceptable receding horizon.

## 7. Conclusion and perspectives

In this paper, the sensitivity analysis of fatigue-oriented open-loop optimal cost and its computational burden with respect to the length of the prediction horizon is conducted for different choices of cost functions. The analysis shows that the use of a non quadratic cost induces significant benefit over standard quadratic (even finely tuned) cost functions provided that prediction horizons longer than 50 seconds are involved. However, the computational burden of the fatigue-oriented optimization becomes rapidly intractable to be incorporated in MPC design. Undergoing works focus on the extraction of fitted static nonlinear feedback from the off-line computation of open-loop formulations with long prediction horizons and non quadratic cost leading to solutions that inherit the benefit from the newly introduced cost while being real-time implementable. Preliminary results on this behavioral cloning solution are encouraging and will hopefully be presented in a forthcoming paper.

## 8. References

- [1] Loew S, Obradovic D and Bottasso C 2019 *Proc. European Control Conference (ECC)* pp 3371–3376
- [2] Collet D, Di Domenico D, Sabiron G and Alamir M 2019 *2019 American Control Conference (ACC)* (IEEE) pp 3500–3505
- [3] Hansen M O 2015 *Aerodynamics of wind turbines* (Routledge)
- [4] Bossanyi E 2003 *Wind Energy* **6** 119–128
- [5] Hammerum K, Brath P and Poulsen N K 2007 *Journal of Physics: Conference Series* **75** 012081
- [6] Barradas-Berglind J and Wisniewski R 2016 *Wind Energy* **19** 2189–2203
- [7] Palmgren A 1924 *Zeitschrift des Vereines Duetsher Ingenieure* **68** 339
- [8] Downing S D and Socie D 1982 *International journal of fatigue* **4** 31–40
- [9] Knudsen T, Bak T and Svenstrup M 2015 *Wind Energy* **18** 1333–1351
- [10] de Jesus Barradas-Berglind J, Wisniewski R and Soltani M 2015 *IET Control Theory & Applications* **9** 1042–1050
- [11] Collet D, Alamir M, Di Domenico D and Sabiron G 2020 A fatigue-oriented cost function for optimal individual pitch control of wind turbines *Unpublished Submitted to IFAC 2020*
- [12] Jonkman J, Butterfield S, Musial W and Scott G 2009 Definition of a 5-MW reference wind turbine for offshore system development Tech. rep. National Renewable Energy Laboratory
- [13] Coleman R P and Feingold A M 1957 *National Advisory Committee for Aerounautics*
- [14] Bir G 2008 *Proc. 46th AIAA aerospace sciences meeting and exhibit* p 1300
- [15] Chang S C 1986 *A hierarchical, temporal decomposition approach to long horizon optimal control problems* Ph.D. thesis University of Connecticut
- [16] Xu W and Anitescu M 2018 *SIAM Journal on Optimization* **28** 2541–2573
- [17] Andersson J A E, Gillis J, Horn G, Rawlings J B and Diehl M 2019 *Mathematical Programming Computation* **11** 1–36