Data-driven fatigue-oriented MPC applied to wind turbines Individual Pitch Control

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

Data-driven fatigue-oriented MPC applied to wind turbines Individual Pitch Control

David Collet[a,b,*], Mazen Alamir[b], Domenico Di Domenico[a], Guillaume Sabiron[a]

[a] *Digital Science and Technology division, IFP Energies Nouvelles, Rond-point de l'échangeur de Solaize, 69360 Solaize, France*

[b] *Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-Lab, 38000 Grenoble, France*

Abstract

Horizontal Axis Wind Turbine (HAWT) rotor is a system whose components are particularly exposed to fatigue damage, due to non-uniform winds and the rotating nature of the rotor. Individual Pitch Control (IPC) can help alleviating the fatigue loads on the rotor blades by rotating the blades independently from each other on their longitudinal axis. However, reducing the blades fatigue damage is often compensated by an increase in pitch activity, damaging the blade pitch actuators. An IPC regulator must thus optimize a trade-off between the fatigue damage of the blades and blade pitch actuators. This paper presents the derivation and implementation of a fatigue-oriented adaptive Model Predictive Control (MPC) IPC regulator using a data-driven fatigue-oriented cost function. This MPC allows to efficiently optimize the IPC fatigue trade-off and significantly reduce the expectancy of an HAWT rotor economic fatigue cost, compared to a finely tuned non-adaptive MPC. The methodology used for the derivation of the presented MPC allows to efficiently reduce an economic fatigue cost, while limiting the sensitivity to controller tuning.

Keywords: Fatigue, Optimal control, Model Predictive Control, Individual Pitch Control

Nomenclature

HAWT	Horizontal Axis Wind Turbine
IPC	Individual Pitch Control
MBC	Multi-Blade Coordinates
MPC	Model Predictive Control Problem
LTI	Linear Time Invariant
OCP	Optimal Control Problem

FO-OCP	Fatigue-Oriented Optimal Control Problem
RFC	RainFlow Counting
QP	Quadratic Programming
NLP	NonLinear Programming
$\mathcal{J}$	Economic fatigue cost
$\mathbf{a}$	Trajectory of a vector a
a_k	k^{th} component of a vector a
$a^\star$	Optimal value of a
x	State vector of an HAWT
u	Input vector to an HAWT
v	Disturbance vector on an HAWT
y	Output vector from an HAWT
N_c	Number of components considered in an HAWT
π	HAWT components prices of replacement vector
$\mathcal{D}_k$	HAWT k^{th} component fatigue damage
m_k	HAWT k^{th} component Wöhler coefficient
L_k^{ult}	HAWT k^{th} component ultimate fatigue load
$\hat{\mathcal{J}}$	Data-riven fatigue-oriented cost function
N_d	Number of time series generated $\hat{\mathcal{J}}$ derivation
$\hat{\mathcal{D}}$	Predicted vector of fatigue damages
$\mathcal{F}$	Vector of functions allowing to predict $\hat{\mathcal{D}}$
w, b	Linear and bias coefficients parameterizing the functions in $\mathcal{F}$
	HAWT rotor azimuth angle
τ_act	Time constant of the blade pitch actuator
$A, B, B_\text{d}, C, D, D_\text{d}$	Parameters of an LTI HAWT system
θ, θ^sp	Blade pitch angles and corresponding setpoints vectors
M	Blade root bending moment vector
$a_\text{yaw}, a_\text{tilt}$	Yawing and tilting component of a vector a expressed in MBC
T	Prediction horizon (s)

Q, R, L Weighting matrices parameterizing a quadratic cost function

p Array of parameters defining a quadratic cost function

$\mathbf{v}_{\text{ol}}$ Disturbance trajectory of a process in an open-loop optimization

$\mathbf{v}_{\text{cl}}$ Disturbance trajectory of a process in a closed-loop simulation

ν Scalar parameter of a parameterized quadratic cost function

$\nu_{\text{ideal}}(\mathbf{v}_{\text{cl}})$ Parameter ν minimizing $\mathcal{J}$ for a disturbance $\mathbf{v}_{\text{cl}}$

ν_{mean} Scalar value of ν minimizing the $\mathcal{J}$ expectancy for a set of disturbances

$\text{MPC}_{\text{ideal}}$ Idealistic parameterized quadratic MPC using $\nu_{\text{ideal}}(\mathbf{v}_{\text{cl}})$ as parameter

MPC_{mean} Finely tuned parameterized quadratic MPC using ν_{mean} as parameter

J_{var} Variance of a signal

μ Mean of a signal

J_{OCP} First order Taylor approximation of $\hat{\mathcal{J}}$

α First order Taylor approximation of $\mathcal{F}$

i_{iter} i^{th} iteration step in a fixed-point algorithm

β Filtering parameter in a fixed-point algorithm

T_{last} Parameter p estimation horizon

ξ Weighting parameter of a weighted formulation of variance

MPC_{filt} Filtered adaptive fatigue-oriented quadratic MPC

$\text{MPC}_{\text{direct}}$ Non-filtered adaptive fatigue-oriented quadratic MPC

1. Introduction

Horizontal Axis Wind Turbine (HAWT) rotor is a rotating system, composed of three blades and blade pitch actuators, allowing to pitch the blades on their longitudinal axis. An HAWT rotor is continuously disturbed by turbulent winds, whose speed and orientation vary in time and space. The recent largest HAWT installations are able to reach diameters of 150 and 200 meters in respectively on-shore and off-shore plants. The wind speed variations over the rotor plane can thus be significant. Therefore, the aerodynamic loads applied on a rotating blade with constant pitch angle will vary with wind speed in the rotor plane during a rotation, inducing oscillatory loads which are a source of fatigue.

The blade pitch actuators allow to pitch the blades on their longitudinal axis, in order to modify the aerodynamic properties of the rotor. Individual Pitch Control (IPC), consists in varying the blades pitch angles during its rotation in the rotor plane, in order to alleviate the oscillatory loads causing fatigue [1]. However, the blades oscillatory loads

reduction has the drawback of increasing blade pitch activity [1, 2, 3, 4] and eventually blade pitch actuators fatigue damage. There is thus an inherent trade-off between the fatigue damage of blades and the one of pitch actuators. A relevant way to evaluate this trade-off, proposed in [5], consists in defining a cost function, denoted by $\mathcal{J}$, as the price-weighted sum of fatigue damages:

$$\mathcal{J}(\mathbf{y}) = \sum_{k=1}^{N_c} \pi_k \mathcal{D}_k(\mathbf{y}_k) \tag{1}$$

where $\mathbf{y}_k$, π_k and $\mathcal{D}_k$ are respectively the output trajectory, price of replacement and fatigue damage of the k^{th} component of a multi-output process, $\mathbf{y}$ contains the process stacked output trajectories and N_c is the number of HAWT components considered in $\mathcal{J}$. The control strategy tailored to minimize a complex cost function such as $\mathcal{J}$ is Model Predictive Control (MPC). MPC can be defined in a nutshell as a control strategy that consists in solving an open-loop Optimal Control Problem (OCP) over a finite receding horizon, parameterized by the initial state of the prediction horizon, objective function and possibly constraints on the system [6]. The first time instant input trajectory of the OCP solution is then given as input to the system by the controller. MPC is based on the fact that solving iteratively an OCP over a finite horizon should allow to approach the solution of the same open-loop OCP over an infinite horizon. The advantages of MPC are that it allows to optimally handle control problems under complex dynamics, constraints specifications and objective functions. Moreover, MPC allows to optimally anticipate disturbances which can be estimated, such as wind with Light Detection and Ranging (LiDAR) devices, in wind turbine control. On the other hand, the main drawback of MPC is the potential computational cost of solving the open-loop OCP. Moreover, the incorporation of the fatigue damages $\mathcal{D}_k$ in the objective function $\mathcal{J}$ of an OCP is challenging due to fatigue estimation [7].

The widely used representation of fatigue consists in counting the number of hysteresis cycles contained in a time series, using a RainFlow Counting (RFC) algorithm [8], and summing the fatigue damages related to each hysteresis cycle with the Palmgrem-Miner linear damage rule [9, 10]. The RFC algorithm has an algorithmic nature which makes its incorporation in an OCP objective challenging. Nevertheless, the methods described in [11, 12], consisting in parameterizing a MPC objective function with the results of a RFC algorithm performed over a previous solution time series, and turning the OCP into a tracking problem. This allows to efficiently reduce the fatigue damage of one process output. The shortcomings of this method are the fact that one RFC evaluation is needed per output, therefore the computational complexity of the method greatly increases with the number of components considered in $\mathcal{J}$. Another fatigue-oriented MPC is designed in [13], where parameters of a quadratic cost function used as objective in the MPC are identified on-line, in order to match an on-line estimation of fatigue.
Thanks to probabilistic considerations on random processes, it is possible to bypass the RFC algorithm and express fatigue damage as nonlinear expression of the zero, second and fourth order spectral moments, which are respectively equivalent to the variance of the zero, first and second time derivative of the process output. This expression is firstly established for narrow-banded processes in [14] and extended to random processes in [15]. Thus, it is possible to relate the fatigue damage to the variance of a signal and its time

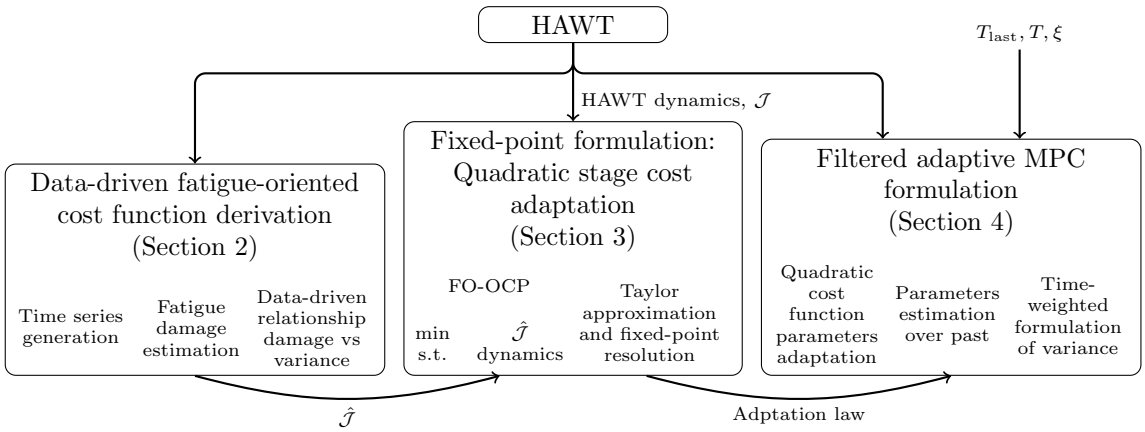

Figure 1: Flow chart summarizing the design of the filtered adaptive MPC and the outline of the paper.

derivatives. Therefore, in [16] a data-driven fatigue-oriented cost function derived from the relationship between variance and fatigue damage of a process output is obtained. This fatigue-oriented cost function is then used in an open-loop OCP, named Fatigue-Oriented Optimal Control Problem (FO-OCP), as a surrogate to $\mathcal{J}$. The solutions of the open-loop FO-OCP show a great potential in reduction of the fatigue cost $\mathcal{J}$ compared to the solution of a finely tuned standard quadratic OCP. This may suggest to use such a fatigue-oriented cost function as objective in a MPC, in order to efficiently reduce the fatigue cost $\mathcal{J}$. However, it is shown in this article that the direct use of the FO-OCP in a MPC dramatically increases the fatigue-oriented cost function when implemented in closed-loop. Hence, a MPC formulation, whose stage cost is adapted based on the FO-OCP structure and filtering features, is proposed.

This article is organized as follows. In Section 2, the derivation of the data-driven fatigue-oriented cost function is presented. In Section 3, a resolution method of the FO-OCP featuring a fixed-point problem, whose MPC stage cost adaptation law is based on, is detailed. In Section 4, the formulation of the MPC with adaptive stage cost, based on the FO-OCP and fixed-point problem formulation is depicted. In Section 5, it is shown that the closed-loop implementation of the MPC with adaptive stage cost presents a great reduction potential of the fatigue cost $\mathcal{J}$. In Section 6, conclusion and perspectives on this approach are given. A summary of the MPC with adaptive stage cost design and the paper outline is given in Figure 1.

2. Data-driven fatigue-oriented cost function derivation

This section presents the derivation of the fatigue-oriented cost function to be used as objective in the FO-OCP. The approach considered here is based on the one defined in [16], where the fatigue-oriented cost function is derived using a data-driven linear regression between the logarithms of variance and fatigue damage of signals. The derivation process comes in three steps:

1. The generation of N_d time series, denoted by
$$\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N_\mathrm{d})}\}$$
representing a wide variety of possible outputs of a closed-loop process.
2. The estimation of variance and fatigue damages for the generated time series, denoted respectively by $\{\mathrm{Var}(\mathbf{y}_k^{(1)}), \ldots, \mathrm{Var}(\mathbf{y}_k^{(N_\mathrm{d})})\}$ and $\{\mathcal{D}_k(\mathbf{y}_k^{(1)}), \ldots, \mathcal{D}_k(\mathbf{y}_k^{(N_\mathrm{d})})\}$ for the k^{th} output of a process.
3. The fit of a regressor for each component k, denoted by $\mathcal{F}_k$, allows to predict a fatigue damage estimation, denoted by $\hat{\mathcal{D}}_k$, from the variance estimates, such that $\hat{\mathcal{D}}_k(\mathbf{y}_k) = \mathcal{F}_k\left(\mathrm{Var}(\mathbf{y}_k)\right) \simeq \mathcal{D}_k(\mathbf{y}_k)$

2.1. Time series generation

The first step consists in generating a set of time series, representative of the variety of outputs which a closed-loop process can yield. However, it is difficult to guess in advance what the dynamics of the closed-loop system will be, as the fatigue-oriented cost function and the OCP solutions depend on the data for which the regressors $\mathcal{F}_k$ have been fitted on. Therefore, it is proposed for the time series generation to simulate a system representative of the HAWT dynamics in closed-loop with an efficient MPC under different wind conditions, as described in this section.

2.1.1. Dynamic system definition

The National Renewable Energy Laboratory (NREL) provides a nonlinear HAWT simulator named Fatigue Aerodynamics Structures and Turbulence (FAST) [17] where a linearization module is integrated. The FAST linearization module allows to obtain a first order linear dynamic model of the HAWT, for a variety of blades azimuth angles, denoted by ψ:

$$\dot{\tilde{x}} = \tilde{A}(\psi)\tilde{x} + \tilde{B}(\psi)\tilde{u} + \tilde{B}_\mathrm{d}(\psi)v \tag{2a}$$
$$\tilde{y} = \tilde{C}(\psi)\tilde{x} + \tilde{D}(\psi)\tilde{u} + \tilde{D}_\mathrm{d}(\psi)v \tag{2b}$$

where $\tilde{x}$, $\tilde{u}$, $\tilde{y}$ and v represent respectively the state, input, output and disturbance of the system, while $\tilde{A}$, $\tilde{B}$, $\tilde{B}_\mathrm{d}$, $\tilde{C}$, $\tilde{D}$ and $\tilde{D}_\mathrm{d}$ are matrices of appropriate dimensions. For IPC, it is often preferred to linearize the rotating blades dynamics in a non-rotating frame of coordinates, named Multi-Blade Coordinates (MBC), more adapted for control design purposes [18, 1]. To pass from the blades rotating coordinates to MBC, the orthogonal MBC transform matrix, denoted by $T(\psi)$, is created:

$$T(\psi) = \frac{2}{3}\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \cos(\psi) & \cos(\psi + \frac{2\pi}{3}) & \cos(\psi + \frac{4\pi}{3}) \\ \sin(\psi) & \sin(\psi + \frac{2\pi}{3}) & \sin(\psi + \frac{4\pi}{3}) \end{pmatrix} \tag{3}$$

The blade pitch angles, denoted by θ_1, θ_2 and θ_3 can be thus turned into collective, yawing and tilting blade pitch angles, denoted respectively by θ_col, θ_yaw and θ_tilt with the following matrix multiplication:

$$\begin{pmatrix} \theta_\mathrm{col} \\ \theta_\mathrm{yaw} \\ \theta_\mathrm{tilt} \end{pmatrix} = T(\psi)\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \tag{4}$$

Table 1: Technical characteristics summary of the Senvion MM82.

Characteristic	Value
Rated power (kW)	2050
Rotor diameter (m)	82
Rotor speed (rpm)	8.5 – 17.1

Table 2: Summary of the parameters considered in the FAST linearization.

Parameter	Value
Operating wind speed (m/s)	12
Actuator time constant τ_{act} (s)	0.1
Disturbances considered in v	Hub-height wind speed

A similar transformation can be performed with the states and outputs of the system relative to the blades. The interest of having the system expressed in MBC for IPC regulation is that the linear system (2) is approximated as a Linear Time Invariant (LTI) system. Moreover, the system is extended in order to take into account the blade pitch actuators dynamics, which are modeled with a first order differential equation:

$$\dot{\theta}_i = -\frac{1}{\tau_{\text{act}}}\theta_i + \frac{1}{\tau_{\text{act}}}\theta_i^{\text{sp}} \tag{5}$$

where θ_i^{sp} is the i^{th} blade pitch angle setpoint and τ_{act} is the time constant of the blade pitch actuator. These transformations eventually yield the following dynamic model:

$$\dot{x} = Ax + Bu + B_{\text{d}}v \tag{6a}$$

$$y = Cx + Du + D_{\text{d}}v \tag{6b}$$

where $x \in \mathbb{R}^{\text{n}_\text{x}}$, $u \in \mathbb{R}^{\text{n}_\text{u}}$ and $y \in \mathbb{R}^{N_\text{c}}$ represent respectively the state, input and output of the system, while A, B, B_{d}, C, D and D_{d} are matrices of appropriate dimensions. It should be noticed that $u = [\theta_{\text{yaw}}^{\text{sp}}, \theta_{\text{tilt}}^{\text{sp}}]^T$ and $y = [M_{\text{yaw}}, M_{\text{tilt}}, \theta_{\text{yaw}}, \theta_{\text{tilt}}]^T$, where $\theta_{\text{yaw}}^{\text{sp}}$ and $\theta_{\text{tilt}}^{\text{sp}}$ are respectively the yawing and tilting blade pitch angles setpoints, M_{yaw} and M_{tilt} are respectively the yawing and tilting blade root bending moments. Moreover, θ_{yaw} and θ_{tilt} are integrated in x.

For the time series generation considered in the article, the LTI system is linearized from FAST with only the first flapwise degree of freedom activated and used as simulator. The turbine simulated is a Senvion MM82 whose characteristics are specified in Table 1. The parameters used for the linearization of the LTI system are summarized in Table 2. For the closed-loop control of this system, parameterized IPC MPCs with different parameters are designed.

2.1.2. Parameterized quadratic MPC definition

In order to obtain time series of the closed-loop process and define a benchmark controller for comparison, a parameterized quadratic MPC regulator is proposed, because its design is standard and has proven to be efficient in this task [3]. The proposed MPC

should solve between each update of the control the following open-loop OCP:

$$\min_{\mathbf{u}} \quad J = \int_t^{t+T} \big(y(\tau)^T Q y(\tau) + u(\tau)^T R u(\tau)\big) d\tau \tag{7a}$$

$$\text{s.t.} \quad \dot{x} = Ax + Bu_+ B_\mathrm{d} v \tag{7b}$$

$$y = Cx + Du + D_\mathrm{d} v \tag{7c}$$

where the weighting matrices $Q \in \mathbb{R}^{N_\mathrm{c} \times N_\mathrm{c}}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are respectively semi definite and definite positive matrices, t is the current time instant, T is the prediction horizon time length. Let $\mathbf{u}$ and $\mathbf{v}_\mathrm{ol}$ be temporal vectors, such that $u(\tau)$ and $v(\tau)$ are respectively the values of $\mathbf{u}$ and $\mathbf{v}_\mathrm{ol}$ at time instant τ over the prediction horizon $[t, t+T]$. In order to efficiently reduce the fatigue cost of the turbine, the MPC weighting parameters must be tuned accordingly. However, it should be noticed that the weighting matrices Q and R contain respectively $\frac{N_\mathrm{c}(N_\mathrm{c}+1)}{2}$ and $\frac{\mathrm{n_u}(\mathrm{n_u}+1)}{2}$ parameters, which leads to 13 parameters to be tuned, as $N_\mathrm{c} = 4$ and $\mathrm{n_u} = 2$. In order to reduce the computational complexity of the weighting matrices Q and R tuning, a relevant parameterization defined in [16] is proposed:

$$Q(\nu) = \mathrm{diag}\big([1, 1, \nu, \nu]\big) \qquad R(\nu) = I_{\mathrm{n_u}} \min(\nu, 1) \times 10^{-10} \tag{8}$$

where ν is the tuning parameter, which penalizes the actuation of the system. It should be noticed that this parameterization relies on the fact that the blades in rotating coordinates have all the same independent dynamics. Hence, the trade-off between pitch activity and blade fatigue loads alleviation must be weighted equally. Then, considering this trade-off in MBC yields the matrix $Q(\nu)$. The role of the matrix $R(\nu)$ is to prevent singularities in the OCP, as the penalization of the system pitch activity is already taken care of by the parameter ν. Therefore, the impact of R on the J value must be negligible. It is thus possible to generate realistic time series of a closed-loop HAWT provided relevant values of ν, T and disturbance trajectory for the closed-loop simulation, denoted by $\mathbf{v}_\mathrm{cl}$.

2.1.3. Realistic trajectories selection

In order that the fatigue-oriented cost function be relevant, it is necessary to consider realistic closed-loop controllers and exogenous disturbance trajectory $\mathbf{v}_\mathrm{cl}$ in the time series generation. For the exogenous disturbance trajectories, a set of N_d wind disturbances, denoted by $\{\mathbf{v}_\mathrm{cl}^{(1)}, \ldots, \mathbf{v}_\mathrm{cl}^{(N_\mathrm{d})}\}$, is generated using the NREL wind generator TurbSim [19], which allows to generate realistic random hub-height winds from given average wind speed and turbulence intensity. For the wind generation, the average wind speed is fixed to 12 m/s, as the LTI system is linearized around this wind speed, and 10000 turbulence intensities are randomly drawn in a realistic distribution of turbulence intensity, obtained from a 10 years long NREL measurement campaign [20]. An histogram of the turbulence intensity distribution is plotted in Figure 2.

In order to have a realistic and performing closed-loop controller, the MPC defined by (7) is implemented with a prediction horizon $T = 2$ s, as longer prediction horizons did not seem to improve further the control quality. The stage cost of this MPC is parmeterized by equation (8), and 50 parameters of ν are tested, spanning in a logarithmic space

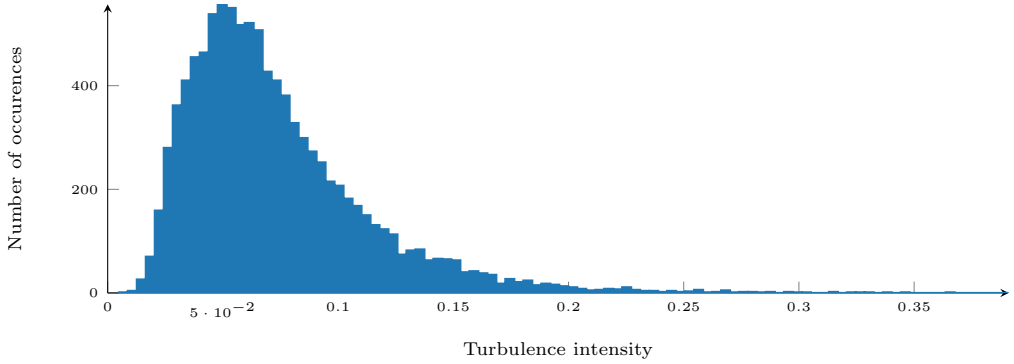

Figure 2: Histogram of the 10000 turbulence intensities used for the set of winds generation.

ranging from 10^1 to 10^4. In Figure 3 is plotted the evolution of the fatigue cost $\mathcal{J}$ with the ν parameter for several winds, normalized by the minimal fatigue cost $\mathcal{J}$ obtained for the corresponding wind. It can be observed that for a given wind, the fatigue cost $\mathcal{J}$ varies with the value of ν and it is not always the same value of ν which minimizes $\mathcal{J}$. Let $\mathcal{J}_{\text{mpc}}(\mathbf{v}_{\text{cl}}, \nu)$ be the fatigue cost obtained in closed-loop with the MPC parameterized by ν, under a given wind disturbance $\mathbf{v}_{\text{cl}}$. Let us define ν_{ideal} and ν_{mean}, the values of ν that respectively minimizes the fatigue cost for a wind disturbance $\mathbf{v}_{\text{cl}}$ and for the set of wind disturbances $\left\{\mathbf{v}_{\text{cl}}^{(1)}, \ldots, \mathbf{v}_{\text{cl}}^{(N_{\text{d}})}\right\}$:

$$\nu_{\text{ideal}}(\mathbf{v}_{\text{cl}}) = \arg\min_{\nu} \mathcal{J}_{\text{mpc}}(\mathbf{v}_{\text{cl}}, \nu) \tag{9a}$$

$$\nu_{\text{mean}} = \arg\min_{\nu} \sum_{i=1}^{N_{\text{d}}} \mathcal{J}_{\text{mpc}}(\mathbf{v}_{\text{cl}}^{(i)}, \nu) \tag{9b}$$

From Figure 3, it can be claimed that

$$\mathcal{J}_{\text{mpc}}(\mathbf{v}_{\text{cl}}, \nu_{\text{mean}}) \geq \mathcal{J}_{\text{mpc}}(\mathbf{v}_{\text{cl}}, \nu_{\text{ideal}}(\mathbf{v}_{\text{cl}}))$$

There is thus an interest in adapting the value of ν depending on $\mathbf{v}_{\text{cl}}$, in order to minimize the fatigue cost in all circumstances. Let MPC_{mean} and $\text{MPC}_{\text{ideal}}$ be controllers whose behaviors correspond to the MPC having respectively ν_{mean} (fixed) and ν_{ideal} (varying) for parameter. It is thus desirable to have a controller approximating the behavior of $\text{MPC}_{\text{ideal}}$ for every wind in the set, as the fatigue cost of $\text{MPC}_{\text{ideal}}$ is always lower or equal than the one of MPC_{mean}. Therefore, in order to approximate the fatigue-oriented cost function corresponding to $\text{MPC}_{\text{ideal}}$, only the time series generated with $\text{MPC}_{\text{ideal}}$ in closed-loop are kept for the fatigue-oriented cost function derivation. It should be noticed that $\text{MPC}_{\text{ideal}}$, as opposed to MPC_{mean}, is not an implementable controller, because the value of the parameter ν_{ideal} is given by an off-line estimation and not an on-line adaptation law.

2.2. Variance and fatigue damage estimations

The fatigue-oriented cost function defined in [16] is based on a data-driven relationship between variance and fatigue damage. The variance, denoted by J_{Var}, is defined as

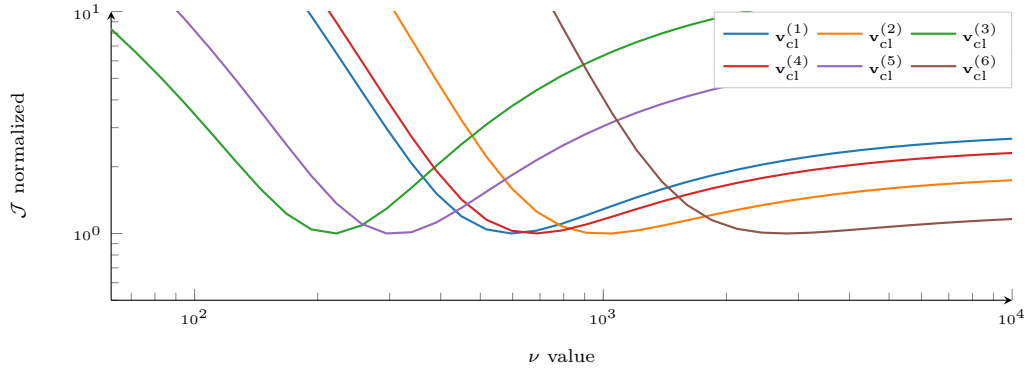

Figure 3: Plot of the fatigue cost $\mathcal{J}$ and the stage cost parameter ν for six different winds, normalized by the minimum fatigue cost $\mathcal{J}$ obtained for each wind, where $\mathbf{v}_{\mathrm{cl}}^{(i)}$ is the i^{th} wind realization.

Table 3: Summary of the parameters used for the estimation of the fatigue damages $\mathcal{D}_k$.

Component	k	L_k^{ult}	m_k
M_{yaw}	1	1000	10
M_{tilt}	2	1000	10
θ_{yaw}	3	$5\frac{8\pi}{180} \times 10^3$	4
θ_{tilt}	4	$5\frac{8\pi}{180} \times 10^3$	4

follows:

$$J_{\mathrm{Var}}(\mathbf{y}_k) = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} y_k(\tau)^2 d\tau - \mu\left(\mathbf{y}_k\right)^2 \tag{10a}$$

$$= \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \left(y_k(\tau) - \mu\left(\mathbf{y}_k\right)\right)^2 d\tau \tag{10b}$$

where

$$\mu\left(\mathbf{y}_k\right) = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} y_k(\tau) d\tau \tag{11}$$

where $y_k(\tau)$ is the value of the trajectory $\mathbf{y}_k$ at time instant τ, t_0 and t_f are respectively the initial and final time instants of the evaluated time series. The damage $\mathcal{D}_k$ is estimated from the time series $\mathbf{y}_k$ with the RFC algorithm [8] and Palmgrem-Miner rule [9, 10]. For the Palmgrem-Miner rule, an ultimate fatigue load and Wöhler exponent, denoted respectively by L_k^{ult} and m_k, must be defined for every components k. These values are selected in order to yield realistic fatigue damages and are summarized in Table 3. .

2.3. Fit of the regressor

For the regression of the fatigue damages $\mathcal{D}_k(\mathbf{y}_k)$ from $J_{\mathrm{Var}}(\mathbf{y}_k)$, it can be observed in Figure 4 that the relationship between the logarithms of the fatigue damage and J_{Var} is approximately linear. Therefore, a logarithmic transformation is performed on both J_{Var} and fatigue damage. Hence, a linear regression is fitted between the logarithms of J_{Var}

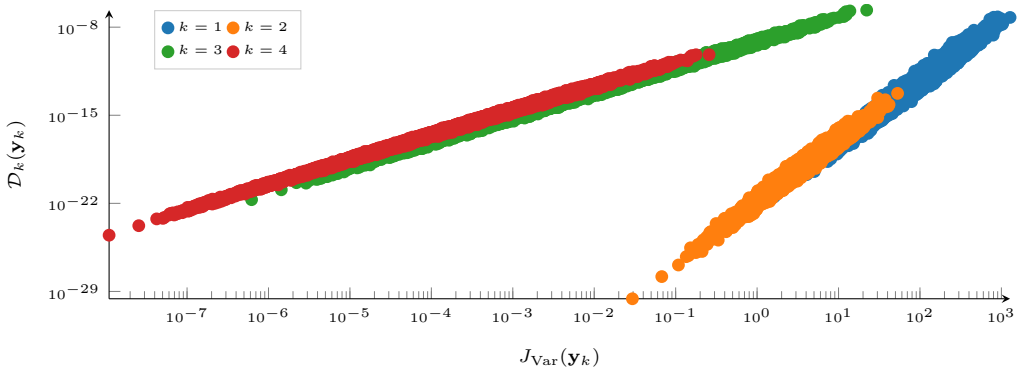

Figure 4: Scatter plot of the variance $J_{\mathrm{Var}}(\mathbf{y}_k)$ and fatigue damage $\mathcal{D}_k(\mathbf{y}_k)$ of each system outputs, obtained from the above described data generation process.

Table 4: Summary of the w_k and b_k values obtained from the regression in the cost function derivation.

Component	k	w_k	b_k	π_k
M_{yaw}	1	4.92	-58.05	1000
M_{tilt}	2	4.90	-57.13	1000
θ_{yaw}	3	1.98	-21.11	1
θ_{tilt}	4	1.96	-20.07	1

and the fatigue damage. Eventually, the predicted logarithm of fatigue damage, denoted by $\hat{\mathcal{D}}_k$ for the k^{th} component, can be expressed as follows:

$$\log\left(\hat{\mathcal{D}}_k\right) = w_k \log J_{\mathrm{Var}}(\mathbf{y}_k) + b_k \tag{12}$$

where w_k is the linear regression coefficient and b_k is the bias term. Therefore, from (1) and (12), $\mathcal{J}$ can be approximated by the fatigue-oriented cost function, denoted by $\hat{\mathcal{J}}$, which is expressed as follows:

$$\hat{\mathcal{J}}(\mathbf{y}) = \sum_{k=1}^{N_{\mathrm{c}}} \pi_k \underbrace{e^{b_k} J_{\mathrm{Var}}(\mathbf{y}_k)^{w_k}}_{\hat{\mathcal{D}}_k(\mathbf{y}_k)} \tag{13}$$

where the values of π_k, w_k and b_k are summarized in Table 4. Moreover, it can be noticed that, as the w_k values are all above 1, the function $\hat{\mathcal{J}}$ is convex, as a sum of compositions of convex functions. In the remainder of this article, the FO-OCP which uses the fatigue-oriented cost function $\hat{\mathcal{J}}$ as objective is presented and a solution allowing to efficiently optimize the FO-OCP in a MPC is detailed.

3. Fixed-point formulation: Quadratic stage cost adaptation

The derivation of the fatigue-oriented cost function $\hat{\mathcal{J}}$ allows to approximate an original OCP using the fatigue cost $\mathcal{J}$ as objective, by using $\hat{\mathcal{J}}$ as cost function. However,

the resulting FO-OCP is not a standard Quadratic Programming (QP) because $w_k \neq 1$ for several components:

$$\min_{\mathbf{u}} \quad \hat{\mathcal{J}}(\mathbf{y}) = \sum_{k=1}^{N_\mathrm{c}} \pi_k e^{b_k} J_{\mathrm{Var}}(\mathbf{y}_k)^{w_k} \tag{14a}$$

$$\text{s.t.} \quad \dot{x} = Ax + Bu + B_\mathrm{d} v \tag{14b}$$

$$y = Cx + Du + D_\mathrm{d} v \tag{14c}$$

where $J_{\mathrm{Var}}(\mathbf{y}_k)$, defined by (10), can be expressed as the integral of a quadratic form over time. In this section, a first order Taylor expansion of $\hat{\mathcal{J}}$ is first presented, showing that $\hat{\mathcal{J}}$ can be approximated with a quadratic form whose weighting matrices depend on an output trajectory $\mathbf{y}^{(i_\mathrm{iter})}$. Then, a fixed-point problem aiming at finding the right $\mathbf{y}^{(i_\mathrm{iter})}$ is derived.

3.1. Successive approximations of the FO-OCP

It is possible to solve the FO-OCP using an NLP solver, after its discretization in time. However, this black-box optimization might not take advantage from the quadratic nature of J_{Var} in the cost functional $\hat{\mathcal{J}}$. Indeed, quadratic forms are known to behave very well in OCP, allowing to have a convex and smooth cost function, with a constant Hessian in gradient descent algorithms. Let us approximate $\hat{\mathcal{J}}$ with a first order Taylor expansion around the trajectory $\mathbf{y}^{(i_\mathrm{iter})}$, where i_iter is the iteration number in the resolution of a fixed-point problem:

$$\begin{aligned}\hat{\mathcal{J}}^{(i_\mathrm{iter})}(\mathbf{y})) = \underbrace{\hat{\mathcal{J}}(\mathbf{y}_k^{(i_\mathrm{iter})})}_{\text{constant}} + \\ \sum_{k=1}^{N_\mathrm{c}} \pi_k \left(\underbrace{e^{b_k} w_k J_{\mathrm{Var}}(\mathbf{y}_k)^{w_k - 1}}_{\alpha_k\left(\mathbf{y}_k^{(i_\mathrm{iter})}\right)} \left(J_{\mathrm{Var}}(\mathbf{y}_k) - \underbrace{J_{\mathrm{Var}}(\mathbf{y}_k^{(i_\mathrm{iter})})}_{\text{constant}} \right) \right) \\ + \underbrace{\varepsilon\left(\mathbf{y}^{(i_\mathrm{iter})}\right)}_{\text{negligible}}\end{aligned} \tag{15}$$

Hence, assuming that $\mathbf{y}^{(i_\mathrm{iter})}$ is known, this leaves us with a linear combination of quadratic forms $J_{\mathrm{Var}}(\mathbf{y}_k)$, for $k \in \{1, \ldots, N_\mathrm{c}\}$. The original FO-OCP using $\mathcal{J}$ as a cost function can thus be approximated by a quadratic form whose weighting matrices

values depend on $\mathbf{y}^{(i_{\text{iter}})}$:

$$\min_{\mathbf{u}} \quad \begin{aligned} \hat{\mathcal{J}}^{(i_{\text{iter}})}(\mathbf{y}) = \int_{t_0}^{t_f} & \left(y(\tau)^T Q\left(\mathbf{y}^{(i_{\text{iter}})}\right) y(\tau) + \right. \\ & L\left(\mathbf{y}^{(i_{\text{iter}})}\right)^T y(\tau) + \\ & \left. u(\tau)^T R\left(\mathbf{y}^{(i_{\text{iter}})}\right) u(\tau) \right) d\tau \\ := & J_{\text{OCP}}\left(p\left(\mathbf{y}^{(i_{\text{iter}})}\right), \mathbf{y}\right) \end{aligned} \tag{16a}$$

$$\text{s.t.} \quad \dot{x} = Ax + Bu + B_{\text{d}} v \tag{16b}$$

$$y = Cx + Du + D_{\text{d}} v \tag{16c}$$

where Q, R and L matrices are respectively definite semi-positive, definite positive matrices and column vector parameterized by $\mathbf{y}^{(i_{\text{iter}})}$, and:

$$p\left(\mathbf{y}^{(i_{\text{iter}})}\right) = \left\{ Q\left(\mathbf{y}^{(i_{\text{iter}})}\right), R\left(\mathbf{y}^{(i_{\text{iter}})}\right), L\left(\mathbf{y}^{(i_{\text{iter}})}\right) \right\}$$

is an array of parameters defining the OCP. The matrices Q and R, along with the vector L, are derived from the α_k, defined in (15), and J_{Var} expressions:

$$Q\left(\mathbf{y}^{(i_{\text{iter}})}\right) = \begin{pmatrix} \pi_1 \alpha_1\left(\mathbf{y}_1^{(i_{\text{iter}})}\right) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \pi_{N_{\text{c}}} \alpha_{N_{\text{c}}}\left(\mathbf{y}_{N_{\text{c}}}^{(i_{\text{iter}})}\right) \end{pmatrix} \tag{17a}$$

$$\begin{aligned} R\left(\mathbf{y}^{(i_{\text{iter}})}\right) = & \min\left(\pi_1 \alpha_1\left(\mathbf{y}_1^{(i_{\text{iter}})}\right), \right. \\ & \left. \dots, \pi_{N_{\text{c}}} \alpha_{N_{\text{c}}}\left(\mathbf{y}_{N_{\text{c}}}^{(i_{\text{iter}})}\right)\right) \times \varepsilon I_{\text{n}_{\text{u}}} \end{aligned} \tag{17b}$$

$$L\left(\mathbf{y}^{(i_{\text{iter}})}\right) = -2 \begin{pmatrix} \pi_1 \alpha_1\left(\mathbf{y}_1^{(i_{\text{iter}})}\right) \mu(\mathbf{y}_1^{(i_{\text{iter}})}) \\ \vdots \\ \pi_{N_{\text{c}}} \alpha_{N_{\text{c}}}\left(\mathbf{y}_{N_{\text{c}}}^{(i_{\text{iter}})}\right) \mu(\mathbf{y}_{N_{\text{c}}}^{(i_{\text{iter}})}) \end{pmatrix} \tag{17c}$$

where $\varepsilon > 0$ must be small enough in order to avoid introducing bias in the approximation of $\hat{\mathcal{J}}$ with R. It should be noticed that R is introduced in (16) only in order to avoid singularities in its resolution. Therefore, the contribution of R in the value of $\hat{\mathcal{J}}^{(i_{\text{iter}})}$ must be negligible compared to the other terms. Hence, p depends on a given output trajectory $\mathbf{y}^{(i_{\text{iter}})}$, while the resulting output trajectory, denoted by $\mathbf{y}^{(i_{\text{iter}}+1)}$, depends on the value of $p(\mathbf{y}^{(i_{\text{iter}})})$ used in the OCP. The approximation of $\hat{\mathcal{J}}$ by J_{OCP} in the OCP defined by (16) is only accurate around $\mathbf{y}^{(i_{\text{iter}})}$. Hence, $\mathbf{y}^{(i_{\text{iter}})}$ must be chosen close to the solution of the FO-OCP. A schematization of the interdependence between $\mathbf{y}^{(i_{\text{iter}})}$ and p is given in Figure 5. This kind of problem is a fixed-point problem whose formulation is detailed in the next subsection.

3.2. Fixed-point problem derivation

For a given $\mathbf{y}^{(i_{\text{iter}})}$, we have an array of parameters p which parameterizes the quadratic cost function J_{OCP}. On the other hand, the solution output trajectory of the QP of (16)

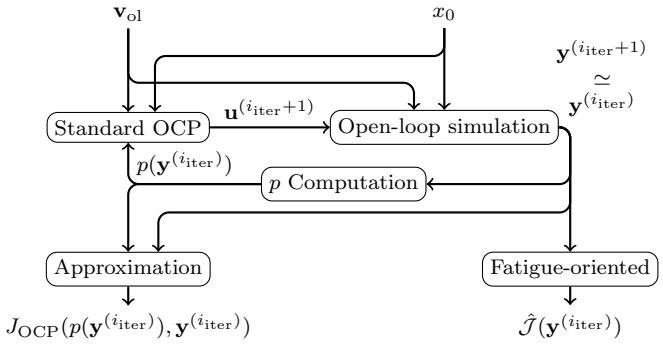

Figure 5: Schematization of the fatigue-oriented open-loop optimization using the fatigue-oriented cost function $\hat{\mathcal{J}}$, where $\mathbf{u}^{(i_{\text{iter}}+1)}$ and x_0 are respectively the solution and initial condition of the OCP.

depends on the value of p. There is thus an algebraic loop that consists in finding the right $\mathbf{y}^{(i_{\text{iter}})}$ and $p(\mathbf{y}^{(i_{\text{iter}})})$ for the FO-OCP approximation. This kind of algebraic loop can be solved with a fixed point algorithm defined by Algorithm 1, for given initial state x_0 and disturbance trajectory $\mathbf{v}_{\text{ol}}$.

3.3. Discussion on the fixed-point convergence

It is not proven that this fixed-point problem will converge. However, it can be assumed that if it does converge, it should converge to a local minimum or saddle point of the FO-OCP. From experience, it was observed that the fixed-point problem can have convergence issues and that filtering the fixed-point using an appropriate parameter $\beta \in]0, 1[$ allows to help the fixed-point in converging, as suggested in [21]. However, the parameter β can slow the convergence down but does not introduce any bias in the fixed-point solution. Hence, finding the global optimum with this method should mainly rely on the convexity of $\hat{\mathcal{J}}$ in the FO-OCP.

Algorithm 1 Fixed-point algorithm allowing to solve $\mathbf{y}^{(i_{\text{iter}})} = \mathbf{y}^{(i_{\text{iter}}+1)}$ for given initial state x_0, disturbance trajectory $\mathbf{v}_{\text{ol}}$ and filtering parameter β.

$i_{\text{iter}} \longleftarrow 0$
$\mathbf{y}^{(i_{\text{iter}})} \longleftarrow$ **Initialize** the output trajectory
$\mathbf{y}^{(i_{\text{iter}}+1)} \longleftarrow$ **Initialize** another output trajectory
while $\mathbf{y}^{(i_{\text{iter}}+1)} \neq \mathbf{y}^{(i_{\text{iter}})}$ **do**
  $i_{\text{iter}} \longleftarrow i_{\text{iter}} + 1$
  $p\left(\mathbf{y}^{(i_{\text{iter}})}\right) \longleftarrow$ **Solve** (17)
  $\mathbf{u}^{(i_{\text{iter}}+1)} \longleftarrow$ **Solve** (16)
  $\hat{\mathbf{y}}^{(i_{\text{iter}}+1)} \longleftarrow$ **Integrate** the system dynamics
  $\mathbf{y}^{(i_{\text{iter}}+1)} \longleftarrow (1-\beta)\hat{\mathbf{y}}^{(i_{\text{iter}}+1)} + \beta\mathbf{y}^{(i_{\text{iter}})}$
end while

The advantage of this method compared to the one solving directly the FO-OCP NLP is that it allows to:

- Express the OCP under a standard quadratic formulation
- Understand the role of the weighting matrices and how they should be adapted to reduce the fatigue cost

It was shown in [16] and [22] that this kind of data-driven quadratic cost function and open-loop optimization featuring a fixed-point resolution, allow to significantly reduce the fatigue cost $\mathcal{J}$ compared to OCP using quadratic forms as objective. It is thus interesting to use the FO-OCP in the internal control problem of a MPC, in order to have a controller able to efficiently reduce $\mathcal{J}$ in closed-loop. However, the computational cost of the fixed-point algorithm in the FO-OCP resolution makes its use in a MPC prohibitive. Therefore, an efficient quadratic MPC stage cost adaptation, based on the J_{OCP} weighting matrices adaptation in the fixed-point algorithm iteration (see equation (17) and Algorithm 1), is derived in the next section.

4. Filtered adaptive MPC formulation

This section presents an adaptive MPC formulation based on the FO-OCP resolution using the fixed-point algorithm 1. The proposed adaptive formulation, featuring several filtering mechanisms, allows an efficient reduction of $\hat{\mathcal{J}}$ when implemented in closed-loop.

It is clear that the natural and naive implementation of a MPC aiming at minimizing $\hat{\mathcal{J}}$ would be to directly use the FO-OCP as open-loop OCP. In the remainder of this article, this implementation is named $\mathrm{MPC}_{\mathrm{direct}}$ and will be used for comparison. However, the fixed-point formulation described in Section 3 highlights the fact that the FO-OCP consists in adapting the stage cost of a quadratic MPC, based on the variance of the optimal output trajectory, or more physically the level of vibration of the system:

$$\min_{\mathbf{u}} \quad J = \int_t^{t+T} \Big(y(\tau)^T Q\left(\mathbf{y}\right) y(\tau) + L\left(\mathbf{y}\right)^T y(\tau) + u(\tau)^T R\left(\mathbf{y}\right) u(\tau) \Big)\, d\tau \tag{18a}$$

$$\text{s.t.} \quad \dot{x} = Ax + Bu + B_{\mathrm{d}} v \tag{18b}$$

$$y = Cx + Du + D_{\mathrm{d}} v \tag{18c}$$

where the weighting matrices Q, L and R are functions of the output trajectory $\mathbf{y}$, which is the solution of the fixed point problem. This kind of open-loop optimization can be efficiently solved by NLP or QP solvers, depending on the nature of the constraints and dynamical system involved. Moreover, this kind of quadratic OCP is well known and widely used in the MPC literature. Therefore, $\mathrm{MPC}_{\mathrm{direct}}$ can result in an adaptive MPC which brutally changes its parameters value. This brutal changes are likely to prevent efficient reduction of $\hat{\mathcal{J}}$ when $\mathrm{MPC}_{\mathrm{direct}}$ is implemented in closed-loop. This is the reason why, along with the prohibitive computational cost, another formulation denoted by $\mathrm{MPC}_{\mathrm{filt}}$, limiting and smoothing the variations of a quadratic MPC parameters, and computationally efficient, is proposed. In order to limit the quadratic MPC stage cost variations, the proposed solutions are:

1. Increase the time length of the trajectory $\mathbf{y}$ for the evaluation of the weighting matrices Q, L and R.

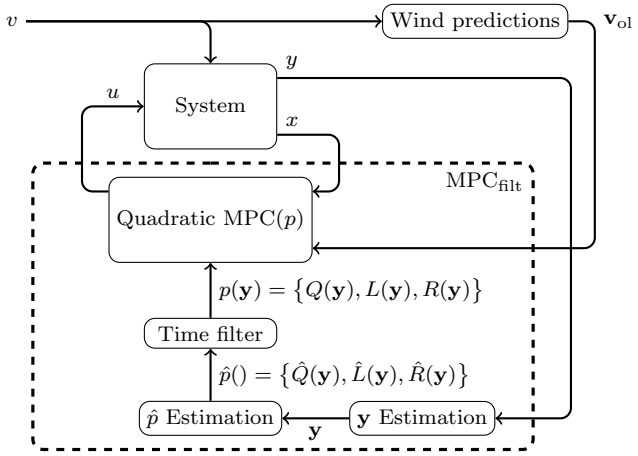

Figure 6: Schematization of the MPC_{filt} architecture, where the vector of parameters p which parameterizes a standard MPC, is a filtered function of the system output trajectory over the previous time instants.

2. Filter through time the variations of the updated matrices in order to limit the effects of noise or outliers on the weighting matrices estimation.

The fixed-point formulation shows implicitly that the stage cost of (18) depends on the variance, or current level of vibration of the closed-loop turbine, and increasing the time length of $\mathbf{y}$ aims at better approximating this level of vibration. However, increasing significantly the prediction horizon would result in increasing the computational burden and trust unreasonably the predictions of the internal model. As a solution for estimating the level of vibration, it is proposed to consider $\mathbf{y}$ on previous time instants of the simulation. Therefore, $\mathbf{y}$ is measured instead of predicted, but the variance estimation is slightly delayed. Moreover, $p(\mathbf{y})$ does not depend on the OCP solution anymore, which breaks the fixed-point problem. Therefore, the computational burden is reduced as the MPC optimization problem becomes the one defined by (18).

A schematization of MPC_{filt} is proposed in Figure 6. The standard MPC array of parameters p is obtained from the filtration of the array of parameters, denoted by $\hat{p}$, given by the approximation of the FO-OCP defined by (15). The derivation of $\hat{p}$ depends on the system output trajectory over the previous time instants, denoted by $\mathbf{y}$. The various operations necessary for the derivation of the matrices Q, L and R, contained in p, are depicted in the remainder of this subsection.

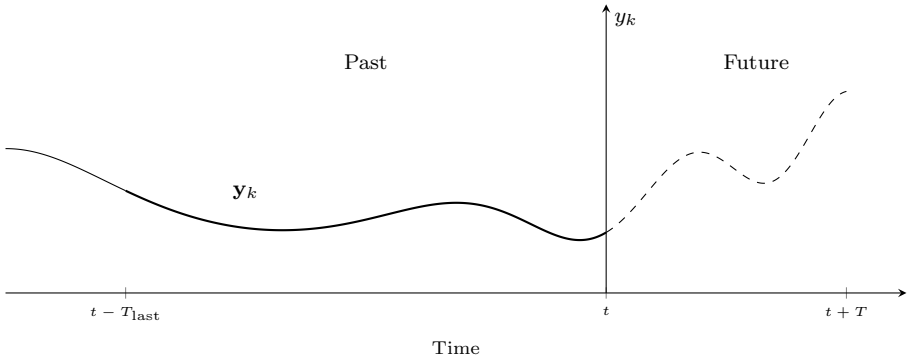

Figure 7: Illustration of the interval considered for evaluating $J_{\text{Var}}(\mathbf{y}_k)$ in MPC_{filt} (thick line), against the one considered in $\text{MPC}_{\text{direct}}$ (dashed line).

4.1. Derivation of the non-filtered array of parameter $\hat{\mathbf{p}}$

From the fixed-point formulation detailed in Section 3, the non-filtered updated values of the weighting matrices Q, L and R, denoted by $\hat{Q}$, $\hat{L}$ and $\hat{R}$, are expressed as follows:

$$\hat{Q}(\mathbf{y})_{ij} = \begin{cases} \pi_k \alpha_k(\mathbf{y}_k) & \text{if} \quad i = j = k \\ 0 & \text{otherwise} \end{cases} \tag{19a}$$

$$\hat{L}(\mathbf{y})_k = -2\pi_k \alpha_k(\mathbf{y}_k) \tag{19b}$$

$$\hat{R}(\mathbf{y}) = \min\left(\text{diag}\left(\hat{Q}(\mathbf{y})\right), 1\right) \times 10^{-3} I_2 \tag{19c}$$

$$\alpha_k\left(\mathbf{y}\right) = e^{b_k} w_k J_{\text{Var}}(\mathbf{y}_k)^{w_k - 1} \tag{19d}$$

where $\hat{Q}(\mathbf{y})_{ij}$ is the i^{th} row and j^{th} column of $\hat{Q}(\mathbf{y})$, $\hat{L}(\mathbf{y})_k$ is the k^{th} row of the vector $\hat{L}(\mathbf{y})$, π_k is the k^{th} component price of replacement and α_k, defined in (15), is a measure of the level of vibration. As the trajectory $\mathbf{y}$ is taken to be the last time instants of the current closed-loop simulation output trajectory, $\mathbf{y}$ is estimated over the interval $[t - T_{\text{last}}, t]$, where t is the current instant and T_{last} is the sliding parameter estimation horizon time length, as illustrated in Figure 7. It should be noticed that the longer T_{last} is, the more J_{Var} should be accurate, but the more delayed is the J_{Var} estimation.

In order to limit the influence of the instants close to $t - T_{\text{last}}$ for J_{Var} estimation in α_k computations, a weighted formulation is considered:

$$J_{\text{Var}}(\mathbf{y}_k) = \frac{1}{T(\xi, T_{\text{last}})} \int_{t-T_{\text{last}}}^{t} e^{\xi(\tau - t)} \Big(y_k(\tau) - \mu(\mathbf{y_k}, \xi, T_{\text{last}})\Big)^2 d\tau \tag{20}$$

where y_k is an instantaneous value of $\mathbf{y}_k$,

$$T(\xi, T_{\text{last}}) = \int_{t-T_{\text{last}}}^{t} e^{\xi(\tau - t)} d\tau$$

$$\mu(\mathbf{y}_k, \xi, T_{\text{last}}) = \frac{1}{T(\xi, T_{\text{last}})} \int_{t-T_{\text{last}}}^{t} e^{\xi(\tau - t)} y_k(\tau) d\tau$$

and $\xi \geq 0$ is the parameter of a weighting function, such that the time instants closer from t are weighted more importantly. It should be noticed that the greater is ξ, the more importance is given to the lasts time instants of the interval $[t - T_{\text{last}}, t]$.

4.2. Filtering of $\hat{p}$ in time

In order to further limit the variations of the weighting matrices Q, L and R due to noise or outliers in $\mathbf{y}$, the weighting matrices are filtered through time, similarly to the filtering step in the fixed-point iteration of Algorithm 1, that helps in converging:

$$Q(\mathbf{y}) = \beta Q_{\text{old}} + (1-\beta)\hat{Q}(\mathbf{y}) \tag{21a}$$

$$L(\mathbf{y}) = \beta L_{\text{old}} + (1-\beta)\hat{L}(\mathbf{y}) \tag{21b}$$

$$R(\mathbf{y}) = \beta R_{\text{old}} + (1-\beta)\hat{R}(\mathbf{y}) \tag{21c}$$

where Q_{old}, L_{old} and R_{old} are the values of Q, L and R obtained at their last update. A summary of the weighting matrices update for a given output trajectory $\mathbf{y}$ and old weighting matrices Q_{old}, L_{old} and R_{old} is given in Algorithm 2.

Algorithm 2 Gives Q, L and R values for a given output trajectory $\mathbf{y}$ and old weighting matrices Q_{old}, L_{old} and R_{old}.

$J_{\text{Var}}(\mathbf{y}_k) \longleftarrow$ Estimate the variance of the different process outputs using equation (20)
$\{\hat{Q}(\mathbf{y}), \hat{L}(\mathbf{y}), \hat{R}(\mathbf{y})\} \longleftarrow$ Update the weighting matrices parameterizing the stage cost with equation (19)
$\{Q(\mathbf{y}), L(\mathbf{y}), R(\mathbf{y})\} \longleftarrow$ Filter the updated weighting matrices through time with equation (21)

4.3. Parameters choice

For the implementation of the control strategy described in this subsection, few parameters must be chosen, i.e. T, β, T_{last} and ξ. The choice of these parameters is depicted in the sequel:

$T = 2$ **s:** The prediction horizon is taken to be 2 s in order to match the one taken for $\text{MPC}_{\text{ideal}}$ and MPC_{mean}. Moreover, ~ 2 s previews are several times considered in the IPC literature [3, 2, 23]. It can be justified by the fact that current remote wind speed measurement devices do not allow accurate previews of wind evolution on longer horizons.

$\beta = 0.95$**:** The time filtering parameter β is inherited from the filtering parameter in the fixed-point Algorithm 1, which was chosen such that the algorithm converges efficiently.

$T_{\text{last}} = 100$ **s,** $\xi = 0.03\ s^{-1}$**:** The parameters T_{last} and ξ were chosen such that the fatigue cost in closed-loop with the filtered MPC is minimized for an highly demanding disturbance.

It should be noticed that the validity of T_{last} and ξ might be limited to the considered dataset. A sensitivity study of MPC_{filt} performances to these parameters for further validation should be the subject of future works. It can also be added that MPC_{filt} depends also on the parameters taken for the fatigue cost function $\mathcal{J}$, i.e. the prices of replacement, the Wöhler exponents and the ultimate loads of the components considered. These parameters could be estimated by the HAWT manufacturer. Now that the adaptive fatigue-oriented adaptive MPC formulation is defined, it will be compared on its ability to reduce the fatigue cost $\mathcal{J}$ to a more classical non-adaptive MPC.

5. Results

The ultimate goal of MPC_{filt} is to efficiently reduce the fatigue cost $\mathcal{J}$ in closed-loop of a system. In order to assess the reduction of $\mathcal{J}$ efficiency, it is proposed in this section to compare the fatigue cost of closed-loop simulations where the HAWT is controlled with MPC_{filt} and benchmark controllers. Quadratic MPCs are the most advanced controllers in the literature of HAWT control for fatigue alleviation [3, 23, 11], therefore the proposed benchmark controllers are:

- MPC_{mean}, a finely tuned parameterized quadratic MPC, presented in Subsection 2.1.3, which is similar to the one designed in [3]
- $\text{MPC}_{\text{ideal}}$, an idealistic parameterized MPC presented in Subsection 2.1.3, whose parameter ν would adapt optimally based on the results of long term closed-loop simulations.

Moreover, a glance is given to the performance of $\text{MPC}_{\text{direct}}$, in order to justify the choice of not considering it further. First of all, the simulation settings used in this comparison is defined. Then, the MPCs are evaluated on their ability to reduce the fatigue-oriented cost $\hat{\mathcal{J}}$, which is the explicit objective function that the FO-OCP must minimize. Eventually, the MPCs are compared on their ability to reduce the fatigue cost $\mathcal{J}$.

5.1. Simulation settings

The simulation settings used for the MPCs comparison is the same as the one used for the time series generation described in Subsection 2.1 and summarized in Table 5. Note, that the simulation settings considered is simplified compared to the one conventionally used for controller validation in industry. This choice was made so that there is no model mismatch between the model used in simulation and the one considered in the MPCs internal optimization. This allows to isolate the fatigue reduction given by the adaptive MPC formulation, without mixing other issues in control systems implementation such as parameters uncertainty. The parameters used for the evaluation of the fatigue-oriented and fatigue cost, $\hat{\mathcal{J}}$ and $\mathcal{J}$, are summarized in Tables 3 and 4. Moreover, it should be noticed that the first 100 seconds of the 600 seconds long time series are removed in order to avoid any unexpected transient due to a bad initialization of one MPC.

Table 5: Summary of the simulation settings used for the MPCs comparison.

Feature	Description
HAWT simulator	MM82 Rotor LTI system in MBC linearized from FAST around 12 m/s
Wind disturbances	10000 TurbSim generated winds of random turbulence intensity and mean wind speed 12 m/s
Initial conditions	LTI system operating conditions
Constraints/Saturations	None
Model mismatch	None
Simulation length	600 s
Sampling time	0.1 s

5.2. Data-driven fatigue-oriented cost function reduction

The cost function that MPC_{filt} and $\text{MPC}_{\text{direct}}$ are supposed to minimize is the fatigue-oriented cost function $\hat{\mathcal{J}}$. As $\hat{\mathcal{J}}$ is an approximation of the fatigue cost $\mathcal{J}$, minimizing $\hat{\mathcal{J}}$ can allow to indirectly minimize $\mathcal{J}$, which is the ultimate objective. In Figure 8 is plotted the scatter of the fatigue-oriented cost $\hat{\mathcal{J}}$ obtained from closed-loop simulations with $\text{MPC}_{\text{ideal}}$, against $\hat{\mathcal{J}}$ with MPC_{filt} and $\text{MPC}_{\text{direct}}$. The black dashed line represents the bisector of the plane. It can be observed that MPC_{filt} allows to have $\hat{\mathcal{J}}$ value equivalent to the one of the fictitious $\text{MPC}_{\text{ideal}}$.

On the other hand for $\text{MPC}_{\text{direct}}$, it can be seen that $\text{MPC}_{\text{direct}}$, whose explicit objective in its open-loop OCP is to minimize $\hat{\mathcal{J}}$, does not succeed to efficiently reduce $\hat{\mathcal{J}}$ in closed-loop. It is fundamental to underline that this is a general drawback in MPC implementation that is rarely considered or analyzed in the MPC literature. To this extent, putting under light that this discrepancy between the open-loop and closed-loop performance can exist for this kind of objective function is a rather important contribution, that a broader audience than the one interested in wind turbine control should look at. Hence, $\text{MPC}_{\text{direct}}$ is not considered further in this study as all hope of efficient $\mathcal{J}$ reduction are lost with this controller in closed-loop.

The controller that would usually be implemented is MPC_{mean} rather than $\text{MPC}_{\text{ideal}}$, which has a fixed parameter ν_{mean}, allowing to minimize the fatigue cost $\mathcal{J}$ expectancy for a given wind distribution. In Figure 9 is plotted the scatter plot of the $\hat{\mathcal{J}}$ value obtained with MPC_{mean}, against the one obtained with MPC_{filt}. It can be noticed that MPC_{filt} allows to yield a lower $\hat{\mathcal{J}}$ than MPC_{mean} in more than 99% of cases, allowing to reduce the expectancy of $\hat{\mathcal{J}}$ of 26%. In summary, these observations let us expect that MPC_{filt} can efficiently reduce the fatigue cost $\mathcal{J}$ as it efficiently reduces $\hat{\mathcal{J}}$.

5.3. True RFC fatigue cost function reduction

The ultimate objective of a fatigue-oriented controller being to efficiently reduce the fatigue cost $\mathcal{J}$ expectancy, MPC_{filt} is compared in this subsection to $\text{MPC}_{\text{ideal}}$ and MPC_{mean} on its ability to reduce the fatigue cost $\mathcal{J}$. In Figure 10 is plotted the scatter of the fatigue cost $\mathcal{J}$ value obtained from closed-loop simulations with MPC_{filt} against the one obtained with $\text{MPC}_{\text{ideal}}$, under the 10000 wind disturbances generated. It can be observed that MPC_{filt} and $\text{MPC}_{\text{ideal}}$ have globally similar performances. However, there is an advantage to $\text{MPC}_{\text{ideal}}$, as MPC_{filt} increases the $\mathcal{J}$ expectancy of 46% compared to $\text{MPC}_{\text{ideal}}$. However, as $\text{MPC}_{\text{ideal}}$ is a fictitious ideal controller, the performances of

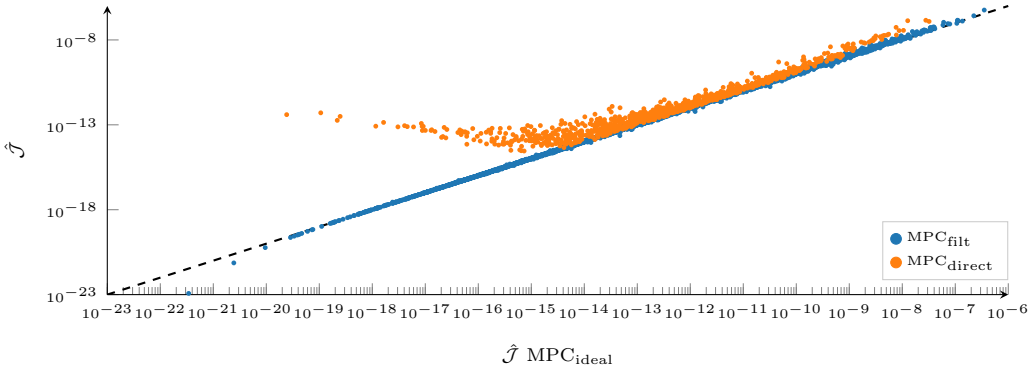

Figure 8: Scatter plot of the fatigue-oriented cost $\hat{\mathcal{J}}$ of the closed-loop MPC$_{\text{direct}}$ and MPC$_{\text{filt}}$ simulation against the one of MPC$_{\text{ideal}}$, under the 10000 winds generated.

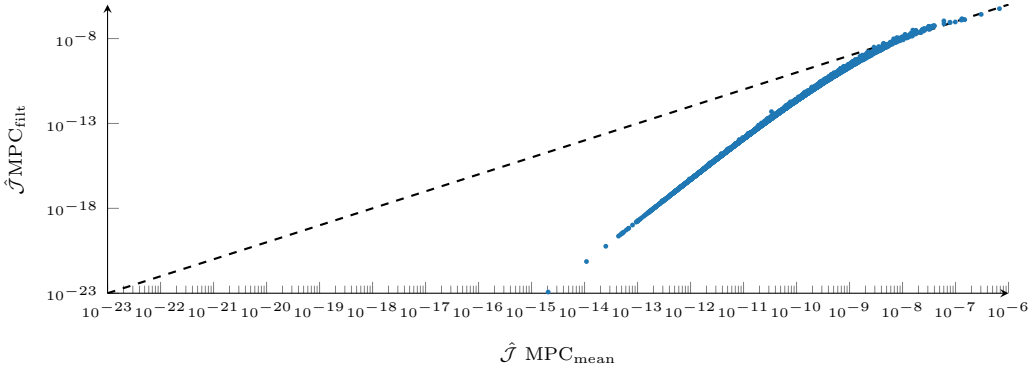

Figure 9: Scatter plot of the fatigue-oriented cost $\hat{\mathcal{J}}$ of the closed-loop MPC$_{\text{filt}}$ simulation against the one of MPC$_{\text{mean}}$, under the 10000 winds generated.

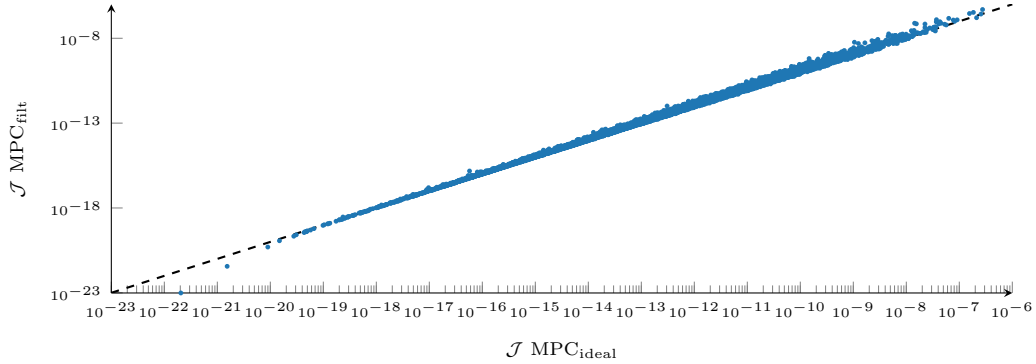

Figure 10: Scatter plot of the closed-loop fatigue costs $\mathcal{J}$ of MPC$_{\text{filt}}$ against the one of MPC$_{\text{ideal}}$, for the 10000 winds generated.

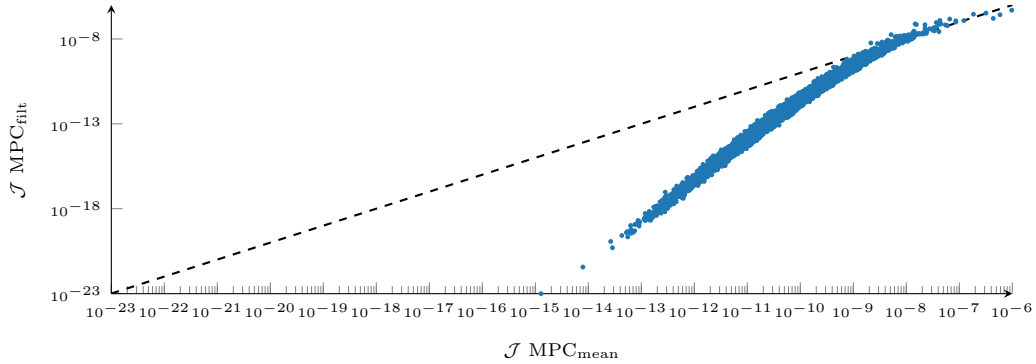

Figure 11: Scatter plot of the closed-loop fatigue costs $\mathcal{J}$ of MPC$_{\text{filt}}$ against the one of MPC$_{\text{mean}}$, for the 10000 winds generated.

MPC$_{\text{filt}}$ can be considered as satisfactory.

In Figure 11 is plotted the scatter of the $\mathcal{J}$ value obtained from closed-loop simulations with MPC$_{\text{filt}}$ against the one obtained with MPC$_{\text{mean}}$. It can be observed that MPC$_{\text{filt}}$ yields generally lower values of $\mathcal{J}$ than MPC$_{\text{mean}}$ and that MPC$_{\text{mean}}$ only matches the performances of MPC$_{\text{filt}}$ for the highest fatigue cost cases. More precisely, MPC$_{\text{filt}}$ allows to reduce $\mathcal{J}$ compared to MPC$_{\text{mean}}$ in more than 99% of the cases and reduce the fatigue cost expectancy of 27%, which is highly satisfactory. A summary of the fatigue-oriented and fatigue cost reductions obtained with MPC$_{\text{filt}}$ and MPC$_{\text{direct}}$ compared to MPC$_{\text{mean}}$ and MPC$_{\text{ideal}}$ is given in Table 6.

The goal of MPC$_{\text{filt}}$ was to approach the fatigue reduction performances of MPC$_{\text{ideal}}$, while being implementable in real life, which is not the case of MPC$_{\text{ideal}}$. It is important to stress out that MPC$_{\text{filt}}$ can yield a lower fatigue cost expectancy than a state-of-the-art HAWT IPC controller such as MPC$_{\text{mean}}$. This is possible thanks to the efficient adaptation of its cost function parameters based on a measure of the current vibrational level of the HAWT, i.e. the variance J_{var} of its various components.

Table 6: Summary of the reductions obtained for the data-driven fatigue-oriented cost function $\hat{\mathcal{J}}$ and the fatigue cost function $\mathcal{J}$, compared to the baseline controllers MPC_{mean} and $\text{MPC}_{\text{ideal}}$.

Cost function	Controller	MPC_{mean}	$\text{MPC}_{\text{ideal}}$
$\hat{\mathcal{J}}$	MPC_{filt}	26%	-23%
	$\text{MPC}_{\text{direct}}$	-117%	-148%
$\mathcal{J}$	MPC_{filt}	27%	-46%

6. Conclusion & Perspectives

This paper proposes a fatigue-oriented adaptive MPC whose adaptation law is based on a fixed-point formulation derived from an OCP using a data-driven fatigue-oriented cost function $\hat{\mathcal{J}}$ as objective. It was shown that the proposed controller allows a 27% reduction of the fatigue cost $\mathcal{J}$, compared to a non-adaptive MPC, finely tuned in order to efficiently reduce $\mathcal{J}$ expectancy. Besides, the proposed MPC formulation allows to better reduce $\hat{\mathcal{J}}$ in closed-loop simulations than a MPC using directly the FO-OCP as its internal open-loop OCP. This observation highlights that it is possible that a MPC do not allow to efficiently reduce the objective given in the open-loop OCP once implemented in closed-loop, even in absence of model mismatch.

The results generated in this paper featured important assumptions, such as the HAWT is modeled as an LTI system without model mismatch, or the wind is only summarized by its hub-height wind speed and perfectly known in advance. In order to fully estimate the potential of this approach in realistic conditions, the following changes must be realized in future studies. The MPCs must be simulated in closed-loop with an HAWT simulator and under a more realistic wind disturbance. It should be noticed that the above changes could affect the performances of the proposed adaptive MPC, as well as the one of the baseline controller used for comparison. Concerning the formulation of the fatigue cost $\mathcal{J}$, the fatigue related to the yawing and tilting blade root bending moments and blade pitch angles has very few physical meaning. Nevertheless, it does not undermine the genericity of this approach and the method should be able to adapt itself efficiently to different fatigue cost functions.

The interest of the presented method, which is more generally the one of fatigue-oriented approaches is to directly consider fatigue damages in the objective, which allows an efficient optimization of the fatigue cost $\mathcal{J}$, more meaningful and explicit than quadratic cost functions. It was shown in this article that the proposed MPC has very good performances compared to a finely tuned non-adaptive MPC, with minimal tuning effort.

References

[1] E. Bossanyi, "Individual blade pitch control for load reduction," *Wind Energy*, vol. 6, pp. 119–128, 2003.

[2] D. Schlipf, S. Schuler, P. Grau, F. Allgöwer, and M. Kühn, "Look-ahead cyclic pitch control using LiDAR," Proc. The Science of Making Torque from Wind, 2010.

[3] M. Mirzaei, M. Soltani, N. Poulsen, and H. Niemann, "An MPC approach to individual pitch control of wind turbines using uncertain LiDAR measurements," pp. 490–495, Proc. European Control Conference (ECC), 2013.

[4] V. Pettas, M. Salari, D. Schlipf, and P. Cheng, "Investigation on the potential of individual blade control for lifetime extension," vol. 1037, p. 032006, Journal of Physics: Conference Series, 2018.

[5] D. Collet, D. Di Domenico, G. Sabiron, and M. Alamir, "A data-driven approach for fatigue-based individual blade pitch controller selection from wind conditions," pp. 3500–3505, 2019 American Control Conference (ACC), 2019.

[6] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[7] J. Barradas-Berglind and R. Wisniewski, "Representation of fatigue for wind turbine control," *Wind Energy*, vol. 19, pp. 2189–2203, 2016.

[8] S. Downing and D. Socie, "Simple rainflow counting algorithms," *International journal of fatigue*, vol. 4, pp. 31–40, 1982.

[9] A. Palmgren, "Die lebensdauer von kugellagern.," *Zeitshrift des Vereines Duetsher Ingenieure*, vol. 68, p. 339, 1924.

[10] M. Miner, "Cumulative damage in fatigue," *Journal of Applied Mechanics*, vol. 3, p. 159, 1945.

[11] S. Loew, D. Obradovic, and C. Bottasso, "Direct Online Rainflow-counting and indirect fatigue penalization methods for Model Predictive Control," pp. 3371–3376, Proc. European Control Conference (ECC), 2019.

[12] S. Löw, A. Anand, and D. Obradovic, "Stage costs formulations of online rainflow-counting for model predictive control of fatigue."

[13] J. Barradas-Berglind, R. Wisniewski, and M. Soltani, "Fatigue damage estimation and data-based control for wind turbines," *IET Control Theory & Applications*, vol. 9, pp. 1042–1050, 2015.

[14] I. Rychlik, "Note on cycle counts in irregular loads," *Fatigue & Fracture of Engineering Materials & Structures*, vol. 16, no. 4, pp. 377–390, 1993.

[15] D. Benasciutti and R. Tovo, "Spectral methods for lifetime prediction under wide-band stationary random processes," *International Journal of fatigue*, vol. 27, no. 8, pp. 867–877, 2005.

[16] D. Collet, M. Alamir, D. Di Domenico, and G. Sabiron, "A fatigue-oriented cost function for optimal individual pitch control of wind turbines," In Proc. of IFAC 2020.

[17] J. Jonkman *et al.*, "Dynamics modeling and loads analysis of an offshore floating wind turbine," *National Renewable Energy Laboratory*, 2007.

[18] G. Bir, "Multi-blade coordinate transformation and its application to wind turbine analysis," p. 1300, Proc. 46th AIAA aerospace sciences meeting and exhibit, 2008.

[19] N. Kelley and B. Jonkman, "NWTC computer-aided engineering tools (TurbSim)," *Last Modified*, vol. 442, 2013.

[20] D. Jager and A. Andreas, "NREL national wind technology center (NWTC): M2 tower; boulder, colorado (data)," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 1996.

[21] M. Alamir, P. Bonnay, F. Bonne, and V. Trinh, "Fixed-point based hierarchical MPC control design for a cryogenic refrigerator," *Journal of Process Control*, vol. 58, pp. 117–130, 2017.

[22] D. Collet, M. Alamir, D. Di Domenico, and G. Sabiron, "Non quadratic smooth model of fatigue for optimal fatigue-oriented Individual Pitch Control." Unpublished, submitted to TORQUE 2020, 2020.

[23] S. Raach, D. Schlipf, F. Sandner, D. Matha, and P. W. Cheng, "Nonlinear model predictive control of floating wind turbines with individual pitch control," pp. 4434–4439, Proc. American Control Conference (ACC), 2014.