Fatigue-oriented data-driven individual pitch control strategies of wind turbines

David Collet

UNIVERSITÉ GRENOBLE ALPES

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE ALPES

Spécialité : **Automatique – Productique**

Arrêté ministériel : 7 août 2006

Présentée par
David COLLET

Thèse dirigée par **Mazen ALAMIR**, Directeur de recherche CNRS, Université Grenoble Alpes

préparée au sein du
Laboratoire Grenoble Images Parole Signal Automatique (GIPSA-Lab)

dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)**

Fatigue-oriented data-driven Individual Pitch Control strategies for wind turbines

Thèse soutenue publiquement le **4 décembre 2020**,
devant le jury composé de :

Pr Jan Willem VAN WINGERDEN
Professeur, TU Delft, Rapporteur
Pr Hervé GUEGUEN
Professeur, CentraleSupelec, Rapporteur
Pr Nicolas PETIT
Professeur, Mines ParisTech, Examinateur
Pr Gildas BESANCON
Professeur, Université Grenoble Alpes, Président du jury
M. Domenico DI DOMENICO
Ingénieur de recherche, IFP Energies Nouvelles, Invité
M. Guillaume SABIRON
Ingénieur de recherche, IFP Energies Nouvelles, Invité
Pr Mazen ALAMIR
Professeur, Université Grenoble Alpes, Directeur de thèse

To Fanny and Paulette

Remerciements

Je tiens tout d'abord à remercier mon directeur de thèse, Monsieur Mazen Alamir qui a toujours répondu présent dans les bons comme les mauvais moments, avec qui j'ai particulièrement apprécié travailler lors de mon séjour à Grenoble et qui m'a grandement inspiré lors de ces trois années. Ensuite, je souhaiterai bien sûr remercier mon encadrement IFPEN, Messieurs Domenico Di Domenico et Guillaume Sabiron, qui ont su me relever et me motiver lorsque je suis tombé dans des écueils, et ont été proches de moi quotidiennement.

J'aimerais aussi exprimer toute ma gratitude à Yann Creff ainsi que toute l'équipe Contrôle, Signal et Système d'IFPEN pour leur accueil durant ces trois années. Je remercie tout particulièrement mes collègues de bureaux qui se sont succédés : l'ancien Johan, les petits jeunes Bassel, Adam et Eduardo, et enfin Lauriane avec qui nous nous sommes mutuellement soutenus du début jusqu'à la fin de nos thèses.

Un grand merci également au département automatique du GIPSA-Lab de m'avoir accueilli pendant trois mois et à tous les collègues qui m'ont accueilli à bras ouverts : Mariana, Kaouther, Makia, Chay, Sohaib, Dimitri, Esteban, Raj, Duc et j'espère que je n'oublie personne. Cela aura été vraiment trois mois très sympathiques passés parmi vous.

I particularly thank Professors van Wingerden and Gueguen, who honoured me to be my reporters, along with Professor Nicolas Petit and Gildas Besançon for being respectively the examiner and president of this thesis committee.

Enfin, j'aimerais remercier mes amis, ma famille et mes parents particulièrement – j'espère que vous êtes fiers – pour votre soutient moral et pour m'avoir poussé. Je ne serais pas ici aujourd'hui sans vous. Bien sûr un grand merci à Fanny et ma petite Paulette pour m'avoir supporté au quotidien pendant ces trois mois de confinement à télétravailler et rédiger, et avoir illuminé ces journées qui auraient pu être mornes.

Résumé de la thèse

Les éoliennes à axe horizontal, qui sont la figure de proue de l'énergie éolienne sont devenues une technologie mature. Bien que l'éolien ait une tendance mondiale à la hausse, l'accroissement de la capacité installée faiblit dans certains pays, à cause des réductions des subventions. Dans un contexte de réchauffement climatique et de transition énergétique, il est de première importance d'optimiser le coût de l'éolien, de manière à rendre cette énergie compétitive face aux énergies fossiles, et ainsi permettre un développement de l'éolien moins dépendant des subventions économiques. Le contrôle des éoliennes peut fortement contribuer à la réponse à cette problématique.

Le contrôle individuel des pâles (IPC) d'éoliennes peut permettre de modifier les propriétés aérodynamiques du rotor, et ainsi réguler les charges déséquilibrées induites par des vents asymétriques. Réguler ces déséquilibres peut aider à réduire la fatigue des pièces de l'éoliennes en rotation, telle que les pâles. Par ailleurs, le contrôle IPC est connu pour accroître l'activité des actionneurs de pâles, induisant une fatigue additionnelle sur les actionneurs ainsi que les roulements portant les pâles. Par conséquent, le contrôle IPC peut avoir des effets positifs sur la fatigue de certaines pièces, mais négatifs sur d'autres. Pour optimiser le coût de l'énergie éolienne, il est nécessaire qu'un contrôleur IPC soit optimisé de manière à pondérer efficacement le compromis entre les fatigues des divers composants de l'éolienne.
Pour répondre à cette problématique, une fonction de coût de la fatigue a été définie comme la somme pondérée de la fatigue de divers composants d'une éolienne. Un contrôleur IPC doit être conçu de manière à minimiser l'espérance du coût de la fatigue. Cependant, l'optimisation de la fatigue est une tâche complexe, car son expression en tant que fonction de coût ne suit pas les formes conventionnelles. Il est montré plusieurs fois au cours de cette thèse que les stratégies de contrôle conventionnelles limitent le potentiel de réduction de la fatigue. Par ailleurs, combiner plusieurs stratégies de contrôle conventionnelles paramétrées différemment pourrait permettre des réductions significatives de l'espérance du coût de la fatigue, par rapport à des stratégies conventionnelles à paramètre fixes. La problématique adressée dans cette thèse est donc l'adaptation des paramètres de stratégies de contrôle IPC conventionnelles, pour une réduction efficace de la fatigue d'éoliennes. Deux approches sont ainsi développées par la suite.

La première consiste à approcher le coût de la fatigue par une fonction de coût orientée fatigue, grâce à une identification basée sur des données. Cette fonction de coût orientée fatigue utilisée dans un problème de contrôle optimal permet de paramétrer un problème d'optimisation standard, qui approche le problème d'optimisation de la fatigue autour de son optimum pour des conditions de vent données. Le problème d'optimisation orienté fatigue permet une réduction efficace du coût de la fatigue.
La commande prédictive (MPC) est une stratégie de contrôle qui permet d'optimiser une fonction de coût spécifique, en résolvant en ligne un problème de contrôle optimal. Ainsi, un contrôleur IPC MPC est obtenu à partir de l'expression du problème d'optimisation orienté fatigue. Ces contrôleurs ont ensuite été mis en œuvre en boucle fermée sur un modèle d'éolienne simplifié, et ont montré un grand potentiel de réduction de l'espérance du coût de la fatigue, par rapport à un contrôleur MPC à paramètres fixés.

La seconde solution est une méthodologie où un superviseur basé sur l'apprentissage, sélectionne les paramètres de contrôleurs candidats à partir des conditions de vent, de manière à réduire efficacement le coût de la fatigue d'une éolienne. Une preuve de concept comportant un superviseur simple a montré que des réductions significatives de la fatigue sont possibles, ce qui encourage aussi le développement de cette seconde approche.

Abstract

Horizontal axis wind turbines are the leading choice in the wind energy industry and are a mature technology. Although wind energy is globally trending, the yearly number of wind turbines installed is decreasing in some countries, and this would seem to be related with reductions in government subsidies. In a context of global warming and energy transition, it is of primal importance to optimize the levelized cost of wind energy in order to make it economically competitive against fossil energy sources and allow wind energy to be less reliant on subsidies. The control of wind turbines can significantly contribute to this challenge.

Individual Pitch Control (IPC) of wind turbines blades can allow us to modify the aerodynamic properties of the rotor and regulate unbalanced loads due to skewed wind patterns in the rotor plane. Regulating these unbalanced loads can help in alleviating fatigue damage to the rotating components of the turbine, such as the blades. On the other hand, IPC is known to induce oscillating loads on the blade pitch actuators, and to increase their excursions, which causes additional fatigue damage to blade pitch actuators and blade bearings. Therefore, IPC can have positive effects on fatigue damage to some components of the turbine, while having negative effects on others. For an efficient optimization of the levelized cost of wind energy, it is necessary that IPC be accurately designed in order to efficiently manage the trade-offs in terms of fatigue damage to various components of a turbine.
To address this issue, a fatigue cost function is defined as a weighted sum of wind turbine components fatigue damage, and possibly economic parameters. An IPC regulator must therefore be designed in order to minimize this fatigue cost. However, the optimization of fatigue is a challenging task, as the fatigue damage formula does not suit standard forms. It is shown several times in this thesis that standard control strategies have a limiting effect on the potential reduction of the fatigue cost expectancy. Moreover, combining several standard control strategies, each designed with different parameters, could allow significant reductions in estimated fatigue costs, compared to standard IPC control strategies with fixed parameters. The challenge addressed in this thesis is thus to adapt the parameters of standard IPC control strategies to efficiently reduce the estimated fatigue cost of wind turbines. Two approaches are developed in order to address this issue.

The first consists in combining the fatigue cost function with a fatigue-oriented one, using data-driven identification based on parameterized quadratic forms. This fatigue-oriented cost function used in an optimal control problem allows us to efficiently parameterize a standard optimization problem, which approximates the fatigue cost optimization problem around its optimum. Therefore, the fatigue-oriented optimal control problem allows efficient reductions in the estimated fatigue cost.
Model Predictive Control (MPC) is a control strategy which allows us to optimize a specific cost function by the online solving of an optimal control problem. An IPC MPC is then generated, based on the fatigue-oriented optimal control problem formula. These controllers are thus standard MPCs whose parameters are adapted for fatigue cost reduction purposes. These controllers are then implemented in a closed loop with a simplified wind turbine model and have shown great potential in reducing estimated fatigue costs, compared to an MPC with fixed parameters.

The second approach is a framework where a supervisory layer selects the parameters of candidate controllers based on wind conditions, in order to efficiently minimize estimated wind turbine fatigue cost. A proof of concept made with a simple supervisory layer showed that significant reductions in fatigue costs are already possible, which also encourages us to develop this second approach further.

Table des matières

Table des figures

Liste des tableaux

List of acronyms

BEM Blade Element Momentum.

COP21 21st Conference Of Parties.
CPC Collective Pitch Control.

DAE Differential Algebraic Equation.
DEL Damage Equivalent Load.
DOF Degrees of Freedom.
DORFC Direct Online RainFlow Counting.

FAST Fatigue, Aerodynamics, Structures and Turbulence.
FO-OCP Fatigue-Oriented Optimal Control Problem.
FOWT Floating Offshore Wind Turbine.

HAWT Horizontal Axis Wind Turbine.

IBC Individual Blade Control.
IPC Individual Pitch Control.

LAC LiDAR Assisted Control.
LCOE Levelized Cost Of Energy.
LiDAR Light Detection And Ranging.
LPV Linear Parameter Varying.
LQR Linear Quadratic Regulator.
LTI Linear Time Invariant.
LTV Linear Time Varying.

MBC Multi-Blade Coordinate.
MBML Multi-Blade Multi-Lag.
MIMO Multiple Inputs Multiple Outputs.
MMPC Multiple Model Predictive Control.
MPC Model Predictive Contro.

NLP NonLinear Programming.
NREL National Renewable Energy Laboratory.
NWTC National Wind Technology Center.

O&M Operation and Maintenance.

PCA Principal Component Analysis.
PDE Partial Differential Equation.
PI Proportional Integral.
PID Proportional Integral Derivative.

QP Quadratic Programming.

RATI Rotor Averaged Turbulence Intensity.
RAWS Rotor Averaged Wind Speed.
RBF Radial Basis Function.
RFC RainFlow Counting.

SISO Single Input Single Output.
SQP Sequential Quadratic Programming.

WFR Wind Field Reconstruction.
WTC Wind Turbine Control.

Chapitre 1

Introduction & State of the art

Wind is defined as 'air in natural motion' due to the atmospheric pressure differential, surface roughness and the Coriolis effect. Air being a fluid, wind power is related to the cube of its speed (which can reach 500 km/h in the most severe tornadoes) and the shape of the object it is acting on. Wind is able to blow down houses, level entire forests and change the shape of mountains. Mankind has used wind energy since earliest antiquity, starting with sails to move ships on water. In the early Middle Ages, windmills were invented to mill grain in arid regions where watermills could not be constructed.
More recently, in 1887 a vertical axis windmill was used to produce electricity in Scotland (see Figure 1.1). In the meantime, in Ohio, the first horizontal axis wind turbine was invented, and by 1891 the first power-regulated wind turbine was operating in Denmark to power the village of Askov (see Figure 1.2). In those days, wind turbines were designed such that their rotor rotational power stalled in high winds, but their aerodynamic efficiency could in no way compared to the design that can be found nowadays [1]. Subsequently, wind energy developed slowly until the 1973 oil price crisis, when countries such as the USA and Denmark started to upgrade their wind turbines to the multi-megawatt scale.

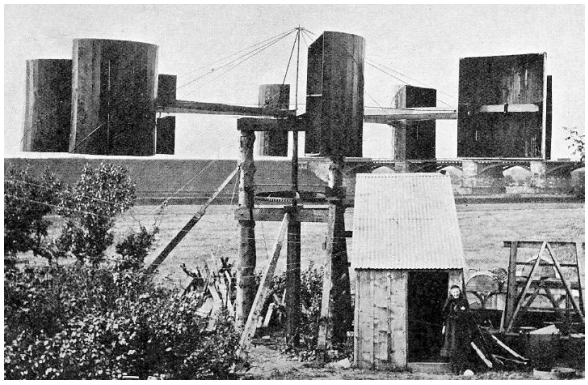

FIGURE 1.1 – First vertical axis wind turbine in Marykirk, Scotland, photographed in 1891. (www.taplondon.co.uk)

FIGURE 1.2 – First horizontal axis wind turbine in Askov, Denmark, photographed in 1897. (www.windsofchange.dk)

In such powerful structures, the old mechanical form of auto regulation was not sufficient, and therefore wind turbine manufacturers started to equip their turbines with blade pitch actuators. This allowed blades to be pitch-regulated on their own axis to vary the aerodynamic properties of the rotor, and so blade pitch control was born. At this time, the actuators allowed every blade to be pitched to the same angle. More recently a new generation of pitch actuator has emerged, allowing an independent pitch angle for each blade. Thanks to this new feature, it is possible to skew the rotor's aerodynamic properties by pitching the blades independently. And so Individual Pitch Control (IPC) was born. In the meantime, wind turbine rotor diameters have increased to reach 220 meters for offshore turbines, and 158 meters for onshore (see Figure 1.3). With such large diameters in play, the wind field over the rotor is skewed due to phenomena such as ground roughness, convection, atmospheric instability, terrain topography or even other Horizontal Axis Wind Turbines (HAWT) in a wind farm. This is illustrated in Figure 1.4, where special atmospheric conditions allowed us to view the eddies caused by the upstream HAWT and impacting the downstream turbines. This skew causes imbalances in the loads on the rotor and can also cause fatigue in rotational elements. IPC is therefore a trending research topic in light of the efforts to counteract asymmetric wind field effects by skewing the aerodynamic properties of the rotor.

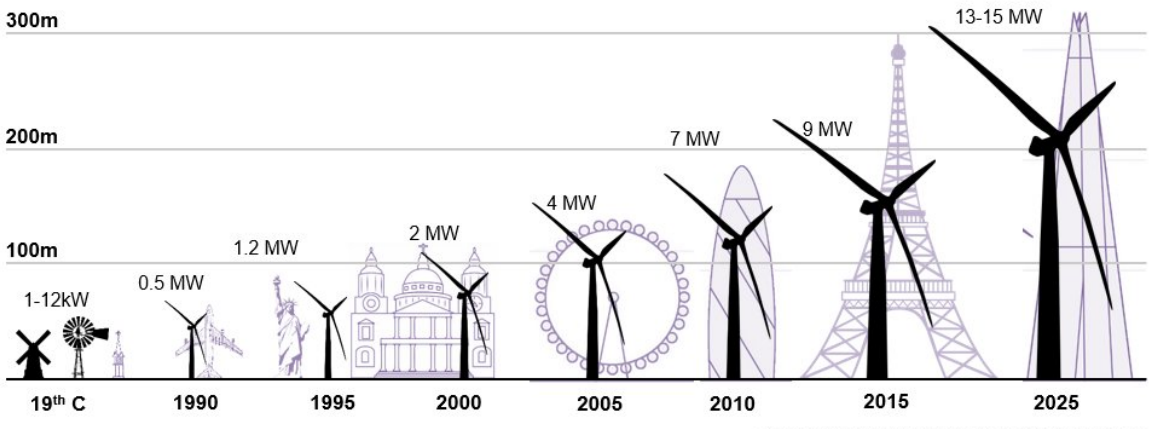

FIGURE 1.3 – HAWT size and power over time. (Bloomberg New Energy Finance)

Beginning in the early 21st century, concerns over global warming have led to the industrial development of wind energy in many countries over the world. Global installed capacity has multiplied by five in the last decade, now exceeding 500 GW [2]. Nowadays, annual investment in the wind power industry represents tens of billions of euros. In order to achieve the 21st Conference Of Parties (COP21) objective, which is to maintain CO_2 (carbon dioxide) emissions under 10^{14} kg until 2100, the wind energy industry is expected to develop even further. The latest report of the Global Wind Energy Council [3] predicts that wind power capacity could reach 2110 GW by 2030, with annual investment reaching 200 billion euros. With such figures, any improvement in HAWT Operation and Maintenance (O&M) would be welcome, and could result in decreased Levelized Cost Of Energy (LCOE) for wind power, saving millions, even billions of euros.
The two methods most frequently employed to decrease LCOE are to maximize the energy collected either by maximizing the running time (e.g. optimizing HAWT predictive maintenance or maximizing power output during operation), or by maximizing HAWT lifespan, i.e. minimizing HAWT fatigue damage in operating conditions (e.g. minimizing fatigue loads on the turbine).

Wind Turbine Control (WTC) optimization can contribute to both of these methods. This thesis addresses the minimization of HAWT fatigue damage. However, as will be clear by the end of this chapter, the optimization of fatigue damage is a challenging task because of the fatigue theory formula, which does not suit standard forms.

FIGURE 1.4 – Visualization of turbulence and eddies that a HAWT can endure. (Christian Steiness)

In the rest of this introduction, IPC control of three-bladed HAWT is introduced with its different frames of coordinates in Section 1.1, where it will be emphasized that IPC is particularly suited to reducing rotor fatigue costs. Next, the state of the art in control-oriented HAWT models and advanced IPC control strategies is presented in Section 1.2. In Section 1.3, fatigue theory is detailed with the state of the art in fatigue-oriented optimal control strategies. Finally, the issue addressed by this thesis is formulated in Section 1.4.

1.1 Introduction to Individual Pitch Control of wind turbines

As mentioned above, wind is a very powerful and complex force. The first HAWT, which were closer to windmills than modern turbines, could be auto-regulated by their rotor aerodynamics. However, the development of wind energy required the addition of a controller to HAWT in order to inject a cleaner power signal into the electricity network, in order to reduce LCOE and prevent failures. Today's HAWT are structurally designed to run at speeds ranging from nominal wind speed v_{nom} up to cut-off wind speed v_{out}, whose values are roughly 10 and 25 m/s respectively in commercial multi-megawatt HAWT. The HAWT generator is limited by its nominal torque T_{nom}, whose value is around 10 kN.m on multi-megawatt turbines, and designed to run at nominal power P_{nom}. The HAWT rotor has high inertia and is limited to its nominal rotational speed, denoted by ω_{nom}, whose value ranges between 12 and 18 rpm on commercial turbines. In order to limit the wind skewness on the rotor and maximize energy production, it is also important to maintain the HAWT rotational axis aligned with the wind direction.

The objectives of WTC are threefold : regulating output power/rotor speed, correcting rotor misalignment with wind direction, and minimizing mechanical strains on the HAWT components. To observe their states, multi-megawatt turbines are covered with sensors :

- Accelerometers on the tower, blades, nacelle, etc. to measure system vibrations.
- Strain gauges on the tower, blades, rotor shaft, etc. to measure loads and monitor fatigue.
- Wind cup anemometer and wind vane located on the rear end of the nacelle such to measure wind speed and direction respectively.

Lidar measurements for HAWT control

However, these sensors can give biased measurements as they are located in the wake of the HAWT. Since the 1970s, LiDAR (Light Detection and Ranging) devices – which were first used for astronomy purposes – have used laser beams to measure distances to solid objects. Using the Doppler effect it is also possible to measure the speed of distant solid objects, which allows us to indirectly measure the air motion by remotely sensing the speed of aerosols advecting in the flow. This technology was first used in wind energy in order to assess the potential wind resources of a site before constructing a wind farm. Nowadays, as it becomes more and more affordable, manufacturers are starting to install LiDAR on top of wind turbine nacelles to estimate upstream wind disturbance at approximately 40 to 200 m in front of the turbine [4]. This technology has the advantage of giving a measurement less disturbed by the rotor wake compared to wind cups and wind vanes, and it allows us to anticipate wind disturbance in the control strategy known as LiDAR Assisted Control (LAC). Moreover, recent developments in LiDAR devices and Wind Field Reconstruction (WFR) algorithms allow us to estimate more wind field

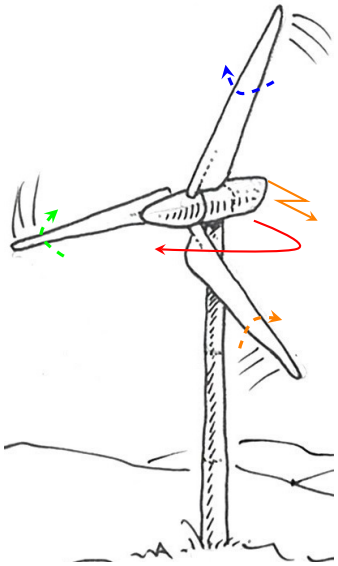

FIGURE 1.5 – Illustration of HAWT actuators. The green, orange and blue dashed arcs represent the action of the blade pitch actuators. The orange arrows represent the action of the electrical generator. The red arc arrow represent the action of the yaw actuator.

characteristics than just horizontal wind speed, such as vertical and horizontal shear, turbulence intensity and turbulence length scale [5]. However, LiDAR sensing has also several drawbacks compared to anemometers. It can give accurate measurements of head-wind approximately every two seconds [6]. But it can also be very sensitive to atmospheric conditions, for example heavy rain or fog, which can affect the LiDAR device's visibility, and to air with a very low density of aerosols, while wind cup anemometers are more robust [7].

HAWT actuators presentation

Commercial HAWT incorporate five actuators (see Figure 1.5) :

Yaw actuator : Rotates the nacelle and rotor on the tower longitudinal axis in order to correct the misalignment between rotor rotational axis and wind direction. Due to the high inertia of the nacelle and rotor, in addition to the significant rotating rotor gyroscopic effect, the actuation rate of this actuator is low compared to the following ones.

Electrical generator : It is possible to vary generator torque in the limit of its nominal value. This torque acts in opposition to the rotational momentum of the rotor, in order to slow down rotor rotation and produce electrical energy.

Three blade pitch actuators : HAWT are equipped with actuators allowing rotation of the blades on their longitudinal axis. Rotating the blades allows us to modify the angle of attack and thus the aerodynamic properties of the rotor, in order to offset the oversupply of wind energy or counteract wind skewness effects.

In this thesis, yaw control of wind turbines will be left out and wind misalignment is assumed to be an exogenous input.

Power regulation of HAWT

In power regulation mode, WTC is usually divided into four virtual regions [1], as shown in Figure 1.6 :

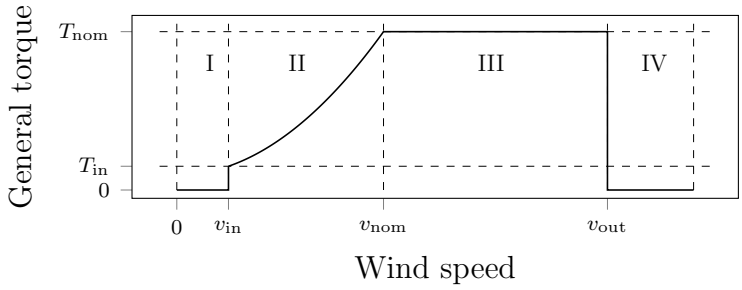

FIGURE 1.6 – The 4 control regions. v_{in}, v_{nom}, v_{out}, T_{in} and T_{nominal} stand for cut-in, nominal, cut-out wind speed, cut-in and nominal torque respectively.

Region I The wind does not supply enough power to produce energy, and so the turbine is stopped.

Region II The wind supplies enough power to produce energy, but less than nominal power. The torque generator is adjusted to maximize rotor aerodynamic efficiency : this control strategy is called Maximum Power Point Tracking (MPPT) and allows rotor speed to vary with wind velocity.

Region III The wind is above nominal speed. The generator torque is at its maximum value, and the rotor aerodynamic properties are adjusted by pitching the blades to feather on their longitudinal axis, in order to regulate rotor speed and output power.

Region IV The wind is above cut-out speed, for safety reasons the blades are pitched to their maximum angle to stop or idle the HAWT.

From collective to individual pitch control

With HAWT, a conventional assumption is that wind speed is uniform over the rotor area. This assumption was particularly relevant while rotor diameters were relatively small (i.e. diameters under 60-80 meters) and wind could only be characterized by its hub-height speed and direction [1]. Accepting this assumption in region III, all the blades must be pitched to the same angle. This technique is called Collective Pitch Control (CPC). However, nowadays, LiDAR technology, WFR algorithms [4] and estimation theory [8, 9] provide a richer description of the wind. Indeed, wind velocity information on the whole rotor plane can now be estimated, which combined with the tremendous growth of rotor diameters justifies the addition of a differential blade pitch angle on each blade, depending on its position in the rotor plane [10].

Various phenomena produce shear and turbulent eddies, making wind velocity and direction vary over the rotor area. Therefore, if the wind speed varies in space over the rotor plane and the blade pitch angles remain constant during rotation of the rotor, their angle of attack will vary as the blades are moving across the rotor plane. Consequently, the aerodynamic loads on the blades may fluctuate according to the azimuth angle, and produce structural fatigue not only on the blades but also on other rotating and non-rotating elements to a lesser extent [10, 11]. By adjusting each blade pitch angle individually depending on the blade position in the rotor plane, it is possible to locally adjust the angle of attack, and thus redress rotor load imbalances [10]. This technique is the aforementioned IPC.

IPC frames of coordinates

In the literature, IPC regulators designed to reduce structural fatigue have one of the two following control objectives : regulating unbalanced loads [10, 12, 13, 14] for vibration and fatigue reduction ; or balancing Floating Offshore Wind Turbines (FOWT) [15, 16]. Therefore blade root bending moments have a prominent role in IPC, as their cyclic variations give an image of rotor unbalanced loads. In order to better observe these unbalanced loads, variables can be changed by projecting the outputs on an alternative frame of coordinates. In the literature, three frames of coordinates can be used for IPC design, namely Multi-Blade Coordinates (MBC) [17], the Clarke coordinates [18], and single-blade coordinates [19].

Multi-Blade Coordinates MBC is the most widely used frame of coordinates in the literature. It uses a transform, named the MBC transform, to project the rotating loads of the blades on a fixed frame of coordinates. These projected loads correspond to the unbalanced loads on the rotor hub. The MBC transform is also known in the literature as the Coleman [20], D-Q or Park transform, and was first used in helicopter aerodynamics. The first appearance of MBC in IPC control, in 2003, was designated the Coleman transform [10]. Since the work of Bir *et al.* [17], published in 2008, the name MBC can also be found in the literature. The MBC transform is defined as follows :

$$\begin{pmatrix} \bar{M} \\ M_{\text{yaw}} \\ M_{\text{tilt}} \end{pmatrix} = \underbrace{\frac{2}{3} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \cos(\psi_1) & \cos(\psi_1 + \frac{2\pi}{3}) & \cos(\psi_1 + \frac{4\pi}{3}) \\ \sin(\psi_1) & \sin(\psi_1 + \frac{2\pi}{3}) & \sin(\psi_1 + \frac{4\pi}{3}) \end{pmatrix}}_{T(\psi_1)} \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix} \tag{1.1}$$

where ψ_1 is the azimuth angle of the first blade, whose value is zero when the blade is horizontal and right of the nacelle when viewing the turbine from the front. The blade root bending moment of blade i is denoted by M_i which can be measured with strain gauges. $\bar{M}$, M_{yaw} and M_{tilt} are the mean, yawing and tilting moments respectively. Thus, the two moments unbalancing the rotor are M_{yaw} and M_{tilt}. The IPC controller is designed to regulate these unbalancing loads by adjusting the yawing and tilting blade pitch angles, denoted respectively by θ_{yaw} and θ_{tilt}. These blade pitch angles do not have physical meaning, but give an image of the blade pitch angle variation from the collective blade pitch angle θ_{col} which must be given to the blades to correct the rotor unbalances. The blade pitch angle, denoted by θ_i, which must be applied to the i^{th} blade is obtained with the inverse Coleman transform :

$$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = T(\psi_1)^{-1} \begin{pmatrix} \theta_{col} \\ \theta_{yaw} \\ \theta_{tilt} \end{pmatrix} \tag{1.2}$$

Recently a new transform, called the Multi-Blade Multi-Lag transform (MBML) [21], closely related to MBC, has been introduced. Similarly to MBC, the starting point is the assumption that the blade root bending moment of blade i can be expressed as :

$$M_i = \bar{M} + M_{\text{yaw}} \cos(\psi_i) + M_{\text{tilt}} \sin(\psi_i) \tag{1.3}$$

which assumes that the three blades are identical and their moments are all varying with the azimuthal variations from an offset moment $\bar{M}$. In real life, because of phenomena such as wear, dust and ice, the offset moment of each blade lags independently from the others. Therefore a new formula for the moments is used in [21] :

$$M_i = \bar{M} + \delta M_i + M_{\text{yaw}} \cos(\psi_i) + M_{\text{tilt}} \sin(\psi_i) \tag{1.4}$$

where δM_i is the lagging offset moment of the i^{th} blade. The authors give an additional constraint, which is to properly split the collective offset $\bar{M}$ from the lag offset :

$$\sum_{i=1}^{3} \delta M_i = 0 \tag{1.5}$$

For the control architecture, MBML differs from MBC because of an additional Multiple Input Multiple Output (MIMO) controller, called the equalizing controller, whose objective is to cancel the δM_i, for $i \in \{1, 2, 3\}$. The output of the equalizing controller is $\delta\theta_i$ and the blade pitch angle of the i^{th} blade is obtained as follows :

$$\theta_i = \theta_{col} + \delta\theta_i + \theta_{yaw}\cos(\psi_i) + \theta_{tilt}\sin(\psi_i) \tag{1.6}$$

The advantages of MBML over MBC is that the δM_i offsets can be cancelled. However in this thesis, the methods developed are tested in simulations where all the blades are identical, therefore MBC is equivalent to the MBML transform.

The first IPC controller presented [10] was a double Single Input Single Output (SISO) Proportional Integral (PI) in the MBC frame of coordinates. The output signals were θ_{yaw} and θ_{tilt}, and the input signals were M_{yaw} and M_{tilt}, expressed as follows :

$$[\theta_{yaw}(t), \theta_{tilt}(t)]^T = -K_P[M_{\text{yaw}}(t), M_{\text{tilt}}(t)]^T - \int_{t_0}^{t} K_I[M_{\text{yaw}}(\tau), M_{\text{tilt}}(\tau)]^T d\tau \tag{1.7}$$

where K_P and K_I are the proportional and integral gains of the controller respectively. This controller made it possible to decrease blade, rotor shaft and yaw bearing fatigue loads by about 15% while increasing pitch activity. This controller serves as the baseline controller in many IPC control papers [12, 13, 22, 23, 24]. It has been implemented in field tests [25] where fatigue load reduction and pitch activity increase were effectively observed. The general IPC controller with MBC transform is illustrated as a block diagram in Figure 1.7. Nevertheless, it is demonstrated in [17] that the MBC transform introduces coupling between θ_{yaw} and M_{tilt}, and between θ_{tilt} and M_{yaw}. Therefore the system is MIMO and controlling such a system with several SISO controllers is not totally reliable.

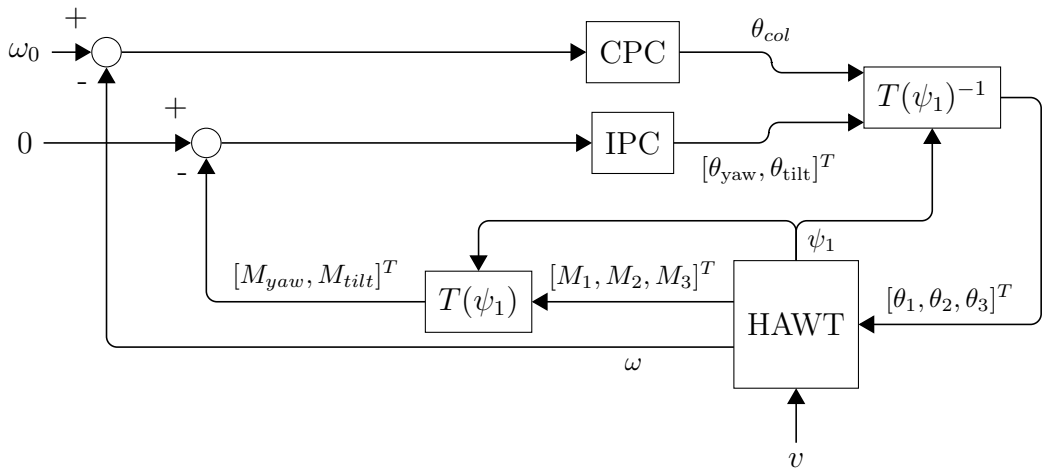

FIGURE 1.7 – Block diagram of the general IPC controller with MBC transform. ω_0 is the rotor rotational speed set-point value and v is the wind disturbance.

The Clarke coordinates An alternative to the MBC transform is the Clarke transform, which comes from electrical engineering [26]. In a similar way to MBC, two moments M_α and M_β are obtained from the moments $M_{1,2,3}$ [18] :

$$\begin{pmatrix} \bar{M} \\ M_\alpha \\ M_\beta \end{pmatrix} = \underbrace{\frac{2}{3}\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix}}_{P} \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix} \tag{1.8}$$

Similar to the MBC transform, the inverse transformation is used to pass from the blade pitch angles $\theta_{1,2,3}$ to the blade pitch angles in Clarke coordinates $\theta_{\alpha,\beta}$:

$$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = P^{-1} \begin{pmatrix} \theta_{col} \\ \theta_\alpha \\ \theta_\beta \end{pmatrix} \tag{1.9}$$

The relation between the moments in Clarke coordinates and the moments in MBC coordinates is the following :

$$\begin{cases} M_\alpha & \propto & M_{\text{yaw}}\cos(\psi_1) & + & M_{\text{tilt}}\sin(\psi_1) \\ M_\beta & \propto & -M_{\text{yaw}}\sin(\psi_1) & + & M_{\text{tilt}}\cos(\psi_1) \end{cases} \tag{1.10}$$

which means that the moments in Clarke coordinates might be oscillating with the blade azimuth angle, while M_{yaw} and M_{tilt} might be constant in MBC coordinates.

With Clarke coordinates, as opposed to the MBC transform, there is no coupling between the blade dynamics. This means that three SISO controllers can be convenient for control, one for the output power regulation (CPC), one to regulate M_α with θ_α and the last one to regulate M_β with θ_β. In Clarke coordinates, two controllers have been proposed so far [18, 27]. They were constituted of a PI controller for the CPC and two proportional [18] or PI resonant [27] controllers in the $\alpha - \beta$ coordinates. They showed that for the rotating components (e.g. blades, rotor bearing, rotor shaft), fatigue load mitigation could be obtained, however pitch activity would increase. The general IPC controller in Clarke coordinates is illustrated as a block diagram in Figure 1.8.

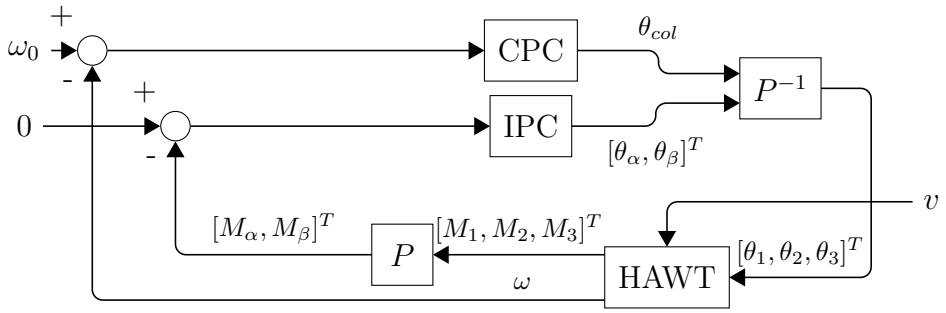

FIGURE 1.8 – Block diagram of the general IPC controller in Clarke coordinates. ω_0 is the rotor rotational speed set-point value and v is the wind disturbance.

Single blade coordinates The single blade coordinates method [19] is the most straightforward frame of coordinates, as every blade is considered as a single system, whose load variations must be regulated. The major difference from the two previously-examined frames of coordinates is the fact that $\theta_1 + \theta_2 + \theta_3 = \theta_{\text{col}}$ is not enforced by the transform.

In the single blade coordinates case, this condition must be ensured by the controller itself, in order to prevent modification of the collective blade pitch angle, which could disturb the CPC regulation.

In single blade coordinates, there are therefore two possible IPC implementations. The first one is to use an IPC in addition to a CPC controller. The IPC takes as input the variations of the blade root bending moments from a set-point, which could be the average moment $\bar{M}$. The IPC yields as output the blade pitch angle variations which must be added to θ_{col} (the CPC output) in order to obtain the blade pitch angle transmitted to the actuators.

The second option is to design an IPC controller which also has the output power or rotor rotational speed as a control objective. The IPC would take as input the blade root bending moments and the rotor rotational speed ω_r to yield as output the blade pitch angles.

In the literature, implementations of PI controllers can be found in [28, 29]. In the most precise paper using this approach [29], a high-pass filter lets through only the blade root bending moments variations which must be alleviated. The Individual Blade Control (IBC) controllers regulate these moment variations by giving a differential pitch angle added to the collective pitch angle θ_{col}, to transmit a pitch angle as input of the turbine. The controller architecture is depicted in Figure 1.9.

The advantage of this approach is that each PI controller is effectively regulating a SISO system, as every single blade's dynamics can be considered as independent from the others. It can be noticed that similarly to the MBML transform, the IBC controller can correct the aerodynamic discrepancies between the blades. However, compared to MBML, nothing ensures that the average of blade pitch angles is equal to the collective blade pitch angle, in the case of non-identical blades. This could result in damage to the CPC regulation. In [19], solutions to prevent damage to the output power are proposed but no actual implementation is shown. This control strategy was implemented in field tests [28] and as expected, blade fatigue loads were reduced and pitch activity increased.

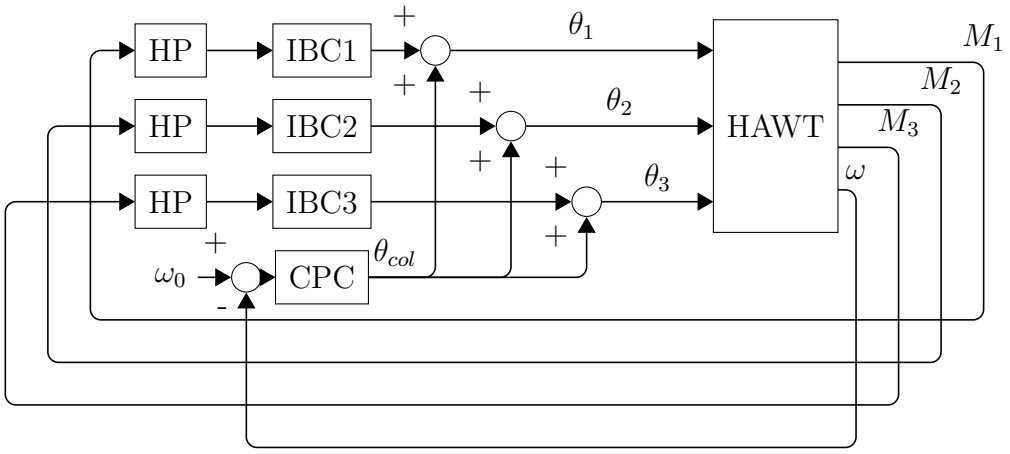

FIGURE 1.9 – Block diagram of the IBC controller as explained in [29]. HP stands for High-Pass filter.

To summarize, three frames of coordinates are available for control design :

— MBC coordinates, which is non-rotating.
— Clarke coordinates, which is rotating.
— Single blade coordinates, which considers the blades in isolation.

Similarities between the frames It was demonstrated in [30] that it is possible to design an IPC controller in any of the previous frames of coordinates and project it onto any other frame of coordinates without control performance losses. Therefore, the choice of frame of coordinates depends on the control strategy and the internal model to be used. It is shown in Section 1.2.2 that some strategies and models are more suited to particular frames of coordinates, e.g. control strategies for linear systems in MBC coordinates or repetitive control strategies in single blade coordinates.

Generalization to nonlinear cases As the HAWT is a highly nonlinear system, it is suggested to use gain scheduling for generalization to the nonlinear case [10]. More recently, controllers using Radial Basis Functions (RBF) neural networks [31] and fuzzy logic PI controllers [32] have been implemented, with the idea to tune PI coefficients online, and avoid the use of gains scheduled PI. In [33], a model reference adaptive control strategy is used in order to optimize online a proportional feedback in order to match the dynamics of a specified model in closed loop. These papers show that fatigue load alleviations are also possible with these kinds of controllers.

1.2 Advanced Individual Pitch Control

In control systems engineering as a whole, Proportional Integral Derivative (PID) and PI controllers are said to solve more than 80% of industrial problems. They are very popular because of their implementation simplicity ; their theory is exhaustive, and off-the-shelf tables and diagrams are available for tuning. However, this technology can have several shortcomings when it comes to optimizing complex processes. It is not possible to specify constraints and objectives on the closed loop, and it does not guarantee robustness margins. This is why some industrial problems can require more advanced control strategies, such as the one presented in the following non-exhaustive list :

Robust control : The robust control objective is to synthesize a controller which is able to respect certain specifications (e.g. closed-loop cost or stability margins) in spite of uncertainties on the internal model or the system state.

Adaptive control : Adaptive control consists in modifying the controllers' parameters dynamically, in order to adapt to changes in the system dynamics.

Optimal control : Optimal control aims at finding the optimal input over a given horizon, such that this input fed to the system minimizes an objective function on the horizon, given initial conditions, dynamics of the system and constraints on states and inputs.

For all the advanced control strategies mentioned above, an internal model is needed for their design. Therefore in this section, a short review of HAWT modelling is firstly provided. Then, an overview of the work using these advanced control strategies is given.

1.2.1 HAWT modelling

In Figure 1.10, an illustration of the onshore HAWT system shows that it can be subdivided into three subsystems, namely the tower-nacelle, drive train and rotor. Note that the rotor can also be subdivided into three subsystems corresponding to each individual blade.

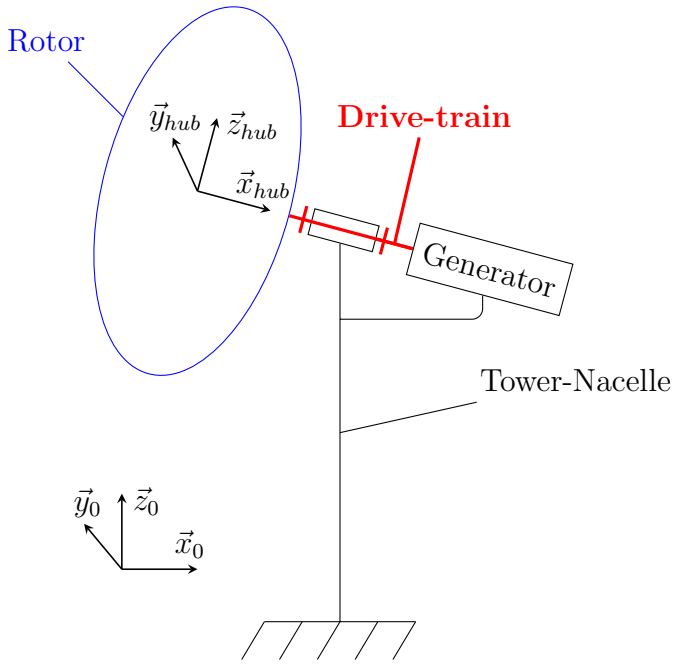

FIGURE 1.10 – HAWT scheme. Each colour corresponds to a different subsystem : black for the tower-nacelle, red for the drive train and blue for the rotor. $(\vec{x}_{hub}, \vec{y}_{hub}, \vec{z}_{hub})$ is the non-rotating coordinate frames of the hub and $(\vec{x}_0, \vec{y}_0, \vec{z}_0)$ those of the tower.

Rotor modelling

The reliable rotor model is called the unsteady aero-elastic Blade Element Momentum (BEM) model and is depicted in appendix A. It uses BEM theory for the aerodynamic part and beam theory for the elastic part. This model yields a linear Partial Differential Equations (PDE) system for the elastic part and a nonlinear Differential Algebraic Equation (DAE) system for the aerodynamic part, containing more than 1000 states for a commercial HAWT turbine. It is used in most HAWT aero-elastic simulators but the implementation of this model in advanced control problems is not straightforward. For controller synthesis, the model is often simplified to algebraic equations which give the steady values of the outputs of interest, or to low-dimensional first-order differential equations. The outputs of interest for CPC are the aerodynamic torque and thrust, denoted respectively by T_a and F_t, and retrieved using the following nonlinear algebraic equations :

$$T_a = \frac{1}{2}\rho v_{\text{hh}}^3 C_p \frac{A}{\omega_r} \tag{1.11a}$$

$$F_t = \frac{1}{2}\rho v_{\text{hh}}^2 C_t A \tag{1.11b}$$

where A is the rotor area, ω_r is the rotational speed of the rotor, v_{hh} is the hub-height wind speed, C_p and C_t are respectively the power and thrust coefficients.

These coefficients are a mapping of tip-speed ratio, denoted by λ, and collective blade pitch angle, denoted by θ_{col} (see Figures 1.11 and 1.12). The tip-speed ratio is defined as the ratio between the speed at blade tip due to rotation and hub-height wind speed v_{hh} :

$$\lambda = \frac{\omega_r R}{v_{\text{hh}}} \tag{1.12}$$

where R is the rotor radius. The mapping is identified using BEM theory, for each combination of ω_r, v_{hh} and θ_{col}. This mapping is dependent on the rotor blade geometry, and

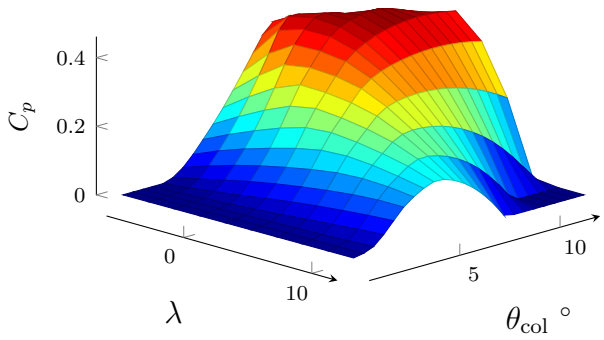

FIGURE 1.11 – Example of power coefficient C_p surface plot.

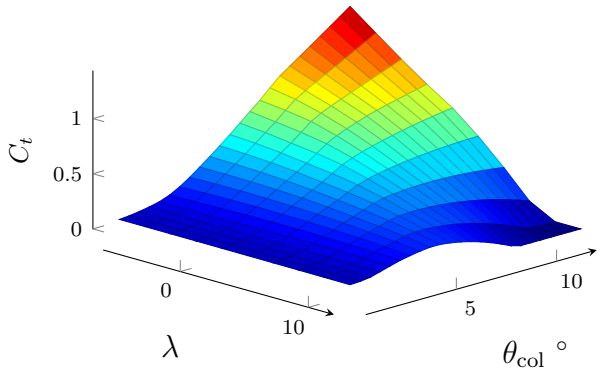

FIGURE 1.12 – Example of thrust coefficient C_t surface plot.

is thus different for each designed rotor. In order to use this mapping to control systems, it can be substituted by look-up tables, neural networks, B-splines or polynomials. Note that these functions can all be differentiated, which is an interesting property for the optimization of the control strategy.

For IPC, the model must be able to relate the impact of pitch angles on the blade root bending moments, depending on the whole rotor information. However, there is no strict consensus concerning rotor models for IPC in the literature. There are two possible models detailed in the literature which consists in identifying an algebraic equation, similarly to the models used for CPC [16], or low-dimensional first-order differential equations, allowing us to account for blade vibrations. This low-order differential equation model can be obtained through linearization of the DAE system obtained from BEM theory and model order reduction of the resulting system, or system identification.

Algebraic models

To the best of my knowledge, two algebraic models can be found in the literature, the first based on a linearization and the second on a nonlinear identification approach. However, physical considerations are present behind their parameterization.

The linear model was presented in [15] and considered that steady aerodynamics BEM is a an algebraic function of :

— The flapwise relative wind speed, denoted by $v_{\text{fl,i}}$ and defined as the relative wind speed at three fourths of the i^{th} blade's radius.
— The i^{th} blade pitch angle θ_i

Considering a minor deformation assumption, $v_{\text{fl,i}}$ can be expressed as follows :

$$v_{\text{fl,i}} = v_i - \dot{x}_{\text{FA}} + \sin(\psi_i)\frac{3}{2H}\frac{3R}{4}\dot{x}_{\text{FA}} \tag{1.13}$$

where v_i is the blade's effective wind speed, ψ_i is the i^{th} blade's azimuth angle, H is the rotor hub-height, R is the rotor radius and $\dot{x}_{\text{FA}}$ is the tower fore-aft displacement velocity. The terms depending on $\dot{x}_{\text{FA}}$ aims at accounting for respectively the translational and rotational displacements of the rotor, due to fore-aft tower deflection. Steady aerodynamics BEM equations are linearized around an operating point depending on the collective

blade pitch angle θ_{col} and v_{hh}. This yields the following linear model :

$$\begin{bmatrix} M^i_{Ip} \\ M^i_{Oop} \\ F^i_{Ip} \\ F^i_{Oop} \end{bmatrix} = \Lambda(v_{\mathrm{hh}}, \theta_{col}) \begin{bmatrix} v_{\mathrm{fl,i}} \\ \theta_i \end{bmatrix} \tag{1.14}$$

where $\Lambda \in \mathbb{R}^{4\times 2}$ is a matrix resulting from the linearization, F^i_{Ip} and F^i_{Oop} are the i^{th} blade root in-plane and out-of-plane forces respectively. The plane which is referred to in the in-plane and out-of-plane notions is the rotor plane.

The nonlinear model was presented in [16] and is based on the CPC aerodynamic model, where aerodynamic forces are nonlinear algebraic functions of hub-height wind speed, collective blade pitch angle and tower displacements. For the CPC model, T_a and F_t (see equation 1.11), are identified using nonlinear coefficients, obtained from data generated with the steady aerodynamic BEM model. In [16], the CPC model is extended to M_{yaw} and M_{tilt}, using other nonlinear coefficients :

$$M_{\mathrm{yaw}} = \frac{1}{2}\rho A \left(C_{S1}(\delta_y - K_{VH}\theta_{yaw}) + C_{S2}\delta_z\right) v^2_{\mathrm{hh}} \tag{1.15a}$$

$$M_{\mathrm{tilt}} = \frac{1}{2}\rho A \left(C_{S1}(\delta_z - K_{VH}\theta_{tilt}) + C_{S2}\delta_y\right) v^2_{\mathrm{hh}} \tag{1.15b}$$

where δ_y (resp. δ_z) is the horizontal (resp. vertical) wind shear, K_{VH} is a nonlinear coefficient depending on the collective blade pitch angle θ_{col}, C_{S1} and C_{S2} are nonlinear coefficients depending on the tip-speed ratio λ, collective blade pitch angle θ_{col} and the misalignment angle of the inflow wind field γ. The mapping of the nonlinear coefficients is approximated with polynomials for nonlinear control in [16].

Note that these models were created with the scope of using IPC to regulate the yawing and tilting displacements of a FOWT. However, alleviating the turbine fatigue loads could come as a secondary objective. In these models, blade flexibility is completely ignored. Therefore, a model of blade vibrations can be useful to a model-based controller aiming at efficiently alleviating the rotor fatigue.

Dynamic models

In most model-based IPC structures, the DAE system that models the rotor aerodynamic is linearized at a given operating point, and the eigenmodes of the rotor blades are identified by solving the eigenmodes of the resulting PDE system that models blade aero-elasticity. Most blade vibrations can be recovered through the two first flapwise and edgewise eigenmodes. This process provides a mass spring damper equation for every mode, which can be transformed into a Linear Time Invariant (LTI) system. An operating point is defined by simple wind characteristics (hub-height speed, direction, shear), rotor rotational speed and blade azimuth. Therefore, an LTI system is obtained for each azimuth angle ψ_1 and for every blade eigenmode :

$$\begin{cases} \dot{x} = A(\psi_1)(x - x_{op}(\psi_1)) + B(\psi_1)(u - u_{op}) + B_d(\psi_1)(d - d_{op}(\psi_1)) \\ y - y_{op}(\psi_1) = C(\psi_1)(x - x_{op}(\psi_1)) + D(\psi_1)(u - u_{op}) + D_d(\psi_1)(d - d_{op}(\psi_1)) \end{cases} \tag{1.16}$$

where $u = [\theta_1, \theta_2, \theta_3] \in \mathbb{R}^3$ is the input vector, $y = [M_1, M_2, M_3] \in \mathbb{R}^3$ the output vector and $d \in \mathbb{R}^m$ is the disturbance vector, with m the number of wind characteristics and $x \in \mathbb{R}^6$ the state vector. Note that the system is of order six because it is composed of

three susbsystems, i.e. the blades, which are systems of order two. A, B, B_d, C, D and D_d are matrices of appropriate dimensions. x_{op}, $u_{op} = [\theta_{col}, \theta_{col}, \theta_{col}]^T$, d_{op} and y_{op} are operating state, input, disturbance and output vectors of appropriate dimensions. Note that the blade azimuth angle ψ_1 depends on the rotor rotational speed ω_r :

$$\psi_1(t) = \psi_1(t_0) + \int_{t_0}^{t} \omega_r(\tau) d\tau \tag{1.17}$$

Therefore, the system (1.16) becomes a Linear Time Variant (LTV) system.

In MBC coordinates According to [17], the variations of the input(B and B_d) and feedthrough (D and D_d) matrix coefficients are reduced using the MBC transform on the state space system. The state space can be thus approximated in MBC coordinates by an LTI system :

$$\begin{cases} \dot{\tilde{x}} & = & \tilde{A}(\tilde{x} - \tilde{x}_{op}) & + & \tilde{B}(\tilde{u} - \tilde{u}_{op}) & + & \tilde{B}_d(d - d_{op}) \\ \tilde{y} - \tilde{y}_{op} & = & \tilde{C}(\tilde{x} - \tilde{x}_{op}) & + & \tilde{D}(\tilde{u} - \tilde{u}_{op}) & + & \tilde{D}_d(d - d_{op}) \end{cases} \tag{1.18}$$

where a matrix or vector X in the blade coordinates has its counterpart $\tilde{X}$ in MBC coordinates. The dynamics of the MBC transform are thus introduced in the state space and because of these dynamics, the system which was made of three SISO systems in the single blade coordinates becomes a MIMO system in the MBC coordinates. Note that therefore $\tilde{u}_{op} = [\theta_{col}, 0, 0]^T$ in MBC coordinates. It is observed in [17] that the MBC transform acts as a filter letting through terms that are integral multiples of NP, where P is the period of one rotor rotation, and filters out the other periodic terms. However, dynamics lying in between would be lost. In order to generalize the model to various winds, a Linear Parameter Varying (LPV) model can be obtained from interpolations of LTI model dynamics [14].

In Clarke coordinates To answer the MBC loss of dynamics issue, the Clarke transform was created. Indeed, the Clarke transform is a static transform, thus no dynamics can be lost. However, since the transform is static, the LTV model (1.16) in blade coordinates stays an LTV model in Clarke coordinates. The main advantage of this transform is that no dynamics are introduced within the transform, thus Single Input Single Output (SISO) controllers can be used [18]. However, a LTV model must be regulated, which is more challenging than an LTI one in the MBC transform. The advantage of using Clarke or MBC transforms before the control of blade root bending moment in single blade coordinates is that the set-points are always zero, while it is varying with wind speed for the blade root bending moment control.

In single blade coordinates The last dynamic rotor model which can be found in the literature is presented in [19]. This model is meant to work in the single blade frame of coordinates. The insight is to consider a single blade as a LPV spring mass system in blade coordinates, with the spring stiffness parameter varying with the blade pitch angle θ_i, in addition to the in-plane and out-of-plane aerodynamic moments, denoted respectively by $M^i_{A,Ip}$ and $M^i_{A,Oop}$, as external forces. However, as blade coordinates is a non-inertial frame, the blade rotates and vibrates due to rotor rotation, drive train vibrations and tower deflections. These inertial motions add inertial forces applied to the blades, depending linearly on blade frame of coordinates accelerations and blade inertial parameters. The i^{th} blade in-plane and out-of-plane inertial moments are denoted respectively by $M^i_{I,Ip}$ and $M^i_{I,Oop}$.

In [19], the model is written with the in-plane (respectively out of plane) blade root bending moments of blade i as states, denoted by M^i_{Ip} (resp. M^i_{Oop}) and θ_i as varying parameter :

$$\begin{pmatrix} \ddot{M}^i_{Ip} \\ \ddot{M}^i_{Oop} \end{pmatrix} = -K(\theta_i) \begin{pmatrix} M^i_{Ip} \\ M^i_{Oop} \end{pmatrix} + K(\theta_i) \begin{pmatrix} M^i_{A,Ip} & + & M^i_{I,Ip} \\ M^i_{A,Oop} & + & M^i_{I,Oop} \end{pmatrix} \tag{1.19}$$

where K is the stiffness matrix. The aerodynamic and inertial moments are called fictitious forces and can be computed from HAWT geometry, nacelle and rotor acceleration measurements. The authors claim that it is possible to estimate the aerodynamic moments by computing the wind speed at an unspecified blade radius (similarly to [15]), provided that wind field characteristics are obtained and a model yielding the aerodynamic moments is available.

It should be noticed that even though the approach is different, it should eventually lead to a model very close to the one obtained using the previous linearization and modal identification method. The drawback of this modelling is the difficulty of implementation and derivation of blade inertial parameters, based on raw data given by the blade manufacturer, i.e. stiffness and mass parameters along the blade.

On the other hand, the previous modelling involving linearization and eigenmodes identification is only valid in the neighbourhood of linearized operating points, while this one could possibly be sufficient for the whole operating range of the turbine. Although the expression of the aerodynamic moments is not specified in [19], it is likely to be highly nonlinear. The modelling in single blade coordinates has also the advantage over the linearization method in MBC coordinates, that no dynamics are filtered out. Possibly due to the difficulty of implementation and because of the lack of off-the-shelf tools able to directly deliver this model, the modelling in single blade coordinates is seldom used in the blade pitch control literature.

In summary, two kinds of rotor models are available, algebraic and dynamic, where blade vibrations are taken into account. Dynamic rotor models are defined for given wind conditions and rotor rotational speed. Therefore, the systems are MIMO LTI in MBC coordinates, SISO LTV in Clarke coordinates and SISO LPV in blade coordinates.

Drive train modelling

The drive train is assumed to be not totally rigid and is modelled with three subsystems, namely rotor shaft, gearbox and generator shaft. In most aero-elastic simulators and control papers [34, 35, 36], the rotor shaft is assumed to be a flexible mass spring damper system with a rigid gearbox (Figure 1.13). The subscripts r refer to the rotor and g to the generator, c, k, J, T and ω stand for the damping, stiffness, inertia, torque and rotational speed respectively. ω_i is the rotational speed at the input of the gearbox, N_{red} is the reduction ratio of the gearbox, such that $N_{\text{red}} = \dfrac{\omega_g}{\omega_i}$, and η is the gearbox efficiency, such that $\eta = \dfrac{T_o \omega_g}{T_i \omega_i}$, where T_o and T_i are respectively the gearbox output and input torque.

Let γ_i be the torsional angle on the flexible shaft, defined as the integral over time of the difference between the rotor and gearbox input rotational speeds :

$$\gamma_i(t) = \int_{t_0}^{t} (\omega_r(\tau) - \omega_i(\tau)) d\tau \tag{1.20}$$

where t_0 is chosen such that $\gamma_i = 0$, when the shaft is at rest. From Newton's second law applied to the rotor and generator shafts, the following differential equations are derived :

$$J_r\ddot{\gamma}_i + c_r\dot{\gamma}_i + k_r\gamma_i = T_a(v_{\text{hh}}, \omega_r, \theta_{col}) - T_0\frac{N_{\text{red}}}{\eta} \quad (1.21)$$

$$J_g\dot{\omega}_g = T_o - T_g \quad (1.22)$$

where $\dot{\gamma}_i$ and $\ddot{\gamma}_i$ are the first and second time derivatives of γ_i. This model is nonlinear with all the nonlinearities contained in T_a, defined in (1.11a). Using this system, the generator rotational speed and the output power can be estimated. For output power regulation in region III, a PID CPC controller is typically used, with the output power regulation error $\varepsilon_P = T_g\omega_g - P_{sp}$ as input, where P_{sp} is the output power set-point, with the pitch angle as output. It can be noticed that as generator torque is constant in region III, CPC objective can also be formulated as rotor rotational speed regulation.

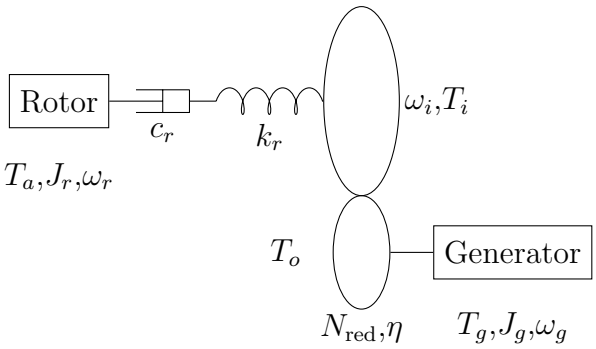

FIGURE 1.13 – Schematic modelling of the drive train.

Tower modelling

In order to get a more accurate HAWT model, the dynamics of the tower can also be integrated in the internal model. The tower can be modelled as a cantilever beam [37, 1] with distributed loading and local loading at the tower tip. This local loading corresponds to the aerodynamic thrust F_t, the generator torque T_g, the rotor unbalanced loads, M_{yaw} and M_{tilt}, the inertial forces and moments of the nacelle plus those acting on the rotor, denoted respectively by $\overrightarrow{F}_{\text{inert}}$ and $\overrightarrow{M}_{\text{inert}}$.

Thus, a resulting moment $\overrightarrow{M}_{tip} = T_g.\vec{x}_{hub} + M_{\text{tilt}}.\vec{y}_{hub} + M_{\text{yaw}}.\vec{z}_{hub} + \overrightarrow{M}_{\text{inert}}$ and force $\overrightarrow{F}_{tip} = F_t.\vec{x}_{hub} + \overrightarrow{F}_{\text{inert}}$ are applied at the tower tip (Figure 1.14), with distributed loading along the tower. From this loading, the resulting moment and force at the base of the tower are denoted by $\overrightarrow{M}_{base}$ and $\overrightarrow{F}_{base}$ respectively. These moments at the tower base yield the fore-aft (mode of the tower deflection in the vertical plane parallel to the rotor axis) and side-side (mode of the tower deflection in the vertical plane, perpendicular to the rotor axis) moments with the following projections, $M_{FA} = \overrightarrow{M}_{base}.\vec{y}_0$ and $M_{SS} = \overrightarrow{M}_{base}.\vec{x}_0$.

Beam theory is used to model tower elasticity, with the tower divided into several tower elements. Thanks to beam theory, it is possible to compute the deflection at the tip of the tower, which when projected onto the $\vec{x}_0$ axis, yields x_{FA}, tower deflection due to the fore-aft mode. The time derivative of the deflection at the tip of the tower, denoted by $\dot{x}_{\text{FA}}$, is used to compute the relative velocity $v_{\text{rel}} = v_{\text{hh}} - \dot{x}_{\text{FA}}$, which can be used instead of v_{hh} for T_a, F_t, M_{yaw} and M_{tilt} computations. The use of v_{rel}, which is necessary for an

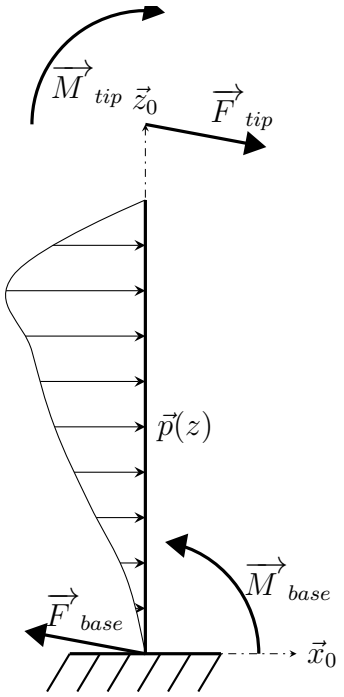

FIGURE 1.14 – Schematic modelling of the tower as a cantilever beam.

accurate modelling, introduces a nonlinear coupling between the drive train, rotor and tower systems.

Similarly to the blades, the first two tower fore-aft and side-side eigenmodes can be identified [1]. Therefore the control-oriented model is the sum of the second order systems obtained from the modal identifications. The external forces acting on this linear system are the aerodynamic and inertial forces transmitted through the rotor and drive train to the tower tip. These external forces are modelled as a nonlinear function of wind characteristics, azimuth angle, blade pitch angles, rotor rotational speed and generator torque. It should be noticed that the aerodynamic loads on the tower can act as aerodynamic damping on the first and second eigenmodes of the side-side and fore-aft modes, making it an LPV system.

To summarize, an HAWT control-oriented model is therefore a complex nonlinear system, made of several possibly LTI, LPV and nonlinear subsystems interacting with each other (see Figure 1.15). Therefore, blade pitch angles have complex consequences on the whole HAWT dynamics, and a single PID is of quite limited effect in controlling such a system. That is why gain scheduling of several PID controllers designed at different operating points is typically used, as in [38]. This confirms again the need for more advanced control strategies allowing us to properly handle nonlinear MIMO systems. Furthermore, as the dynamics of the rotor depend on wind characteristics, it has been demonstrated that LiDAR combined with blade pitch control can help in reducing fatigue loads and improve output power accuracy [39] ; these results have been confirmed with field tests in [40]. CPC control is still an evolving research topic, even though various advanced control techniques have been proposed so far. A good review of the CPC controllers already implemented can be found in [41]. An overview of the advanced control strategies proposed for HAWT IPC is given below.

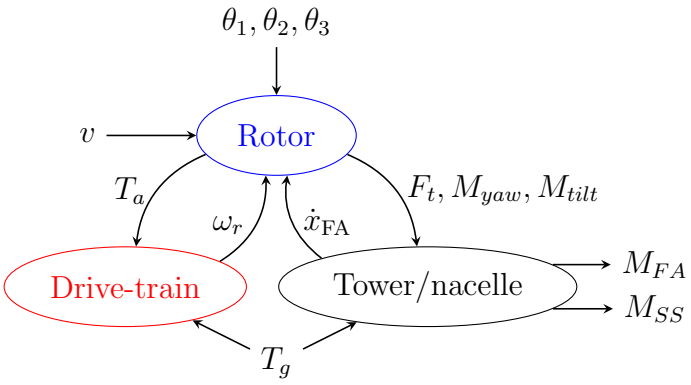

FIGURE 1.15 – Scheme of interactions between HAWT subsystems.

1.2.2 Overview of advanced control strategies application to IPC

In this section, the main advanced control strategies proposed for HAWT IPC are presented. The design of these strategies is based on the measurements given by the sensors mentioned on page 4 of Section 1.1, i.e. strain gauges, accelerators, wind cup anemometers, wind vane and possibly LiDAR. These control concepts can be divided into three categories, namely adaptive, robust and optimal controls.

Adaptive control

Adaptive control aims at adapting controller parameters to a system whose dynamics are evolving or uncertain. In the case of HAWT, both features are present : wind characteristics vary, and due to all the simplifications made in order to obtain a control-oriented model (linearization, modal identification etc.), there is a significant amount of uncertainties in the resulting model's dynamics.

An HAWT rotor is designed to rotate millions of times during its lifetime, and therefore the IPC controller is designed to repeat itself many times. When a system exhibits periodic behaviour, adaptive control strategies such as repetitive control, which is tailored to track periodic signals and reject periodic disturbances, are of interest.

These kinds of techniques can be used for IPC, as the rotating blades are periodically enduring local wind turbulence due to wind shear, wind misalignment and tower shadow during their revolution. Moreover, this control strategy can be easily adapted for periodic LTV system such as an HAWT rotor. Two papers can be found using this technique for IPC [42, 22]. Nevertheless, the assumption of constant periodicity in time can be violated in the HAWT system for several reasons :

1. The CPC might not regulate the rotor rotational speed perfectly, and therefore the time period could slightly vary.
2. As wind disturbances are passing through the rotor, wind field skewness might fluctuate with time, and the disturbance would be pseudo-periodic.
3. The nacelle is moving longitudinally, because of the tower fore-aft mode. The resonant frequencies of the tower are not equal to the rotational frequency of the rotor for tower safety reasons. Therefore, when considering tower movements, the pseudo-periodic disturbance might be polluted by the frequencies of the tower modes.

These issues are pointed out by the authors of [42]. However, because of the stochastic characteristics of their controller, for slowly varying disturbances and efficient rotational

speed regulation, the controller rejects the disturbances – like classical IPC controllers, but with a smoother input. The strategy used in [22] is interesting because it identifies a linear model from previous repetitions and uses it to control the turbine with a constraints-free MPC. Note that with such a method, the blade coordinates must be preferred over others. In terms of model, the one used in [42] is the linearized model derived with modal identification, while in [22] the model is identified online.

Robust control

In robust control, the objective is to synthesize a controller such that the closed-loop behaviour respects some given specifications in spite of uncertainties in the internal model. Even though there exists a nonlinear robust control solution in the control theory literature, it has not reached the stage of industrial application yet. On the other hand, there is an extensive theory of robust control for LTI, LTV and LPV systems. Therefore, robust control can be particularly relevant for IPC, as uncertainties are present on the LTI and LPV models.

Two very popular methods in robust control theory are H_∞ mixed-sensitivity and loop shaping. Both methods were implemented for IPC respectively in [43, 12, 44, 45] and [46, 30], with LTI systems in MBC coordinates. Note that [12] and [45] used feedforward with LiDAR information in order to anticipate variations in wind characteristics. In [30], a H_∞ loop-shaping IPC controller is also synthesized in single blade and Clarke coordinates, using the same LTI model, in order to show the equivalence of the closed-loops obtained. The authors claim that the choice of coordinates is only a matter of ease of implementation. In all these papers, for the extension to the nonlinear case a gain scheduling approach is suggested. In [14], a H_∞ mixed-sensitivity synthesis is performed on the LPV system, and validation is performed on a broad set of operating conditions.

Note that these robust control strategies yield a solution that can meet specifications with a certain degree of robustness, yielding important stability margins. However, this solution may be conservative in terms of closed-loop performance, compared to solutions obtained with optimal control.

Optimal control

The principle of optimal control is simple. For given initial conditions and system dynamics, optimal control needs to find the control input that minimizes an objective function on a time horizon subject to constraints on states and inputs. Moreover, optimal control strategies can be very efficient at scheduling and anticipating the optimal rejection of a disturbance if the latter can be measured.

The simplest and most conventional control regulator is the Linear Quadratic Regulator (LQR), which can be derived from an LTI system and a quadratic objective function. The LQR solves an optimal control problem on an infinite horizon. It delivers an optimal gain K, which, multiplied by the current state x, gives the optimal control input to be fed to the system, denoted by $u^\star = -Kx$. The optimal gain is the solution of an algebraic Riccati equation, which can be solved using several methods : readers are referred to [47] for more information.

LQR was one of the first controllers implemented for IPC along with a PI controller in [10], for a linearized LTI model in MBC coordinates. LQR was also used in [15], using a whole HAWT model with the linearized algebraic rotor model in MBC coordinates. It

used current wind speed information on every blade in addition to the current state, which makes it possible to better compensate for wind disturbance, but not to anticipate it. This aims at balancing a FOWT, as opposed to IPC implementations that aim at alleviating oscillating loads in blade root bending moments. Therefore the controller outputs were the three blade pitch angles, instead of differential blade pitch angles to be added to θ_{col}. Note that LQR has several shortcomings, as it is restricted to LTI systems and it cannot optimally use the information LiDAR might provide.

Model Predictive Control (MPC) is a much more complete optimal control technique, as it allows us to specify a more complex control problem than LQR, possibly with non-linear systems, non-quadratic objective function, complex constraints and integrate feed-forward information on the horizon [48]. However, it is limited to finite time horizons and it can require a solver to estimate the solution of the optimal control problem at each control update, which implies a discrete-time implementation and can be time consuming. Moreover, depending on the objective function, constraints and system specified, the optimization problem might not be convex any more [48, 49]. Therefore, the solver might get stuck in the saddle points and local optima [49]. Nevertheless, it allows us to optimally integrate LiDAR information and respects the HAWT constraints, and this is particularly interesting for industrial problems such as optimal LiDAR-assisted control of wind turbines.

MPC can be found in several places in the LiDAR-assisted IPC literature. Beginning with [13], where an IPC MPC is implemented using a quadratic objective function, with an identified greybox LPV rotor model scheduled on wind speed in MBC coordinates. In [50], a study merging repetitive control and MPC is presented. It is assumed that turbulence goes through the rotor relatively slowly compared to its rotational speed and that blade dynamics are identical. Therefore, the repetitive control feature allows us to estimate the disturbance from the previous blade and use this information as feedforward in a constrained MPC. The model used is an identified greybox LTI model of the whole turbine, using the blade coordinates for the rotor part. Therefore, this controller can directly yield the generator torque and the three blade pitch angles. In [16] an MPC using LiDAR is used in order to balance a FOWT and regulate its power production. A whole FOWT model is used with a nonlinear algebraic rotor model in MBC coordinates, with a quadratic objective function. The MPC outputs are the yawing and tilting blade pitch angles, the blade pitch velocity and the time derivative of the generator torque.

1.2.3 Summary

This section presented the state of the art on the existing HAWT control-oriented models and advanced IPC control strategies. What emerges from this state of the art is that there is no strict consensus on coordinates to express the model in, or the model to select. The chosen coordinates do not influence control quality, but they do influence the choice of the control strategy and its ease of implementation. The choice of the model, on the other hand, depends on the objective of the control designer and the control strategies to be implemented. For instance, the single blade model expressed in blade coordinates is particularly well suited to repetitive control strategies, while algebraic rotor models expressed in MBC coordinates are useful for balancing FOWT. Therefore, everything depends on the specifications that the HAWT must respect and the objectives it must achieve.

1.3 Fatigue of Wind Turbines

The main objective of IPC is to alleviate the effects of skewed winds on the rotor plane. One of the undesirable effects of skewed winds, which can be alleviated with IPC, is the surge of oscillatory loads that will stress the turbine and can severely damage some of its components, possibly leading to their premature failure. The latter can impact significantly the turbine OPEX and even the LCOE of wind energy.

IPC allows us to alleviate these oscillatory loads caused by skewed winds, thereby reducing the fatigue of the rotating components. However, in order to alleviate the oscillatory loads on the blades, IPC needs to pitch the blades depending on the azimuth angle and the skewness of the winds. Therefore, in order to reduce fatigue on the blades, pitch activity is increased and the blade pitch actuators sustain more damage than before. Thus, while IPC can reduce blade damage, it does not necessarily reduce turbine OPEX and the LCOE of wind energy. With intensive IPC actuation, the actuators may be damaged very rapidly, with costs in excess of a usual CPC implementation. There is thus a trade-off when optimizing pitch actuation and blade oscillatory loads. In order to correctly manage this trade-off and correctly address this issue, we need to understand what the wind energy industry needs. In today's world, maximizing the profit of wind energy and thus minimizing its LCOE will encourage the financial sector to invest in wind energy rather than carbon-emitting energy resources, thereby reducing the carbon footprint of the energy mix.

To do so, we need to design a controller that is able to minimize cost. A possible solution would be to use economic MPC [51, 52], provided an objective function that approximates the economic cost can be obtained. Fatigue theory can help in estimating this cost, as fatigue damage can give an estimation of a component lifetime ratio which is consumed during a simulation. Obviously, when one component fails, it needs to be replaced. This replacement has an economic cost, which is the price of the component plus the price of replacing it, including machinery, transport, labour etc. However, estimating this replacement price is outside the scope of this paper ; also, it is likely to be site-dependent and must be determined by the turbine owner. More importantly, a means of minimizing the fatigue damage weighted by the price of replacement is required, in order to meet the needs of the turbine owner.

This section first examines fatigue theory and fatigue estimation using the widely-accepted Palmgrem-Miner linear damage rule [53], which was simplified for engineering purposes in [54]. An important feature of the Palmgrem-Miner fatigue theory is the rainflow counting algorithm, which is also discussed below. Then, methods for the integration of fatigue damage as an objective function in optimal control are given.

1.3.1 Palmgrem-Miner fatigue theory

Fatigue theory is based on the assumption that every material contains tiny internal cracks. When the material is submitted to cyclic loading, the cracks propagate until the material fails. An hysteresis cycle is a cyclic load defined by its mean and amplitude. For example, the hysteresis cycle of index s (illustrated in Figure 1.16) has a mean load denoted by $\bar{L}_s = \frac{1}{2}\left(y(t_{\max,s} + t_{\min,s})\right)$ and amplitude $L_s^R = y(t_{\max,s}) - y(t_{\min,s})$, where $t_{\max,s}$ and $t_{\min,s}$ are respectively the time instant of the minimum and maximum of the hysteresis cycle s. For a mechanical component of index k we can determine from its material and geometry :

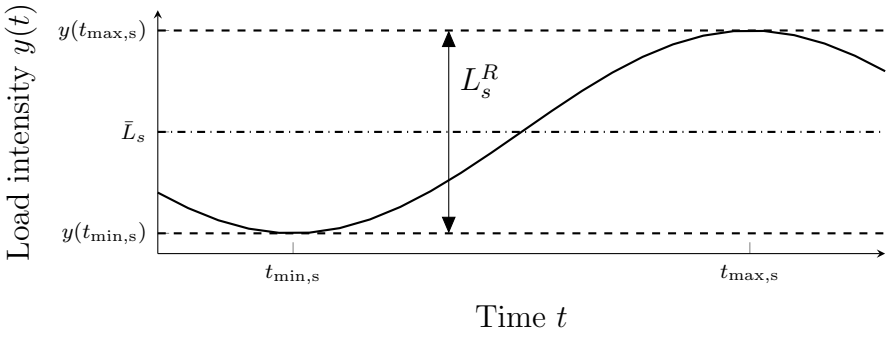

FIGURE 1.16 – Illustration of the hysteresis cycle s and its parameters.

L_k^{ult} The ultimate load, which is the maximum load that the mechanical component can bear. If the actual load surpasses the ultimate load, the component will fail instantly.

m_k The Wöhler coefficient, which indicates how sensitive a material is to mechanical fatigue. Its value depends on the material, for instance $m_k = 4$ for steel and $m_k = 10$ for glass fibre.

L_k^{MF} The fixed load-mean, which is the average load that a mechanical component supports.

Fatigue damage of one component is defined as the ratio of the component's consumed lifetime. Using the Palmgrem-Miner linear damage rule, it is possible to estimate the fatigue damage corresponding to an hysteresis load cycle. This linear damage rule is based on empirical observation of the number of sinusoidal cycles, denoted by N, of load amplitude S that a component can endure until its failure. This experiment yielded the S-N curves, relating N and S in logarithmic scale. It can be experimentally observed that the relationship between N and S is :

$$\log N = b - a \log S \tag{1.23}$$

where b is the intercept and $a > 0$ is the proportional coefficient of the relationship, which is constant for a given material and corresponds to the Wöhler coefficient defined above, so $m_k = a$. Using the definition of L_k^{ult}, the load that would break the component for a single cycle is the one that reaches the ultimate load. There are thus two possibilities for b. If one considers that approximately all the cycles are centred on L_k^{MF}, it is possible to assume that :

$$b = a \log \left(L_k^{\text{ult}} - |L_k^{\text{MF}}| \right) \tag{1.24}$$

considering that $S = \frac{1}{2} L_s^R$. However, more generally one cycle can reach L_k^{ult} if

$$\frac{1}{2} L_s^R = L_k^{\text{ult}} - |\bar{L}_s|$$

This is called the Goodman correction and it requires that :

$$b = \log \left(L_k^{\text{ult}} - |\bar{L}_s| \right) \tag{1.25}$$

Hence, the number of cycles that the mechanical component of index k can endure, denoted by N_k, is a function of the hysteresis cycle mean and load, whose formula is the following :

$$N_k(\bar{L}_s, L_s^R) = \left(\frac{L_k^{\text{ult}} - |L_k^{\text{MF}}|}{\frac{1}{2} L_s^R} \right)^{m_k} \tag{1.26}$$

and using the Goodman correction, N_k becomes :

$$N_k(\bar{L}_s, L_s^R) = \left(\frac{L_k^{\text{ult}} - |\bar{L}_s|}{\frac{1}{2} L_s^R} \right)^{m_k} \tag{1.27}$$

The Rainflow Counting algorithm (RFC) [55] counts the number of hysteresis cycles endured by one component from load time series. It returns for each hysteresis cycle contained in the load time series, the number of times this cycle occurs, along with its amplitude and mean. For the hysteresis cycle of index s, its number of occurrences is denoted by n_s. In order to quantify the fatigue damage endured by the mechanical component of index k, denoted by $\mathcal{D}_k$, the damage contribution of each hysteresis cycle, whose index ranges from 1 to N_{cyc}, must be added. Therefore $\mathcal{D}_k$ is expressed as follows :

$$\mathcal{D}_k = \sum_{s=1}^{N_{\text{cyc}}} \frac{n_s}{N_k(\bar{L}_s, L_s^R)} \tag{1.28}$$

Fatigue damage is a non-dimensional metric which is very meaningful, as it represents the fraction of a component's lifetime consumed during a time series. However, the main HAWT components are designed to last at least 20 years and control engineers are interested in simulations of 5 to 10 minutes long. The fatigue damage of such simulations should thus be a number in the range $[10^{-7}, 10^{-10}]$, which can seem negligible for someone not familiar with fatigue theory. Moreover, it can be very sensitive and should be better observed and compared in the logarithmic scale ; it may require normalization by the length of the simulation for comparison. Therefore, another metric called Damage Equivalent Load (DEL), which is easier to understand and visualize, was designed. DEL is defined as the amplitude of a sinusoidal loading of specified frequency around a specified fixed load-mean that would inflict the same damage on the component as the evaluated load time series. This definition is translated by the following equation :

$$\mathcal{D}_k = \frac{f^{\text{eq}} T}{N_k(L_k^{\text{MF}}, \text{DEL}_k)} \tag{1.29}$$

where f^{eq} is the specified frequency of the DEL, T is the length of the simulation, L_k^{MF} is the specified fixed load-mean and DEL_k is the DEL corresponding to $\mathcal{D}_k$. Hence, using equation (1.26) or (1.27), it is possible to formulate DEL_k :

$$\text{DEL}_k = 2(L_k^{\text{ult}} - |L_k^{\text{MF}}|) \left(\frac{\mathcal{D}_k}{f^{\text{eq}} T} \right)^{\frac{1}{m_k}} \tag{1.30}$$

Therefore, DEL can be seen as a standard deviation of load intensity, as both are in the same order of magnitude, provided that f^{eq} and possibly L_k^{MF} are judiciously chosen. However, if f^{eq} is too high, DEL_k will be relatively small compared to standard deviation and vice versa. The same goes for $|L_k^{\text{MF}}|$, and while it may be a good idea to take the load time series average as the L_k^{MF} value, if this average varies significantly between simulations the DEL may be biased and therefore not comparable.

To summarize, the DEL value depends greatly on tuning but is easier to manipulate for comparison purposes. Moreover, it has appealing properties, such that without Goodman correction it does not depend on L_k^{ult} anymore, which is useful when the ultimate load value is not available. It may be worth noting that for two load time series x and y, the ratio of their damages is equal to the ratio of their DEL power with the corresponding Wöhler coefficient :

$$\frac{\mathcal{D}_k(x)}{\mathcal{D}_k(y)} = \left(\frac{\text{DEL}_k(x)}{\text{DEL}_k(y)} \right)^{m_k} \tag{1.31}$$

This confirms that $\mathcal{D}_k(x) > \mathcal{D}_k(y)$ is equivalent to $\text{DEL}_k(x) > \text{DEL}_k(y)$, provided that $m_k \geq 1$, which is the case for most of usual materials. It also shows that if DEL can be compared in linear scale, fatigue damage should be compared in logarithmic scale. This

comparison explains the high sensitivity of fatigue damage to load time series standard deviation variations. Therefore, this ratio infers that a quadratic cost function, which is proportional to the square of standard deviation, might not be suitable for modelling fatigue damage where Wöhler coefficients are higher than two. What further separates quadratic forms or standard deviation from fatigue damage expression is the method used to count the hysteresis cycles, which is presented in the next subsection.

1.3.2 Rainflow counting algorithm

The RainFlow Counting (RFC) algorithm illustrated in [55] is widely accepted in the fatigue community as a method for counting the number of hysteresis cycles in a time series for fatigue damage estimation. It aims at counting the hysteresis cycles that a mechanical component endures over a time series. Note that in fatigue theory it is assumed that fatigue damage depends only on the number of cycles that one material endures, no matter the evolution rate of the loads. Therefore, for a given single hysteresis cycle and mechanical component, if the hysteresis cycle occurs in a few microseconds or several days, the same ratio of the mechanical component's lifetime is consumed. Of course, the damage rate, which is the rate at which a mechanical component deteriorates, is higher in the first case. Hence, a preprocessing step in the RFC algorithm is to only consider the reversal (maxima and minima, or peaks and valleys) of the time series. A vector $S = [s_1, \dots, s_n]$ is thus obtained from a time series, where n is the number of reversals, and s_i is the i^{th} reversal. An illustration of vector S is given in Figure 1.17. It should be pointed out that vector S respects the following conditions :

— s_1 is the first value in the time series.
— s_n is the last value in the times series.
— $\text{sign}(s_{i-1} - s_i) \neq \text{sign}(s_i - s_{i+1}) \quad \forall i \in \{2, \dots, n-1\}$

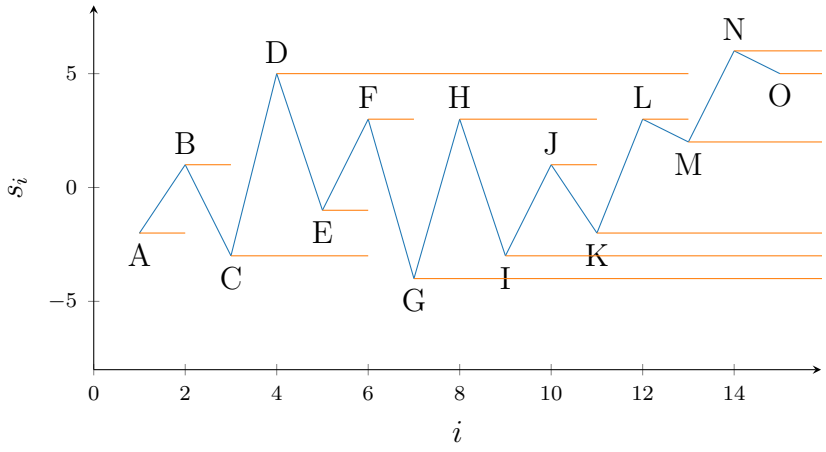

FIGURE 1.17 – Illustration of the vector S and the RFC algorithm. The full cycles correspond to the pair of orange lines enclosed by another pair of lines like $\{EF, HI, JK, LM\}$ while the half cycles are the remaining pairs $\{AB, BC, CD, DG, GN, NO\}$.

In Figure 1.17 shows how the RFC algorithm works. RFC takes its name from an analogy with rain drops falling off a pagoda roof. To visualize the analogy, the pagoda roof is the blue line that represents the vector S reversals in the figure. Then, we imagine the x-axis oriented vertically with its positive end towards the bottom : we can see the "rain", i.e. the orange lines in the figure, "dripping" from the pagoda. The full cycles correspond to each pair of lines that is enclosed by another pair of lines, like $\{EF, HI, JK, LM\}$,

while the half cycles are the remaining pairs $\{AB, BC, CD, DG, GN, NO\}$. The RFC algorithm is detailed in Appendix B.

Note that cycles can occur within cycles and that the cycles remaining open are counted as half cycles. These unclosed cycles may possibly be closed later on, but this cannot be seen in this finite time horizon. Let us consider a time horizon T split into two smaller time horizons T_1 and T_2, such that T is the concatenation of T_1 and T_2. Let us then denote the damage over a time horizon T by $\mathcal{D}(T)$. Due to the unclosed cycles, this theory of fatigue damage has a property which can seem counter-intuitive :

$$\mathcal{D}(T) \geq \mathcal{D}(T_1) + \mathcal{D}(T_2) \tag{1.32}$$

Indeed, the unclosed cycles in the time horizon T_1 cannot be closed during the time horizon T_2 if the latter is considered in isolation. Therefore :

$$\mathcal{D}(T_2) \leq \mathcal{D}(T) - \mathcal{D}(T_1) \tag{1.33}$$

because unclosed cycles of T_1 can be closed during T_2, with increased fatigue damage as a consequence. This means that the fatigue damage added during T_2 depends on what happened in the previous load time series :

$$\mathcal{D}(T) - \mathcal{D}(T_1) = \mathcal{D}(T_2|T_1) \tag{1.34}$$

To summarize this subsection, estimating fatigue damage using the Palmgrem-Miner linear damage rule and RFC algorithm has the following consequences :

— Fatigue damage cannot be explicitly expressed as an integral over time.
— Fatigue damage on a given load time series depends on the previous load time series.
— Fatigue damage due to an event happening at the end of the time series can be significantly amplified by an event placed at the beginning of the time series.

Note that the effects of the second assertion can be alleviated by considering a relatively long time horizon. 'Relatively long' means having a large number of cycles, and possibly full cycles. It should be stressed that if no cycles are remaining before that, the RFC algorithm counts the unclosed cycles as half cycles, and the second assertion is not a problem any more.

In summary, this RFC algorithm problematizes the incorporation of fatigue damage as an objective function in an optimal control problem. However, alternative methods for quantifying fatigue damage have been developed in attempts to consider fatigue damage in an optimal control problem.

1.3.3 Fatigue in Optimal Control of Wind Turbines

It was previously demonstrated that the incorporation of fatigue damage estimation using the Palmgrem-Miner linear damage rule with RFC into an optimal control problem can be problematic. The purpose of this subsection is to examine the state of the art in methods aimed at incorporating a fatigue reduction objective in an optimal control problem.

Quadratic forms

The most widely-used method in WTC to substitute fatigue damage in optimal control problems is using quadratic forms as an objective function [16, 13, 56, 57, 10, 58]. A

quadratic cost function J is in continuous time an integral of a second order polynomial of the system states and inputs, denoted respectively by x and u, over the time horizon considered $[t_0, t_f]$:

$$J = \int_{t_0}^{t_f} \left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) \right) d\tau \tag{1.35}$$

where Q and R are respectively semi-definite and definite positive matrices. In discrete-time, this cost function can be turned into a sum over N instants sampled over the horizon :

$$J = \sum_{k=1}^{N} x(t_k)^T Q x(t_k) + u(t_k)^T R u(t_k) \tag{1.36}$$

where t_k is the k^{th} time instant of the optimization horizon. This kind of cost function is especially preferred in optimization problems because it is convex and infinitely differentiable, which allows the cost function to be smooth. The convexity of the cost function allows the possibility that the optimization problem be convex, provided that the equality and inequality constraints form, respectively, a hyperplane and a convex set. Moreover, the Hessian of the optimal control problem is constant, which is important for the resolution of the problem. The infinite differentiability property ensures the smoothness of the cost function, which can help avoid instabilities in numerical optimization.

In [57], the DEL of various load time series are plotted against their corresponding standard deviation, for two Wöhler coefficients of 3 and 10. It was observed that the relationship between standard deviation and DEL is almost linear. Therefore the authors concluded that minimizing variance is equivalent to minimizing standard deviation, which is equivalent to minimizing DEL, as their relationship is almost linear. The authors suggested it is equivalent to minimizing fatigue damage from the DEL definition.

In [56], the fatigue damages of various load time series are plotted against their variances, and in this case the relationship is not linear any more, as fatigue damage is proportional to DEL power the Wöhler exponent, see (1.31). However, the authors draw the same conclusion, as it is approximately monotonically increasing, minimizing variance is equivalent to minimizing fatigue damage. This is approximately true for fatigue damage optimization of a single mechanical component ; however, when it comes to optimizing a fatigue damage trade-off between multiple mechanical components, it is quite false.

Let J_{fat} be a cost function expressed as the sum of the fatigue caused by two outputs of a process, denoted byy_1 and y_2 :

$$J_{\text{fat}}(y_1, y_2) = \mathcal{D}(y_1) + \mathcal{D}(y_2) \tag{1.37}$$

where $\mathcal{D}(y_i)$ is the fatigue damage of y_i. It is assumed that the standard deviation of y_1 and y_2 is always very close to their DEL, therefore :

$$\text{DEL}(y_1) = \text{std}(y_1) + \varepsilon(y_1) \tag{1.38a}$$
$$\text{DEL}(y_2) = \text{std}(y_2) + \varepsilon(y_2) \tag{1.38b}$$

where $\text{std}(y_i)$ is the standard deviation of y_i and

$$\varepsilon(y_i) = \text{DEL}(y_i) - \text{std}(y_i) \ll \min(\text{DEL}(y_1), \text{std}(y_i))$$

is a small error. Let J be the sum of y_1 and y_2 quadratic means, another cost function :

$$J(y_1, y_2) = ||y_1||_2^2 + ||y_1||_2^2 \tag{1.39a}$$
$$= \text{std}(y_1)^2 + \text{std}(y_1)^2 + K \tag{1.39b}$$
$$= \text{DEL}(y_1)^2 + \text{DEL}(y_1)^2 + K + \varepsilon \tag{1.39c}$$

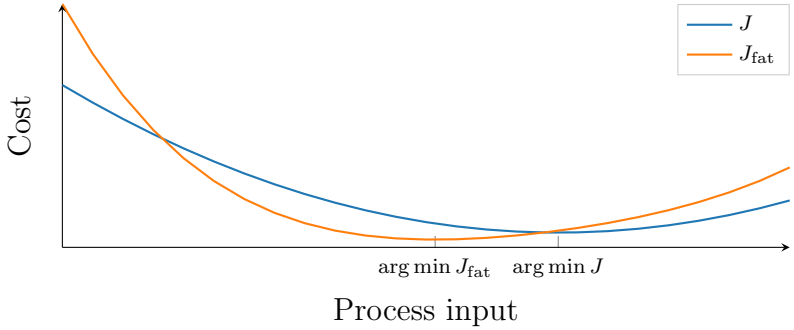

FIGURE 1.18 – Illustration of the difference between J and J_{fat} for $m_1, m_2 = 4$.

where $K \in \mathbb{R}$ is a constant depending on the average of y_1 and y_2, $\varepsilon = \varepsilon(y_1) + \varepsilon(y_2)$ and $||y_i||$ is the L_2 norm of y_i. From equation (1.31), there exists $w_1, w_2 \in \mathbb{R}^+$ such that :

$$J_{\text{fat}}(y_1, y_2) = w_1 \text{DEL}(y_1)^{m_1} + w_2 \text{DEL}(y_2)^{m_2} \tag{1.40}$$

This relation shows that if $w_1, w_2 \neq 2$:

$$J_{\text{fat}}(y_1, y_2) \neq J(y_1, y_2) \tag{1.41}$$

as illustrated in Figure 1.18. Therefore an optimization problem using J as cost function should not yield the same result as an optimization problem using J_{fat} as objective function. Note that for very specific weighting of $||y_1||_2^2$ and $||y_2||_2^2$, J can approximate J_{fat}. However, this is not true in the general case, which is why other methods have been created. They are presented below.

Spectral approach

It was mentioned previously that because of residual unclosed cycles in the RFC algorithm, fatigue damage of one mechanical component over a given time interval depends on the previous loading of the mechanical component. However, unclosed cycles might be closed sooner or later, which would add this fatigue damage at an unspecified time instant. These uncertainties show the stochastic nature of fatigue damage and RFC algorithm. In [59], load time series of a narrow-banded process are considered as a stochastic process. A stochastic process can be represented by its spectral moments [60], whose m^{th} order for the signal x is denoted by λ_m^x and defined as follows :

$$\lambda_m^x = \int_0^\infty f^m S_x(f) df \tag{1.42}$$

where S_x is the positive power spectrum density of the signal x. Note that λ_0^x, λ_2^x and λ_4^x are respectively the variances of signals x, $\dot{x}$ and $\ddot{x}$, i.e. the first and second time derivatives of x. In [59] and [60], the authors show that it is possible for a narrow-banded process to estimate its damage rate, denoted by $\dot{D}$, as follows :

$$\dot{\mathcal{D}} = \frac{1}{2\pi} \sqrt{\frac{\lambda_4^x}{\lambda_2^x}} \left(\frac{\sqrt{2\lambda_0^x}}{L^{\text{ult}} - |L^{\text{MF}}|} \right)^m \Gamma\left(1 + \frac{m}{2}\right) \tag{1.43}$$

where Γ is the Gamma function. As indicated above, this formula is limited to narrow-banded processes. For its general application to random processes, a correction is proposed in [60], involving a fraction of polynomials of λ_0^x, λ_2^x and λ_4^x. The damage rate can thus be approximated by a nonlinear function of spectral moments, ultimate load and fixed-mean

load, whose formulation depends on the process examined.

In [61], the authors show how to relate the transfer function of a process to its spectral moments. This relationship is then used in an optimization to determine the feedback gain that minimizes the expectation of fatigue of the closed-loop system. This method is then employed for the synthesis of a CPC controller, with the additional constraint that it must not deteriorate the rotor rotational velocity regulation of a benchmark controller. Eventually, fatigue damage reduction is effectively obtained.

The advantage of such a controller is that fatigue minimization is explicitly expressed as a control objective, using a spectral-based fatigue damage modelling. However, it is limited to a simple state feedback structure and does not allow the incorporation of feedforward information that could be obtained with LiDAR measurements. It is important to note that not only the variance of a load time series is involved in fatigue estimation, but also the variances of the first and second time derivatives.

Quadratic cost with dynamically varying parameters

Quadratic cost functions have interesting properties for optimization nevertheless, as they are convex and infinitely differentiable. In [62], the authors relied on the assumption that quadratic cost functions can be used for fatigue damage approximation in an MPC framework, but the parameters of the quadratic cost function must be adapted throughout the simulation, in order to match an online fatigue estimation.

In [62], fatigue damage is estimated online using a Preisach hysteresis operator [63]. In [64], the equivalence between this method of fatigue estimation and the one using symmetric RFC (neglecting residual half cycles) with the Palmgrem-Miner linear damage rule is provided. The authors of [62] claim that the hysteresis method has an advantage over the RFC algorithm method because it allows a closed-form cost function and can be estimated online. The closed-form property enables us to consider this kind of cost function in a continuous optimal control problem [65, 66]. However, this optimal control problem is hard to solve, and to the best of my knowledge this solution has not been implemented yet in the literature on fatigue optimization. Moreover, the classic RFC with linear damage rule method can be implemented online, if the residual reversals are kept in memory in order to possibly close them later, as is done in [67]. Therefore, this online RFC method could have been used in place of the hysteresis operator method for online fatigue estimation.

The quadratic cost function supposed to approximate fatigue was considering a single mechanical component, whose load history is denoted by $x \in \mathbb{R}$. The quadratic cost function, denoted by $\mathcal{F}$, is parameterized using parameters a and g :

$$\mathcal{F}(a, g, t) = ax(t)^2 + g\dot{x}(t)^2 \tag{1.44}$$

It is important to note that the authors used the first derivative of the load history information $\dot{x}$ in the cost function, as suggested in [59, 60, 61]. The fatigue damage endured by the mechanical component over an horizon $[0, T_k]$ is denoted by h_k and could be estimated using either of the online fatigue estimation methods. The fatigue damage approximation using the quadratic cost function is denoted by $\hat{h}(k, a, g)$ and expressed as follows :

$$\hat{h}(k) = \int_0^{T_k} \mathcal{F}(a, g, t)dt \tag{1.45}$$

Eventually, the authors used an autoregressive model with exogenous input for the dynamic identification of the couple (a, g) that minimizes the following cost function :

$$V_N(a,g) = \frac{1}{N}\sum_{i=1}^{N}\Big(h_i - \hat{h}(i,a,g)\Big)^2 \tag{1.46}$$

where N is the horizon considered for the identification of (a, g). Therefore, the dynamic parameters a_k and g_k, which are the parameters identified at the k^{th} time instant, are used in an MPC objective function, in order to account for fatigue damage cost. The fatigue damage cost J_{fat} is taken into account in the MPC objective function over the prediction horizon $[t_k, t_k + T]$ as follows :

$$J_{\text{fat}} = \int_{t_0}^{t_f} \mathcal{F}(a_k, g_k, t)dt \tag{1.47}$$

where $T = 3\ s$ in [62].

This MPC featuring a dynamically identified parameterized cost function was successfully implemented on a HAWT aero-elastic simulator in order to regulate the rotor rotational speed, while alleviating tower bending moment fatigue loads. Significant fatigue reduction was eventually obtained compared to a baseline MPC controller, whose tuning parameters seemed to have been chosen arbitrarily.

Optimization with dynamic objective function

The latest contribution on the incorporation of fatigue damage in the cost function of an optimal control problem aims at directly using the outputs of the RFC algorithm. Additional outputs that the RFC algorithm can provide are the time instants corresponding to the minimum and maximum reversals of the cycle s, denoted respectively by $t_{\text{max,s}}$ and $t_{\text{min,s}}$ [68]. Using this information, it is possible to derive $\bar{L}_s$ and L_s^R as follows :

$$\bar{L}_s(y_k, t_{\text{max,s}}, t_{\text{min,s}}) = \frac{y_k(t_{\text{max,s}}) + y_k(t_{\text{min,s}})}{2} \tag{1.48a}$$

$$L_s^R(y_k, t_{\text{max,s}}, t_{\text{min,s}}) = \frac{y_k(t_{\text{max,s}}) - y_k(t_{\text{min,s}})}{2} \tag{1.48b}$$

where y_k is the k^{th} output trajectory of a dynamic system whose fatigue damage must be minimized. It is thus possible to express the fatigue damage cost of the output y_k denoted by $J_{\text{fat}}(y_k)$ as follows :

$$J_{\text{fat}}(y_k, \text{RFC}(y_k)) = \sum_{s=1}^{N_{\text{cyc}}} n_s \left(\frac{1}{2}\frac{L_s^R(y_k, t_{\text{max,s}}, t_{\text{min,s}})}{L_k^{\text{ult}} - |\bar{L}_s(y_k, t_{\text{max,s}}, t_{\text{min,s}})|}\right)^{m_k} \tag{1.49}$$

where $\text{RFC}(y_k)$ is the result of the RFC algorithm on the time series y_k, which gives for each hysteresis cycle s the corresponding $t_{\text{max,s}}$ and $t_{\text{min,s}}$. It is thus possible to compute the gradient of $J_{\text{fat}}(y_k)$ with respect to the input trajectory of a dynamic system using the chain rule, and estimate its Hessian. The estimation of the gradient and the Hessian allows us to approximate this NonLinear Programming (NLP) with a Quadratic-Programming (QP) and obtain a new input trajectory. This procedure, involving RFC, must be repeated for each gradient descent step. This method is referred to as Direct Online RainFlow Counting (DORFC) MPC in the literature. It should be pointed out that in this formulation of J_{fat} (1.49), only the time instants corresponding to reversals are considered in the cost function, and therefore this optimal control problem can only be solved using

single-shooting [69], which is less computationally efficient than multiple-shooting [69].

The authors of [68] claim that the advantages of this method are that it directly incorporates the estimation of fatigue damage using RFC as a control objective and optimizes it efficiently, which allows good control performance with minimal tuning effort. On the other hand, they claim that the method's drawbacks are the difficulty of its implementation. The MPC formulation is not standard, as its objective function is not the integral of a stage cost over time. Moreover, it does not take into account unclosed cycles, and therefore closed-loop control performance could suffer from cycles closing at unexpected instants.

In my personal opinion, there is also the risk that the more components are considered, the more RFC estimations are required (one per component per gradient descent step), which could significantly increase the computational cost. Moreover, in [68], components with Wöhler coefficients only up to 4 were considered in the simulations. Therefore, it is not guaranteed that this method would behave appropriately, due to numerical issues during optimization, with Wöhler coefficients as high as 10 for components made of glass fibre, such as rotor blades.

Summary

The main takeaways from the state of the art on the inclusion of the fatigue reduction objective in an optimal control problem are the following :

— Quadratic forms do not allow us to accurately represent fatigue damage.
— The damage rate is nonlinearly related to the spectral moments of a signal, and thus to quadratic forms of the signal and its time derivatives.
— The state-of-the-art optimal control strategies aimed at minimizing fatigue damage, presented in Subsection 1.3.3, are both MPCs with dynamic objective functions.

1.4 Thesis objectives and outline

In this chapter, it was first shown that IPC can effectively help reducing fatigue of HAWT blades. Nevertheless, it significantly increases the fatigue of the blade pitch actuators and possibly other components, which makes IPC controversial, and its tuning difficult. As CPC and generator torque controls are respectively designed to regulate and maximize power production, IPC in addition to CPC must be designed to minimize the fatigue cost of the turbine while minimizing the impact of IPC on power production. The fatigue cost of the turbine, denoted by $\mathcal{J}$, can be approximated by the weighted sum of fatigue damages $\mathcal{D}_k$ and prices of replacement π_k, which is turbine-dependent, where k is an index corresponding to the turbine components considered. Therefore, the mathematical expression of $\mathcal{J}$ is the following :

$$\mathcal{J}(\mathbf{y}) = \sum_{k=1}^{N_c} \pi_k \mathcal{D}_k(\mathbf{y}_k) \tag{1.50}$$

where $\mathbf{y}$ is the output trajectory of the turbine, $\mathbf{y}_k$ is the k^{th} component output trajectory and N_c is the number of components considered.

1.4.1 The issue

To summarize, on the one hand there is fatigue theory, presented in Section 1.3 ; its estimation involves a complex algorithm, making the explicit incorporation of fatigue in an optimal control problem highly challenging. On the other hand, there are diverse control strategies, such as PID, LQR, MPC or H_∞, all depending on several design parameters, which can be summarized by a vector of parameters denoted by p.

Let $\mathbf{y}_{\mathrm{CL}}$ be the output of a closed-loop HAWT controlled by an IPC regulator $K(p)$ and disturbed by an exogenous force v. The output $\mathbf{y}_{\mathrm{CL}}$ depends thus on the vector of parameters p and the exogenous disturbance trajectory $\mathbf{v}$. For an efficient fatigue cost reduction, the vector of parameters $\bar{p}^\star$ must minimize the fatigue cost expectancy :

$$\bar{p}^\star = \arg\min_p \mathbb{E}\Big[\mathcal{J}\left(\mathbf{y}_{\mathrm{CL}}(\mathbf{v}, p)\right)\Big] \tag{1.51}$$

where $\mathbf{v}$ has a given probabilistic distribution.

However, the approaches presented in Subsection 1.3.3 suggest that this parameter p could be adapted with time in order to better reduce fatigue. Let $p^\star$ be the optimal vector of parameters for an exogenous disturbance $\mathbf{v}$:

$$p^\star(\mathbf{v}) = \arg\min_p \mathcal{J}\left(\mathbf{y}_{\mathrm{CL}}(\mathbf{v}, p)\right) \tag{1.52}$$

then an adaptive $p^\star$ could help in further reducing the fatigue cost expectancy $\mathcal{J}$.

An illustration of the above insight is given in Figures 1.19 and 1.20 ; where possible surface plots of fictitious $\mathcal{J}(\mathbf{y}_{\mathrm{CL}}(\mathbf{v}, p))$ are plotted.

— In Figure 1.19, $p^\star(\mathbf{v})$ which is denoted by the red curve is constant. Therefore, $\bar{p}^\star = p^\star$ and adapting $p^\star$ with $\mathbf{v}$ will not allow further reduction of the fatigue cost expectancy.

— In Figure 1.20, $p^\star$ varies with $\mathbf{v}$. Therefore, selecting a constant $\bar{p}^\star$, which minimizes the fatigue cost expectancy, requires using a p value which does not minimize $\mathcal{J}$ for most values of $\mathbf{v}$. It is thus obvious in this case that an adaptive $p^\star$ such as the one denoted by the red curve can help in reducing the fatigue cost expectancy $\mathcal{J}$.

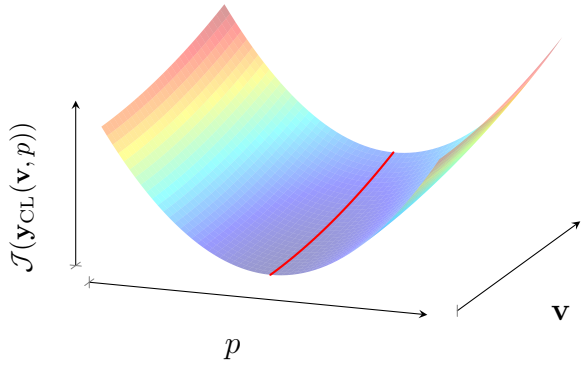

FIGURE 1.19 – Surface plot of $\mathcal{J}(\mathbf{y}_{\mathrm{CL}}(\mathbf{v}, p))$ for a case where $p^\star(\mathbf{v}) = \bar{p}^\star$ is constant. The red curve shows the evolution of $p^\star$ with the wind trajectory $\mathbf{v}$.

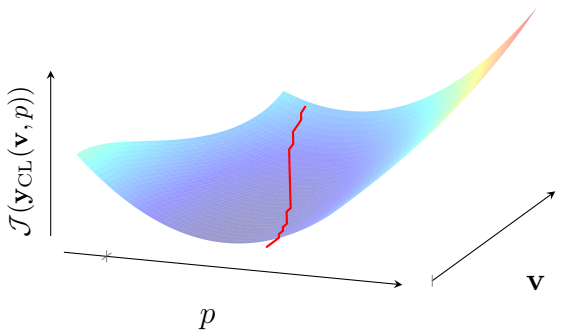

FIGURE 1.20 – Surface plot of $\mathcal{J}(\mathbf{y}_{\mathrm{CL}}(\mathbf{v}, p))$ for a case where $p^\star(\mathbf{v})$ varies. The red curve shows the evolution of $p^\star$ with the wind trajectory $\mathbf{v}$.

The issues that this thesis tries to resolve are the following :

— Is there an advantage in adapting the controller vector of parameters p for an efficient reduction of the fatigue cost expectancy $\mathcal{J}$?
— If yes, is the fatigue cost expectancy reduction significant ? How could the vector of parameters p be efficiently adapted, knowing that the expression of the fatigue cost $\mathcal{J}$ is not suited to be set as an optimal control problem objective function ?

1.4.2 The outline

The outline of this thesis is the following :

Chapter 2 This chapter shows, via a study of the influence of MPCs parameters on the fatigue cost, that there is no universal MPC design configuration with a standard form for fatigue cost reduction. Therefore, it would be interesting to adapt the controller parameters, but how ?

Chapter 3 In this chapter, a data-driven cost function is derived, allowing us to accurately approximate the fatigue cost $\mathcal{J}$ in a wide range of operating conditions. Moreover, this new cost function can be efficiently used in open-loop optimal control problems for fatigue cost reduction.

Chapter 4 This chapter aims at using the fatigue-oriented cost function, presented in the previous chapter, for an online adaptation of MPC parameters p. Two implementations are presented, one of which is very efficient in reducing fatigue cost expectancy, compared to an individual controller with fixed parameters.

Chapter 5 It was shown several times that there is an interest in adapting the vector of parameters based on the disturbance trajectory $\mathbf{v}$. Therefore, a data-driven solution is proposed to learn the function $p^\star(\mathbf{v})$, defined by (1.52), in order to adapt online the vector of parameters based on the system environment.

Chapter 6 Eventually, the conclusion is drawn, the main results are detailed and perspectives on the approaches presented in this thesis are given.

Chapitre 2

Study on the influence of MPC prediction horizon and design on closed-loop fatigue cost

Model Predictive Control (MPC) is an optimal control strategy that consists in iteratively solving an optimal control problem, whose solution is used to update the input to the system. The optimal control problem aims at minimizing a cost function, denoted J, with respect to the input trajectory of a given dynamic system over a prediction horizon T, for a given initial condition x_0. This optimal control problem can be subjected to constraints on system inputs or states. Moreover, if known, feedforward trajectory information can be added, which allows us to optimally anticipate the rejection of disturbances. Let $(P^{(T)})$ be the open-loop optimal control problem to be solved in an MPC over a prediction horizon of length T. MPC is based on the assumption that the system, in closed-loop, using the solution of the open-loop optimal control problem $(P^{(T)})$, approximates the solution of $(P^{(T_{\text{long}})})$ where $T_{\text{long}} \gg T$.

MPC is thus an obvious solution to the problem stated in Section 1.4, which is minimizing fatigue cost $\mathcal{J}$. Indeed, a LiDAR-assisted MPC IPC strategy gave one of the best results among the studies aiming at alleviating fatigue loads in [13]. MPCs for IPC regulation of HAWT are designed in several other studies [50, 70, 22, 71] using various design settings for both blades and MBC coordinates.

In [72], a study on the influence of prediction horizons for LiDAR-assisted CPC MPCs leads to the conclusion that in some cases there are no clear reductions of the fatigue cost expectancy $\mathcal{J}$ with long horizons, which can seem surprising. Moreover, there is no clear consensus on the MPC design setting to be used for an IPC regulation aiming at reducing the fatigue cost, and many MPC design settings could be implemented for an efficient IPC regulation. It should be highlighted that both the design settings and the prediction horizon of an MPC can be seen as a vector of parameters p defining the MPC, which must be tuned appropriately in order to efficiently reduce fatigue cost.

In this chapter, a study presenting the influence of the MPC horizon and design settings for IPC regulation on the fatigue cost $\mathcal{J}$ is conducted. This study eventually shows that no universal configuration or optimal prediction horizon for fatigue cost minimization exists. Moreover, it brings to light that **depending on the wind and turbine conditions some parameters are more efficient at reducing fatigue cost $\mathcal{J}$ than others**.

The outline of this chapter is the following :

— In Section 2.1, the material needed to understand the remainder of this chapter is defined, i.e the general MPC formulation, the parameters used in the fatigue cost estimation, and the derivation of the dynamic model used in the MPCs.
— In Section 2.2, the different MPC design settings used in the study are presented. These design settings are drawn from the MPC literature. Moreover, two parameterizations of the MPC's stage cost are proposed.
— In Section 2.3, the influence of the horizon and the MPC settings on the fatigue cost is studied. Moreover, as an MPC can be computationally expensive, some attention is also given to the computational complexity of the different MPC design settings.
— In Section 2.4, a discussion on this study is proposed.

2.1 Introduction to the MPC environment for IPC of HAWT

This section introduces all the tools, models and procedures needed. Thus, the MPC's general formulation is presented in Subsection 2.1.1. Then the parameters used for the fatigue cost function are given in Subsection 2.1.2. Finally, the process to obtain the LTI and LPV models used in the design of the following MPCs is detailed and compared to a HAWT aero-elastic simulator in Subsection 2.1.3.

2.1.1 General MPC formulation

The general formulation of an MPC continuous optimal control problem, denoted $(P^{(T)})$ is the following :

$$\min_{\mathbf{u}} \quad J(\mathbf{x},\mathbf{u},\mathbf{v}) = \int_{t_0}^{t_0+T} L(x(t),u(t),v(t),t)dt \tag{2.1a}$$
$$\dot{x}(t) = f(x(t),u(t),v(t),t) \qquad \forall t \in [t_0, t_0+T] \tag{2.1b}$$
$$x(t_0) = x_0 \tag{2.1c}$$
$$0 = h(x(t),u(t),v(t),t) \qquad \forall t \in [t_0, t_0+T] \tag{2.1d}$$
$$0 \geq g(x(t),u(t),v(t),t) \qquad \forall t \in [t_0, t_0+T] \tag{2.1e}$$

where :

- $t_0 \in \mathbb{R}$ is the initial time of the optimization.
- $\mathbf{x} = \{x(t) \in \mathbb{R}^{\mathrm{n_x}} | \forall t \in [t_0, t_0+T]\}$ is the state trajectory.
- $\mathbf{u} = \{u(t) \in \mathbb{R}^{\mathrm{n_u}} | \forall t \in [t_0, t_0+T]\}$ is the input trajectory.
- $\mathbf{v} = \{v(t) \in \mathbb{R}^{\mathrm{n_v}} | \forall t \in [t_0, t_0+T]\}$ is the feedforward information trajectory.
- $L : \mathbb{R}^{\mathrm{n_x}\times\mathrm{n_u}\times\mathrm{n_v}\times 1} \longrightarrow \mathbb{R}$ is the stage cost.
- $f : \mathbb{R}^{\mathrm{n_x}\times\mathrm{n_u}\times\mathrm{n_v}\times 1} \longrightarrow \mathbb{R}^{\mathrm{n_x}}$ is the map describing the dynamics.
- $h : \mathbb{R}^{\mathrm{n_x}\times\mathrm{n_u}\times\mathrm{n_v}\times 1} \longrightarrow \mathbb{R}^{\mathrm{n_x}}$ is a function that models the equality constraint on the system.
- $g : \mathbb{R}^{\mathrm{n_x}\times\mathrm{n_u}\times\mathrm{n_v}\times 1} \longrightarrow \mathbb{R}^{\mathrm{n_x}}$ is a function that models the inequality constraint.

This optimal control problem can be discretized in time and turned into a NonLinear Programming (NLP) with equality and inequality constraints, denoted $(P_{\mathrm{d}}^{(N)})$:

$$\min_{\{u_0,\ldots,u_N\}} \quad J_{\mathrm{d}}(\mathbf{x},\mathbf{u},\mathbf{v}) = \sum_{l=0}^{N} L(x_l,u_l,v_l,t_l)T_s \tag{2.2a}$$
$$x_{l+1} = f_{\mathrm{d}}(x_l,u_l,v_l,t_l) \qquad \forall l \in \{0,\ldots,N\} \tag{2.2b}$$
$$x(t_0) = x_0 \tag{2.2c}$$
$$0 = h(x_l,u_l,v_l,t_l) \qquad \forall l \in \{0,\ldots,N\} \tag{2.2d}$$
$$0 \geq g(x_l,u_l,v_l,t_l) \qquad \forall l \in \{0,\ldots,N\} \tag{2.2e}$$

where J_{d} and f_{d} are the discrete versions of J and f respectively, T_s is the sampling time, $NT_s = T$, $t_l = lT_s$, $x_l = x(t_l)$, $u_l = u(t_l)$ and $v_l = v(t_l) \quad \forall l = \{0,\ldots,N\}$. The input of the system is updated using the first value of the optimal input sequence solution, denoted $u_0^\star$.

2.1.2 Fatigue cost function

In this chapter, the fatigue cost function will concern only the components related to M_{yaw}, M_{tilt}, θ_{yaw} and θ_{tilt}, in order to avoid disturbing the CPC regulation. If the CPC

regulation was considered, then power regulation and production should also be accounted for in the economic cost function. Note that the fatigue damage estimations performed on the yawing and tilting blade pitch angles are considered to be fatigue damage of the blade actuators, as they are an image of these actuators' cyclic behaviour. However, it is not totally accurate and a better image of fatique in the blade pitch actuators might have been obtained by considering the fatigue damage caused by actuator torque and pitch bearing travel. The fatigue cost function to minimize throughout this chapter is thus the following :

$$\mathcal{J}(\mathbf{y}) = \sum_{k=1}^{N_c=4} \pi_k \mathcal{D}_k(\mathbf{y}_k) \tag{2.3}$$

where the parameters of the fatigue estimation, (e.g. ultimate loads, price of replacement, Wöhler coefficients and fixed-mean load) are summarized in Table 2.1. These parameters are chosen in order to induce realistic fatigue costs, as they are not available from the turbine manufacturer.

Component	k	π_k	L_k^{ult}	m_k	L_k^{MF}
M_{yaw}	1	1000	1000	10	0
M_{tilt}	2	1000	1000	10	0
θ_{yaw}	3	1	$5\frac{8\pi}{180} \times 10^3$	4	0
θ_{tilt}	4	1	$5\frac{8\pi}{180} \times 10^3$	4	0

TABLE 2.1 – Summary of the parameters used for fatigue estimation of Chapter 1 simulations.

It should be noticed that other parameters could have been used for fatigue cost estimation. However, changing the parameters is not expected to modify the methodology or the qualitative conclusions given in this chapter.

2.1.3 Models

The National Renewable Energy Laboratory (NREL) provides an open-source HAWT simulator named Fatigue, Aerodynamic, Structures and Turbulence (FAST) [38, 34]. This software allows us to parameterize and simulate a HAWT using various modules interacting with each other for aerodynamics, elastic deformation and controls. It also permits us to linearize the dynamics of the parameterized HAWT with respect to a set of inputs, outputs and Degrees Of Freedom (DOF). It further offers the possibility of directly processing the MBC transform to automatically yield an LTI system (1.18) for given operating collective pitch angle, hub-height wind speed and rotor rotational speed, denoted respectively by $\bar{\theta}_{\text{col}}$, $\bar{v}_{\text{hh}}$ and $\bar{\omega}_r$.

This subsection presents the process used to obtain the LTI and LPV models which can later be used as MPC internal models. First of all, it is shown how $\bar{\theta}_{\text{col}}$ and $\bar{\omega}_r$ are related to $\bar{v}_{\text{hh}}$, which ultimately results in having linear models parameterized by $\bar{v}_{\text{hh}}$ only. Then I explain how the blade pitch actuator dynamics are integrated into the LTI systems resulting from the linearizations. Finally, the parameterization of the LPV system is detailed and compared to the HAWT nonlinear simulator FAST.

Parameterization of the linearizations

Due to the specifications of power regulation in regions II and III (see page 5 of Section 1.1), the rotor rotational speed must be regulated to the value ω_{nom} which is expected

to maximize the aerodynamic power in region II and regulate the aerodynamic power to its nominal value, denoted by P_{nom}, in region III. Therefore, in region II $\theta_{\text{col}} = 0$, as CPC is inactive, and $\omega_{\text{nom}}(v_{\text{hh}}) = \frac{\lambda^\star_0 v_{\text{hh}}}{R}$, where $\lambda^\star_{\theta_{\text{col}}}$ is the tip speed ratio value, defined in (1.12), that maximizes the aerodynamic power coefficient for a given collective pitch angle θ_{col}. The electrical generator must thus vary its torque in order to regulate ω_r around ω_{nom}.

In region III, the generator torque is constant and ω_r must be regulated around its nominal value, denoted by ω_{nom}, by varying θ_{col}. This problem is equivalent to regulating the aerodynamic power coefficient (see Figure 1.11) around $C_p(\lambda_{\text{nom}}(v_{\text{hh}}), \theta_{col}) = \frac{P_{\text{nom}}}{\frac{1}{2}\rho A v_{\text{hh}}^3}$, where $\lambda_{\text{nom}}(v_{\text{hh}}) = \frac{\omega_{\text{nom}} R}{v_{\text{hh}}}$ depends on v_{hh}. Hence, if the HAWT is correctly regulated by the CPC controller, then the operating collective blade pitch angle and rotor rotational speed, $\bar{\theta}_{\text{col}}$ and $\bar{\omega}_r$, are functions of $\bar{v}_{\text{hh}}$.

To summarize, it is not necessary to perform FAST linearizations for all existing triplets $(\bar{\theta}_{\text{col}}, \bar{v}_{\text{hh}}, \bar{\omega}_r)$, as $\bar{\theta}_{\text{col}}$ and $\bar{\omega}_r$ can be parameterized by $\bar{v}_{\text{hh}}$ due to the torque and CPC control. Therefore, the linearizations can be limited to the triplets $(\bar{\theta}_{\text{col}}(\bar{v}_{\text{hh}}), \bar{v}_{\text{hh}}, \bar{\omega}_r(\bar{v}_{\text{hh}}))$, which ultimately depend only on the operating hub-height wind speed $\bar{v}_{\text{hh}}$, where $\bar{\theta}_{\text{col}}(\bar{v}_{\text{hh}})$ and $\bar{\omega}_r(\bar{v}_{\text{hh}})$ are defined as follows :

$$\bar{\theta}_{\text{col}}(\bar{v}_{\text{hh}}) = \begin{cases} C^{-1}_{\text{p},\lambda_{\text{nom}}(\bar{v}_{\text{hh}})}\left(\frac{P_{\text{nom}}}{\frac{1}{2}\rho A \bar{v}_{\text{hh}}^3}\right) & \text{if} \quad v_{\text{nom}} < \bar{v}_{\text{hh}} < v_{\text{out}} \\ 0 & \text{if} \quad v_{\text{in}} < \bar{v}_{\text{hh}} \leq v_{\text{nom}} \end{cases} \tag{2.4a}$$

$$\bar{\omega}_r(\bar{v}_{\text{hh}}) = \begin{cases} \omega_{\text{nom}} & \text{if} \quad v_{\text{nom}} < \bar{v}_{\text{hh}} < v_{\text{out}} \\ \frac{\lambda^\star_0 \bar{v}_{\text{hh}}}{R} & \text{if} \quad v_{\text{in}} < \bar{v}_{\text{hh}} \leq v_{\text{nom}} \end{cases} \tag{2.4b}$$

where $C^{-1}_{\text{p},\lambda}$ is the inverse function of $C_p(\lambda, \theta)$ for fixed λ.

LTI model

The FAST linearization module allows us to obtain LTI systems for every blade azimuth angle specified. In order to avoid the dependency of the systems on the azimuth, it is common practice in wind turbine control to average the dynamics of the systems over the azimuth angles in order to obtain a single LTI model [71], depending on the operating wind speed $\bar{v}_{\text{hh}}$. Note that averaging the dynamics can be tricky, but it will be shown later in this paper that in this case the model obtained matches the nonlinear model FAST very well. According to (1.18) the LTI model is expressed as follows :

$$\dot{\tilde{x}} = \tilde{A}(\bar{v}_{\text{hh}})\delta\tilde{x}(\bar{v}_{\text{hh}}) + \tilde{B}(\bar{v}_{\text{hh}})\delta\tilde{u}(\bar{v}_{\text{hh}}) + \tilde{B}_d(\bar{v}_{\text{hh}})\delta d(\bar{v}_{\text{hh}}) \tag{2.5a}$$

$$\tilde{y} = \tilde{C}(\bar{v}_{\text{hh}})\delta\tilde{x}(\bar{v}_{\text{hh}}) + \tilde{D}(\bar{v}_{\text{hh}})\delta\tilde{u}(\bar{v}_{\text{hh}}) + \tilde{D}_d(\bar{v}_{\text{hh}})\delta d(\bar{v}_{\text{hh}}) + \tilde{y}_{op}(\bar{v}_{\text{hh}}) \tag{2.5b}$$

where $\delta\tilde{x}(\bar{v}_{\text{hh}}) = \tilde{x} - \tilde{x}_{op}(\bar{v}_{\text{hh}})$, $\delta\tilde{u}(\bar{v}_{\text{hh}}) = \tilde{u} - \tilde{u}_{op}(\bar{v}_{\text{hh}})$ and $\delta d(\bar{v}_{\text{hh}}) = d - d_{op}(\bar{v}_{\text{hh}})$.

In this chapter, the only degree of freedom considered for the linearization is the first eigenmode of the blade flapwise vibrations. The inputs are the collective, yawing and tilting blade pitch angles, and the outputs are the average, yawing and tilting out-of-plane blade root bending moments. Therefore, $\tilde{x} \in \mathbb{R}^6$, $\tilde{u} = [\theta_{col}, \theta_{yaw}, \theta_{tilt}]^T$ and $\tilde{y} = [\bar{M}, M_{\text{yaw}}, M_{\text{tilt}}]$. The full disturbance vector d is composed of :

— The hub-height wind speed
— The angle of the wind direction with the rotor axis and the vertical component of the wind velocity, which define the spatial direction of the wind.
— Horizontal and vertical wind shear, which parameterizes the spatial variations of the wind velocity in the rotor plane.

All these wind features can be estimated using LiDAR measurements with WFR algorithms ; their definitions can be found in [73]. Note that at a later stage, a reduced version of this vector might be considered instead.

Integrate blade pitch actuator dynamics

First of all, it should be noticed that FAST linearizations do not integrate the dynamics of the blade pitch actuators, as they are external to the system described by the simulator. The dynamics considered for the blade pitch actuators are a first-order low-pass filter :

$$\tau\dot{\theta}_i + \theta_i = \theta_i^{\text{sp}} \quad i \in \{1,2,3\} \tag{2.6}$$

where τ is the blade pitch actuator time constant equal to 1 second and θ_i^{sp} is the set-point value given to the actuator by the controller. It must be underlined that these dynamics are expressed in the blade coordinates and the MBC transform must be applied to them, in order to integrate blade dynamics in the LTI model obtained from FAST linearization expressed in MBC coordinates. After having applied the MBC transform, defined in (1.1), equation (2.6) becomes :

$$\tau\left(\omega_r\frac{dT(\psi)^{-1}}{d\psi}\tilde{u} + T(\psi)^{-1}\dot{\tilde{u}}\right) + T(\psi)^{-1}\tilde{u} = T(\psi)^{-1}u \tag{2.7}$$

where $u = T(\psi)[\theta_1^{\text{sp}}, \theta_2^{\text{sp}}, \theta_3^{\text{sp}}]^T$ is the vector of blade pitch angle set-points in the MBC coordinates given by the controller and θ_i^{sp} is the i^{th} blade pitch angle set-point. By multiplying (2.7) by $T(\psi)$ on the left hand side, it follows that :

$$\dot{\tilde{u}} = \underbrace{\left(-\frac{1}{\tau}I_3 - \omega_r T(\psi)\frac{dT(\psi)^{-1}}{d\psi}\right)}_{\tilde{\tau}}\tilde{u} + \frac{1}{\tau}u \tag{2.8}$$

where I_3 is the identity matrix of dimension 3 and $\tilde{\tau}$ is formulated as follows :

$$\tilde{\tau} = \begin{pmatrix} -\frac{1}{\tau} & 0 & 0 \\ 0 & -\frac{1}{\tau} & -\omega_r \\ 0 & \omega_r & -\frac{1}{\tau} \end{pmatrix} \tag{2.9}$$

Note that for a linearized system, ω_r is fixed to its operating value and becomes $\bar{\omega}_r(\bar{v}_{\text{hh}})$. Therefore, $\tilde{\tau}$ becomes a function of $\bar{v}_{\text{hh}}$.

Extension of the LTI model

It is possible to extend the LTI system of equations (2.5) with the actuator dynamics. To do this, let us define $\bar{A}$, $\bar{B}$, $\bar{B}_d$, $\bar{C}$, $\bar{D}$, $\bar{D}_d$, x and y :

$$\bar{A}(\bar{v}_{\text{hh}}) = \begin{pmatrix} \tilde{A}(\bar{v}_{\text{hh}}) & \tilde{B}(\bar{v}_{\text{hh}}) \\ 0 & \tilde{\tau}(\bar{v}_{\text{hh}}) \end{pmatrix} \quad \bar{B}(\bar{v}_{\text{hh}}) = \frac{1}{\tau}\begin{pmatrix} 0 \\ I_3 \end{pmatrix} \quad \bar{B}_d(\bar{v}_{\text{hh}}) = \begin{pmatrix} \tilde{B}_d(\bar{v}_{\text{hh}}) \\ 0 \end{pmatrix} \tag{2.10a}$$

$$\bar{C}(\bar{v}_{\text{hh}}) = \begin{pmatrix} \tilde{C}(\bar{v}_{\text{hh}}) & \tilde{D}(\bar{v}_{\text{hh}}) \\ 0 & I_3 \end{pmatrix} \quad \bar{D}(\bar{v}_{\text{hh}}) = \frac{1}{\tau}\begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \bar{D}_d(\bar{v}_{\text{hh}}) = \begin{pmatrix} \tilde{D}_d(\bar{v}_{\text{hh}}) \\ 0 \end{pmatrix} \tag{2.10b}$$

$$x = \begin{pmatrix} \tilde{x} \\ \tilde{u} \end{pmatrix} \qquad y = \begin{pmatrix} \tilde{y} \\ \tilde{u} \end{pmatrix} \tag{2.10c}$$

The extended system can be expressed as follows :

$$\dot{x} = \bar{A}(\bar{v}_{\text{hh}})(x - \bar{x}_{op}(\bar{v}_{\text{hh}})) + \bar{B}(\bar{v}_{\text{hh}})(u - \bar{u}_{op}(\bar{v}_{\text{hh}})) + \bar{B}_d(\bar{v}_{\text{hh}})(d - d_{op}(\bar{v}_{\text{hh}})) \tag{2.11a}$$

$$y = \bar{C}(\bar{v}_{\text{hh}})(x - \bar{x}_{op}(\bar{v}_{\text{hh}})) + \bar{D}(\bar{v}_{\text{hh}})(u - \bar{u}_{op}(\bar{v}_{\text{hh}})) + \bar{D}_d(\bar{v}_{\text{hh}})(d - d_{op}(\bar{v}_{\text{hh}})) + \bar{y}_{op}(\bar{v}_{\text{hh}}) \tag{2.11b}$$

where $\bar{x}_{op}$, $\bar{u}_{op}$ and $\bar{y}_{op}$ are the corresponding steady state, input and output of the HAWT system linearized around $\bar{v}_{\text{hh}}$.

Definition of the LPV model

A HAWT can be approximated by an LPV system, defined by a set of LTI systems which are polytopes of the LPV system. In order to constitute a basis of LTI systems defining the LPV system, ten linearizations are performed for different values of $\bar{v}_{\mathrm{hh}}$, ranging from 12 to 24m/s and equally spaced. Note that this range of wind speeds stays in region III.

In order to smoothly switch from one LTI system to another, a Gaussian kernel is used, whose i^{th} weight, denoted w_i, is defined as follows :

$$w_i(v_{\mathrm{hh}}) = \frac{e^{-\sigma||v_{\mathrm{hh}}-\bar{v}_{\mathrm{hh,i}}||_2^2}}{\sum_{j=1}^{10} e^{-\sigma||v_{\mathrm{hh}}-\bar{v}_{\mathrm{hh,j}}||_2^2}} \tag{2.12}$$

where $\sigma > 0$ parameterizes the radius of the kernel and $\bar{v}_{\mathrm{hh,i}}$ is the hub-height wind speed value for the i^{th} linearization, which parameterizes the kernel centre. Thus the LPV dynamic system is defined as follows :

$$\dot{x} = A(v_{\mathrm{hh}})(x - x_{op}(v_{\mathrm{hh}})) + B(v_{\mathrm{hh}})(u - u_{op}(v_{\mathrm{hh}})) + B_d(v_{\mathrm{hh}})(d - d_{op}(v_{\mathrm{hh}})) \tag{2.13a}$$

$$y = C(v_{\mathrm{hh}})(x - x_{op}(v_{\mathrm{hh}})) + D(v_{\mathrm{hh}})(u - u_{op}(v_{\mathrm{hh}})) + D_d(v_{\mathrm{hh}})(d - d_{op}(v_{\mathrm{hh}})) + y_{op}(v_{\mathrm{hh}}) \tag{2.13b}$$

where $x = \bar{x}$. Any matrix $X \in \{A, B, B_d, \cdot\}$ is defined as a weighted sum of kernel coefficients :

$$X(v_{\mathrm{hh}}) = \sum_{i=1}^{10} w_i(v_{\mathrm{hh}})\bar{X}_i(\bar{v}_{\mathrm{hh,i}}) \tag{2.14}$$

where $\bar{X}_i$ is a matrix or vector present in one of the LTI systems of type (2.11) at operating wind speed $\bar{v}_{\mathrm{hh,i}}$.

Comparison to a HAWT simulator

To evaluate how the LPV system behaves compared to a HAWT aero-elastic simulator, the LPV system is compared to FAST under a wind defined by the full disturbance vector presented in page 38. This wind is generated using the NREL wind generator TurbSim [73], for an average wind speed of 16 m/s and 10% turbulence intensity, conditions in which the turbine is likely to run during the study. The resulting hub-height wind speed time series is plotted in Figure 2.1 and varies over 5 polytopes of the LPV system. Therefore the wind is sufficiently turbulent to permit evaluation of the behaviour of the LPV system in several operating conditions of its polytopes.

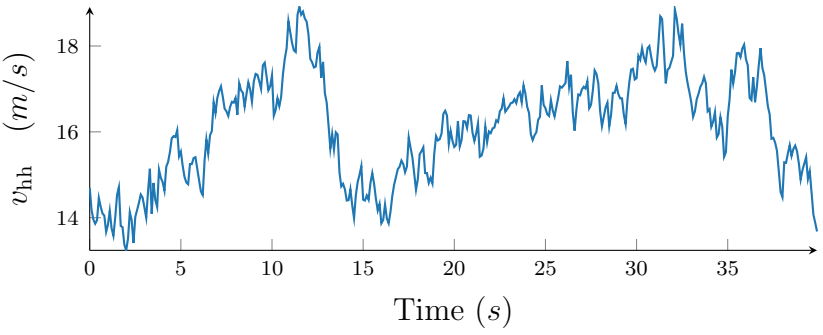

FIGURE 2.1 – Time series of v_{hh} used for the comparison of FAST and the LPV system.

The collective blade pitch angle is varied as a function of v_{hh}, while the yawing and tilting blade pitch angles are varied using an integrated Gaussian noise. These collective, yawing and tilting blade pitch angles are fed to the inverse Coleman transform defined by (1.1), yielding the three blade pitch angles whose time series are plotted in Figure 2.2. These blade pitch angles are given as input to the HAWT simulator, while the LPV system takes the collective, yawing and tilting blade pitch angles as inputs.

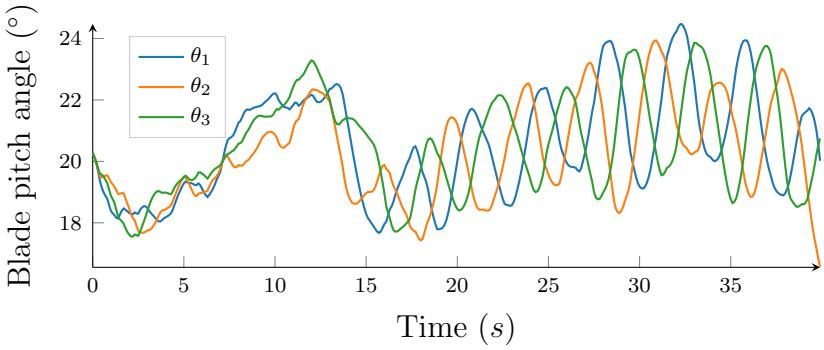

FIGURE 2.2 – Time series of the blade pitch angles used for the comparison of FAST and the LPV system.

The system is simulated during 40 seconds on the HAWT simulator FAST with only the first flapwise mode of the blades' DOF activated. The kernel radius parameter for the LPV model was set to $\sigma = 3$, which was found to be a good value. As can be seen in Figure 2.3, every kernel value is approximately equal to 1 at its centre and 0 for other kernel centres.

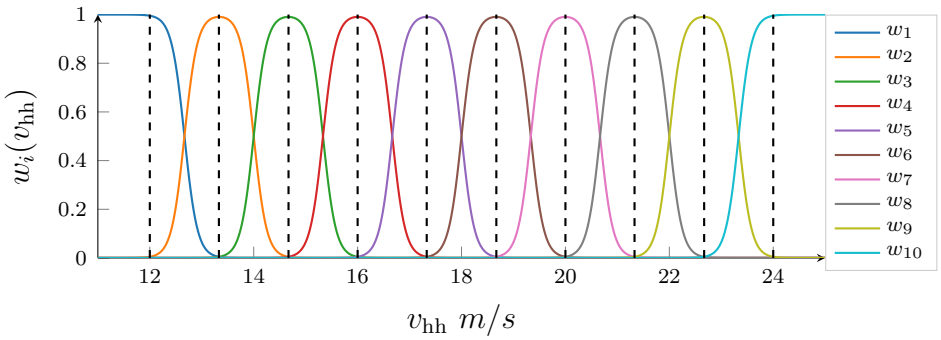

FIGURE 2.3 – Evolution of the Gaussian kernel values with the hub-height wind speed v_{hh} for $\sigma = 3$. The black dashed lines denote the centre of every kernel.

The time series of the obtained average, yawing and tilting blade root bending moments are plotted in Figure 2.4. Some discrepancies can be observed at the beginning of the time series, which can be explained by the fact that for the considered initial conditions, the HAWT blade vibrational states are far from their steady states. This kind of nonlinearity cannot be modelled by an LPV system, and therefore the system was not expected to model these dynamics. Nevertheless, this is not an issue, as an IPC regulator is supposed to regulate the blade states to a region relatively close to their steady states.

It is obvious that an LPV system can accurately model the system around operating points used for linearization. Therefore, once the HAWT simulator behaviour gets closer to the operating condition of the linearization, after approximately 5 seconds, it can be

observed that the LPV outputs closely match the HAWT simulator outputs.

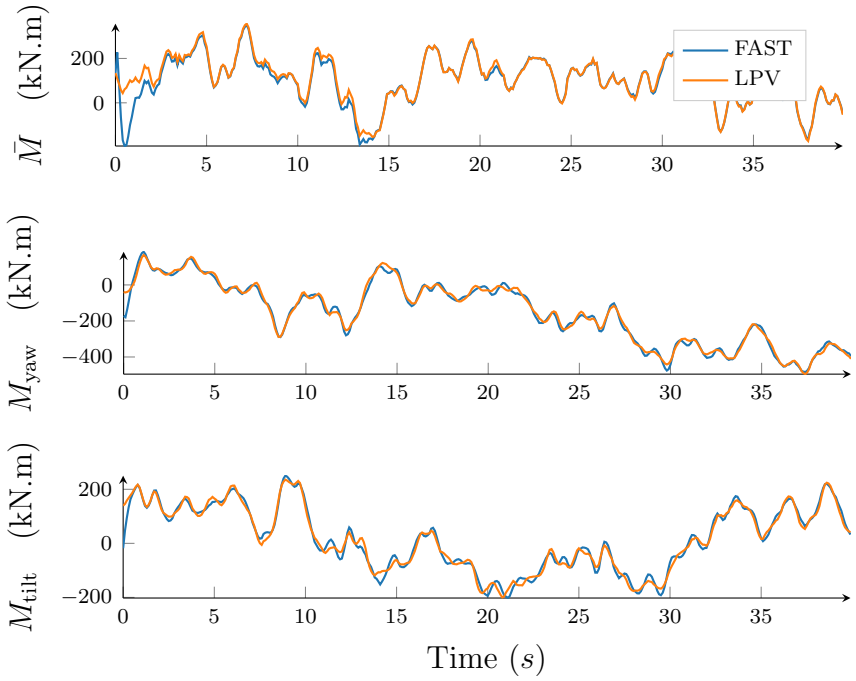

FIGURE 2.4 – Time series of the average, yawing and tilting moments obtained during the comparison of FAST and the LPV system.

It can be concluded that the LPV system obtained from FAST linearization represents relatively accurately the HAWT dynamics in open-loop simulations around the operating conditions of its polytopes. This is convenient for its use as an internal model in closed-loop MPC, because the MPC will regulate the system around the operating points, where the model is the most accurate. Moreover, potential model mismatch is not an issue for a regularly updated MPC.

2.2 Presentation of the different MPC design settings

This section describes the various MPC design settings used in the comparative study. Two of them are taken from the IPC literature in MBC coordinates, while another two are based on the more general MPC literature on nonlinear systems, for which no studies in the IPC literature could be found. The MPC design settings presented are the following :

Linear MPC MPC with a linear internal model, similar to the one designed in [71, 70], using a single operating point and model, already applied in the IPC literature.

Multiple MPC MPC using the weighted outputs of individual linear MPCs, based on [74, 75, 76].

Gain scheduled MPC linear MPC parameterized by the current wind speed v_{hh}, based on [77].

LPV MPC MPC with an LPV internal model, varying with the predicted wind speed over the prediction horizon, similar to the one designed in [13], already applied in the IPC literature.

Finally, two parameterizations of the MPC stage cost, based on the fatigue cost formulation, are depicted in Subsection 2.2.5, which doubles the number of design settings.

Moreover, a summary of all the MPC design settings defined later is given in Table 2.2 at the end of this section.

2.2.1 Linear MPC

The first presented MPC uses an LTI internal model, obtained from a FAST linearization for a given hub-height wind speed $\bar{v}_{\text{hh}}$. The cost function of the MPC is a quadratic cost function and with only linear equality constraints, making thus the open-loop optimization problem convex. In this subsection, as a single LTI system is considered, every dependence on $\bar{v}_{\text{hh}}$ from equation (2.11) will be omitted in order to alleviate the notations.

The continuous problem is addressed first, in line with the approach outlined in Subsection 2.1.1. The continuous problem is then discretized in time and then a brief description of the MPC analytical solution is given, which is possible as no inequality constraints are present in the formulation.

Continuous problem

The continuous open-loop optimal control problem which must be solved by the MPC at time instant t_0 is the following :

$$\min_{\bar{\mathbf{u}}} \quad J = \int_{t_0}^{t_0+T} \Big((y(\tau) - y_{\text{ref}})^T Q\,(y(\tau) - y_{\text{ref}}) + \delta u(\tau)^T R \delta u(\tau) \Big)\, d\tau \tag{2.15a}$$

$$\text{s.t.} \quad \dot{x}(t) = \bar{A}\delta x(t) + \bar{B}\delta u(t) + \bar{B}_d \delta d(t) \quad \forall t \in [t_0, t_0+T] \tag{2.15b}$$

$$y(t) = \bar{C}\delta x(t) + \bar{D}\delta u(t) + \bar{D}_d \delta d(t) + \bar{y}_{op} \quad \forall t \in [t_0, t_0+T] \tag{2.15c}$$

$$x(t_0) = x_0 \tag{2.15d}$$

$$\theta_{\text{col}}(t) = \theta_{\text{col,CPC}} \quad \forall t \in [t_0, t_0+T] \tag{2.15e}$$

where :

- $y_{\text{ref}} = [\bar{M}_{op}, 0, 0, \bar{u}_{op}^T]^T \in \mathbb{R}^6$ is the reference output that the system must track.
- $\theta_{\text{col,CPC}}$ is the collective blade pitch angle value given by the CPC.
- x_0 is the current observed or measured state.
- $Q \in \mathbb{R}^6$ and $R \in \mathbb{R}^3$ are respectively positive semi-definite and positive definite matrices, which weighs the MPC stage cost.
- The $\delta x = (x - \bar{x}_{op})$, $\delta u = (u - \bar{u}_{op})$ and $\delta d = (d - \bar{d}_{op})$ vectors represent the difference of the state, input and disturbance values from their respective operating values.

The information on the trajectory of $d(t)$ over the prediction horizon is assumed to be known with precision. Note that the equality constraint (2.15e) is added in order to limit the influence of the IPC regulator on the CPC regulator. The collective blade pitch angle is thus considered as an external parameter, which will ensure that the individual blade pitch angles oscillate around the collective blade pitch angle value given by the CPC regulator.

Note too that the constraints on the blade pitch actuators, which mainly correspond to a saturation on the blade pitch actuator rotating speed, are not considered. This choice is made because considering the nonlinear formulation of these constraints on θ_{col}, θ_{yaw} and θ_{tilt} significantly increases the computational burden of the MPCs. Moreover, considering a sufficient penalization on the control inputs allows us to avoid activating this saturation.

Discrete problem

In order to solve the optimal control problem (2.15) efficiently, it is proposed to discretize it and convert it into a finite QP problem. The discretized version of (2.15) is expressed as follows :

$$\min_{\{u_0,\dots,u_N\}} \quad J_\mathrm{d} = \sum_{l=1}^{N} \left(y_l - y_\mathrm{ref}\right)^T Q \left(y_l - y_\mathrm{ref}\right) T_s + \sum_{l=0}^{N} \delta u_l^T R \delta u_l T_s \tag{2.16a}$$

$$\text{s.t.} \quad x_{l+1} = \bar{A}_\mathrm{z}\delta x_l + \bar{B}_\mathrm{z}\delta u_l + \bar{B}_\mathrm{d,z}\delta d_l + \bar{x}_{op} \quad \forall l \in \{0,\dots,N\} \tag{2.16b}$$

$$y_l = \bar{C}\delta x_l + \bar{D}\delta u_l + \bar{D}_d\delta d_l + \bar{y}_{op} \quad \forall l \in \{0,\dots,N\} \tag{2.16c}$$

$$x(t_0) = x_0 \tag{2.16d}$$

$$\theta_\mathrm{col,l} = \theta_\mathrm{col,CPC} \quad \forall l \in \{0,\dots,N\} \tag{2.16e}$$

where for any vector or scalar X, $X_l = X(t_l)$. Moreover, $\bar{A}_\mathrm{z}$, $\bar{B}_\mathrm{z}$ and $\bar{B}_\mathrm{d,z}$ are respectively the discrete versions of $\bar{A}$, $\bar{B}$ and $\bar{B}_d$. Note that the solution of (2.16) is equivalent to (2.15), provided that T_s is sufficiently small. This quadratic problem can be solved using QP solvers such as `CVX` [78] or NLP solvers such as `CasADi` [79] or `Acado-toolkit` [80].

Analytical solution

A QP such as (2.16) without inequality constraint can also be solved analytically. The solution to this QP, denoted by $\mathbf{u}_\mathrm{z}^\star$, can be expressed as a linear combination of δx_0, $\theta_\mathrm{col,CPC}$ and $\delta\mathbf{d}_\mathrm{z}$ with additional constant terms :

$$\mathbf{u}_\mathrm{z}^\star = -K_1\delta x_0 - K_2\left(\theta_\mathrm{col,CPC} - \bar{\theta}_\mathrm{col,op}\right) - K_3\delta\mathbf{d}_\mathrm{z} - K_4\left(\bar{y}_\mathrm{op} - y_\mathrm{ref}\right) + \mathbf{1}_\mathrm{N,3}\bar{u}_\mathrm{op} \tag{2.17}$$

where K_1, K_2, K_3 and K_4 are matrices of appropriate dimension. Details about the derivation of these matrices can be found in Appendix C.

Hence, for the online implementation, only the matrix multiplications of (2.17) are needed. This formulation of the solution allows us to reduce significantly the computational burden of the online optimization. For instance, the resolution of this optimization for a prediction horizon of $N = 20$ and a disturbance vector of dimension three takes on average about 100 ms using `CasADi` against 1 ms using the analytical solution. However, it should be noted that this analytical solution cannot be obtained if inequality constraints are involved in the formulation.

To summarize, this linear MPC gives an optimal input $u_0^\star$, which is the first value of the optimal input sequence $\mathbf{u}_\mathrm{z}^\star$, for :
— The current state x_0
— The collective blade pitch angle $\theta_\mathrm{col,CPC}$
— Discrete receding horizon disturbance trajectory $\mathbf{d}_\mathrm{z}$
The optimal input of the MPC using the LTI system linearized around the i^th hub-height wind speed $\bar{v}_\mathrm{hh,i}$ is denoted by $u_{0(\mathrm{i})}^\star(x_0, \theta_\mathrm{col,CPC}, \mathbf{d}_\mathrm{z})$.

2.2.2 Multiple MPC

The internal model of the linear MPC is obtained from a linearization of the nonlinear HAWT simulator around a given operating point, parameterized by $\bar{v}_\mathrm{hh}$. The linear MPC is thus supposed to regulate the latter properly around the same operating point, where the linear model matches the HAWT simulator's nonlinear one. In order to implement such a controller on a nonlinear system, a possible approach, called Multiple MPC

(MMPC), consists in weighting the outputs of several linear MPCs designed around different operating points and depending on a scheduling variable [74, 75, 76].

The MMPC presented here considers ten linear MPCs and weighs their outputs with a Gaussian kernel similar to the one defined in (2.12). The MMPC output, denoted by u_{MMPC}, is thus a function of the current wind speed v_{hh}, current state x_0 and receding horizon disturbance trajectory $\mathbf{d}_{\mathrm{z}}$, expressed as follows :

$$u_{\mathrm{MMPC}}(v_{\mathrm{hh}}, x_0, \mathbf{d}_{\mathrm{z}}) = \sum_{i=1}^{10} \tilde{w}_i(v_{\mathrm{hh}}) u^{\star}_{0(\mathrm{i})}\left(x_0, \theta_{\mathrm{col,CPC}}, \mathbf{d}_{\mathrm{z}}\right) \tag{2.18}$$

where $\tilde{w}_i$ has a definition similar to w_i in (2.12) :

$$\tilde{w}_i(v_{\mathrm{hh}}) = \frac{e^{-\sigma_{\mathrm{MMPC}}||v_{\mathrm{hh}} - \bar{v}_{\mathrm{hh,i}}||_2^2}}{\sum_{j=1}^{10} e^{-\sigma_{\mathrm{MMPC}}||v_{\mathrm{hh}} - \bar{v}_{\mathrm{hh,j}}||_2^2}} \tag{2.19}$$

where σ_{MMPC} is the kernel radius parameter, which parameterizes the smoothness of switches between controllers.

2.2.3 Gain scheduled MPC

Gain-scheduled MPC is a regulator whose open-loop optimal control problem is parameterized by a scheduling variable, which is the current hub-height wind speed, denoted by v_{hh}. In the MPC optimal control problem, both the internal model and the objective function are parameterized by v_{hh}. It should be noted that in the optimal control problem, the current hub-height wind speed is referred as $v_{\mathrm{hh,0}}$. This gain-scheduled MPC must not be confused with MMPC, which is a linear combination of the results of several linear MPCs.

The discrete open-loop optimization problem defined in (2.16) is thus transformed as follows :

$$\min_{\{u_0,\ldots,u_N\}} \quad \begin{aligned} J_{\mathrm{d}} = &\sum_{l=1}^{N} \left(y_l - y_{\mathrm{ref}}(v_{\mathrm{hh,0}})\right)^T Q(v_{\mathrm{hh,0}}) \left(y_l - y_{\mathrm{ref}}(v_{\mathrm{hh,0}})\right) T_s + \\ &\sum_{l=0}^{N} \delta u_l(v_{\mathrm{hh,0}})^T R(v_{\mathrm{hh,0}}) \delta u_l(v_{\mathrm{hh,0}}) T_s \end{aligned} \tag{2.20a}$$

$$\text{s.t.} \quad x_{l+1} = A_{\mathrm{z}}(v_{\mathrm{hh,0}})\delta x_l(v_{\mathrm{hh,0}}) + B_{\mathrm{z}}(v_{\mathrm{hh,0}})\delta u_l(v_{\mathrm{hh,0}}) + B_{\mathrm{d,z}}(v_{\mathrm{hh,0}})\delta d_l(v_{\mathrm{hh,0}}) + x_{op}(v_{\mathrm{hh,0}}) \quad \forall l \in \{0,\ldots,N\} \tag{2.20b}$$

$$y_l = C(v_{\mathrm{hh,0}})\delta x_l(v_{\mathrm{hh,0}}) + D(v_{\mathrm{hh,0}})\delta u_l(v_{\mathrm{hh,0}}) + D_d(v_{\mathrm{hh,0}})\delta d_l(v_{\mathrm{hh,0}}) + y_{op}(v_{\mathrm{hh,0}}) \quad \forall l \in \{0,\ldots,N\} \tag{2.20c}$$

$$x(t_0) = x_0 \tag{2.20d}$$

$$\theta_{\mathrm{col,l}} = \theta_{\mathrm{col,CPC}} \quad \forall l \in \{0,\ldots,N\} \tag{2.20e}$$

where $A_{\mathrm{z}}(v_{\mathrm{hh,0}})$, $B_{\mathrm{z}}(v_{\mathrm{hh,0}})$ and $B_{\mathrm{d,z}}(v_{\mathrm{hh,0}})$ are the discrete versions of $A(v_{\mathrm{hh,0}})$, $B(v_{\mathrm{hh,0}})$ and $B_{\mathrm{d}}(v_{\mathrm{hh,0}})$ respectively. The weighting matrices $Q(v_{\mathrm{hh,0}})$ and $R(v_{\mathrm{hh,0}})$ can be parameterized by the current wind speed.

This QP problem can be solved using QP solvers, NLP solvers or analytically. However, using QP solvers and the analytical method requires the problem to be re-built between each update of the MPC, which can be time consuming. Using NLP solvers, it is possible

to build a single problem parameterized by the current hub-height wind speed. Therefore, only the resolution of this NLP problem is required between each update of the MPC. In this thesis, the NLP solution implemented in `CasADi` was preferred to the two other possible implementations.

2.2.4 LPV MPC

The last MPC design for controlling a HAWT system presented in this chapter consists in directly using the LPV model derived from FAST linearizations (2.13) as an internal model. Therefore, the hub-height wind speed information over the prediction horizon, which can be obtained using LiDAR, is used to parameterize the evolution of the system dynamics in the open-loop optimal control problem :

$$\min_{\{u_0,\dots,u_N\}} \quad \begin{aligned} J_{\mathrm{d}} = &\sum_{l=1}^{N} \left(y_l - y_{\mathrm{ref}}(v_{\mathrm{hh,l}})\right)^T Q(v_{\mathrm{hh,l}}) \left(y_l - y_{\mathrm{ref}}(v_{\mathrm{hh,l}})\right) T_s + \\ &\sum_{l=0}^{N} \delta u_l(v_{\mathrm{hh,l}})^T R(v_{\mathrm{hh,l}}) \delta u_l(v_{\mathrm{hhl}}) T_s \end{aligned} \tag{2.21a}$$

$$\text{s.t.} \quad x_{l+1} = A_{\mathrm{z}}(v_{\mathrm{hh,l}})\delta x_l(v_{\mathrm{hh,l}}) + B_{\mathrm{z}}(v_{\mathrm{hh,l}})\delta u_l(v_{\mathrm{hh,l}}) + B_{\mathrm{d,z}}(v_{\mathrm{hh,l}})\delta d_l(v_{\mathrm{hh,l}}) + x_{op}(v_{\mathrm{hh,l}}) \quad \forall l \in \{0,\dots,N\} \tag{2.21b}$$

$$y_l = C(v_{\mathrm{hh,l}})\delta x_l(v_{\mathrm{hh,l}}) + D(v_{\mathrm{hh,l}})\delta u_l(v_{\mathrm{hh,l}}) + D_d(v_{\mathrm{hh,l}})\delta d_l(v_{\mathrm{hh,l}}) + y_{op}(v_{\mathrm{hh,l}}) \quad \forall l \in \{0,\dots,N\} \tag{2.21c}$$

$$x(t_0) = x_0 \tag{2.21d}$$

$$\theta_{\mathrm{col,l}} = \theta_{\mathrm{col,CPC}} \quad \forall l \in \{0,\dots,N\} \tag{2.21e}$$

where $v_{\mathrm{hh,l}}$ is the hub-height wind speed at time instant t_l. Note that the hub-height wind speed can also parameterize the cost function and the reference trajectory over the prediction horizon. Therefore (2.21) can be seen as an NLP or as a QP parameterized by the hub-height wind speed over the horizon $[v_{\mathrm{hh,0}},\dots,v_{\mathrm{hh,N}}]$. This remark is important, because it justifies that the optimization problem is convex and therefore the solver should converge on the global solution. This optimization problem will be solved using `CasADi` below.

2.2.5 Cost function parameterizations

The weighting matrices $Q \in \mathbb{R}^{\mathrm{n_y}}$ and $R \in \mathbb{R}^{\mathrm{n_u}}$ of the objective function J_{d}, depend respectively on $\frac{n_y(n_y+1)}{2}$ and $\frac{n_u(n_u+1)}{2}$ parameters. In order to have a closed-loop MPC which efficiently reduces the fatigue cost $\mathcal{J}$ defined in (2.3), these weighting matrices must be optimized in line with this latter objective. However, optimizing such a high number of parameters can be computationally expensive and highly time-consuming. Therefore, two parameterizations based on the system and fatigue cost function observations are proposed in this subsection, in order to reduce the number of parameters to optimize. Finally, a method for optimizing the reduced number of parameters is presented.

System and fatigue cost observations

Note that the HAWT blades considered in the simulation all have the same dynamics. Therefore, the pairs of outputs $(M_{\mathrm{yaw}}, M_{\mathrm{tilt}})$ and $(\theta_{\mathrm{yaw}}, \theta_{\mathrm{tilt}})$ have respectively the same dynamics and range of variations, due to the MBC transform defined in (1.1). Moreover, the fatigue cost function parameters defined in Table 2.1 are constant for each pair. Hence,

it is assumed that each pair should be weighted equally in the cost function J_{d}.

On the other hand, the IPC should avoid disturbing the CPC regulation. Therefore, the $\bar{M}$ and θ_{col} should be weighted to 0 in the cost function J_{d}. Moreover, due to the equality constraint (2.16e), the decision variable relative to the collective pitch angles is completely parameterized by $\theta_{\mathrm{col,CPC}}$, given by the CPC regulator. Hence, it is not an issue to weigh all the terms relative to the collective pitch angle in J_{d}.

Parameterized cost function 1

The weighting matrices Q and R are thus parameterized by $\nu \in \mathbb{R}^+$, in order to approximate the fatigue cost $\mathcal{J}$ to the cost function J_{d} defined in (2.16c) :

$$Q(\nu) = \mathrm{diag}([0, 1, 1, 0, \nu, \nu]) \tag{2.22a}$$
$$R(\nu) = \mathrm{diag}([0, 1, 1]) \min(\nu, 1) \times 10^{-10} \tag{2.22b}$$

It should be noted that the parameterization of R is designed such that the penalization on the decision variables in J_{d} is not significant, while avoiding singularity issues in the optimization by weighting it to zero.

Parameterized cost function 2

MPC regulators yield an output after every resolution of the open-loop optimization problem. Errors in system modelling, state estimation and disturbance measurements can result in deviation from the optimal solution. Therefore, high variations can appear on the MPC control input. These variations could excite vibrational modes of the turbine which are not modelled and impose an important additional amount of fatigue on the system, which must be avoided. The second parameterized cost function gives a solution to prevent this issue.

The solution consist in extending the output of the internal model with the derivative, defined in (2.11), of the MBC blade pitch angles $\dot{\tilde{u}}$ in order to penalize the latter in the cost function. The parameterized cost function defined in (2.22) thus becomes :

$$Q(\nu_1, \nu_2) = \mathrm{diag}([0, 1, 1, 0, \nu_1, \nu_1, 0, \nu_2, \nu_2]) \tag{2.23a}$$
$$R(\nu_1, \nu_2) = \mathrm{diag}([0, 1, 1]) \min(\nu_1, \nu_2, 1) \times 10^{-10} \tag{2.23b}$$

where ν_1 and ν_2 are the new parameters of the cost function, respectively penalizing the yawing and tilting blade pitch angle set-point and their derivatives.

This change in the cost function is expected to limit the variations in the MPCs set-points, and therefore have less abrupt changes. Moreover, it was proven in [81] that penalizing the state variations of a system contributes to the stability of MPC without terminal constraints.

Optimizing the weighting parameters

The ultimate objective of this MPC is to minimize the fatigue cost defined in (2.3) in closed loop with the aero-elastic HAWT simulator. A closed-loop simulation with the parameterized MPC is defined by its initial state, the wind time series applied to the system, denoted by $\mathbf{v}$, and the objective cost function parameters, denoted by $\tilde{\nu}$. Therefore $\tilde{\nu} = \nu$ for the first cost function (2.22) and $\tilde{\nu} = [\nu_1, \nu_2]$ for the second one (2.23).

Also, if relatively long simulations are considered, then the influence of the initial conditions can be ignored if those chosen are in the vicinity of the steady state. Moreover, what matters in this thesis is to alleviate system fatigue loads, which is equivalent to regulating the system loads around a set-point. There is thus little interest in considering the transient from one state to another in the fatigue estimation. Therefore, the initial conditions of every simulation are taken to be the system's steady state.

We can therefore consider the system closed-loop simulation fatigue cost as a function of $\mathbf{v}$ and $\tilde{\nu}$, denoted by $\mathcal{J}(\mathbf{v}, \tilde{\nu})$. Hence, the optimization problem that must be solved in order to optimally tune the MPC parameters is the following :

$$\min_{\tilde{\nu}} \quad \mathbb{E}\left(\mathcal{J}(\mathbf{v}, \tilde{\nu})\right) \tag{2.24a}$$

$$\text{s.t.} \quad \mathbf{v} \sim \mathcal{W} \tag{2.24b}$$

$$\tilde{\nu} > 0 \tag{2.24c}$$

where $\mathcal{W}$ is a relevant distribution of the wind parameters. This optimization problem is solved using the Nelder–Mead simplex search method described in [82], through the `Matlab` `fminsearch` function, for every linear MPC.

This optimization yields the vector of optimal parameters $\tilde{\nu}^\star$, which is indexed by the linearization parameter $\bar{v}_{\text{hh}}$ considered. Next, the weighting matrices Q and R, used in the optimizations of the gain-scheduled (2.20) and LPV (2.21) MPCs, are parameterized as follows :

$$Q(v_{\text{hh}}) = Q\left(\tilde{\nu}^\star(v_{\text{hh}})\right) \tag{2.25a}$$

$$R(v_{\text{hh}}) = R\left(\tilde{\nu}^\star(v_{\text{hh}})\right) \tag{2.25b}$$

The range of wind speeds considered in the following section means the system cannot be assumed to be linear. Therefore the linear MPCs are not implemented alone, as they cannot have a relevant behaviour for all the winds considered. The linear MPCs are thus only used in the MMPC design, which is considered to be a generalization of linear MPCs to nonlinear systems. A summary of the MPC design settings considered in this chapter is given in Table 2.2, with the corresponding labels used in the next section.

2.3 Results

Next, the MPCs presented above are implemented in closed loop with the HAWT simulator FAST to study the influence of the MPC prediction horizon and design settings on the closed-loop fatigue cost, under a realistic wind distribution. In order to draw a parallel with the issue described in Subsection 1.4.1, the vector of parameters p is composed of the design settings used, i.e. the MPC architecture, the parameterization of the stage cost and the MPC prediction horizon. The comparative study analyzed in this section will show that there is no ideal vector of parameters for fatigue cost reduction, but that its optimal value can vary significantly with respect to the wind conditions.

The outline of this section is the following :

- In Subsection 2.3.1, the various parameters used in the simulations are indicated.
- In Section 2.3.2, we check the CPU time needed for one step of each MPC, in order to give an idea of the relative computational complexity of the different MPC design settings.

	Cost function 1 : Penalty on the moments and blade pitch angles	**Cost function 2 :** Penalty on the moments, blade pitch angles and their first time derivative
Linear MPC : One single operating point, one single model	Not considered alone under the HAWT simulator	Not considered alone under the HAWT simulator
MMPC : Weighted sum of individual linear MPCs	$\text{MPC}_{\text{M},1}$	$\text{MPC}_{\text{M},2}$
Gain-scheduled MPC : Linear MPC parameterized by the current wind speed	$\text{MPC}_{\text{GS},1}$	$\text{MPC}_{\text{GS},2}$
LPV MPC : Time varying linear model over the prediction horizon	$\text{MPC}_{\text{LPV},1}$	$\text{MPC}_{\text{LPV},2}$

TABLE 2.2 – Summary of the MPC design settings defined in this chapter, and their respective labels used in the next section.

— In Subsection 2.3.3, the influence of the MPC prediction horizon and design settings on the fatigue cost is analyzed. Moreover, it is shown that adapting the parameters of the MPC to wind conditions can allow a significant reduction in the fatigue cost expectancy.

2.3.1 Simulation settings

In this section, the MPC design settings presented above are implemented for different prediction horizons in closed-loop with the HAWT simulator FAST. The parameters used for the simulations are summarized in Table 2.3.

Parameter	Value
MPC horizon steps N	$\{2, 6, 10, 20, 30\}$
MPC updating period	0.1 second
Simulation length	200 seconds
Simulation sampling time	0.0125 second

TABLE 2.3 – Summary of the parameters used for the closed-loop simulations.

Also, the HAWT only has its blades' first flapwise modes DOF activated, which means that the turbine rotor rotational speed does not need to be regulated with CPC. Indeed, the rotor is set to rotate at fixed rotational speed, as if it was driven by a motor and the collective blade pitch angle $\theta_{\text{col,CPC}}$ is varied as a static function of the wind.

The feedback state, which is needed in every MPC open-loop optimization, is directly given by the FAST output, which prevents the design of a state observer. The disturbance

predictions are free of errors and take into account the hub-height wind horizontal speed, direction, vertical speed and vertical shear, which are most of the features defining the wind in the following simulations.

The closed-loop MPCs are tested under turbulent hub-height winds generated by the TurbSim wind generator[73]. Wind features such as mean wind speed, turbulence intensity and power law exponent are randomly drawn from realistic distributions of an actual wind site, i.e. the NREL National Wind Technology Center (NWTC). The distributions are obtained from a three-year measurement campaign [83].

In the end, only the winds in region III, i.e. winds having a mean wind speed over 1 minute above 12 m/s, where blade pitch control is supposed to be activated, are kept for simulation. This makes 70633 samples available in the dataset, whose mean wind speed and turbulence intensity estimated over 1-minute histograms are plotted in Figure 2.5 and 2.6 respectively. Nevertheless, it should be noted that IPC could also be of interest in region II for fatigue reduction, but discussion of this matter is outside the scope of this study.

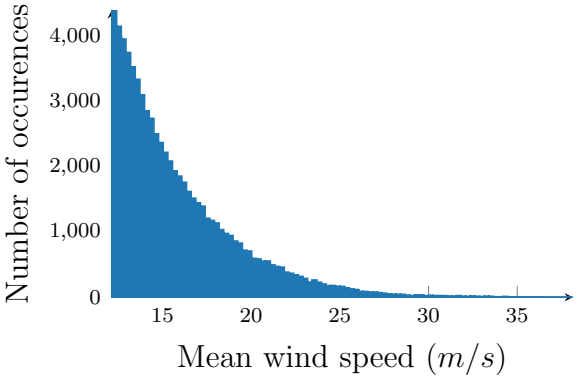

FIGURE 2.5 – Histogram of the wind speed averaged over 1 minute for 70633 samples.

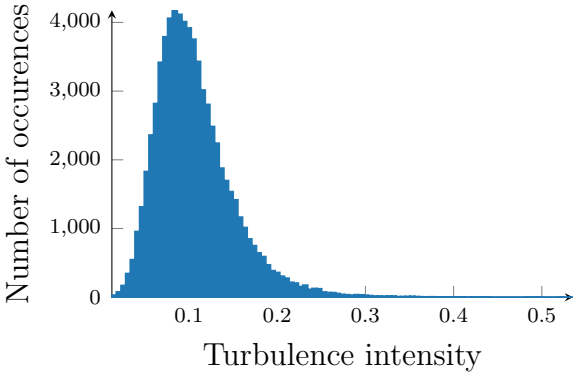

FIGURE 2.6 – Histogram of the turbulence intensity estimated over 1 minute for 70633 samples.

The parameterized cost functions (2.24) are optimized using 10 TurbSim-generated winds with randomly-drawn mean wind speed and turbulence intensity in the distribution corresponding to the histograms presented in Figures 2.5 and 2.6. In order to obtain statistics on the closed-loop fatigue cost, 96 other winds, whose parameters are randomly drawn from the same distributions, are used as disturbance in the HAWT simulator.

2.3.2 A glance at the CPU times

One drawback of MPC is the computational cost necessary to solve its optimization problem, which can be prohibitively high if the updating period is too short to make the computations. The CPU time needed for one optimization is thus a relevant indicator of the MPC's complexity. The CPU times given below are obtained for optimizations performed under `Matlab`, with a standard laptop using a 16 GB RAM and 2.60 GHz processor.

The average CPU times needed for the presented MPCs with various prediction horizons are summarized in Table 2.4. With this laptop configuration, which is not the same as a HAWT controller computer, the following observations can be made :

— They are real-time feasibility issues for $N \geq 6$ with the gain-scheduled and LPV MPCs.

— The CPU times needed for the gain-scheduled and LPV MPCs are two to three orders of magnitude greater than for the MMPCs.

It should be noted that on a HAWT computer unit, the computational capacities might be reduced, but the controller would be programmed under a lower-level language such as C or C++, which could speed up the computations. Therefore, the transition to a HAWT computer unit could play both ways : it could increase or reduce the computational time.

Moreover, the computational complexities of the presented MPC design settings are not equivalent, and this is another important aspect which must be taken into account for real-time implementation. However, there are solutions to alleviate the computational workload of these MPCs, e.g. parameterizing the MPC control sequence, distributing its open-loop optimization over the horizon, or learning its behaviour from data.

N	2	6	10	20	30
MPC_M	2.6×10^{-4}	3.4×10^{-4}	3.6×10^{-4}	3.5×10^{-4}	3.5×10^{-4}
MPC_GS	8.7×10^{-2}	1.6×10^{-1}	2.5×10^{-1}	4.4×10^{-1}	6.3×10^{-1}
MPC_LPV	8.7×10^{-2}	1.6×10^{-1}	2.5×10^{-1}	4.4×10^{-1}	6.3×10^{-1}

TABLE 2.4 – Summary of the average computational time in seconds needed for every MPC, depending on the number of prediction steps considered.

2.3.3 Fatigue cost analysis

Every closed-loop simulation is analyzed with the fatigue cost function $\mathcal{J}$. Therefore, every vector of parameters p of the MPCs, defined by the MPC design settings and the prediction horizon, have 96 fatigue cost values. Depending on the objective, it may be interesting to tune the vector of parameters p in order to :

— Minimize the **expectancy** of $\mathcal{J}$, which is in this case the average of the $\mathcal{J}$ corresponding to a given p, because the winds are generated such that they all have equal probability.

— Minimize the **median** of $\mathcal{J}$, because of the Rayleigh distribution of $\mathcal{J}$, the median might give less importance to rare but highly-damaging cases.

— Maximize the probability that a vector of parameters p gives lower fatigue cost value $\mathcal{J}$ than other parameter vectors.

It should be noted that a universal controller which efficiently minimizes the fatigue cost should be valid for all the above criteria. Moreover, for the design of an IPC aimed at minimizing the fatigue cost $\mathcal{J}$ expectancy, the first criterion must be met as a priority.

In the remainder of this subsection, the controllers simulated in closed loop will be analyzed in terms of the above criteria, while the influence of prediction horizons on fatigue cost for the various design settings is also examined.

Fatigue cost expectancy

In Figure 2.7, the evolution of the fatigue cost expectancy $\mathcal{J}$ with the prediction horizon T is plotted for the six MPCs. The main observations are summarized as follows :

— For the shortest prediction horizon, MPCs using the first parameterized cost function (2.22) have lower fatigue cost averages than the one using the second cost function (2.23).

— For prediction horizons above 2 seconds, all the MPCs using the second parameterized cost function yield lower fatigue cost averages than the one using the first cost function (2.23).
— The MPCs using the second parameterized cost function (2.23) all have a globally decreasing average fatigue cost with the prediction horizon.
— The MPCs using the first parameterized cost function (2.22) have an increasing or almost constant fatigue cost.
— The MPCs using the second parameterized cost function (2.23) yield generally lower fatigue cost than the ones using the first cost function (2.22).
— The controller to be selected for fatigue cost expectancy minimization is obviously $\text{MPC}_{\text{M,2}}$, for a time horizon of 2 seconds.

To summarize these observations, there is no clear trend concerning the influence of MPC design settings or prediction horizon on the fatigue cost average. However, it can be seen that the MPC using the second parameterized cost function (2.23) generally behaves better than the one using the first cost function (2.22) for prediction horizons longer than 2 seconds.

Nevertheless, the fatigue cost average might be dominated by the simulations yielding the highest fatigue costs, because of the fatigue cost Rayleigh distribution. Therefore, it indicates how an MPC behaves in terms of fatigue cost in the highly-damaging cases. Hence, analyzing the evolution of the fatigue cost median might allow us to understand how the MPCs behave in most cases.

Fatigue cost median

Figure 2.8 plots the evolution of the $\mathcal{J}$ median with the prediction horizon, for the six MPCs. The similarities which can be found with the average case are the following :

— The MPCs using the first parameterized cost function (2.22) are still performing better than the one using the second cost function (2.23) for short prediction horizons.
— The $\text{MPC}_{\text{M,1}}$ shows good performance for the shortest prediction horizon.

On the other hand, the differences from the average case are :

— The fatigue cost median of all the MPCs excepting $\text{MPC}_{\text{M,1}}$ is globally decreasing with the prediction horizon.
— The fatigue cost median of the gain-scheduled and LPV MPCs are monotonically decreasing with the prediction horizon.
— The controller to be selected for fatigue cost expectancy minimization is $\text{MPC}_{\text{GS,1}}$ or $\text{MPC}_{\text{LPV,1}}$, for a time horizon greater or equal to 2 seconds.
— The controllers minimizing the fatigue cost median have fatigue cost expectancy several orders of magnitudes greater than the one minimizing the fatigue cost expectancy, and vice versa.

To summarize these observations, for the MPC design settings and prediction horizon, trends are clearer in the average than in the median. It can be assumed that **fatigue cost average is dominated by the simulations yielding the highest fatigue costs**. If this hypothesis is confirmed, it would suggest that some *MPCs prediction horizons and design settings are better suited to reducing fatigue cost for the highly-damaging cases*. It would also suggest that some *MPCs prediction horizons and design settings are better suited to reducing fatigue cost for cases causing medium damage* , which represent most cases.

In order to evaluate the veracity of this hypothesis, it is proposed to analyze the number of times where an MPC minimizes the fatigue cost for a given vector of parameters p.

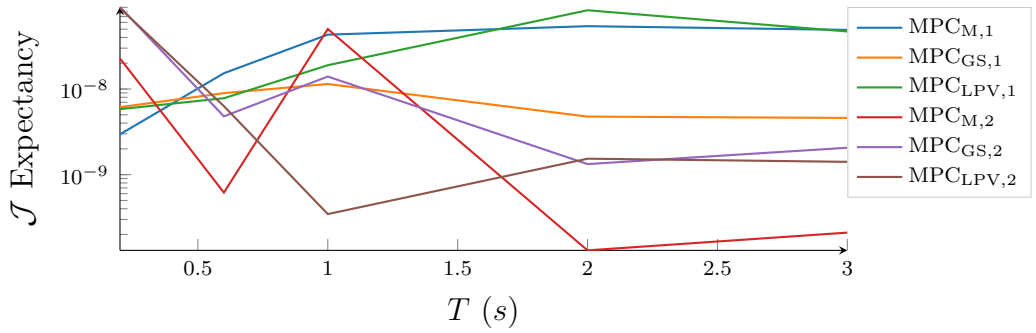

FIGURE 2.7 – Evolution of the MPC design settings fatigue cost expectancy with the prediction horizon. There is no clear trend concerning the influence of MPC design settings or prediction horizons on the fatigue cost average.

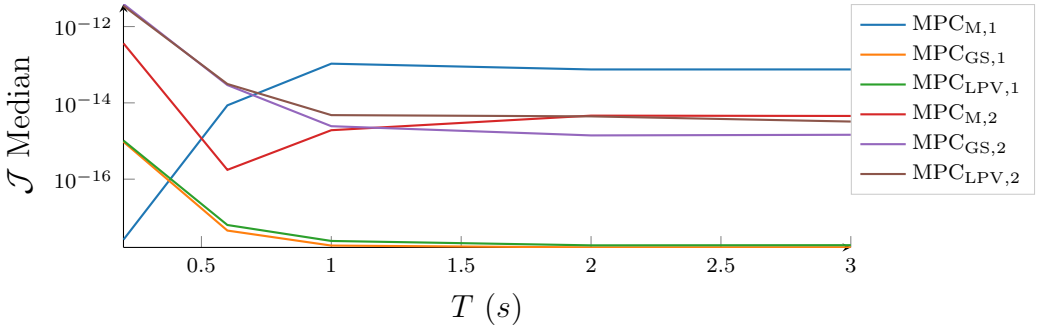

FIGURE 2.8 – Evolution of the MPCs fatigue cost median with the prediction horizon. Trends are emerging for both the prediction horizon and the MPC design settings.

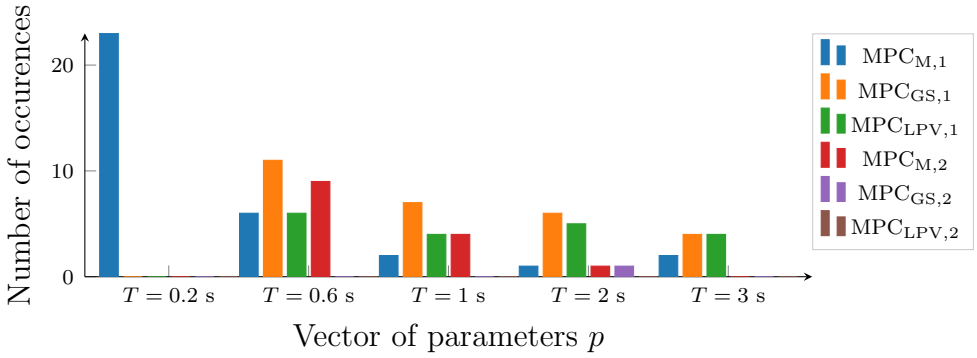

FIGURE 2.9 – Histogram of the number of times a vector of parameters p minimizes $\mathcal{J}$ in closed-loop simulations, under 96 winds.

Probability of minimizing the fatigue cost

Figure 2.9 plots the histogram of the vector of parameters which minimizes the fatigue cost $\mathcal{J}$, under the 96 generated winds, where the vector of parameters is defined by a design setting and a prediction horizon. Thus :

- Only 17 parameter vectors out of 30 are minimizing the fatigue cost of at least one simulation.
- The controller which minimizes the fatigue cost most often is $\text{MPC}_{\text{M},1}$ for a prediction horizon of 0.2 seconds, which minimizes the fatigue cost in almost 25% of cases.
- The controller minimizing the fatigue cost expectancy minimizes the fatigue cost in only one case, which might be the most damaging case.
- The controller minimizing the fatigue cost median minimizes the fatigue cost in about 6% of the cases.

Summary

To summarize these observations, the hypothesis that fatigue cost average is dominated by the simulations yielding the highest fatigue costs is confirmed, as the controller minimizing the fatigue cost expectancy minimizes the fatigue cost in only one case.

Therefore, on this realistic wind distribution, if a parameter vector must be tuned in order to minimize the fatigue cost expectancy, this is equivalent to tuning it in the most damaging simulation. Moreover, controllers which minimize the fatigue cost in a large amount of cases have very poor performance on fatigue cost expectancy, because they might have poor performance in the high-damage simulations.

This suggests that adapting the MPCs parameter vector, such that for every wind trajectory $\mathbf{v}$ the vector of parameters is selected as $p^\star(\mathbf{v})$, defined by (1.52), could indeed allow us to further reduce the fatigue cost expectancy. **In these** 96 **closed-loop simulations, the fatigue cost expectancy could have been reduced by 23% using** $p^\star(\mathbf{v})$ instead of $\bar{p}^\star$, defined by (1.51), i.e. the fixed parameter vector which minimizes the fatigue cost expectancy.

Moreover, in regard to the sensitivity of the various MPC design settings to the prediction horizon, these examples show that **the best MPC design setting** for fatigue cost reduction **depends on the MPC prediction horizon**. Conversely, **the best prediction horizon** for fatigue cost reduction **depends on the MPC design setting**. Hence, it is difficult to generalize on the optimal prediction horizon for fatigue cost reduction in our search for a universal optimal value.

2.4 Discussion

In this chapter, several MPC design settings and parameterizations of the MPC cost function were presented. Then, a study of the influence of the MPC prediction horizon and design settings on the closed-loop fatigue cost was performed. The principal results obtained were the following :

- **Depending on the MPC design setting used, the prediction horizon does not have the same influence on the closed-loop fatigue cost** and the optimal prediction horizon can vary significantly. For example, the MPCs which are using more accurate internal models, such as gain-scheduled or LPV MPCs, can efficiently schedule the disturbance rejection over a long horizon.
- The **weighting in the cost function changes also the influence of the prediction horizon on the resulting fatigue cost**. For instance, the second parameterized cost function (2.23), where the yawing and tilting blade pitch angles derivatives are penalized, allowed the MMPC fatigue cost to decrease with the prediction horizon. On the other hand, the fatigue cost of the MMPC using the first parameterized cost function (2.22) increased with the prediction horizon.
- **There is no universal MPC prediction horizon and design setting for fatigue cost reduction**, using classical MPC formulations. The appropriate MPC prediction horizon and design setting also depends on the wind conditions. For example, for long prediction horizons, MPCs using the first cost function (2.22) yield lower fatigue cost than those using the second cost function (2.23) in most cases. However, the MPCs using the second cost function for long prediction horizons have the lowest fatigue cost on average, because the MPCs using the second cost function are the most efficient in wind conditions determining the highest

fatigue costs.

— In the example presented, **adapting the parameter vector p which parameterizes the MPC can reduce the fatigue cost expectancy by** 23%, compared to an MPC with a fixed parameter tuned to minimizing the fatigue cost expectancy.

To conclude this chapter, **there is no ideal standard MPC design setting and prediction horizon for a realistic wind distribution**. These findings and the observations made in this chapter suggest that the best MPC design setting and prediction horizon in terms of fatigue cost depend on the exogenous disturbances acting on the system.

Thus, the next chapters address the prospects for fatigue-oriented controllers aimed at efficiently adapting the parameter vector p for fatigue cost reduction. The solutions presented are MPCs whose stage cost weighting matrices vary regarding a data-driven cost function and wind predictions, along with a framework selecting a controller from several candidates, based on data. All these solutions aim at finding a controller which reduces the fatigue cost expectancy more than an individual controller with fixed parameters optimized for fatigue cost expectancy reduction.

Chapitre 3

Data-driven fatigue-oriented cost function

Chapter 2 concluded that a standard MPC formulation, whose design setting and prediction horizon are components of the array of parameters p defining the MPC, cannot reduce the fatigue cost efficiently for a wide range of disturbances with a fixed array of parameter p. The discussion in this latter chapter also suggested that in order to significantly reduce the fatigue cost expectancy, the array of parameters should be varied with the wind and turbine conditions. This is the reason why several works have aimed at designing fatigue-oriented MPCs [62, 68], whose main attention was focused on the cost function design.

Indeed, it was also shown through equations (1.37) to (1.41) that quadratic forms, which are commonly used in standard MPC formulations, are not suitable for modelling the fatigue cost $\mathcal{J}$ which should be minimized. Therefore it is not surprising that standard MPC formulations fail to consistently reduce fatigue cost on a broad spectrum of wind disturbances.

However, the quadratic forms used as cost functions in standard MPC formulations, when representative of the cost to be minimized, are convenient since they are convex, smooth and their hessian is constant. This allows us to have time-efficient optimizations, ensuring convergence and global optimum, provided that the dynamic system is linear and the inequality constraint forms a convex set.

Therefore, this chapter presents a framework that consists in deriving a data-driven fatigue-oriented cost function based on quadratic features, denoted by $\hat{\mathcal{J}}$, that approximates the fatigue cost $\mathcal{J}$ defined by (1.50). Moreover, $\hat{\mathcal{J}}$ can be efficiently used as an objective function in an optimal control problem. It is eventually shown that this approach consists ultimately in adapting the parameters of a standard optimal control problem using a quadratic cost function, based on the wind information over the optimization horizon.

The outline of this chapter is the following :

— In Section 3.1, a methodology to derive a data-driven fatigue-oriented cost function, approximating the fatigue cost $\mathcal{J}$, defined by (1.50) and based on quadratic features, is detailed.
— In Section 3.2, a method to optimize efficiently an optimal control problem using the fatigue-oriented cost function based on quadratic features is presented. This method involves a fixed-point problem which is solved by resolving successive optimal control problems using a quadratic cost function.
— In Section 3.3, this fatigue-oriented cost function is used in an open-loop optimal control problem and compared to one using parameterized quadratic cost function such as the one defined by (2.2.5). The open-loop optimization uses an LTI system and is performed under a realistic ensemble of wind conditions. The open-loop optimizations using the fatigue-oriented cost function show great potential for fatigue cost expectancy reduction.
— In Section 3.4, a discussion is proposed on the possible shortcomings, extension and perspectives of this framework.

3.1 Data-driven fatigue-oriented cost function derivation

In this section a methodology aimed at deriving a data-driven fatigue-oriented cost function approximating the fatigue cost $\mathcal{J}$, defined by (1.50) and based on quadratic features, is presented. This methodology is divided into the following steps and schematized in Figure 3.1 :

1. Generation of N_d time series, denoted by $\{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(N_d)}\}$, representing a wide variety of the possible outputs of a closed-loop process (Subsection 3.1.1)
2. Define a basis of quadratic forms to be used as features in a regression of fatigue damage (Subsection 3.1.2). The basis of quadratic forms is denoted by $\mathbf{J}(\mathbf{y}_k^{(i)})$ for the k^{th} output of the i^{th} time series.
3. Fit parametric regressions of fatigue damage from the previously defined basis of quadratic forms, allowing us to predict the fatigue damage $\hat{\mathcal{D}}_k$ of the k^{th} component (Subsection 3.1.3)
4. From the regressions, derive a fatigue-oriented cost function, denoted $\hat{\mathcal{J}}$, such that it approximates the fatigue cost $\mathcal{J}$ (Subsection 3.1.4)

Subsection 3.1.5 discusses how the fatigue-oriented cost function can be related to the spectral approach of fatigue, presented in page 28 of Subsection 1.3.3.

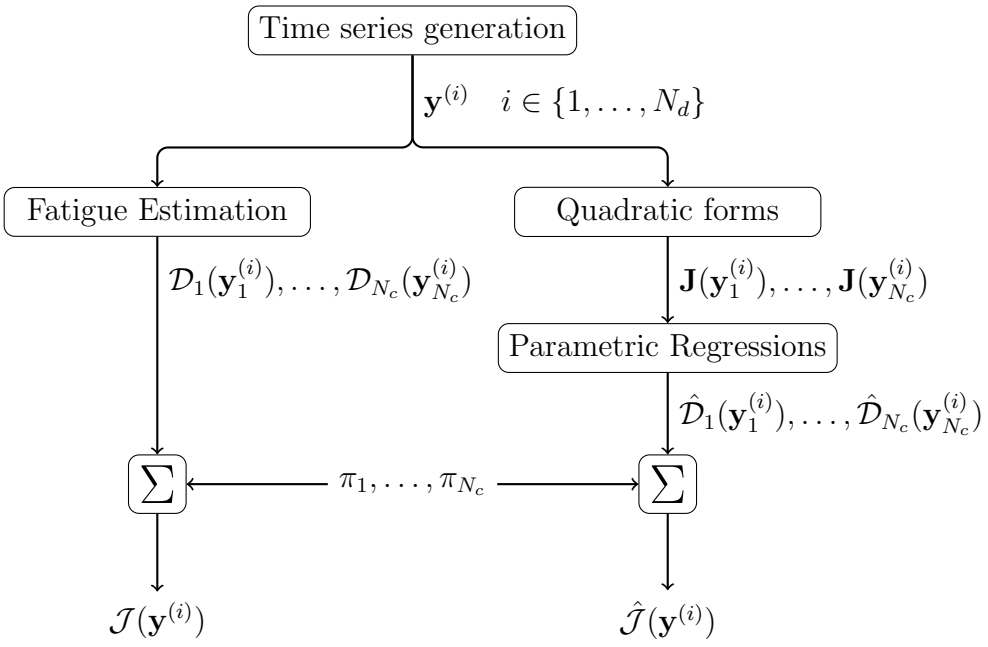

FIGURE 3.1 – Schematization of the methodology aiming at deriving a data-driven fatigue-oriented cost function $\hat{\mathcal{J}}$, approximating the fatigue cost $\mathcal{J}$.

3.1.1 Data generation

The first step of the fatigue-oriented cost function derivation consists in generating a set of time series, representative of the variety of outputs which a closed-loop process can yield. Therefore, a simulator allowing us to model the dynamics of the system to be controlled in closed loop is needed.

However, it is difficult to guess in advance what the dynamics of the closed-loop system will be, as the optimal control problem solutions depend on the data generation. Hence,

simulating a variety of processes allowing us to cover a sufficiently wide range of possible dynamics for the closed-loop system may be preferable.

Each time series generated will be post-processed and yield an individual sample of fatigue damage and quadratic cost function features, which will be needed for the fatigue damage regression. Therefore, the simulator must generate a sufficient number of time series, denoted N_d, such that the relation between fatigue damage and the quadratic cost function features is statistically representative and allows an accurate regression. In the following, $\mathbf{y}_k^{(i)}$ denotes the trajectory of the k^{th} process output of the i^{th} time series.

3.1.2 Basis of linear and quadratic cost functions

Once all the time series are generated, the fatigue damage of each output with its corresponding ultimate load and Wöhler exponent are estimated for each time series, in order to constitute a set of target values for the fatigue damage regression. For the k^{th} output of the i^{th} time series, the damage is $\mathcal{D}_k(\mathbf{y}_k^{(i)})$, defined by (1.28).

A set of linear and quadratic forms, which are to be used as cost functions in an optimal control problem, is also defined. These linear and quadratic forms will also constitute a set of features for the fatigue damage regression. The set of features can be composed of the following elementary linear and quadratic cost functions, defined for a signal $\mathbf{y}_k^{(i)}$:

— The integral of the difference between the signal and a constant $\varepsilon \in \mathbb{R}$, denoted $J_{\text{L},\varepsilon}(\mathbf{y}_k^{(i)})$

— The variance of the signal, denoted $J_{\text{Var}}(\mathbf{y}_k^{(i)})$

— The mean square of the p^{th} time derivative of the signal, denoted $J_{\text{MS,p}}(\mathbf{y}_k^{(i)})$, for $p \in [\![0, +\infty[\![$.

whose mathematical expressions are :

$$J_{\text{L},\varepsilon}(\mathbf{y}_k^{(i)}) = \int_{t_0}^{t_0+T} (y_k^{(i)}(t) - \varepsilon)dt \tag{3.1a}$$

$$J_{\text{Var}}(\mathbf{y}_k^{(i)}) = \frac{1}{T}\int_{t_0}^{t_0+T} y_k^{(i)}(t)^2 dt - \left(\frac{1}{T}\int_{t_0}^{t_0+T} y_k^{(i)}(t)dt\right)^2 \tag{3.1b}$$

$$J_{\text{MS,p}}(\mathbf{y}_k^{(i)}) = \frac{1}{T}\int_{t_0}^{t_0+T} \left(\frac{d^p y_k^{(i)}(t)}{dt^p}\right)^2 dt \tag{3.1c}$$

where $y_k^{(i)}(t)$ is the value of the trajectory $\mathbf{y}_k^{(i)}$ at time instant t. t_0 and $t_0 + T$ are respectively the first and last time instants in the time series.

All the features considered are stacked in the column vector

$$\mathbf{J}(\mathbf{y}_k^{(i)}) = [J_1(\mathbf{y}_k^{(i)}), \ldots, J_{N_{\text{ft}}}(\mathbf{y}_k^{(i)})]^T$$

where J_j is the j^{th} feature and N_{ft} is the total number of features considered. It should be noted that as many features as needed can be considered.

However, the more features are considered, the more complex the resulting optimal control problem will be. Moreover, as modelling and measurements errors can generate noise on $\mathbf{y}_k^{(i)}$, this noise would be amplified by the time derivatives. Hence, if on the one hand adding time derivatives as features might increase the accuracy of a fatigue damage regression, on the other hand, and in practice, the potentially important error in

their estimation could affect the solution of the resulting optimal control problem. The conclusion is that one should keep the 'simpler is better' rule in limiting the number of features and the use of time derivatives, provided that the method yields acceptable results. Although it is possible to filter a part of the noise contained in the outputs' time derivatives.

3.1.3 Parametric regression of fatigue damage

In order to approximate the fatigue cost $\mathcal{J}$ with a cost function based on linear and quadratic cost functions, it is proposed to first approximate fatigue damage from the previously generated features, and this for each output of the process.

Therefore, a parametric and differentiable regression function $\mathcal{F}_k$ is fitted, for the k^{th} component, on the dataset of fatigue damages $\mathcal{Y}_k = \{\mathcal{D}_k(\mathbf{y}_k^{(1)}), \dots, \mathcal{D}_k(\mathbf{y}_k^{(N_d)})\}$ from the dataset of column vectors of features $\mathcal{X}_k = \{\mathbf{J}(\mathbf{y}_k^{(1)}), \dots, \mathbf{J}(\mathbf{y}_k^{(N_d)})\}$. Fatigue damage of the k^{th} output can be approximated with $\hat{\mathcal{D}}_k$ using the fitted parametric function $\mathcal{F}_k$:

$$\hat{\mathcal{D}}_k(\mathbf{y}_k^{(i)}) = \mathcal{F}_k(\mathbf{J}(\mathbf{y}_k^{(i)})) \tag{3.2}$$

where $\mathbf{y}_k^{(i)}$ is the k^{th} output of the i^{th} time series or trajectory.

It should be noticed that $\mathcal{F}_k$ could be any regression function, provided that it is parametric, such as :

— Linear regression
— Ridge regression
— Lasso regression
— Elasticnet regression
— Feedforward neural network with differentiable activation function
— Support vector machine
— Decision trees, random forests, gradient boosting decision trees

These regressions can possibly be combined with linear transformations on the features or target values, which leaves a larger latitude to the designer. However, once again, one should keep the 'simpler is better' principle so that simple structures should be preferable provided that they enable correct prediction performances.

3.1.4 Fatigue-oriented cost function derivation

The approximation of fatigue damage using parametric regression functions must be performed for every considered component in the fatigue cost $\mathcal{J}$ expression, defined by (1.50). Therefore, a set of parametric regression functions is obtained $\{\mathcal{F}_1, \dots, \mathcal{F}_{N_c}\}$, where N_c is the number of components considered in the fatigue cost estimation. The set of parametric regression functions allows us to approximate the fatigue damage of every component using equation (3.2), given $\mathbf{y}^{(i)}$.

It is thus possible to approximate the fatigue cost $\mathcal{J}$, with the fatigue-oriented cost function, denoted $\hat{\mathcal{J}}$, expressed as follows :

$$\hat{\mathcal{J}}(\mathbf{y}^{(i)}) = \sum_{k=1}^{N_c} \pi_k \underbrace{\mathcal{F}_k(\mathbf{J}(\mathbf{y}_k^{(i)}))}_{\hat{\mathcal{D}}_k(\mathbf{y}_k^{(i)})} \tag{3.3}$$

where $\mathbf{y}$ is the vector of outputs of a given time series or trajectories.

3.1.5 Relation with the spectral approach

In subsection 1.3.3, a spectral approach to fatigue is briefly presented, relating damage rate to the spectral moments of a narrow-banded process in equation (1.43). The relationship is a posynomial [1] of the zero, second and fourth spectral moments of the considered process, which depend on the Wöhler coefficient, ultimate load and fixed-mean load. From probability theory, it can be shown that for an infinite time length signal of a random process :

- The zero order spectral moment is equivalent to J_{Var}, defined by (3.1b).
- The second and fourth order spectral moments are equivalent to $J_{\mathrm{MS},1}$ and $J_{\mathrm{MS},2}$ respectively, defined by (3.1c).
- The fixed-mean load is equivalent to J_{L}, defined by (3.1a).

From these properties, it can be concluded that :

- There is a relation between the features defined in Subsection 3.1.2 and fatigue damage.
- There exists an explicit expression of the relation between the basis of quadratic forms defined in Subsection 3.1.2 and fatigue damage, for narrow-banded processes.
- If the process is **narrow-banded**, the functions $\mathcal{F}_k$ could be directly estimated, without data generation or regression, as the expression is explicit.

However, if the process is **not narrow-banded**, a correction coefficient, based on a posynomial expression of the spectral moments and observations on the bandwidth, must be put in factor of equation (1.43). Hence, the study of one process bandwidth can still impose an empirical approach, which justifies the data-driven methodology depicted in this chapter, intended to be the more general. Nevertheless, the spectral approach yields important information for the derivation of the fatigue-oriented cost function, the process being narrow-banded or not :

- There is a relation between the features defined in Subsection 3.1.2 and fatigue damage, which justifies the choice of the features.
- This relation is based on posynomials. This suggests to consider logarithmic transformations on the features and fatigue damages in order to obtain a linear relation.
- The relation between the features and fatigue damage is nonlinear. Therefore the fatigue-oriented cost function is likely to be non-quadratic.
- The fatigue-oriented cost function being non-quadratic, its quadratic approximation in the fixed-point problem is suspected to vary.
- If the cost function of the optimal control problem varies with the turbine and wind conditions, the placement of the closed-loop system poles will vary accordingly.
- Therefore, the closed-loop process might not be narrow-banded, even if the open-loop process is narrow-banded, which justifies once more the use of data-driven fatigue-oriented cost function.

Once the fatigue-oriented cost function is derived, it can be used in an optimal control problem as an objective function. Several examples of derivation are given in Appendix D. However, as the fatigue-oriented cost function will not be quadratic, a specific method allowing to obtain intermediate optimal control problems using quadratic forms, in order to solve a fixed-point problem, is depicted in the next section.

1. Product of variables power real constants, it becomes an hyperplane in logarithmic space.

3.2 Optimization of the fatigue-oriented cost function

The ultimate objective being to efficiently solve an optimal control problem whose cost function is the fatigue cost $\mathcal{J}$ defined by (1.50). The reasons for deriving a data-driven fatigue-oriented cost function based on quadratic features are :

- Approximating the fatigue cost $\mathcal{J}$, defined by (1.50), with a cost function $\hat{\mathcal{J}}$ allows us to easily handle an approximation of $\mathcal{J}$ in an optimal control problem
- Allowing us to easily and efficiently turn a non-quadratic optimal control problem into a sequence of standard problems, thanks to the basis of linear and quadratic forms defined in Subsection 3.1.2
- This resolution is based on the methods consisting in solving a general optimization problem via successive QP, which is a very standard approach and widely used for the resolution of NLP.

The derivation of the fatigue-oriented cost function $\hat{\mathcal{J}}$ allows us to approximate an original optimal control problem using the fatigue cost $\mathcal{J}$ as objective function, with the data-driven fatigue-oriented cost function $\hat{\mathcal{J}}$. However, the resulting Fatigue-Oriented Optimal Control Problem (FO-OCP) is not a standard QP, as the cost function $\hat{\mathcal{J}}$ might not be quadratic :

$$\min_{\mathbf{u}} \quad \hat{\mathcal{J}}(\mathbf{y}) = \sum_{k=1}^{N_c} \pi_k \mathcal{F}_k(\mathbf{J}(\mathbf{y}_k)) \tag{3.4a}$$

$$\text{s.t.} \quad \dot{x} = f(x,u,v,t) \tag{3.4b}$$

$$x(t_0) = x_0 \tag{3.4c}$$

$$0 = h(x,u,v,t) \tag{3.4d}$$

$$0 \geq g(x,u,v,t) \tag{3.4e}$$

$$y = h_{\mathrm{y}}(x,u,v,t) \tag{3.4f}$$

where the functions f, g and h are defined in (2.1), y is the instantaneous output of the process and h_{y} is a function that maps the output from the current state x, input u, disturbance v and time instant t.

The outline of this section is the following :

- In Subsection 3.2.1, a Taylor expansion of the data-driven fatigue-oriented cost function $\hat{\mathcal{J}}$ is presented. This shows that $\hat{\mathcal{J}}$ can be approximated with a quadratic form whose weighting matrices depend on an output trajectory.
- In Subsection 3.2.2, a fixed point-problem aiming at finding the right output trajectory is detailed.
- In Subsection 3.2.3, it is shown that solving the previously-presented fixed-point problem is equivalent to solving the FO-OCP defined by (3.4). Moreover, the convergence of the fixed-point problem is discussed.
- In Subsection 3.2.4, the consequences of this methodology on the optimal control problem using quadratic cost functions are discussed. Moreover, the similarities and differences between the other fatigue-oriented optimizations for the optimal control problem and the methodology presented in this chapter are also discussed.

3.2.1 Approximation of the optimal control problem with quadratic forms

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 of the linear and quadratic forms used for the derivation of the data-driven fatigue-oriented cost function.

Indeed, the main motivation for the use of these quadratic forms as features is that they are known to behave very well in optimal control problems. Therefore, let us approximate $\hat{\mathcal{J}}$ with a first-order Taylor expansion of $\mathcal{F}_k$, around the trajectory $\mathbf{y}^{(N_{\text{iter}})}$, where N_{iter} is the iteration number in the resolution of a fixed-point problem resolution :

$$\hat{\mathcal{J}}^{(N_{\text{iter}})}(\mathbf{y})) = \sum_{k=1}^{N_c} \pi_k \left(\underbrace{\mathcal{F}_k\left(\mathbf{J}\left(\mathbf{y}_k^{(N_{\text{iter}})}\right)\right)}_{\text{constant}} + \sum_{j=1}^{N_{\text{ft}}} \underbrace{\frac{\partial \mathcal{F}_k}{\partial J_j(\mathbf{y}_k)}\left(\mathbf{J}\left(\mathbf{y}_k^{(N_{\text{iter}})}\right)\right)}_{\alpha_{k,j}\left(\mathbf{y}_k^{(N_{\text{iter}})}\right)\in\mathbb{R}} \left(J_j(\mathbf{y}_k) - \underbrace{J_j\left(\mathbf{y}_k^{(N_{\text{iter}})}\right)}_{\text{constant}} \right) \right) + \underbrace{\varepsilon\left(\mathbf{y}^{(N_{\text{iter}})}\right)}_{\text{negligible}} \quad (3.5)$$

Hence, $\hat{\mathcal{J}}$ can be approximated by a linear combination of quadratic forms $J_j(\mathbf{y}_k)$, for $k \in \{1, \dots, N_c\}$ and $j \in \{1, \dots, N_{\text{ft}}\}$. The original FO-OCP using $\mathcal{J}$ as a cost function is thus approximated by a quadratic form whose weighting matrices depend on $\mathbf{y}_k^{(N_{\text{iter}})}$:

$$\begin{aligned}
\min_{\mathbf{u}} \quad & \hat{\mathcal{J}}^{(N_{\text{iter}})}(\mathbf{y}) = J_{\text{OCP}}\left(p\left(\mathbf{y}^{(N_{\text{iter}})}\right), \mathbf{y}\right) = \int_{t_0}^{t_0+T} \Big(y(\tau)^T Q\left(\mathbf{y}^{(N_{\text{iter}})}\right) y(\tau) + \\
& \qquad L\left(\mathbf{y}^{(N_{\text{iter}})}\right)^T y(\tau) + \\
& \qquad u(\tau)^T R\left(\mathbf{y}^{(N_{\text{iter}})}\right) u(\tau) \Big)\, d\tau && \text{(3.6a)} \\
\text{s.t.} \quad & \dot{x} = f(x, u, v, t) && \text{(3.6b)} \\
& x(t_0) = x_0 && \text{(3.6c)} \\
& 0 = h(x, u, v, t) && \text{(3.6d)} \\
& 0 \geq g(x, u, v, t) && \text{(3.6e)} \\
& y = h_{\text{y}}(x, u, v, t) && \text{(3.6f)}
\end{aligned}$$

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

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

is an array of parameters of the optimal control problem.

The matrices Q and R, along with the vector L, are derived from the equations (3.1), (3.2) and (3.5). Their expression in terms of $\mathbf{y}^{(N_{\text{iter}})}$ depends on the quadratic forms and regression functions used. Let us assume that the j^{th} quadratic feature of $\mathbf{J}$ can be expressed :

$$J_j(\mathbf{y}_k) = \int_{t_0}^{t_0+T} \left(q_j y_k^2(t) + l_j y_k(t) \right) dt \quad (3.7)$$

where $q_j \geq 0$ and l_j are respectively quadratic and linear weighting coefficients. The matrices Q, R and L can thus be expressed as follows :

$$Q\left(\mathbf{y}^{(N_{\text{iter}})}\right) = \begin{pmatrix} \pi_1 \sum_{j=1}^{N_{\text{ft}}} \alpha_{1,j}(\mathbf{y}_1^{(N_{\text{iter}})}) q_j & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \pi_{N_c} \sum_{j=1}^{N_{\text{ft}}} \alpha_{N_c,j}(\mathbf{y}_{N_c}^{(N_{\text{iter}})}) q_j \end{pmatrix} \tag{3.8a}$$

$$R\left(\mathbf{y}^{(N_{\text{iter}})}\right) = \min\left(\pi_1 \sum_{j=1}^{N_{\text{ft}}} \alpha_{1,j}(\mathbf{y}_1^{(N_{\text{iter}})}) q_j \quad \dots \quad \pi_{N_c} \sum_{j=1}^{N_{\text{ft}}} \alpha_{N_c,j}(\mathbf{y}_{N_c}^{(N_{\text{iter}})}) q_j\right) \times \varepsilon I_{n_{\text{u}}} \tag{3.8b}$$

$$L\left(\mathbf{y}^{(N_{\text{iter}})}\right) = \begin{pmatrix} \pi_1 \sum_{j=1}^{N_{\text{ft}}} \alpha_{1,j}(\mathbf{y}_1^{(N_{\text{iter}})}) l_j \\ \vdots \\ \pi_{N_c} \sum_{j=1}^{N_{\text{ft}}} \alpha_{N_c,j}(\mathbf{y}_{N_c}^{(N_{\text{iter}})}) l_j \end{pmatrix} \tag{3.8c}$$

where $\varepsilon > 0$ must be small enough in order to not induce a bias in the approximation of $\hat{\mathcal{J}}$ with R. It should be noted that R is introduced in (3.6) only in order to avoid singularities in the resolution. Therefore, its contribution must be negligible compared to the other terms in the objective function.

A schematization of the insight developed in this Subsection is given in Figure 3.2. This optimization thus consists in finding p such that the cost function of the standard OCP, denoted by J_{OCP}, approximates $\hat{\mathcal{J}}$ around the solution of the FO-OCP. Eventually, as $\hat{\mathcal{J}}$ also approximates $\mathcal{J}$, the array of parameters p obtained allows the standard optimal control problem to efficiently reduce the fatigue cost $\mathcal{J}$.

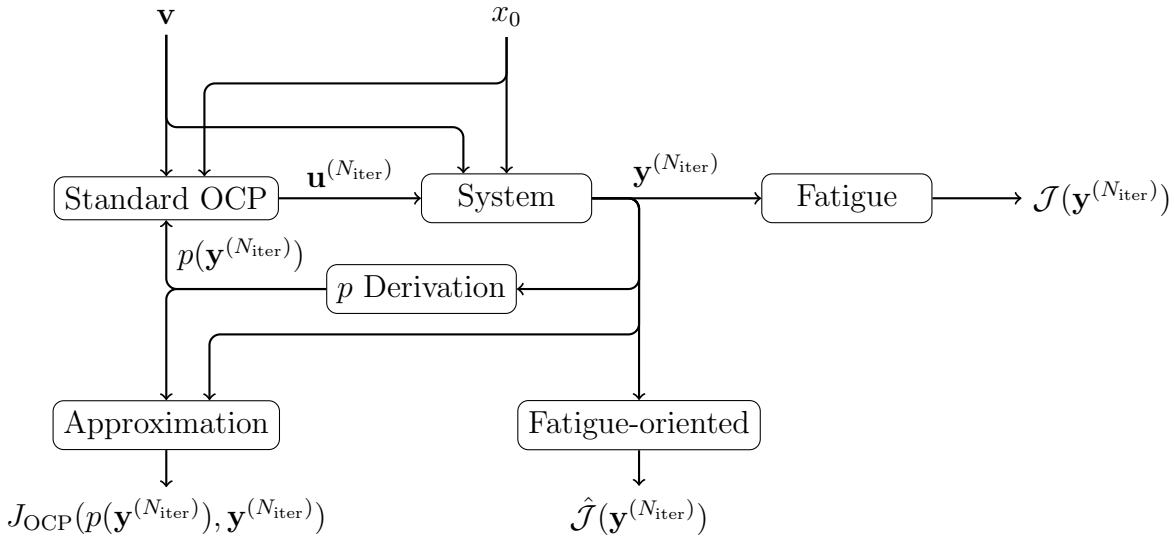

FIGURE 3.2 – Schematization of the fatigue-oriented optimization using the fatigue-oriented cost function $\mathcal{J}$.

Therefore, p depends on $\mathbf{y}^{(N_{\text{iter}})}$, while $\mathbf{y}^{(N_{\text{iter}})}$ depends on the p value used in the optimal control problem. This kind of problem is thus a fixed-point problem whose formulation is detailed in the next subsection.

3.2.2 Fixed-point problem formulation

For a given $\mathbf{y}^{(N_{\text{iter}})}$, we have an array of parameters p which parameterizes the quadratic cost function J_{OCP}, approximating the fatigue-oriented cost function $\hat{\mathcal{J}}$ for the optimal

control problem (3.6). Therefore, (3.6) will yield an optimal solution, denoted by $\mathbf{u}^{(N_{\text{iter}})}$. Integrating this solution with the dynamic system equations (3.6b) and (3.6f) will give a new $\mathbf{y}^{(N_{\text{iter}}+1)}$, which would thus yield a new array of parameters p for (3.6), therefore a new optimal solution will be obtained, and so on.

Therefore, in order to solve the FO-OCP, defined by (3.4), with the approximated QP optimal control problem (3.6), the right $\mathbf{y}^{(N_{\text{iter}})}$ or $p(\mathbf{y}^{(N_{\text{iter}})})$ must be found. Indeed, it will allow us to correctly approximate $\hat{\mathcal{J}}$ around the solution of the FO-OCP by deriving the appropriate array of parameters. Hence, the solutions to the FO-OCP and (3.6) should be equivalent.

Hence, solving the FO-OCP is equivalent to solving the fixed-point problem :

$$\mathbf{y} = g_{x_0,\mathbf{v}}(\mathbf{y})$$

where the function $g_{x_0,\mathbf{v}}$ is defined by the aAlgorithm 1, for given initial state x_0 and disturbance trajectory $\mathbf{v}$.

Algorithm 1: Function $g_{x_0,\mathbf{v}} : \mathbf{y}^{(N_{\text{iter}})} \longrightarrow \mathbf{y}^{(N_{\text{iter}}+1)}$

Result: Estimate a new system output trajectory $\mathbf{y}^{(N_{\text{iter}}+1)}$ from an initial output trajectory $\mathbf{y}^{(N_{\text{iter}})}$, given an initial state x_0 and a disturbance trajectory $\mathbf{v}$.

$\mathbf{y}^{(N_{\text{iter}})} \longleftarrow$ **Get** the initial output trajectory

$p\left(\mathbf{y}^{(N_{\text{iter}})}\right) \longleftarrow$ **Solve** (3.8)

$\mathbf{u}^{(N_{\text{iter}})} \longleftarrow$ **Solve** (3.6)

$\mathbf{y}^{(N_{\text{iter}}+1)} \longleftarrow$ **Integrate** the system dynamic equations (3.6b) and (3.6f)

Note that this fixed-point problem used to solve the FO-OCP is very similar to the Sequential Quadratic Programming (SQP) approach and it is exactly SQP if f and h are similar functions. However, if f or h are not similar, (3.6) is also an NLP but with a quadratic cost function. The advantage of this method compared to the one solving directly the FO-OCP NLP is that it allows us to :

— Express the optimal control problem under a standard formulation
— Understand the role of the weighting matrices and how they should be varied in a fatigue cost optimization using quadratic forms
— Understand how the weighting matrices are modified by the wind and turbine conditions, in order to use directly the right parameters in an optimal control problem using a standard formulation

3.2.3 Convergence of the fixed-point problem

Concerning the convergence of this fixed-point problem, it cannot be proven that the fixed-point method will converge. However, it can be assumed that if it does converge, it will probably converge on a local minimum or saddle point of the FO-OCP. It should be noted that, as $\hat{\mathcal{J}}$ is only an approximation of the fatigue cost $\mathcal{J}$, converging on a local minimum of the FO-OCP does not mean that the fatigue cost $\mathcal{J}$ is minimized locally.

From my experience, it was observed that the fixed-point problem can have convergence issues and that filtering the fixed point using an appropriate parameter $\beta \in]0, 1[$,

such that :

$$\mathbf{y}^{(N_{\text{iter}}+1)} = (1-\beta) g_{x_0,\mathbf{v}}\left(\mathbf{y}^{(N_{\text{iter}})}\right) + \beta \mathbf{y}^{(N_{\text{iter}})}$$

allows us to help the fixed point in converging, as suggested in [84]. 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 $\hat{\mathcal{J}}$. However, if the system is nonlinear, there might be other convergence issues due to the non-convexity of the NLP (3.6).

3.2.4 Consequences on the optimal control problem and similarities with the literature

To summarize this section, the fatigue cost $\mathcal{J}$ is approximated with a data-driven cost function $\hat{\mathcal{J}}$ based on quadratic features. This allows us to turn the fatigue trade-off optimization problem into a series of standard optimal control problems using quadratic cost functions, while solving the fixed-point problem $\mathbf{y} = g_{x_0,\mathbf{v}}(\mathbf{y})$.

The solution to this fixed-point problem, denoted by $\mathbf{y}^{(\infty)}$, depends on the values of x_0 and $\mathbf{v}$. Therefore, the array of parameters parameterizing the standard optimal control problem defined by (3.4), $p(\mathbf{y}^{(\infty)})$ is also a function of x_0 and $\mathbf{v}$.

This is reminiscent of the fact that, **in order to minimize a fatigue cost such as $\mathcal{J}$, the weighting matrices of a standard optimal control problem using a quadratic cost function should vary with respect to the wind and turbine conditions**.

The insight of varying the weighting matrices of quadratic cost functions in order to match a fatigue cost was already suggested and implemented in [62], where the parameters of an MPC using a linear quadratic cost function were varied dynamically. However, this approach was 'blind' as it was using an identification algorithm and relied on the fact that fatigue damage is a linear combination of the variance of the signal and its first derivative, which is not generally true.

In the presented methodology here, the relationship between quadratic cost functions of the signal and its fatigue damage is intended to be more general and is thus nonlinear. The idea of adapting a cost function in order to match fatigue damage, depending on the initial conditions and disturbance characteristics, is to some extent also present in [68]. Indeed the cost function parameters depend on the results given by the RFC algorithm run on the output given by the last iteration in the optimization, which could make one think about the successive resolution of standard optimal control problems in the fixed point.

The difference in the methodology presented in this chapter, compared to [68], is that the RFC algorithm is not necessary, and therefore the computational complexity is better scaled to the number of components considered in the fatigue trade-off optimization. Moreover, both of these works suggest once again that **control strategies with adaptive features are best suited to manage fatigue trade-off problems**.

3.3 Application to an LTI system

This section aims at evaluating the potential of this methodology, consisting in deriving a data-driven fatigue-oriented cost function based on quadratic features for its use in an optimal control problem.

The optimization of (3.6) using the fixed-point method is first compared to an NLP solver, in order to verify that the fixed-point efficiently solves the optimization problem. Then, the solution of the optimization (3.6) using the fixed-point method is compared to the one of an optimal control problem using a quadratic cost function (2.22) with fixed parameters, optimized for fatigue cost reduction. The comparison is performed on a realistic wind distribution, under a simplified simulation setting.

The outline of this section is the following :

— In Subsection 3.3.1, the environment used for the optimization is detailed, e.g. the system, wind distribution and optimal control problem considered.
— In Subsection 3.3.2, the derivation of the fatigue-oriented cost function $\hat{\mathcal{J}}$ is presented.
— In Subsection 3.3.3, open-loop optimizations of (3.6) using the fixed-point method and an NLP solver are compared in terms of fatigue cost and CPU time. Then a decomposition of the horizon is proposed, in order to reduce significantly the time needed for the optimizations using the fixed-point method.
— In Subsection 3.3.4, the performance of the FO-OCP in terms of fatigue cost expectancy reduction is analyzed. Then, the interest of having a controller that adapts the weighting matrices of its cost function in order to reduce the fatigue cost expectancy is once more highlighted.
— In Subsection 3.3.5, as several choices made in this section are not optimal in hindsight, possibilities of improvements of the presented results are given.

3.3.1 Simulation settings

This subsection presents the environment used for the optimization in this section. First of all, the information relative to the optimal control problem is given : e.g. parameters of the fatigue cost function, system considered, constraints. Then, the fatigue-oriented cost function derivation process is depicted and a word is given on the disturbances considered in the sequel.

The fatigue cost optimal control problem

The fatigue cost $\mathcal{J}$ considered in the following is the same as the one used in Chapter 2, defined by (2.3).

The LTI system considered in this section is derived from FAST linearization and augmented with actuator dynamics, such as (2.11) :

$$\dot{x} = \bar{A}(\bar{v}_{\text{hh}})(x - \bar{x}_{op}(\bar{v}_{\text{hh}})) + \bar{B}(\bar{v}_{\text{hh}})(u - \bar{u}_{op}(\bar{v}_{\text{hh}})) + \bar{B}_d(\bar{v}_{\text{hh}})(d - d_{op}(\bar{v}_{\text{hh}}))$$
$$y = \bar{C}(\bar{v}_{\text{hh}})(x - \bar{x}_{op}(\bar{v}_{\text{hh}})) + \bar{D}(\bar{v}_{\text{hh}})(u - \bar{u}_{op}(\bar{v}_{\text{hh}})) + \bar{D}_d(\bar{v}_{\text{hh}})(d - d_{op}(\bar{v}_{\text{hh}})) + \bar{y}_{op}(\bar{v}_{\text{hh}})$$

The parameters used for the LTI system derivation are summarized in Table 3.1.

The reasons motivating the use of an LTI system free of constraints rather than a HAWT model or an LPV model such as (2.13) with actuator saturation are the following :

Parameter	Value
Operating wind speed $\bar{v}_{\text{hh}}$	12 m/s
Actuator time constant	0.1 s
Sampling time	0.1 s
Disturbances considered in the feedforward	hub-height wind speed
Constraints	None

TABLE 3.1 – Summary of the LTI system parameters.

— The convexity of the optimization problem relies only on the fatigue-oriented cost function one and avoids added pitfalls in this first analysis.
— It makes the first implementation of this method easier.
— This configuration allows us to break down the horizon into several subproblems, which reduces significantly the CPU time and RAM needed for every solution.

Nevertheless, it is possible to have more accurate results and extend the validation of this fatigue-oriented cost function by considering a more realistic internal model and constraints on the system.

Wind disturbances

The set of winds considered is based on the data gathered in the NREL NWTC measurement campaign described in [83]. All the winds are generated for a 12 m/s mean wind speed with the wind generator TurbSim [73], in order to oscillate around the LTI system operating point. It should be noticed that the winds considered are 50 seconds long.

The parameters that change between wind generations are the seed used for the generation of random numbers in the TurbSim algorithm and the turbulence intensity. The random seed is a random integer drawn in a uniform distribution. The turbulence intensity is randomly drawn in a probability distribution corresponding to the probability of having a turbulence intensity given that the mean wind speed is 12 m/s. Figure 3.3 plots the histogram of the 1000 turbulence intensities considered for wind generation.

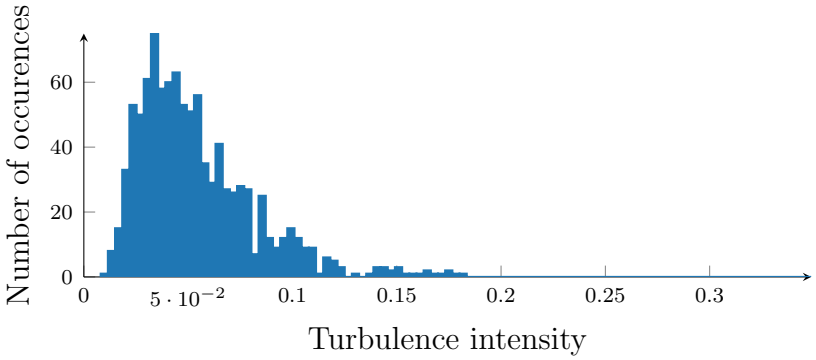

FIGURE 3.3 – Histogram of the 1000 turbulence intensities used for the set of wind generation.

3.3.2 Fatigue-oriented cost function derivation

Remember that the derivation of the fatigue-oriented cost function $\hat{\mathcal{J}}$ is divided in two parts :

— The generation of the time series, whose fatigue damage $\mathcal{D}_k$ and quadratic features $\mathbf{J}$ are estimated, yielding pairs of $\mathcal{D}_k$ and $\mathbf{J}$.
— The nonlinear regression between the fatigue damage $\mathcal{D}_k$ and the quadratic features $\mathbf{J}$, in order to be able to predict the fatigue damage from a signal's quadratic features.

Data generation

The fatigue-oriented cost function $\hat{\mathcal{J}}$ might be more accurate if it was derived from data which could be trajectories given by solutions of the FO-OCP. However, it is not possible to know in advance what these trajectories would be, as the FO-OCP themselves depend on this data generation.

Generate realistic trajectories Nevertheless, the trajectories given by the FO-OCP are derived from open-loop optimizations using quadratic cost functions, whose weighting matrices are obtained by solving the fixed-point problem defined in Subsection 3.2.2. Therefore, it is proposed to use trajectories of open-loop optimizations, using the integral of a quadratic stage cost over time as an objective function, for time series generation. These time series will then be used for the fatigue damage estimation and quadratic features, necessary for the regression of the fatigue damage $\mathcal{D}_k$ from $\mathbf{J}$.

Stage cost parameterization However, the stage cost in the objective function must be chosen such that it is close to solutions given by the fixed-point method. As very few prior on the stage cost weighting matrices values is available, it is proposed to use the parameterized quadratic stage cost defined by (2.22), where the weighting matrices Q and R are parameterized by ν :

$$Q(\nu) = \mathrm{diag}\left([0, 1, 1, 0, \nu, \nu]\right) \tag{3.10a}$$
$$R(\nu) = \mathrm{diag}\left([0, 1, 1)\min(\nu, 1) \times 10^{-10}\right. \tag{3.10b}$$

For the data generation, various values of the ν parameter are selected in order to observe different dynamics in the resulting time series. Moreover, it will allow us to constitute a database allowing us to analyze the influence of the ν parameter on the fatigue cost.

These open-loop optimizations are using the parameterized quadratic stage cost in their cost function, for 50 values of the parameter ν, selected in a logarithmic space ranging from 10^1 to 10^4. A summary of the parameters used for the open-loop optimizations is given in Table 3.2.

Parameter	Value
Time length	50 s
Sampling time	0.1 s
Parameter ν	$\{10^1, 10^{1.0612}, \ldots, 10^4\}$
Internal model	LTI
Operating wind speed	12 m/s
Constraints	None
Initial conditions	Origin

TABLE 3.2 – Summary of the parameters used for the open-loop optimizations using the parameterized quadratic stage cost defined by (2.22).

It should be noted that these parameters are selected in order to match the simulation settings defined in Subsection 3.3.1. The disturbances to be considered in the open-loop optimizations are also the ones defined in Subsection 3.3.1. Therefore, the only varying parameters between disturbances are the turbulence intensity and the random seed parameters of the TurbSim generator.

Tuning the stage cost parameter value Figure 3.4 plots the evolution of the fatigue cost $\mathcal{J}$ with the ν parameter for 9 winds, normalized by the lowest fatigue cost $\mathcal{J}$ obtained with each wind. It can be observed that for a given wind, the fatigue cost $\mathcal{J}$ varies with the value of ν. Moreover, for the same wind, there are values of ν for which the fatigue cost $\mathcal{J}$ is significantly higher than others. However, for a given wind, if the FO-OCP should converge on one of these samples, it should be near the one corresponding to the ν giving the lowest fatigue cost $\mathcal{J}$.

In order to optimize $\mathcal{J}$ efficiently in the FO-OCP, the fatigue-oriented cost function $\hat{\mathcal{J}}$ should accurately approximate the fatigue cost $\mathcal{J}$ around its minimum for a given wind. Therefore, considering the ν values for which $\mathcal{J}$ is very high for a given wind in the regression between the $\mathcal{J}$ and J_{var}, is of limited interest and could deteriorate the quality of the regression. Hence, these irrelevant cases are removed from the generated dataset in the following regression.

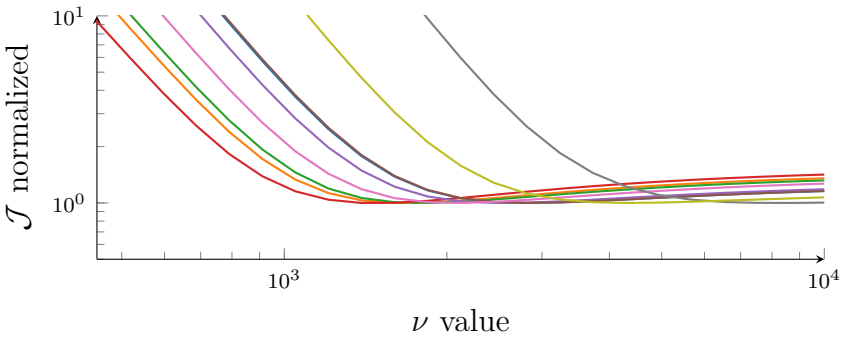

FIGURE 3.4 – Plot of the fatigue cost $\mathcal{J}$ and stage cost parameter ν for 9 different winds, normalized by the minimum fatigue cost $\mathcal{J}$ obtained for each wind, each colour representing a wind realization.

Regression between fatigue damage and quadratic forms

Quadratic features Concerning the quadratic features, only J_{Var} is selected, because it allows us to have already relatively good results and avoids the use of time derivatives which can be a source of noise in the optimization. However, it might be interesting in future works to consider more features and observe the behaviour of the resulting optimizations.

Parametric regression function For the regression of the fatigue damage from J_{Var}, it can be observed in Figure 3.5 that the relationship between the logarithms of the fatigue damage and J_{var} is linear. Therefore, a logarithmic transformation is performed on both J_{var} and the normalized damage. Then, a linear regression is fitted between the logarithms of J_{var} and the fatigue damage. Therefore, the predicted logarithm of fatigue damage, denoted by $\hat{\mathcal{D}}_k$ for the k^{th} component, can be expressed as follows :

$$\log\left(\hat{\mathcal{D}}_k\right) = w_k \log J_{\text{Var}}(\mathbf{y}_k) + b_k \tag{3.11}$$

where w_k is the linear coefficient and b_k is the bias term. The values w_k and b_k obtained for the four components considered in the fatigue cost are summarized in Table 3.3.

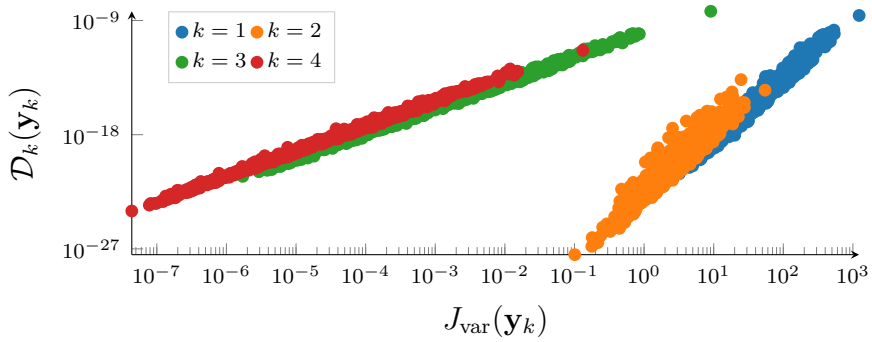

FIGURE 3.5 – Scatter plot of the variance $J_{\text{var}}(\mathbf{y}_k)$ and fatigue damage $\mathcal{D}_k(\mathbf{y}_k)$ of each system output, obtained from the above-described data generation process.

Component	k	w_k	b_k
M_{yaw}	1	4.92	-53.70
M_{tilt}	2	5.01	-52.13
θ_{yaw}	3	1.99	-22.75
θ_{tilt}	4	1.99	-21.48

TABLE 3.3 – Summary of the w_k and b_k values obtained from the regression in the cost function derivation.

Therefore, the expression of the fatigue-oriented cost function $\hat{\mathcal{J}}$ is the following :

$$\hat{\mathcal{J}}(\mathbf{y}) = \sum_{k=1}^{N_c} \pi_k \underbrace{e^{b_k} J_{\text{Var}}(\mathbf{y}_k)^{w_k}}_{\hat{\mathcal{D}}_k(\mathbf{y}_k)} \tag{3.12}$$

where π_k, w_k and b_k are respectively the price of replacement, linear coefficient and bias term for the k^{th} component. The values of π_k are parameters of the fatigue cost function $\mathcal{J}$, summarized in Table 2.1. Moreover, it can be seen that as the w_k are all above 1, the fatigue-oriented cost function is thus convex, as a sum of compositions of convex functions.

In order to evaluate the potential performance of the FO-OCP using this newly derived fatigue-oriented cost function $\hat{\mathcal{J}}$, it is possible to compare its solutions to the ones given by the open-loop optimizations used in the above-described data generation process.

3.3.3 Comparison of the fixed point method and NLP open-loop optimizations

In order to verify that the fixed-point method allows us to efficiently solve the FO-OCP, it is proposed to compare the optimization solved with the fixed-point method to the one using the NLP solver `CasADi` [79]. The features interesting for open-loop optimization of the FO-OCP within the scope of reducing the fatigue cost $\mathcal{J}$ are thus :

— The ability of the optimization to minimize $\hat{\mathcal{J}}$
— The potential for reducing $\mathcal{J}$ and the match between $\hat{\mathcal{J}}$ and $\mathcal{J}$
— The computational cost of the optimization

The latter point had major shortcomings compared to the NLP solver, especially for very long optimization horizons, where the RAM is filled and the CPU time is getting tremendously long. Therefore, a decomposition of the intermediate optimization problems is proposed in Subsection 3.3.3, in order to significantly decrease the computational burden.

It should be noted that the fixed-point method is using a filtering parameter $\beta = 0.7$, defined in Subsection 3.2.3, which was found to be a good value. Moreover, these optimizations are tested under two horizons of 10 and 100 seconds, under generated winds.

Minimizing the data-driven fatigue-oriented cost function $\hat{\mathcal{J}}$

The first objective of the fixed-point method is to minimize the value of the data-driven fatigue-oriented cost function $\hat{\mathcal{J}}$.

Therefore, Figures 3.6 and 3.7 plot the evolution of $\hat{\mathcal{J}}$ with the iterations of the fixed-point method, for optimization horizons of respectively 10 and 100 seconds, which is denoted by FP in the legend. The NLP curves correspond to the value of $\hat{\mathcal{J}}$ obtained with the optimum found by the NLP solver. The following observations can be made :

- The fixed-point method allows us to obtain lower values for $\hat{\mathcal{J}}$ than the NLP solver in both cases, which should be due to the stopping criteria implemented.
- The fixed-point method effectively converges, which means that a local optimum or saddle point is probably reached.
- Moreover, as $\hat{\mathcal{J}}$ is convex in this example, it has reached the global optimum.
- The cost $\hat{\mathcal{J}}$ is monotonically decreasing for the 10 seconds case, while for the 100 seconds case, some oscillations are present for the first iterations and then $\hat{\mathcal{J}}$ keeps on decreasing to convergence.

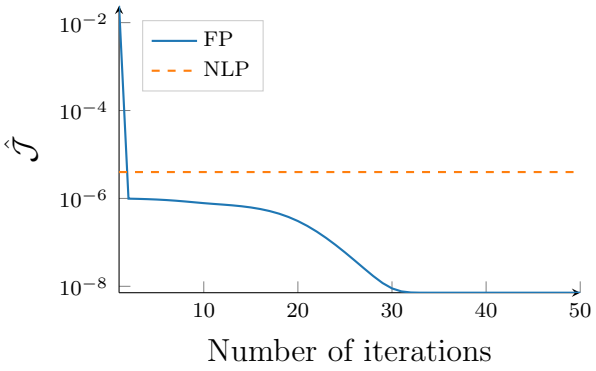

FIGURE 3.6 – Evolution of $\hat{\mathcal{J}}$ with the fixed point iterations in the semilogarithmic scale for a time horizon of 10 seconds.

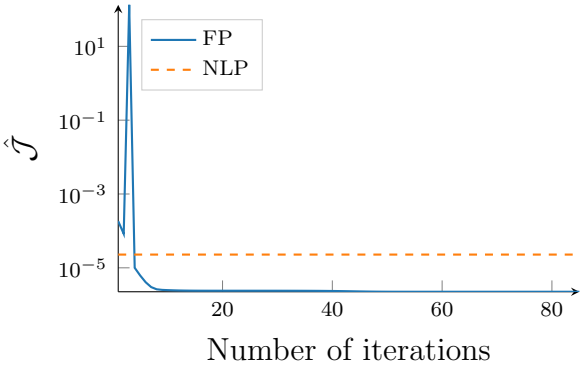

FIGURE 3.7 – Evolution of $\hat{\mathcal{J}}$ with the fixed-point iterations in the semilogarithmic scale for a time horizon of 100 seconds.

Minimizing the fatigue cost $\mathcal{J}$

The second objective, which is expected to be induced by the introduction of the fatigue-oriented cost function, is to efficiently reduce the fatigue cost $\mathcal{J}$.

Hence, Figures 3.8 and 3.9 plot the evolution of $\mathcal{J}$ with the fixed-point iterations and the cost obtained with the NLP solver. Similar observations to the one made previously on the evolution of $\hat{\mathcal{J}}$ can be done :

- The fatigue cost $\mathcal{J}$ globally decreases with the number of iterations and eventually converges to values much lower than the one found by the NLP solver.

— The fatigue cost $\mathcal{J}$ is several orders of magnitude lower with the fixed-point method than the NLP solver for the 10 seconds case.

Therefore, the expected reduction of the fatigue cost $\mathcal{J}$ is reached, which supposes that the data-driven fatigue-oriented cost function matches the fatigue cost in both cases.

Figures 3.10 and 3.11 plot the scatter of $\hat{\mathcal{J}}$ and $\mathcal{J}$ during the fixed-point iterations. It is thus confirmed that in these examples, the predictions made by the fatigue-oriented cost function $\hat{\mathcal{J}}$ matches the effective fatigue cost $\mathcal{J}$ relatively well.

However, the observations on the fatigue cost minimization are only valid for these two examples and might not be a general truth. Indeed, if the fatigue-oriented cost function does not allow us to generalize the predictions to all the conditions that the system can encounter, then there might be a mismatch between $\hat{\mathcal{J}}$ and $\mathcal{J}$. If there is such a mismatch, then the optimization of (3.4) will still minimize $\hat{\mathcal{J}}$ but the minimization of $\mathcal{J}$ might not be induced, which is the ultimate objective.

Therefore, **it is of primal importance to have a fatigue-oriented cost function that allows us to match the fatigue cost in every condition that the closed-loop system can encounter**. Otherwise, the fatigue cost optimization can become erratic in some conditions.

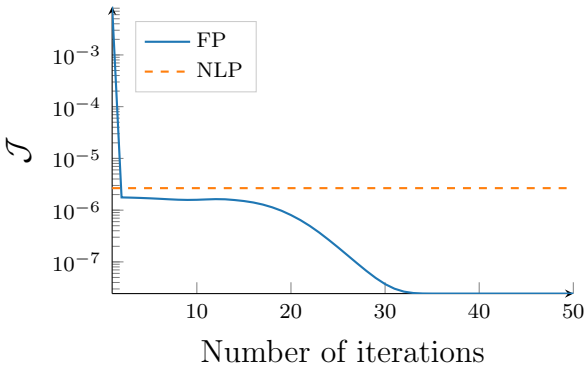

FIGURE 3.8 – Evolution of $\mathcal{J}$ with the fixed-point iterations in the semilogarithmic scale for a time horizon of 10 seconds.

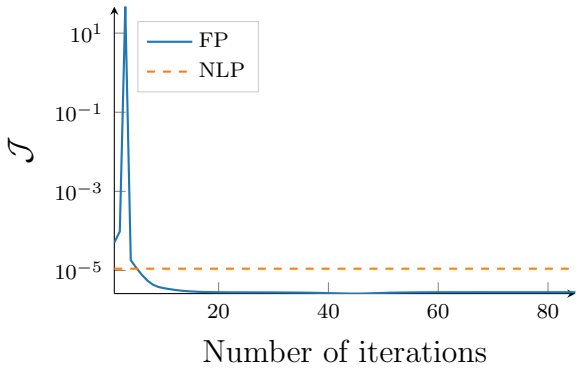

FIGURE 3.9 – Evolution of $\mathcal{J}$ with the fixed-point iterations in the semilogarithmic scale for a time horizon of 100 seconds.

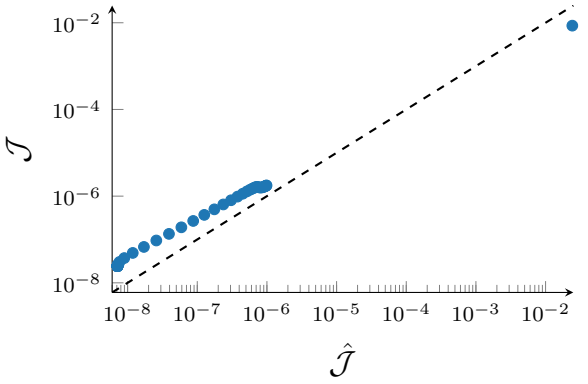

FIGURE 3.10 – Scatter plot of $\mathcal{J}$ and $\hat{\mathcal{J}}$ along the optimization iterations in the logarithmic scale for a time horizon of 10 seconds.

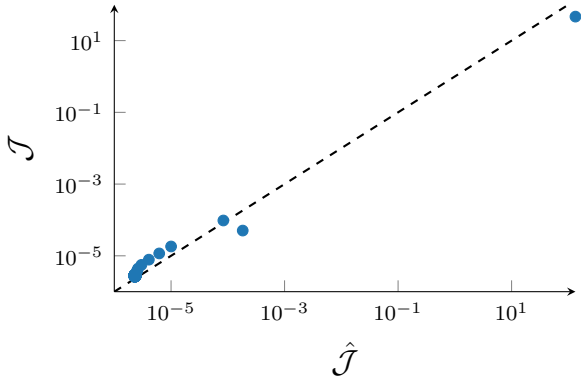

FIGURE 3.11 – Scatter plot of $\mathcal{J}$ and $\hat{\mathcal{J}}$ along the optimization iterations in the logarithmic scale for a time horizon of 100 seconds.

Computational cost shortcomings

In the case presented in this Section, the system is LTI and free of inequality constraints. The fixed-point problem consists thus in solving successive QP, whose solution can be obtained analytically, similarly to the method presented in equation (2.17).

The resolution of the analytical solution requires matrix inversions of dimension $N \times \mathrm{n_u}$, which must be performed for every iteration of the fixed-point method, where N is the number of time steps in the horizon and n_u is the dimension of the input. These matrix inversions require a tremendous amount of RAM, especially for very long optimization horizons such as 100 seconds where $N = 1000$.

In Table 3.4, the time needed for the compilation and resolution of the optimization of (3.4) using the fixed-point or the NLP solver are given. It should be noted that the fixed-point method does not require a compilation and that optimizations are run on a desktop PC equipped with a 2.60GHz processor and 16Gb RAM. It can be seen that the time needed for the resolution is one to two orders of magnitude greater than the time required by the NLP solver. The resolution time is particularly high for a very long optimization horizon of 100 seconds, where the RAM is completely filled for each matrix inversion required in the fixed-point iterations.

Metric	T	N	NLP solver	Fixed-point
Compilation time (s)	10	100	17	None
	100	1000	5833	None
Resolution time (s)	10	100	0.21	3.2
	100	1000	26.5	2769.4

TABLE 3.4 – Summary of the compilation and solving time for the fixed-point and NLP solver, bringing to light that the resolution of the fixed-point method is highly time-consuming compared to the NLP one.

Therefore, the computational burden of the fixed-point method compared to that of the NLP solver is a major drawback for an implementation of the fixed-point method optimization in an MPC. However, there is still a possibility to significantly reduce the computational cost of the QP by decomposing its optimization horizon.

Decomposition of the optimization horizon

In order to decrease both the CPU time and RAM needed for the fixed-point method optimization, a solution consisting in breaking down the time horizon into several sub-horizons constrained between each other is proposed, similarly to what is done in [85, 86].

The insight is to break down the optimization problem (3.6) into M intervals. For every interval a QP, denoted by QP_j for the j^{th} interval, constrained on its initial and final states, denoted respectively by $x_0^{(j)}$ and $x_0^{(j+1)}$, is solved analytically. Then, the solution of the initial QP, whose horizon is not broken down, can be retrieved by solving a Master QP, managing the initial states of every interval.

For the case of variance it is a little more complex, as the stage cost depends on the average on the full horizon, denoted by μ. Therefore μ is considered as an exogenous input of the local QP and can be retrieved by adding an equality constraint to the QP managing

the initial states of every interval. The details of this method and an illustration can be found respectively in Appendix E and Figure 3.12.

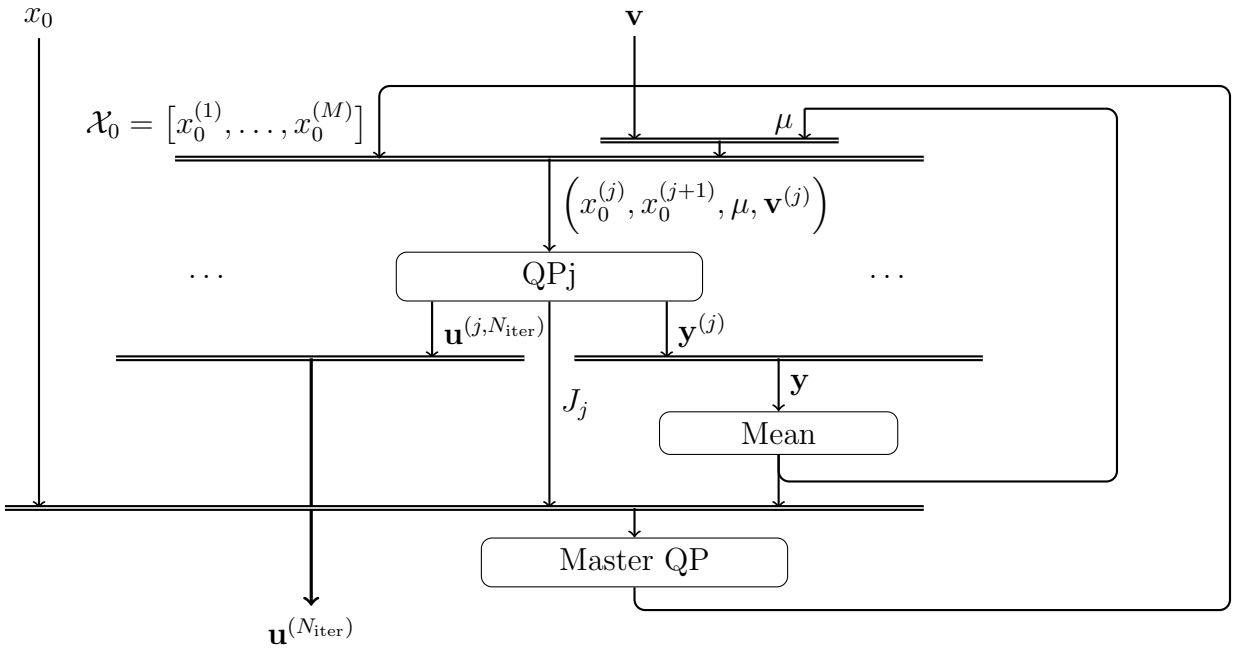

FIGURE 3.12 – Schematization of the horizon decomposition used, further details can be found in Appendix E.

This decomposition of the optimization horizon method allows us to significantly reduce the RAM needed, if the number of intervals M is judiciously chosen :

— The M local QP performed on every interval needs a matrix inversion of dimension $\frac{N}{M} \times \mathrm{n_u}$.

— The QP managing the initial states needs a matrix inversion of dimension $M \times \mathrm{n_x}$.

Let us bear in mind that the initial QP needed a single matrix inversion of dimension $N \times \mathrm{n_u}$. As an example, for a 10 seconds prediction horizon, where $N = 100$, $\mathrm{n_x} = 6$ and $\mathrm{n_u} = 2$, the value of M allowing us to minimize the computational cost is 25. Therefore instead of performing a single matrix inversion of dimension 200, the decomposition of the horizon allows us to perform 25 matrix inversions of dimension 8 for the local QPs and one of dimension 150 for the Master QP.

In terms of CPU time and RAM, there is a trade-off managed by M. Indeed, a greater M means more matrix inversions of smaller matrices and the inversion of a bigger matrix for the QP managing the initial states. Thus Figures 3.13 and 3.14 plot the evolution of the CPU time with the number of intervals M considered in the temporal decomposition, for time horizons of respectively 10 and 100 seconds. It can be seen in the figures that :

— There is a trade-off managed by M for the CPU times

— The optimal values are $M = 25$ and 50, for optimization horizons of respectively 10 and 100 seconds.

— The corresponding CPU times are respectively 0.098 and 2.044 seconds.

— The CPU times are decreased respectively by one and three orders of magnitude, compared to the initial QP where $M = 1$.

Therefore, this decomposition of the horizon allows us to have much faster optimization than the NLP solver does, as can be seen in Table 3.5.

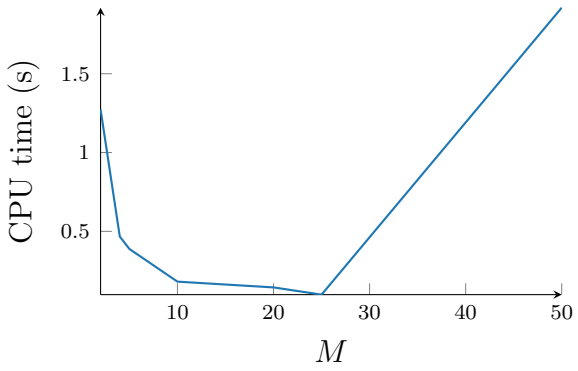

FIGURE 3.13 – Evolution of the CPU time needed for the fixed-point method with the number of intervals M considered in logarithmic scale for a time horizon of 10 seconds.

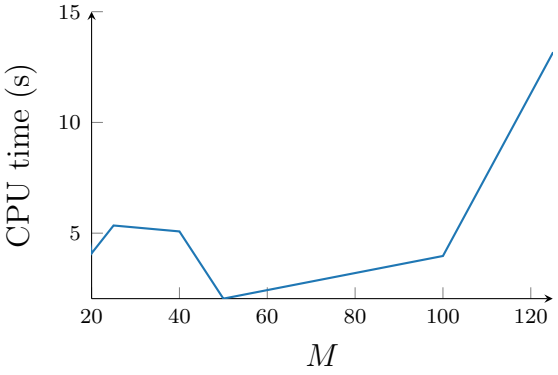

FIGURE 3.14 – Evolution of the CPU time needed for the fixed-point method with the number of intervals M considered for a time horizon of 100 seconds.

Metric	T	N	NLP solver	Fixed-point	FP with optimally decomposed horizon
Compilation time (s)	10	100	17	None	None
	100	1000	5833	None	None
Resolution time (s)	10	100	0.21	3.20	0.10
	100	1000	26.50	2769.40	2.01

TABLE 3.5 – Summary of the compilation and solving time for the NLP solver, fixed-point and fixed-point method with a decomposed horizon, showing that the resolution of the fixed-point method with a decomposed horizon is much less time-consuming than the other methods.

However, it should be noted that there are also possibilities of directly decomposing the optimization horizon of the NLP, but it might not be as easy with an NLP as it is with a QP. Moreover, it should be noted that the presented decomposition of the horizon can only be performed if the system is linear and free of inequality constraints.

In the sequel, this temporal decomposition scheme is used in the fixed-point iterations for solving the optimal control problems using $\hat{\mathcal{J}}$ as cost function.

3.3.4 Evaluation of the fatigue reduction potential

The aim of the FO-OCP, using the fatigue-oriented cost function $\hat{\mathcal{J}}$ as objective, is to efficiently reduce the fatigue cost $\mathcal{J}$. In this Subsection, its ability to reduce the fatigue cost is thus compared to that of parameterized quadratic open-loop optimizations.

As will be clear by the end of this subsection, this comparison leads to the conclusion that adapting the value of p that parameterizes a quadratic open-loop optimization can allow a tremendous fatigue cost reduction, compared to a quadratic open-loop optimization with fixed parameters. It will be shown that efficient adaptations can be obtained by means of FO-OCP or gain scheduling of the turbulence intensity.

Comparison of the FO-OCP and parameterized quadratic optimization fatigue costs

If the FO-OCP allows us to perfectly minimize the fatigue cost $\mathcal{J}$, then for a given wind, no optimal control problems can yield a solution with a lower fatigue cost. As it is not possible to test the solutions of all existing optimal control problems, this property is verified against the parameterized quadratic open-loop optimizations used in the data generation of Subsection 3.3.2. However, before this comparison, let us first introduce some mathematical formalism.

Let $\mathcal{J}_{\text{quad}}(\mathbf{v}, \nu)$ be the fatigue cost corresponding to the trajectories given by the parameterized quadratic open-loop optimization with the parameter ν, under the disturbance $\mathbf{v}$. Let ν_{best} and $\nu_{\text{adapt}}(\mathbf{v})$ be respectively the fixed ν value which minimizes the fatigue cost expectancy and the ν value which minimizes the fatigue cost for a given wind trajectory $\mathbf{v}$:

$$\nu_{\text{best}} = \arg\min_{\nu} \sum_{i=1}^{N_d} \mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu) \tag{3.13a}$$

$$\nu_{\text{adapt}}(\mathbf{v}) = \arg\min_{\nu} \mathcal{J}_{\text{quad}}(\mathbf{v}, \nu) \tag{3.13b}$$

where $\mathbf{v}_i$ is the i^{th} wind generated and N_d is the number of winds generated. It should be noted that ν_{best} and ν_{adapt} are analogous to $\bar{p}^\star$ and $p^\star(\mathbf{v})$, defined respectively by (1.51) and (1.52). Finally, let $\mathcal{J}_{\text{FO}}(\mathbf{v})$ be the fatigue cost $\mathcal{J}$ corresponding to the trajectories given by the FO-OCP.

If the FO-OCP allows us to perfectly minimize the fatigue cost $\mathcal{J}$, the following property should be true $\forall i \in \{1, \dots, N_d\}$:

$$\mathcal{J}_{\text{FO}}(\mathbf{v}_i) \leq \mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu_{\text{adapt}}(\mathbf{v}_i)) \leq \mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu_{\text{best}}) \tag{3.14}$$

A second property on the fatigue cost expectancy $\mathcal{J}$ follows by summing (3.14) $\forall i \in \{1, \dots, N_d\}$:

$$\sum_{i=1}^{N_d} \mathcal{J}_{\text{FO}}(\mathbf{v}_i) \leq \sum_{i=1}^{N_d} \mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu_{\text{adapt}}(\mathbf{v}_i)) \leq \sum_{i=1}^{N_d} \mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu_{\text{best}}) \tag{3.15}$$

Figure 3.15 and 3.16 plot the scatter of $\mathcal{J}_{\text{FO}}(\mathbf{v}_i)$ and respectively, $\mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu_{\text{adapt}}(\mathbf{v}_i))$ and $\mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu_{\text{best}})$ $\forall i \in \{1, \dots, N_d\}$. It can be noted from the figures and the associated data that :

- The FO-OCP matches very well the performance of the parameterized quadratic optimization using ν_{adapt}, in terms of fatigue cost reduction, and outperforms that of the parameterized quadratic optimization using ν_{best}
- In 88% of the cases $\mathcal{J}_{\text{FO}}(\mathbf{v}_i) \leq \mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu_{\text{adapt}}(\mathbf{v}_i))$ and in 100% of the cases $\mathcal{J}_{\text{FO}}(\mathbf{v}_i) \leq \mathcal{J}_{\text{quad}}(\mathbf{v}_i, \nu_{\text{best}})$.
- The fatigue cost expectancy with the FO-OCP is 5% greater than the one obtained with the optimization using ν_{adapt}, but **50% lower than the one obtained with the optimization using** ν_{best}.

Therefore, it can be claimed that the FO-OCP using the fatigue-oriented cost $\hat{\mathcal{J}}$ derived in this chapter reduces very efficiently the fatigue cost expectancy compared to a parameterized quadratic open-loop optimization with fixed parameters. Moreover, the FO-OCP performance is equivalent to that of a parameterized quadratic open-loop optimization, whose parameters are adapted in order to minimize the fatigue cost as a function of the wind.

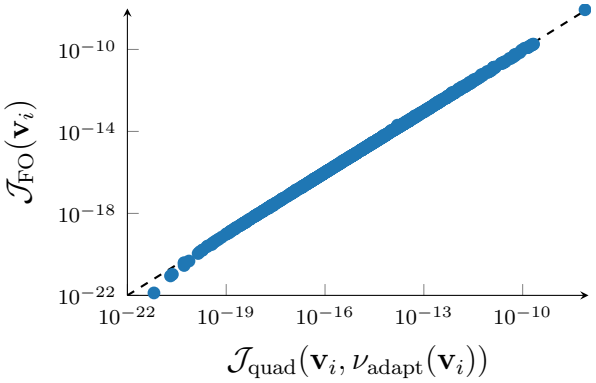

FIGURE 3.15 – Scatter plot of the fatigue costs $\mathcal{J}_{\mathrm{quad}}(\mathbf{v}_i, \nu_{\mathrm{adapt}}(\mathbf{v}_i))$ and $\mathcal{J}_{\mathrm{FO}}(\mathbf{v}_i)$, for $i \in \{1, \ldots, 1000\}$.

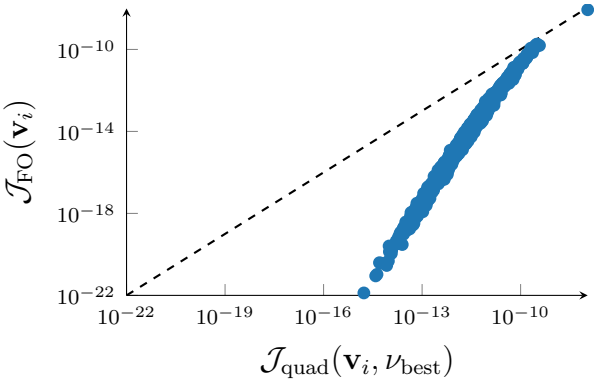

FIGURE 3.16 – Scatter plot of the fatigue costs $\mathcal{J}_{\mathrm{quad}}(\mathbf{v}_i, \nu_{\mathrm{best}})$ and $\mathcal{J}_{\mathrm{FO}}(\mathbf{v}_i)$, for $i \in \{1, \ldots, 1000\}$.

Interest of adapting the stage cost of a quadratic open-loop optimization

It can be noted that the FO-OCP can be solved using a fixed point method, presented in Subsection 3.2.2, which consists in successively solving quadratic open-loop optimizations, whose parameters are adapted as a function of the previous solution. The solution of the FO-OCP is thus also a solution of a quadratic open-loop optimization problem, whose parameters are obtained from the solution of the fixed-point method, which depends on the wind disturbance and initial condition.

Therefore, there is a great similarity between the FO-OCP and the parameterized quadratic open-loop optimization using ν_{adapt}, as they both result in being a quadratic open-loop optimization whose parameters are adapted based on the wind disturbance. Moreover, they have very similar performance in reducing the fatigue cost.

Note that on the one hand ν_{adapt} is a mapping based on the data generated, and the generalization of the relationship between ν_{adapt} and $\mathbf{v}$ might be challenging. On the other hand for the FO-OCP, the parameters of the quadratic optimization are automatically parameterized by means of the fixed-point method, which depends on the structure of $\hat{\mathcal{J}}$ and $\mathbf{v}$. Having an explicit method allowing us to obtain automatically these parameters, while efficiently reducing the fatigue cost $\mathcal{J}$, constitutes a considerable advantage for the FO-OCP over the parameterized quadratic open-loop optimization using ν_{adapt}.

Turbulence intensity gain scheduling

The above study shows that the parameters of a controller must be adapted as a function of the wind. In this study, the only varying parameters between each wind generation is the turbulence intensity. Therefore, this means that a turbulence intensity might be associated to a $\nu_{\mathrm{adapt}}(\mathbf{v})$.

Figure 3.17 plots the evolution of $\nu_{\mathrm{adapt}}(\mathbf{v})$ value with the corresponding turbulence intensity, estimated from the wind time series. It can be seen that there is a linear relationship in the logarithmic space, but it is not a perfect line, which means that turbulence intensity cannot explain by itself all the variations of $\nu_{\mathrm{adapt}}(\mathbf{v})$.

This observation suggests approximating $\nu_{\mathrm{adapt}}(\mathbf{v}_i)$ with a linear regression of the turbulence intensity of $\mathbf{v}_i$ in the logarithmic space and that a gain scheduling of turbulence intensity could help in estimating ν_{adapt}. However, turbulence intensity is not an instan-

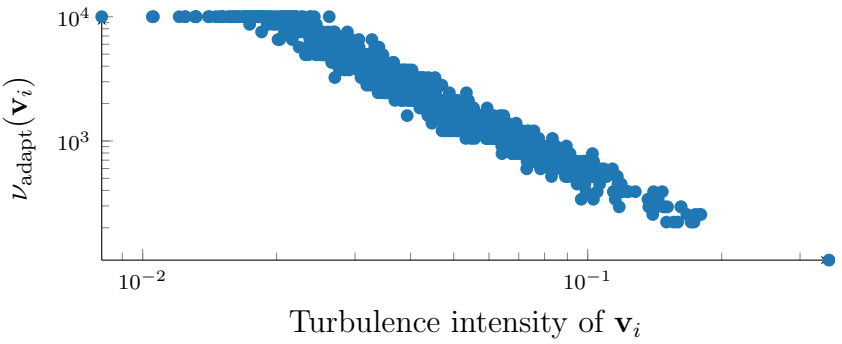

FIGURE 3.17 – Scatter plot of the turbulence intensity of $\mathbf{v}_i$ and $\nu_{\text{adapt}}(\mathbf{v}_i)$, for $i \in \{1, \dots, 1000\}$.

taneous value as opposed to wind speed, direction or shear.

If the exact turbulence intensity was known in advance, using the linear relationship between the logarithms of turbulence intensity and ν_{adapt}, the fatigue cost expectancy reduction compared to the parameterized quadratic open-loop optimizations using ν_{best} would be limited to 7.7%. This fatigue cost expectancy reduction is already good, but far from the potential fatigue reduction achieved with the FO-OCP. This suggests that the ν_{adapt} value does not only depend on the turbulence intensity but on other parameters which could be contained in the wind.

It should be noted that machine learning and feature selection tools could help in finding a more accurate and complex scheduling law from wind characteristics, but this is not the purpose of this chapter. However, this subject will be further studied and discussed in Chapter 5.

Summary

To summarize the results set out in this Subsection :

— A parameterized quadratic open-loop optimization which could efficiently adapt its parameters, in order to minimize the fatigue cost for a given wind disturbance, could reduce the fatigue cost expectancy by 50% compared to one which does not adapt its parameters.

— The **FO-OCP can be seen as an adaptive quadratic open-loop optimization** and achieves this 50% fatigue cost expectancy reduction.

— A summary of the fatigue cost expectancy reductions that the strategies proposed in this subsection could provide, compared to the parameterized quadratic open-loop optimization using ν_{best},is given in Table 3.6.

— **None of the strategies proposed above can be used as a controller in real life**, because they need the information on the disturbance over the whole optimization horizon of 50 seconds.

Strategy	$\mathcal{J}$ expectancy reduction
Parameterized quadratic with ν_{adapt}	51%
FO-OCP	50%
Scheduling of turbulence intensity	7.7%

TABLE 3.6 – Summary of the fatigue cost expectancy reductions compared to the parameterized quadratic optimization using ν_{best}.

This subsection aimed at highlighting the potential of the FO-OCP in efficiently reducing the fatigue cost expectancy and **the need to adapt the parameters of a quadratic cost function in order to reduce further the fatigue cost expectancy**.

The drawback is that the results obtained here are not realistic, as an accurate forecast of the future wind over 50 seconds would be necessary. However, these results suggest that it might be possible to find a control strategy allowing us to approach such results. As the FO-OCP is the strategy showing the highest potential in both closed-loop implementation and fatigue cost expectancy reduction, MPCs based on its expression are outlined in the following chapter.

3.3.5 Some suggestions to go further

Throughout this section, it was emphasized that several choices, in particular on the simulation conditions and on the derivation of the fatigue-oriented cost function, were not optimal in hindsight. Therefore, this subsection summarizes some improvement prospects for the results set out in this section.

Concerning the improvements to the simulation conditions :

- Consider using a more realistic internal model for simulations, representing the nonlinearities of a HAWT, such as an LPV model or a an aero-elastic simulator, instead of an LTI system.
- Consider constraints such as actuator saturation in the dynamic system.
- Consider uncertainties on the disturbance estimation.

These suggestions for improvements might not be going the way of the fatigue-oriented cost function, as many of the improvements concerning the simulation conditions would challenge even more the optimizations using the fatigue-oriented cost function. However, it could also be more challenging for the optimizations using quadratic cost functions.

Therefore, all these proposals for improvements are aimed at the same point, i.e. improving the quality of the results and having a more accurate idea of the data-driven fatigue-oriented cost function potential. The results presented in this section can thus be considered as preliminary, and further studies must be conducted in order to appreciate fully the potential of the methodology described in this chapter. Unfortunately, lack of time prevented the generation of these results and their presentation in this paper.

3.4 Summary and explanations

This chapter presented a methodology consisting in deriving a fatigue-oriented cost function from time series assumed to represent the dynamic behaviour of a closed-loop HAWT. From these time series we extracted features corresponding to quadratic cost functions and fatigue damage. Then, a regression is applied to the extracted data, in order to give a fatigue-oriented cost function that approximates the fatigue cost of a HAWT, defined by (1.50), from quadratic features.

It is shown that from the spectral approach to fatigue theory, described in Subsection 1.3.3, it is possible to relate these quadratic cost function features to fatigue damage with a posynomial expression. Therefore, the fatigue-oriented cost function is non-quadratic and used in the optimal control problem, defined by (3.4), in order to reduce

the HAWT fatigue cost $\mathcal{J}$ in an optimal control problem.

Thanks to the structure of the fatigue-oriented cost function, it is possible to solve the FO-OCP, defined by (3.4), with a fixed-point method, involving intermediate optimal control problems using quadratic cost functions. This allows us to emphasize once more that **in order to efficiently reduce fatigue in an optimal control problem using quadratic cost functions, the parameters of the cost function must be varied according to the turbine and wind conditions**.

Finally, the methodology presented in this chapter is tested under a simple example where the fatigue-oriented cost function helps in reducing the fatigue cost expectancy by about 50%, compared to the optimized parameterized quadratic cost function, defined by (2.22), with fixed parameters.

However, the results presented here are not totally representative of HAWT behaviour, as only an LTI system free of constraints was considered, with a perfect knowledge of the future wind. Therefore, it is important to test the potential of this approach on a more realistic HAWT simulator, under a wider spectrum of winds whose time series are not fully known in advance.

The study conducted in this chapter showed that the FO-OCP can help in significantly reducing the fatigue cost $\mathcal{J}$ in an open-loop optimal control problem. The issue that follows from this study is : How could this FO-OCP be used in closed-loop, for an efficient reduction of the closed-loop fatigue cost ? This issue is addressed in Chapter 4.

Chapitre 4

Closed-loop implementation of the fatigue-oriented cost function for wind turbine control

As a reminder, it was suggested in Chapter 2 that there is no universal fixed array of parameters p defining a standard MPC that allows us to efficiently optimize the closed-loop fatigue cost $\mathcal{J}$, defined by (1.50). Moreover, it was shown that adapting the array of parameters p can allow us to significantly reduce the closed-loop estimated fatigue cost compared to an individual controller with a fixed p value, optimized for $\mathcal{J}$ expectancy minimization.

Besides in Chapter 3, thanks to a data-driven relationship between a quadratic cost function such as variance and fatigue damage, the FO-OCP, defined by (3.4), is defined based on this relationship, in order to efficiently reduce the fatigue cost $\mathcal{J}$. Then, a fixed-point method based on the expressions of $\hat{\mathcal{J}}$ and the FO-OCP is derived in order to efficiently adapt an array of parameters p, parameterizing an open-loop optimal control problem, for fatigue cost reduction.

The efficiency of the FO-OCP in reducing fatigue cost $\mathcal{J}$ suggests using FO-OCP in an MPC. However, it is shown in this chapter that the FO-OCP in an MPC does not always allow us to efficiently reduce the fatigue-oriented cost function $\hat{\mathcal{J}}$ in closed-loop simulations.

Therefore, a second MPC formulation alleviating the pitfalls of the previous one is proposed. This second MPC uses the integral of a quadratic stage cost over time as the objective function, whose stage cost is adapted based on the fatigue-oriented cost function $\hat{\mathcal{J}}$ formulation and previous time series. It is shown in this chapter that this second MPC mitigates the pitfalls of the previous one. This MPC allows us to better reduce the fatigue-oriented cost function $\hat{\mathcal{J}}$ and indirectly the fatigue cost $\mathcal{J}$.

The outline of this chapter is the following :

— In Section 4.1, the simulation settings, i.e. system, set of disturbances, fatigue-oriented cost function, used for the closed-loop implementation are defined.

— In Section 4.2, a first intuitive MPC formulation using directly the FO-OCP as its open-loop optimization problem is presented. Then, it is shown that the MPC cannot always efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$, which prevents us from efficiently reducing the fatigue cost $\mathcal{J}$.

— In Section 4.3, a second MPC formulation, alleviating the pitfalls of the first formulation, is described. It is eventually shown that this MPC can allow us to significantly reduce the fatigue cost $\mathcal{J}$ in closed-loop simulations, compared to a standard quadratic MPC.

— Finally, in Section 4.4, a discussion of the results is given, along with a description of the future studies which must be conducted in order to fully appreciate the potential of the presented approaches.

4.1 Simulation settings

This section presents the settings used for the simulations of the closed-loop MPCs. In order to simplify the computations and only analyze the ability of the MPCs to reduce the fatigue cost, the same simulation settings as used in Chapter 3 are selected in this chapter. Therefore, it avoids the presence of any subtleties such as non-convexity, model-mismatch or estimation errors.

It should be noted that for simplicity, the system is simplified by an LTI model linearized around 12 m/s, therefore the winds considered in the data generation and tests of

the closed-loop must vary around 12 m/s as well.

The two main blocks of the simulation settings are :
- The system model, described in Subsection 4.1.1.
- The set of winds used to disturb the model, described in Subsection 4.1.2.

In Subsection 4.1.3, the values of other simulation setting parameters are defined.

4.1.1 System model

In order to model the HAWT dynamics, an LTI model linearized around 12 m/s is considered, similar to the one defined by (2.11) :

$$\dot{x} = \bar{A}(\bar{v}_{\text{hh}})(x - \bar{x}_{op}(\bar{v}_{\text{hh}})) + \bar{B}(\bar{v}_{\text{hh}})(u - \bar{u}_{op}(\bar{v}_{\text{hh}})) + \bar{B}_d(\bar{v}_{\text{hh}})(d - d_{op}(\bar{v}_{\text{hh}}))$$
$$y = \bar{C}(\bar{v}_{\text{hh}})(x - \bar{x}_{op}(\bar{v}_{\text{hh}})) + \bar{D}(\bar{v}_{\text{hh}})(u - \bar{u}_{op}(\bar{v}_{\text{hh}})) + \bar{D}_d(\bar{v}_{\text{hh}})(d - d_{op}(\bar{v}_{\text{hh}})) + \bar{y}_{op}(\bar{v}_{\text{hh}})$$

up to the difference that the offsets are neglected, in order to simplify the MPC derivations. Nevertheless, in order to implement such MPCs on FAST, it should be noted that the offsets must be carefully taken into account and several linearizations might be considered. Moreover, the only exogenous disturbance considered on the system is the hub-height wind speed.

In the remainder of this chapter, the MPCs will use this LTI system as an internal model, with a perfect knowledge of the exogenous disturbance over the predicted horizon. Therefore the MPCs will have all the information needed to make exact predictions of the system's evolution. Moreover, the optimizations used in both MPCs will be convex provided that the cost function and the set of constraints are convex.

4.1.2 Wind disturbances

The set of winds considered is based on the data gathered in the NREL NWTC measurement campaign described in [83]. All the winds are generated for a 12 m/s mean wind speed with the TurbSim wind generator [73], in order to oscillate around the LTI system operating point.

The parameters that change between wind generations are the seed used for the generation of random numbers in the TurbSim algorithm and the turbulence intensity. The random seed is a random integer drawn in a uniform distribution and the turbulence intensity is randomly drawn in a probability distribution corresponding to the probability of having a turbulence intensity given that the mean wind speed is 12 m/s. Figure 4.1 plots the histogram of the 1000 turbulence intensities considered for the wind generation.

4.1.3 Miscellaneous

The remaining parameters considered for the simulation settings, such as the simulation time length, the sampling time or initial conditions, are summarized in Table 4.1. The simulation time length and sampling time values were selected by hand, in order to manage a trade-off between accuracy of the fatigue cost estimation and computational cost of the simulations. A signal must be sufficiently long and finely sampled to yield accurate fatigue cost estimations.

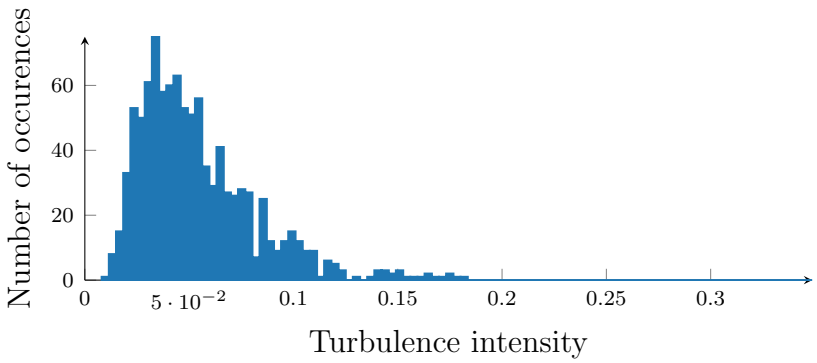

FIGURE 4.1 – Histogram of the 1000 turbulence intensities used for the set of wind generation.

In the remainder of this chapter, MPCs based on the fatigue-oriented cost function architecture will be considered. The simulation time length is of 48 seconds, because the winds generated are 50 seconds long and the MPC prediction horizon is 2 seconds long.

Parameter	Value
Simulation time length	48 s
Simulation sampling time	0.1 s
Initial conditions	Steady state
MPC prediction horizon	2 s

TABLE 4.1 – Summary of the remaining parameters used for the closed-loop simulations.

All the simulations are initialized around the steady state of the system, because it is only relevant to analyze fatigue damage when the closed loop is regulating the outputs around steady-state and not during transients. The fatigue-oriented cost function $\hat{\mathcal{J}}$ used in the remainder of this chapter is the same as the one defined in Subsection 3.3.2.

4.2 MPC implementation based on the fatigue-oriented optimization : the intuitive formulation

This section presents the most intuitive formulation of an MPC based on the FO-OCP, which directly uses the FO-OCP as the MPC optimization problem. However, it will be shown that such an MPC does not always allow us to efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$ in closed-loop simulations, which will abort any expectation of an efficient fatigue cost reduction. Therefore the pitfalls of this MPC are analyzed in order to design a second formulation, mitigating the latter, which will be presented in the next section.

The outline of this section is the following :

— In Subsection 4.2.1, the straightforward implementation of the FO-OCP in an MPC is briefly described. However, it will be shown that the closed-loop trajectories resulting from its implementation might significantly differ from the open-loop trajectories (even without model mismatch).

— In Subsection 4.2.2, a comparison of the fatigue-oriented cost $\hat{\mathcal{J}}$ reduction efficiency between the previously-defined MPC in closed-loop simulation and the FO-OCP in open-loop optimization is presented. This comparison shows that the MPC does not allow us to efficiently reduce $\hat{\mathcal{J}}$.

— In Subsection 4.2.3, the pitfalls of the MPC which are the cause of poor closed-loop performance are analyzed and solutions are proposed in order to mitigate these pitfalls.

4.2.1 Implementation and possible issues

The most straightforward way to use the FO-OCP in an MPC is to directly use the FO-OCP as the MPC optimization problem. The fixed-point method described in Subsection 3.2.2, allows us to solve the FO-OCP using successive standard optimization problems under a quadratic cost function. It should be noted that as only J_{var} is used in the fatigue-oriented cost function derivation, which is a quadratic term, the linear term can be ignored. The general formulation of the optimization problem which must be solved between each iteration of the fixed-point can be expressed as follows :

$$\min_{\mathbf{u}} \quad J(\mathbf{y}) = \int_{t_0}^{t_0+T} \left(y(\tau)^T Q(\tilde{\mathbf{y}})\, y(\tau) + u(\tau)^T R(\tilde{\mathbf{y}})\, u(\tau) \right) d\tau \tag{4.2a}$$

$$\text{s.t.} \quad \dot{x} = f(x, u, v, t) \tag{4.2b}$$

$$x(t_0) = x_0 \tag{4.2c}$$

$$0 = h(x, u, v, t) \tag{4.2d}$$

$$0 \geq g(x, u, v, t) \tag{4.2e}$$

$$y = h_{\text{y}}(x, u, v, t) \tag{4.2f}$$

where the weighting matrices Q and R are parameterized by a given output trajectory $\tilde{\mathbf{y}}$, which is the solution of the fixed point problem and thus a function of the initial condition x_0 and the predicted disturbance $\mathbf{v}$. It should be noted that the matrices Q and R are parameterizing the MPC and are contained in the array of parameters p which defines the MPC. This kind of open-loop optimization can be efficiently solved by NLP or QP solvers, depending on the nature of the constraints and dynamic system involved. Moreover, they are well known and widely used in the MPC literature.

A schematization of this MPC using the FO-OCP in its open-loop optimization problem, denoted by $\text{MPC}_{\text{direct}}$, is given in Figure 4.2. A standard MPC using the open-loop optimization problem defined by (4.2) is parameterized by an array of parameters p, whose derivation depends on the MPC solution output trajectory $\tilde{\mathbf{y}}$. Hence, the fixed-point problem defined in Subsection 3.2.2 must be solved between every update of the MPC, in order to find the right $p(\tilde{\mathbf{y}})$ for the standard MPC. It should be noted that this implies solving several times the standard MPC before the fixed-point converges. Moreover, the MPC solution depends on the current state of the system x and the measured disturbance trajectory over the prediction horizon $\mathbf{v}$. Therefore, the fixed-point solution and the parameter p depend also on x and $\mathbf{v}$.

Therefore, $\text{MPC}_{\text{direct}}$ can be seen as an MPC using the integral of a quadratic stage cost over time as objective, whose array of parameters p varies with the current state and the disturbance trajectory over the prediction horizon. On the other hand, when in Section 3.3.2 the FO-OCP is performed on a long open-loop optimization horizon, the optimal control problem also uses the integral of a quadratic stage cost over time as objective, whose array of parameters p is parameterized by another disturbance trajectory and another initial condition.

Hence, the closed-loop trajectories which are effectively visited by the closed-loop system might differ from the open-loop trajectory (even without model mismatch). This is

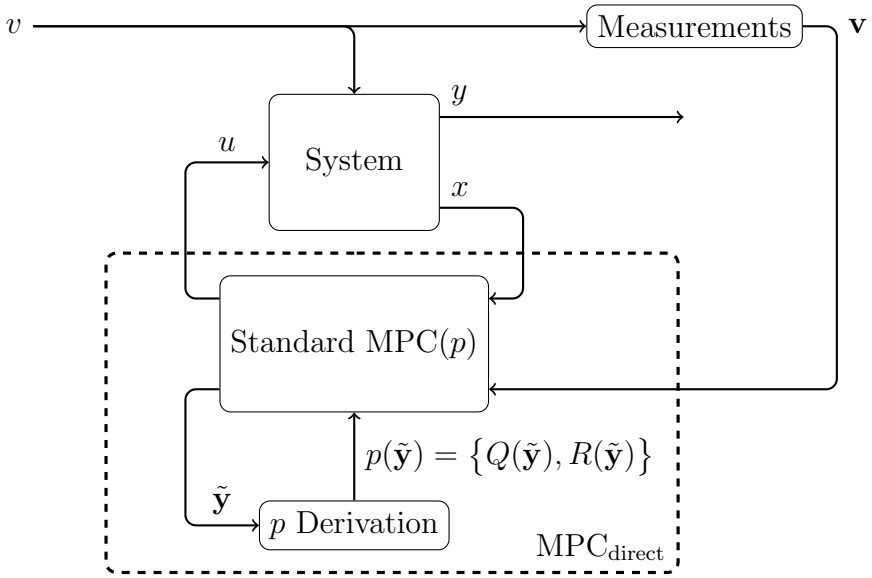

FIGURE 4.2 – Schematization of the $\text{MPC}_{\text{direct}}$ architecture, where the array of parameters p which parameterizes a standard MPC is the solution of a fixed-point problem. The fixed-point problem depends on the state of the turbine x and predicted disturbance trajectory over the prediction horizon $\mathbf{v}$.

due to the finite character of the prediction horizon and the fact that the optimal parameterization depends on the closed-loop trajectory. This induces a difference in performance between the open-loop ideal case with the FO-OCP and the closed-loop system, for which the optimal solution is never the one that corresponds to the trajectory that will be really encountered.

Therefore, the $\text{MPC}_{\text{direct}}$ might not be able to efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$ in closed-loop simulations. Moreover, if $\text{MPC}_{\text{direct}}$ fails to efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$, there is no reason for $\text{MPC}_{\text{direct}}$ to succeed in efficiently reducing the fatigue cost $\mathcal{J}$, which is the ultimate objective of the controller.

In the next Subsection, closed-loop simulations with $\text{MPC}_{\text{direct}}$ are performed and the resulting fatigue-oriented costs $\hat{\mathcal{J}}$ are compared to open-loop optimization solutions under the same disturbance.

4.2.2 Comparison of closed-loop simulations and open-loop optimizations

In this subsection, $\text{MPC}_{\text{direct}}$ is simulated in closed-loop of the system under a given wind disturbance 48 seconds long. Moreover, the FO-OCP is performed to solve an open-loop optimization problem under the same disturbance. The disturbance trajectories considered are the 1000 winds generated described in Subsection 4.1.2. The time series resulting from the closed-loop simulations and open-loop optimizations are then evaluated with the fatigue-oriented cost $\hat{\mathcal{J}}$.

It should be noted that MPC theory relies on the assumption that the optimal control problem solved in the MPC over a receding horizon should approximate the optimal control problem solution over an infinite horizon. However with the fatigue-oriented cost function $\hat{\mathcal{J}}$ as objective, it was suggested in Subsection 4.2.1 that the closed-loop MPC and the corresponding open-loop optimization over a long horizon might solve **different**

time varying quadratic optimal control problems and thus visit **different** trajectories. Therefore, $\text{MPC}_{\text{direct}}$ might not efficiently reduce the fatigue-oriented cost function $\hat{\mathcal{J}}$ over a long horizon.

In order to check this issue, the scatter of the fatigue-oriented cost $\hat{\mathcal{J}}$ given by the FO-OCP open-loop optimizations and the closed-loop simulations using $\text{MPC}_{\text{direct}}$ is plotted in Figure 4.3. It should be noted that if the closed-loop MPC matches the fatigue-oriented cost $\hat{\mathcal{J}}$ of the corresponding open-loop optimal control problem solution, the samples should follow the plane bisector. However, from Figure 4.3 and the corresponding data, it can be observed that :

— There are samples with a strong mismatch between the fatigue-oriented cost $\hat{\mathcal{J}}$ of the FO-OCP and that of the closed-loop simulations.
— The closed-loop simulations increase $\hat{\mathcal{J}}$ compared to the FO-OCP in 94% of cases.
— The closed-loop simulations increase $\hat{\mathcal{J}}$ compared to the FO-OCP by more than 100% in 45% of cases.

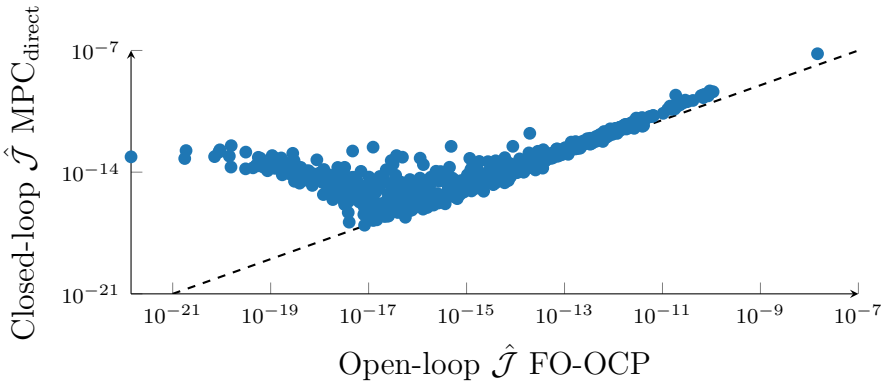

FIGURE 4.3 – Scatter plot of the fatigue-oriented cost $\hat{\mathcal{J}}$ of the closed-loop $\text{MPC}_{\text{direct}}$ simulation against that of the FO-OCP, under the 1000 winds generated.

To summarize, in the majority of cases, $\text{MPC}_{\text{direct}}$ does not reduce the fatigue-oriented cost $\hat{\mathcal{J}}$ efficiently. Therefore, if $\text{MPC}_{\text{direct}}$ fails to efficiently reduce $\hat{\mathcal{J}}$, there is no reason for $\text{MPC}_{\text{direct}}$ to efficiently reduce the fatigue cost $\mathcal{J}$ in closed-loop simulations, which is the ultimate objective.

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 **this discrepancy between the open-loop and closed-loop performance** is a rather important contribution, that deserves the attention of a broader audience than the one interested in wind turbine control.

In order to understand these poor closed-loop performances, the pitfalls of $\text{MPC}_{\text{direct}}$ must be analyzed. Therefore, in the next Subsection, two cases yielding good and bad performances are analyzed in order to identify the pitfalls of $\text{MPC}_{\text{direct}}$. Then, solutions allowing us to mitigate their influence in closed-loop simulations are proposed.

4.2.3 Pitfalls of $\text{MPC}_{\text{direct}}$

The major shortcoming of $\text{MPC}_{\text{direct}}$ in closed-loop simulation is that it does not allow us to efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$, compared to the FO-OCP open-loop optimizations performed over a long horizon. It was suggested in the previous subsections

that such behavior could be due to the fact that $\text{MPC}_{\text{direct}}$ is equivalent to a quadratic MPC, whose stage cost is parameterized by the current state and predicted disturbance trajectory. Indeed, this feature means the optimal control problem solved in the MPC could be significantly different from the one solved in the FO-OCP, which minimizes the fatigue-oriented cost $\hat{\mathcal{J}}$.

It is thus proposed to observe the evolution of the parameters given by the fixed-point solution which parameterizes the stage cost approximating the FO-OCP of $\text{MPC}_{\text{direct}}$ around the optimum. This is done for two cases, one where $\hat{\mathcal{J}}$ is efficiently reduced by the $\text{MPC}_{\text{direct}}$ and one where it is significantly increased.

The parameters given by the fixed-point problem can be summarized by four out of the six diagonal terms of the Q matrix, defined by (4.2). These diagonal terms are denoted by Q_{22}, Q_{33}, Q_{55} and Q_{66} in the following. They are plotted in Figures 4.4 and 4.5 as a function of the simulation time, for cases where $\hat{\mathcal{J}}$ is respectively reduced by 2 and increased by 10^7, compared to the FO-OCP open-loop optimization. From the Figures 4.4 and 4.5, the following observations can be made :

— In Figure 4.4, it can be seen that the stage cost parameters evolve smoothly around a value and never converge.
— In Figure 4.5, the stage cost parameters evolve also around a value until the 39^{th} second, where they start to converge around another value before coming back to the initial value around the 44^{th} second.
— The main difference between the two time series is the fact that the stage cost parameters converge around a single value in the case with good performance, while they switch around two points of convergence in the case with poor performance. These switches are likely to be related to convergence issues in the fixed-point algorithm.

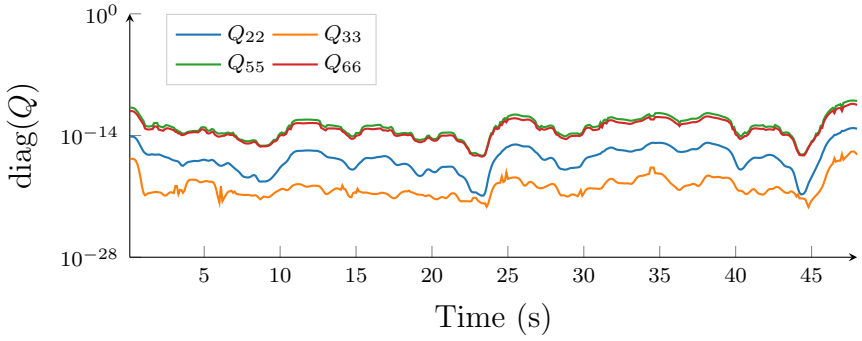

FIGURE 4.4 – Time evolution of the Q weighting matrix diagonal terms for $\text{MPC}_{\text{direct}}$, on a wind where the closed-loop yields good performance.

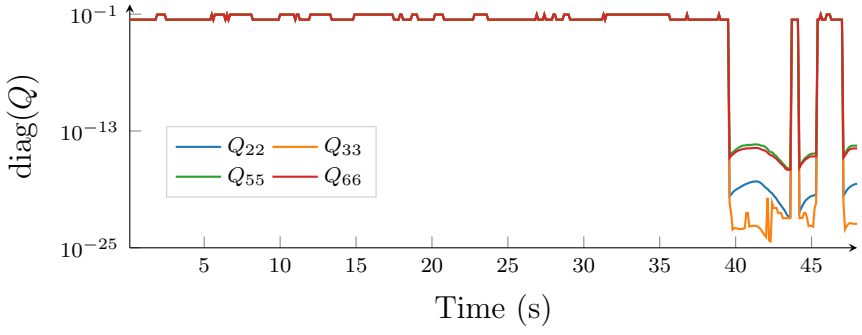

FIGURE 4.5 – Time evolution of the Q weighting matrix diagonal terms for $\text{MPC}_{\text{direct}}$, on a wind where the closed-loop yields bad performance.

This analysis suggests that the poor performances obtained in the presented case are related to convergence issues in the fixed-point algorithm, which induces gaps in the stage cost evolution observed in Figure 4.5. Indeed, it seems unlikely that abruptly switching from one set of stage cost parameters to another helps in reducing the fatigue cost of the closed-loop system.

The evolution of the stage cost parameters is not plotted for every closed-loop simulation with bad performance in this chapter, due to the lack of space. However, in-depth investigation shows that most poor performance cases featured this kinds of gap.

The issue is thus to understand why these gaps are showing up ? There are several possible reasons, which may be related to each other :

— The FO-OCP performed in $\text{MPC}_{\text{direct}}$ at each time step are completely independent from each other.
— The fixed-point solution which parameterizes the stage cost depends partly on the current state, which depends itself on the previous solutions found by $\text{MPC}_{\text{direct}}$. Therefore, there might be stability issues or several local solutions to the fixed-point method.
— The fatigue-oriented cost function $\hat{\mathcal{J}}$ was derived from signals which were regulated around steady-state. Therefore, the FO-OCP might have issues when only transient is observed on the prediction horizon. Moreover, variance is not really meaningful on signals with only transient.

The possible solutions to remedy these pitfalls are the following :

— Filter the stage cost parameters evolution in time, in order to alleviate their variations and smooth their time evolution. Moreover, this could help to compensate for the possible convergence issues.
— Perform the fixed-point problem over a longer horizon, free of transients.

In the next section, a second MPC formulation based on the FO-OCP is proposed, aiming at compensating for all the pitfalls stated above.

4.3 MPC implementation based on the fatigue-oriented optimization : a filtered formulation

This section presents a second MPC formulation based on the FO-OCP, aiming at mitigating the pitfalls of the first formulation. The new formulation allows us to mitigate a great part of these pitfalls and proves to be more efficient in reducing the fatigue-oriented cost function $\hat{\mathcal{J}}$ in closed-loop simulations.

Finally, the MPC presented in this section is compared to two benchmark parameterized quadratic MPCs, on its ability to efficiently reduce the fatigue cost $\mathcal{J}$.

The outline of this section is the following :

— In Subsection 4.3.1, the formulation of this second MPC is presented, using several filtering features in order to alleviate the variations of the stage cost parameters.
— In Subsection 4.3.2, closed-loop simulations using this second MPC formulation are compared to the FO-OCP open-loop optimizations, in order to verify that the pitfalls of the first MPC formulation are mitigated.
— In Subsection 4.3.3, the potential of this second MPC fatigue cost expectancy reduction is compared to the benchmark MPCs.

4.3.1 MPC formulation

It was observed in Subsection 4.2.3, that the bad closed-loop performances of $\text{MPC}_{\text{direct}}$ are probably related to convergence issues in the fixed-point algorithm, inducing high variations of the stage cost parameters. The solutions which could mitigate this pitfall are presented in this subsection.

It was suggested in the conclusion of Subsection 4.2.3 that possible solutions would be :

— Filtering the stage cost parameters in time.
— Increasing the prediction horizon's time length, in order to limit the presence of transients.

The first proposition can be easily implemented, but the second one, i.e. increasing significantly the prediction horizon, will result in increasing significantly the computational burden, which is not reasonable. By looking further into the fixed-point formulation described in Subsection 3.2.2, it can be noticed that the stage cost parameter is adapted as a function of the level of vibration of the system outputs. In the FO-OCP, a fixed point is used because the level of vibration is quantified on the optimization horizon.

In $\text{MPC}_{\text{direct}}$, the level of vibration is evaluated over the prediction horizon which might be too short for an accurate evaluation. However, it is not mandatory to evaluate the level of vibration over the prediction horizon only. Indeed, it is also possible to identify the level of vibration from the previous time instants of the closed-loop simulation. This level of vibration should be close to the one evaluated over the prediction horizon, provided that the wind conditions vary slowly and there are very few uncertainties in the MPC predictions.

Moreover, the trajectory $\tilde{\mathbf{y}}$ necessary for the derivation of the stage cost parameters, defined by (4.2), is directly obtained from the previous time instants of the simulation and is thus fixed. There is thus no inter-dependency between the stage cost parameters and $\tilde{\mathbf{y}}$ anymore, which breaks the fixed-point problem. Therefore, the computational burden can also be reduced as the MPC optimization problem becomes the one defined by (4.2).

Therefore, several solutions implemented together are proposed in order to limit the variations of the stage cost parameters, i.e. the weighting matrices Q and R defined by (4.2), while being able to adapt them efficiently :

— Estimate the variance J_{var} over the previous time instants of the simulations. The estimation of J_{var} is thus performed over a longer but delayed previous horizon, compared to the prediction horizon of only 2 seconds.
— Weight the variance estimation through time, in order to limit the influence of occurrences far away in time on the current instant and slightly reduce the delay in the estimation of J_{var}.
— Filter through time the variations of the updated matrices in order to limit the effects of noise or outliers on the weighting matrices estimation

A schematization of this MPC based on the FO-OCP, denoted by MPC_{filt}, is proposed in Figure 4.6. The standard MPC array of parameters p is obtained from the filtration of the non-filtered array of parameters, denoted by $\hat{p}$, given by the approximation of the FO-OCP defined by (3.5). The derivation of $\hat{p}$ depends on the system output trajectory over the previous time instants, denoted by $\tilde{\mathbf{y}}$. The various operations necessary for the derivation of the matrices Q and R, contained in p, are described in the remainder of this subsection.

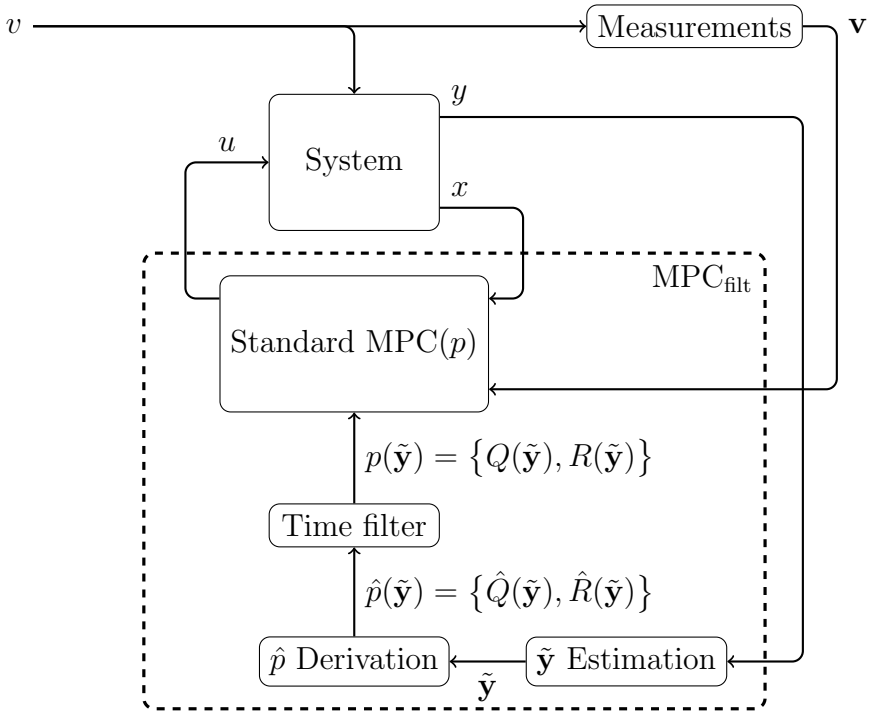

FIGURE 4.6 – Schematization of the MPC_{filt} architecture, where the vector of parameters p which parameterizes a standard MPC is a filtered function of the system output trajectory on the previous time instants.

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

From the fixed-point formulation detailed in Section 3.2.2, the non-filtered updated values of the weighting matrices Q and R, denoted by $\hat{Q}$ and $\hat{R}$, are expressed as follows :

$$\hat{Q}(\tilde{\mathbf{y}})_{ij} = \begin{cases} \pi_k e^{b_k} w_k \left(J_{\text{var}}(\tilde{\mathbf{y}}_k)\right)^{w_k - 1} & \text{if} \quad i = j = k \\ 0 & \text{otherwise} \end{cases} \tag{4.3a}$$

$$\hat{R}(\tilde{\mathbf{y}}) = \min\left(\text{diag}\left(\hat{Q}(\tilde{\mathbf{y}})\right), 1\right) \times 10^{-3} I_2 \tag{4.3b}$$

where $\hat{Q}(\tilde{\mathbf{y}})_{ij}$ is the i^{th} row and j^{th} column of $\hat{Q}(\tilde{\mathbf{y}})$, π_k, w_k and b_k are respectively the k^{th} component price of replacement, regression linear coefficient and intercept.

In order to limit the variations of $J_{\text{var}}(\tilde{\mathbf{y}}_k)$, the latter must be evaluated over a sufficiently long previous horizon. The trajectory $\tilde{\mathbf{y}}$ is thus taken to be the last time instants of the current closed-loop simulation output trajectory, instead of the prediction horizon. $\hat{Q}(\tilde{\mathbf{y}})$ corresponds to the interval $[t_0 - T_{\text{last}}, t_0]$, where t_0 is the current instant and T_{last} is the sliding horizon time length. T_{last} should be longer than the prediction horizon otherwise this method has no interest. It should be noted that the longer T_{last} is, the more J_{var} should be accurate, but the more time delay there is in the J_{var} estimation.

In order to limit the influence of the instants close to $t_0 - T_{\text{last}}$ on the J_{var} value and therefore alleviate the delay in the J_{var} estimation, a weighted formulation of the variance J_{var} is considered :

$$J_{\text{var}}(\tilde{\mathbf{y}}_k) = \frac{1}{T(\xi, T_{\text{last}})} \int_{t_0 - T_{\text{last}}}^{t_0} e^{\xi(\tau - t_0)} \tilde{y}_k^2(\tau) d\tau - \left(\frac{1}{T(\xi, T_{\text{last}})} \int_{t_0 - T_{\text{last}}}^{t_0} e^{\xi(\tau - t_0)} \tilde{y}_k(\tau) d\tau \right)^2 \tag{4.4}$$

where

$$T(\xi, T_{\text{last}}) = \int_{t_0 - T_{\text{last}}}^{t_0} e^{\xi(\tau - t_0)} d\tau$$

and $\xi \geq 0$ is the parameter of the weighting function, which monotonically increases with time. It should be noted that the greater ξ is, the more importance is given to the previous time instants of the interval $[t_0 - T_{\text{last}}, t_0]$. Note too that for the first time instants while $t_0 < T_{\text{last}}$, the sliding horizon is shortened and begins at the start of the simulation.

Filtering of $\hat{\mathbf{p}}$ in time

In order to limit once again the variations of the weighting matrices Q and R due to noise or outliers in $\tilde{\mathbf{y}}$, the weighting matrices are filtered through time :

$$Q(\tilde{\mathbf{y}}) = \beta Q_{\text{old}} + (1 - \beta)\hat{Q}(\tilde{\mathbf{y}}) \tag{4.5a}$$

$$R(\tilde{\mathbf{y}}) = \beta R_{\text{old}} + (1 - \beta)\hat{R}(\tilde{\mathbf{y}}) \tag{4.5b}$$

$$\tag{4.5c}$$

where Q_{old} and R_{old} are the values of Q and R obtained at their last update. It should be noted that for the first time instant, Q and R are initialized with the results of the fixed-point method, performed on the predicted disturbance $\mathbf{v}$ over the prediction horizon of 2 seconds long. A summary of the weighting matrices update for a given output trajectory $\tilde{\mathbf{y}}$ and old weighting matrices Q_{old} and R_{old} is given in Algorithm 2.

Algorithm 2: Algorithm summarizing the updating process of the weighting matrices Q and R.

Result: Gives Q and R values for a given output trajectory $\tilde{\mathbf{y}}$ and old weighting matrices Q_{old} and R_{old}.

Estimate the value of $J_{\text{var}}(\tilde{\mathbf{y}}_k)$ using equation (4.4);

Compute $\hat{Q}(\tilde{\mathbf{y}})$ and $\hat{R}(\tilde{\mathbf{y}})$ with equation (4.3);

Filter the update values through time to obtain $Q(\tilde{\mathbf{y}})$ and $R(\tilde{\mathbf{y}})$ with equation (4.5);

The values of the various parameters described in this subsection, i.e. T_{last}, β and ξ, are tuned by trial and error method in order to yield good results. The values used in the remainder of this chapter are $T_{\text{last}} = 20$ s, $\beta = 0.3$ and $\xi = 0.25$ s^{-1}. These values might not be optimal and further study on their influence on the closed loop should be conducted.

This second MPC formulation will be denoted by MPC_{filt} in the following. Now that MPC_{filt} is defined, it must be verified that MPC_{filt} allows us to efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$ compared to the FO-OCP and mitigate the pitfalls of $\text{MPC}_{\text{direct}}$.

4.3.2 Comparison of closed-loop simulations and open-loop optimizations

In this Subsection, a similar experiment to the one conducted in Subsection 4.2.2 is performed. Therefore, MPC_{filt} is simulated in closed-loop of the system, under a given wind disturbance 48 seconds long. Moreover, the FO-OCP is performed to solve an open-loop optimization problem under the same disturbance. The disturbances considered are the 1000 winds generated described in Subsection 4.1.2. The time series resulting from the

closed-loop simulations and open-loop optimizations are then evaluated with the fatigue-oriented cost $\hat{\mathcal{J}}$.

The aim of this experiment is to check whether or not MPC_{filt} in closed-loop with the system allows us to efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$. Therefore, the scatter of the fatigue-oriented cost $\hat{\mathcal{J}}$ given by the FO-OCP open-loop optimizations and the closed-loop simulations using MPC_{filt} is plotted in Figure 4.7. From the figure and the corresponding data, it can be observed that :

— The closed-loop simulations allow us to generally and efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$, compared to the open-loop FO-OCP.
— There are still samples with a strong mismatch between the fatigue-oriented cost $\hat{\mathcal{J}}$ of the FO-OCP and that of the closed-loop simulations.
— The closed-loop simulations increase $\hat{\mathcal{J}}$ compared to the FO-OCP in 83% of cases.
— The closed-loop simulations increase $\hat{\mathcal{J}}$ compared to the FO-OCP by more than 100% in 16% of cases.

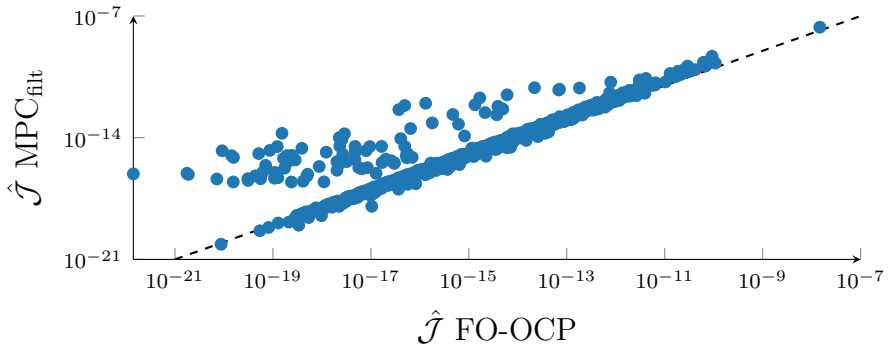

FIGURE 4.7 – Scatter plot of the fatigue-oriented cost $\hat{\mathcal{J}}$ of the closed-loop MPC_{filt} simulation against that of the FO-OCP, under the 1000 winds generated.

To summarize, in the majority of cases, MPC_{filt} succeeds in obtaining a fatigue-oriented cost $\hat{\mathcal{J}}$ in the same order of magnitude as that of the FO-OCP, which is very good. This leaves more hope for an efficient reduction of the fatigue cost $\mathcal{J}$ in closed-loop simulations.

However, it can be observed that there are still about 15% of cases where $\hat{\mathcal{J}}$ is significantly increased compared to that of the FO-OCP open-loop optimization. It is thus proposed to observe the evolution of the stage cost parameters during a closed-loop simulation with MPC_{filt}. In Figures 4.8 and 4.9, the stage cost parameter evolution is plotted for two cases where MPC_{filt} shows respectively good and poor performances. From these figures, the following observations can be made :

— For the good case, the stage cost parameters evolve very smoothly and adapt slowly their values with time.
— For the bad case, there are strong transients of the stage cost parameters, one from 0 to 5 seconds and another from 20 to 25 seconds. It can also be noticed that the stage cost parameters stabilize several times to different values.

To summarize these observations, the high variations of the stage cost parameters are again responsible for the poor closed-loop performances. There is an explanation to these bad performances, deduced from the above observations :

— The transients are due to a poor initialization of the stage cost parameters, which is performed by solving the fixed-point problem over the prediction horizon. Therefore, this issue might fade away with time for longer simulations.
— Nevertheless, the influence of time filtering on the stage cost parameters can be appreciated during these transients, where first order steps can be observed.

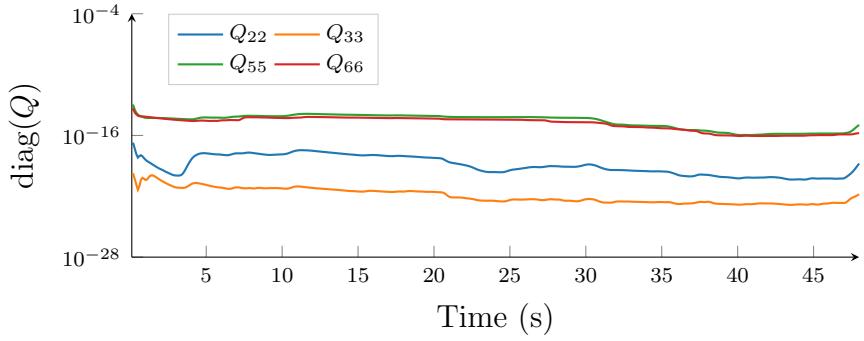

FIGURE 4.8 – Time evolution of the Q weighting matrix diagonal terms for MPC_{filt}, on a wind where the closed-loop yields good performance.

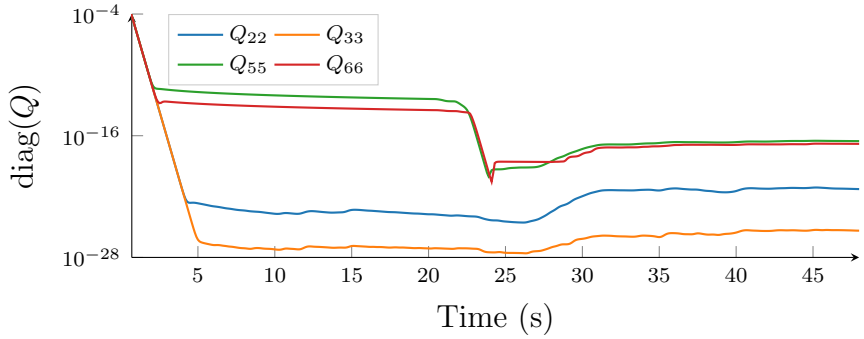

FIGURE 4.9 – Time evolution of the Q weighting matrix diagonal terms for MPC_{filt}, on a wind where the closed-loop yields poor performance.

— The 'stair' effect, where two transients delayed by 20 seconds are observed, is due to the fact that the stage cost parameters use the variance evaluated on the last 20 seconds. Therefore, while the first transient was still included in the last 20 seconds, the stage cost parameters converged on given values. Then, once the transient was not included anymore, a new transient started in order to converge on another value. It should be noted that this effect should also fade away with time.

To summarize, all the pitfalls of MPC_{filt} are due to a poor initialization of the stage cost parameters. However, these pitfalls should fade away with time. Therefore, in order to fully appreciate the potential of MPC_{filt}, longer simulations should be considered. Moreover, the first instants of the simulations should be removed as they may contain transients. However, it is considered that as the bad performance cases are already few enough, this data will be thus processed to evaluate the potential fatigue cost expectancy reduction of MPC_{filt} compared to benchmark controllers.

4.3.3 Potential fatigue cost reduction with MPC_{filt}

The ultimate goal behind the design of MPC_{filt} is to efficiently reduce the fatigue cost expectancy of the system in closed-loop simulations. This subsection presents the potential fatigue cost $\mathcal{J}$ expectancy reduction which would be achieved compared to two benchmark MPCs.

Therefore, the benchmark MPCs are first defined. Then, a comparison between closed-loop simulations using MPC_{filt} and the benchmark MPCs in terms of fatigue cost reduction is presented.

Definition of the benchmark MPCs

The MPCs chosen as benchmarks for the evaluation of the potential fatigue cost expectancy reduction that MPC_{filt} could provide are quadratic MPCs, analogous to the open-loop optimizations with fixed and adaptive parameters, defined by (3.13). The weighting matrices of these quadratic MPCs stage costs are parameterized by ν, following (2.22) :

$$Q(\nu) = \text{diag}\left([0, 1, 1, 0, \nu, \nu]\right) \tag{4.6a}$$
$$R(\nu) = \text{diag}\left([0, 1, 1)\min(\nu, 1) \times 10^{-10}\right. \tag{4.6b}$$

Let $\mathcal{J}_{\text{MPC}}(\mathbf{v}, \nu)$ be the fatigue cost obtained for a **closed-loop simulation** with a quadratic MPC parameterized by ν, under a given wind disturbance $\mathbf{v}$. The two benchmark MPCs differ from each other by the ν parameter used. The first benchmark MPC, denoted by MPC_{best}, uses the parameter $\nu_{\text{best,CL}}$, which minimizes the fatigue cost expectancy, corresponding to $\bar{p}^\star$, defined by (1.51). The second benchmark MPC, denoted by $\text{MPC}_{\text{adapt}}$, uses the parameter $\nu_{\text{adapt,CL}}(\mathbf{v})$, which minimizes the fatigue cost for a given wind disturbance on the closed-loop simulation, corresponding to $p^\star(\mathbf{v})$, defined by (1.52). It should be noted that the parameters $\nu_{\text{best,CL}}$ and $\nu_{\text{adapt,CL}}$ are also respectively the closed-loop counterparts of the parameters ν_{adapt} and ν_{best}, defined by (3.13). The parameters $\nu_{\text{best,CL}}$ and $\nu_{\text{adapt,CL}}$ are defined as follows :

$$\nu_{\text{best,CL}} = \arg\min_{\nu} \sum_{i=1}^{N_{\text{data}}} \mathcal{J}_{\text{MPC}}(\mathbf{v}_i, \nu) \tag{4.7a}$$
$$\nu_{\text{adapt,CL}}(\mathbf{v}) = \arg\min_{\nu} \mathcal{J}_{\text{MPC}}(\mathbf{v}, \nu) \tag{4.7b}$$

Therefore, MPC_{best} is a classical parameterized quadratic MPC whose parameter is optimized and easily implementable. On the other hand, $\text{MPC}_{\text{adapt}}$ is an adaptive quadratic MPC which needs the disturbance information on the whole closed-loop simulation to adapt its parameter properly. The latter MPC is thus not implementable but is considered in the comparison as an ideal controller, giving some of the lowest fatigue costs that MPC_{filt} could reach.

Comparison of the MPCs on fatigue cost

The ultimate objective of the MPCs designed in this chapter is to efficiently reduce the fatigue cost $\mathcal{J}$ in closed-loop simulations. Therefore, a comparison of the fatigue cost $\mathcal{J}$ obtained with the closed-loop simulations of the above presented MPCs under the 1000 winds generated is proposed. A summary of the MPCs used in the comparison is given in Table 4.2. Note that in Table 4.2, $\text{MPC}_{\text{direct}}$ is missing, as it does not allow us to efficiently reduce the fatigue-oriented cost $\hat{\mathcal{J}}$. Therefore $\text{MPC}_{\text{direct}}$ was not considered in this comparison on fatigue costs.

Figures 4.10 and 4.11 plot the scatter of the fatigue costs of MPC_{filt} against those of respectively $\text{MPC}_{\text{adapt}}$ and MPC_{best}. The following observations can be made :

- In 97% of cases, MPC_{filt} reduces the fatigue cost more than MPC_{best}.
- In 13% of cases, MPC_{filt} reduces the fatigue cost more than $\text{MPC}_{\text{adapt}}$.
- In terms of fatigue cost expectancy reduction, MPC_{filt} allows us to have a fatigue cost expectancy 30% lower than MPC_{best}, while $\text{MPC}_{\text{adapt}}$ allows us to reach a fatigue cost expectancy 52% lower than MPC_{best}.

MPC name	Description
MPC_{filt}	MPC based on the fatigue-oriented cost function $\hat{\mathcal{J}}$, where the stage cost parameter variations are alleviated by several filtering features. (see Subsection 4.3.2)
MPC_{best}	Parameterized quadratic MPC whose ν parameter minimizes the fatigue cost expectancy. (see Subsection 4.3.3)
$\text{MPC}_{\text{adapt}}$	Parameterized quadratic MPC whose ν parameter minimizes the fatigue cost for a given wind. (see Subsection 4.3.3)

TABLE 4.2 – Table summarizing the different MPCs used in the following comparison.

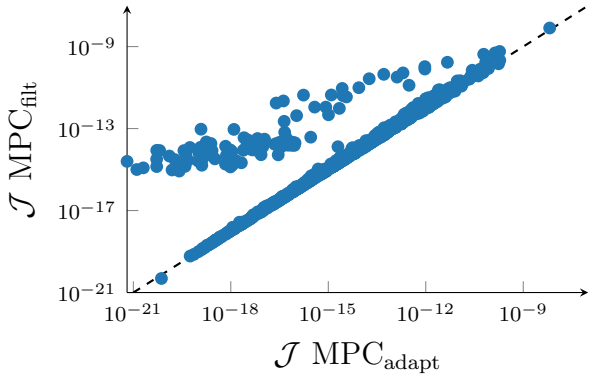

FIGURE 4.10 – Scatter plot of the closed-loop fatigue costs $\mathcal{J}$ of MPC_{filt} against those of $\text{MPC}_{\text{adapt}}$, for the 1000 winds generated.

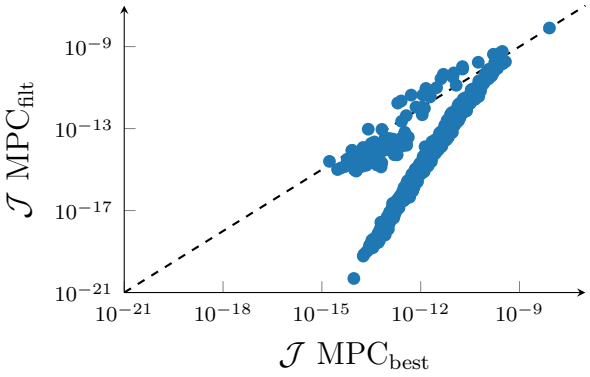

FIGURE 4.11 – Scatter plot of the closed-loop fatigue costs $\mathcal{J}$ of MPC_{filt} against those of MPC_{best}, for the 1000 winds generated.

Summary

To summarize these results, MPC_{filt} is very efficient in reducing the fatigue cost expectancy and the fatigue cost in general. It can even match the performances of $\text{MPC}_{\text{adapt}}$ in many cases, which is a fictitious ideal controller, allowing us to achieve outstanding fatigue cost reduction. Moreover, the performances of $\text{MPC}_{\text{adapt}}$ and MPC_{filt} compared to that of MPC_{best} show that quadratic MPCs whose stage cost parameters are efficiently adapted allow us to tremendously reduce the fatigue cost expectancy of a system, compared to a quadratic MPC with fixed stage cost parameters.

It should be noted that there are still some closed-loop simulations where the fatigue cost is significantly increased compared to $\text{MPC}_{\text{adapt}}$. These simulations suffer from poor initializations of the stage cost parameters, which induce transients in the stage cost parameters and closed-loop simulations, eventually producing additional fatigue costs. These simulations could be thus considered as outliers and it can be expected that by considering longer simulations, these transients would shade away, as well as the gaps with the $\text{MPC}_{\text{adapt}}$ fatigue cost observed in Figure 4.10.

Finally, it should be remembered that simplified simulation settings were considered in this section. Therefore, these promising results obtained with MPC_{filt} should be received with caution. Indeed the simulation settings considered :

— An LTI system without model mismatch.
— A wind only summarized by its hub-height wind speed.
— No uncertainties on the disturbance prediction nor the state estimation.

In order to fully estimate the potential of this approach in realistic conditions, the following

improvements must be made :

- Simulate the closed-loop on a realistic LPV or an aero-elastic HAWT simulator system
- Test the system under a wind defined by a full wind vector as the one defined in Subsection 2.1.3 or a full field disturbance
- Consider uncertainties on the wind prediction or the state estimation
- Simulate the closed-loop on a longer time scale

It should be noted that the above improvements could affect the performance of MPC_{filt}, as well as MPC_{best}.

4.4 Discussion

This chapter presented two MPCs based on the fatigue-oriented cost function $\hat{\mathcal{J}}$:

- The first MPC, denoted $\text{MPC}_{\text{direct}}$, directly uses the FO-OCP in its optimization problem.
- The second MPC, denoted MPC_{filt}, uses a standard quadratic MPC formulation, where the stage cost parameters are adapted online, based on previous information and the structure of the fatigue-oriented cost function $\hat{\mathcal{J}}$.

It was shown in Subsection 4.2.1 that both MPCs could be seen as MPCs minimizing the integral of a time varying/adaptive quadratic stage cost over time. However, the difference in the adaptation of these MPCs is that MPC_{filt} contains filtering features which allow us to regulate the stage cost parameter variations, while $\text{MPC}_{\text{direct}}$ adapts its stage cost independently of the other time instants.

The fatigue-oriented MPCs were first evaluated on their ability to reduce their objective function, i.e. the fatigue-oriented cost function $\hat{\mathcal{J}}$. It was observed that $\text{MPC}_{\text{direct}}$ could not properly reduce $\hat{\mathcal{J}}$ in closed-loop simulations. However, its analysis enabled us to derive a better solution as shown in this chapter.

Then, MPC_{filt} was evaluated on its ability to reduce the fatigue cost expectancy given a realistic and comprehensive distribution of winds. Moreover, it was compared to benchmark quadratic MPCs with a fixed and an adaptive stage cost, optimized in order to minimize the fatigue cost reduction expectancy. In this comparison, MPC_{filt} proves to be particularly efficient compared to the MPC with a fixed stage cost.

Main results

The main results of this chapter are the following :

- $\text{MPC}_{\text{adapt}}$ which is a quadratic MPC whose stage cost is efficiently adapted, denoted by $\text{MPC}_{\text{adapt}}$, can allow a **reduction of the fatigue cost expectancy of 52%**, compared to MPC_{best}, under the presented simulation settings. MPC_{best} is a non-adaptive MPC whose stage cost parameters are optimized in order to minimize the fatigue cost expectancy.
- The excessive stage cost parameter variations with time are responsible for the poor closed-loop performance of the fatigue-oriented MPCs. Therefore, $\text{MPC}_{\text{direct}}$ fails to reduce the fatigue-oriented cost $\hat{\mathcal{J}}$ in closed-loop, which is its objective function.
- The formulation of MPC_{filt} allows us to alleviate the stage cost parameter variations, which permits a **reduction of the fatigue cost expectancy of 30%** compared to MPC_{best}, in the presented simulation settings.

The results of this chapter are thus strongly encouraging for the development of MPCs with an adaptive stage cost, which is reminiscent of the MPCs developed in [68], described respectively in Subsection 1.3.3 and 1.3.3, where :

— In [62], the stage cost parameters of an MPC are updated based on an online identification of fatigue from a quadratic cost function, which has similarities with MPC_{filt}.

— In [68], the cost function parameter is adapted based on the rainflow counting algorithm performed iteratively on the prediction horizon. This strategy has more similarities to $\text{MPC}_{\text{direct}}$ than MPC_{filt}, where the cost function is adjusted only from the prediction horizon information.

Similarities between MPC_{filt} and the MPC presented in [62]

Many similarities could be found between MPC_{filt} and the MPC presented in [62], where its stage cost parameters were adjusted based on an online identification of a fatigue cost function. The idea is indeed globally the same, but the difference in methods is clarified below :

— In [62], the stage cost parameters are updated based on an online system identification. It needs thus to continuously estimate the fatigue damage on-line. Moreover, with this implementation, the stage cost parameters might end up converging and yield an optimal stage cost with fixed parameters.

— MPC_{filt} is quite different, as the derivation of a fatigue-oriented cost function, which is assumed to be true for the whole life of the system, is performed offline. The stage cost parameters are then **updated based on the structure of the fatigue-oriented cost function** and the system information over several previous seconds only. Therefore, the stage cost parameters should not converge to a fixed value and always be able to adapt the stage cost to various wind conditions.

Nevertheless, it should be very interesting to compare these methods in several examples, in order to observe which one performs better and if the other MPCs have the same drawbacks as $\text{MPC}_{\text{direct}}$.

Conclusion

To conclude this chapter, it confirms once again that there is a **strong interest in adapting one set of standard controller parameters to turbine and wind conditions in order to efficiently optimize the system fatigue cost**. MPC_{filt} could be an efficient solution for the fatigue trade-off optimization of a HAWT. However, there are still several remaining design steps before implementing MPC_{filt} on a realistic HAWT aero-elastic simulator. Moreover, there might be many issues to solve when uncertainties are present in wind prediction, state estimation or model dynamics.

All these uncertainties and mathematical complexities could discourage one turbine designer from using such control strategies. The tuning of a standard IPC controller for fatigue mitigation appears as an insurmountable problem, as there is no universal controller allowing us to efficiently minimize fatigue for every wind condition using a standard control strategy.

Moreover, most of the results presented in this thesis suggest that the parameters of one controller should be adapted to the wind conditions. This also suggests that the parameters of one controller which minimizes the fatigue cost for a given wind $\mathbf{v}$ depends on $\mathbf{v}$.

Therefore, it is proposed in Chapter 5 to identify the function $p^\star(\mathbf{v})$, which gives the parameters of the controller minimizing the fatigue cost under the wind $\mathbf{v}$. The data-driven identification of this function allows us to select online the parameters of controller candidates for an efficient reduction of the fatigue cost $\mathcal{J}$.

Chapitre 5

Data-driven on-line controller selection with a supervisory layer

It was highlighted through Chapters 2, 3 and 4 that :

- Designing a universal controller which minimizes the fatigue cost $\mathcal{J}$ is highly challenging.
- No universal controller allowing us to efficiently reduce a HAWT fatigue cost for a large variety of wind conditions has been designed yet.
- Standard control strategies need to adapt their parameters depending on wind conditions, which are not defined by the instantaneous hub-height wind speed alone.
- Adapting efficiently the parameter vector p parameterizing a standard control strategy with respect to wind conditions could allow us to reduce significantly the fatigue cost expectancy, compared to a controller optimized in order to minimize the fatigue cost expectancy with fixed parameters.
- The parameter vector p of a controller which allows us to minimize the fatigue cost for a given wind should depend on the current wind conditions.

On the other hand, a study of the wind spectrum [87] allowed us to observe that the spectrum is divided into several peaks :

- The turbulent peak which accounts for variations of a period of approximately one minute, due to eddies formed by the roughness of the complex terrain nearby.
- The diurnal peak, which accounts for variations of a period of approximately 12 hours, mainly due to the temperature variations between day and night.
- The synoptic peak, which accounts for variations of a period of roughly $2-3$ days, due to changes in atmospheric conditions and the movements of air mass on the Earth's surface.

Moreover, it was shown in [88] that on a single wind site, the turbine inflow wind can be clustered into several classes of winds, using horizontal wind speed and additional atmospheric conditions (e.g. temperature stratification) throughout the year. These considerations are going far beyond control theory, but the influence of meteorological and atmospheric conditions on a HAWT is significant.

The insight developed in this chapter is to **design a supervisory layer** featuring a surrogate model, which allows us to **select a controller from several candidates based on wind conditions, in order to minimize the fatigue cost** $\mathcal{J}$. The surrogate model is then used for **selecting online the candidate controller which would yield the lowest fatigue cost** for the current wind conditions. This supervisory layer consists thus in approximating the function $p^\star(\mathbf{v})$, defined by (1.52), which yields the parameters of a controller allowing us to minimize the fatigue cost given the current wind conditions, using a data-driven solution.

The controllers forming the set of candidates are designed using **classical control theory** and **tuned with parameter vector showing good performance**, in order to regulate the variations due to the turbulent peak, i.e. turbulence in the boundary layer. Selection of the controller is thus based on wind features, or even atmospheric conditions, relative to diurnal and synoptic peak variations.

For an efficient selection of the controllers, a mapping of the fatigue cost $\mathcal{J}$ as a function of wind conditions and candidate controllers is needed. In order to obtain such a mapping of the fatigue cost $\mathcal{J}$, the various candidate controllers must be simulated off-line in closed loop with the system, under various wind conditions. Then, the fatigue cost of each simulation is evaluated with the fatigue cost function $\mathcal{J}$. Ultimately, there are two possibilities for the derivation of the data-driven surrogate model :

- It can be a regression fitted to the fatigue cost mapping, then the predictions allow

us to select the candidate controller yielding the lowest predicted fatigue cost.
— Similar to classification approaches, it can be a regression of the probability density function that a candidate controller minimizes the fatigue cost, based on the fatigue cost mapping $\mathcal{J}$.

The difficulty thus shifts from the control design to the surrogate model design. Several works [89], [90], [91], [92] and [93] have looked into designing data-driven surrogate models relating wind and fatigue loads for a given controller. The scopes of these works were to monitor fatigue or assess HAWT lifetime, but controller selection was not mentioned. Nevertheless, these works have shown that it is possible to relate wind and fatigue of a HAWT with a surrogate model.

The innovation proposed in this chapter is to use the predictions made by a data-driven surrogate model to select the candidate controller which is the most likely to yield the lowest fatigue cost, under the current wind conditions. The features defining the current wind conditions can be any characteristic of the wind signal or spectrum, which could be estimated using any sensor or estimation strategy.

It should be stressed out that the work presented in this chapter is quite exploratory. It consists in a few guidelines for the supervisory layer design, based on the preliminary observations and investigations conducted so far. Moreover, a first proof of concept is given in order to show the interest of such a method. Finally, remaining interrogations, prospects for future developments and improvements are detailed, in order to give a start to anyone wishing to explore further these kinds of approaches in the future.

The outline of this chapter is the following :
— In Section 5.1, an overview of the methodology with its main design steps is given.
— In Section 5.2, a proof of concept highlighting the preliminary results obtained and the potential fatigue cost expectancy reduction is presented.
— In Section 5.3, solutions to improve the preliminary results and the methodology are proposed. Moreover, a discussion of the implementation issues of this methodology on a commercial wind turbine is given, along with possible solutions to overcome these issues.
— In Section 5.4, the interest, shortcomings and possible improvements of this methodology are discussed.

5.1 Methodology description

As a reminder, the ultimate goal of all the controllers presented in this thesis is to minimize the fatigue cost $\mathcal{J}$, defined by (1.50). In order to decide online which controller is the most suitable in a finite list of candidates, denoted by $\mathcal{K}_{\text{list}}$, a mapping of the fatigue cost from wind conditions and candidate controller must be derived. It should be noted that for given wind conditions, many wind scenarios are possible due to the wind's stochastic nature. Therefore, from a few metrics representing the wind conditions, the fatigue cost could take different values as well and has also a stochastic nature.

In order to avoid the computationally-expensive prediction of all possible fatigue costs from a set of wind conditions using physical laws, a data-driven surrogate model can be derived. For this derivation, machine learning techniques appear to be promising solutions, allowing us to predict a fatigue cost or a probability density function from current wind conditions, in an acceptable computational time [94].

Once this surrogate model is designed, a 'Selector' component will select the controller which is the most likely to reduce the fatigue cost from among the candidates, for current wind conditions. This selection is based on the predictions performed by the surrogate model and the list of candidate controllers.

This supervisory layer approach is thus composed of two main components which are :

— The surrogate model, allowing us to predict from wind conditions the fatigue cost of every candidate controllers, or the probability density that a candidate controller minimizes the fatigue cost. Guidelines for its derivation are described in Subsection 5.1.1.

— The 'Selector', which based on the surrogate model predictions, selects and switches the controller the most likely to reduce the fatigue cost. The challenge in Selector design is to switch the candidate in closed-loop of the system, while mitigating the undesirable transient effects which could occur during the switch. A possible design of the 'Selector' function is given in Subsection 5.1.2.

5.1.1 Surrogate model derivation guidelines

In order to derive a data-driven surrogate model allowing us to predict the fatigue cost value of the candidate controllers, or the probability that a candidate controller minimizes the fatigue cost for given wind conditions, a mapping of the fatigue cost $\mathcal{J}$ as a function of wind conditions and candidates is needed.

In order to generate the surrogate model, the following procedure, where the mapping is generated using off-line simulations, is proposed :

1. **Control design** A set of N_{cand} candidate controllers defined by their parameter vector p, denoted by :
$$\mathcal{K}_{\text{list}} = \{p_1, ..., p_{N_{\text{cand}}}\}$$
must be designed, where p_j is the j^{th} candidate controller parameter vector. Each controller in $\mathcal{K}_{\text{list}}$ must ensure the appropriate regulation of the closed-loop HAWT and show good performance in reducing fatigue cost for at least one wind condition. It should be noted that each parameter vector p_j might not only be composed of gains parameterizing a given control strategy, but can also contain the control strategy associated, i.e. PI, MPC, and so on.
2. **Wind generation** A comprehensive set of N_{wind} wind time series must be generated for various conditions, corresponding to the wind distribution of the wind site where the turbine might be installed.
3. **Closed-loop simulations** Closed-loop simulations of the HAWT with each candidate controller are run under the previously-generated set of winds. The point of the data generation process is to gather triplets of wind time series, HAWT output time series and candidate controller.
4. **Fatigue cost estimation** The HAWT output trajectories are evaluated using the fatigue cost function $\mathcal{J}$, in order to yield the fatigue cost mapping, denoted by $\mathcal{Y}$. The fatigue cost mapping is defined such that $\mathcal{Y}(\mathbf{v}_i, p_j)$ is the fatigue cost of the HAWT in closed-loop with the controller parameterized by $p_j \in \mathcal{K}_{\text{list}}$, under the wind time series $\mathbf{v}_i$.
5. **Features extraction** Wind features are extracted from the wind time series generated, denoted by $\mathbf{v}$, yielding the wind features column vector, denoted by $X(\mathbf{v})$ which summarizes the wind time series into several characteristics. These wind

feature vectors are rearranged to form the wind feature space, denoted by $\mathcal{X} = \{X(\mathbf{v}_1), ..., X(\mathbf{v}_{N_{\text{wind}}})\}$.

6. **Surrogate model derivation** The surrogate model, denoted by f_{sm}, must be designed to predict :
   — Either the fatigue cost of the HAWT in closed-loop with a controller parameterized by $p_j \in \mathcal{K}_{\text{list}}$, under the wind time series $\mathbf{v}_i$, in case of regression.
   — Or the probability that the controller parameterized by $p_j \in \mathcal{K}_{\text{list}}$ in closed-loop with the HAWT minimizes the fatigue cost $\mathcal{J}$ for the set of controllers $\mathcal{K}_{\text{list}}$, under the wind time series $\mathbf{v}_i$, in case of classification.

   The inputs of the surrogate model are thus the vector of wind features $X(\mathbf{v}_i)$ and the candidate controller parameterized by p_j.

The surrogate model can be fitted using the previously generated mapping. The surrogate model can either approximate $\mathcal{Y}$ as follows, in case of regression, and be denoted by $f_{\text{sm}}^{\text{reg}}$:

$$f_{\text{sm}}^{\text{reg}}(X(\mathbf{v}_i), p_j) \simeq \mathcal{Y}(\mathbf{v}_i, p_j) \tag{5.1}$$

or approximates the probability that $\mathcal{Y}(\mathbf{v}_i, p_j)$ is lower than or equal to $\mathcal{Y}(\mathbf{v}_i, p_l)\ \forall p_l \in \mathcal{K}_{\text{list}}$, in case of classification, and denoted by $f_{\text{sm}}^{\text{clf}}$:

$$f_{\text{sm}}^{\text{clf}}(X(\mathbf{v}_i), p_j) \simeq \mathbb{P}\Big[\mathcal{Y}(\mathbf{v}_i, p_j) \leq \mathcal{Y}(\mathbf{v}_i, p_l)\Big] \quad \forall p_l \in \mathcal{K}_{\text{list}} \tag{5.2}$$

It should be noted that data could also be generated online by running the turbine in closed-loop with various candidate controllers under fluctuating wind conditions. This would allow us to explore the set of candidates and refine the surrogate model needed for the controller selection as new data is coming in. This approach is close to reinforcement learning and could be indeed useful to refine the surrogate model online with field data, once the turbine is running. However, it might be risky to start with no prior at all on the surrogate model.

Moreover, it should be noted that using a surrogate model predicting the probability that a candidate minimizes the fatigue cost cannot learn from data gathered online. Indeed, when one controller is tested, only the fatigue cost associated to the current wind conditions and candidate controller is known. Therefore, it cannot be determined how the candidate performs compared to the other candidates under the tested wind conditions. Hence, the probability that a candidate minimizes fatigue cost under given wind conditions cannot be estimated for the regression. Nevertheless, the mapping generation from off-line simulations allows us to obtain a convenient prior surrogate model in order to run the turbine.

In a wind farm, due to the site topology and turbine wake interactions, the wind conditions experienced by the turbines might be significantly different. This means that the data gathered to synthesize the surrogate model could also be significantly different. Therefore, in a wind farm, the surrogate model is likely to be turbine/site dependent.

Online implementation

In order to better understand how the surrogate model is to be used online, a schematization is proposed in Figure 5.1. In order to obtain the best candidate controller in $\mathcal{K}_{\text{list}}$, the wind information is processed as follows :

1. The vector of wind features X is estimated from a wind time series $\mathbf{v}$, using any sensor available such as LiDAR, wind vane, or estimation techniques such as Wind Field Reconstruction (WFR) algorithms.
2. The surrogate model f_{sm} can predict the fatigue cost $\mathcal{J}$ with $f_{\text{sm}}^{\text{reg}}$ or the probability that a controller minimizes the fatigue cost with $f_{\text{sm}}^{\text{clf}}$, for every candidate controller in $\mathcal{K}_{\text{list}}$ under the vector of wind features X.
3. The candidate controller the most likely to reduce efficiently the fatigue cost, parameterized by $p^\star \in \mathcal{K}_{\text{list}}$, is selected by the 'Selector' block of the supervisory layer and then implemented in closed-loop with the HAWT.

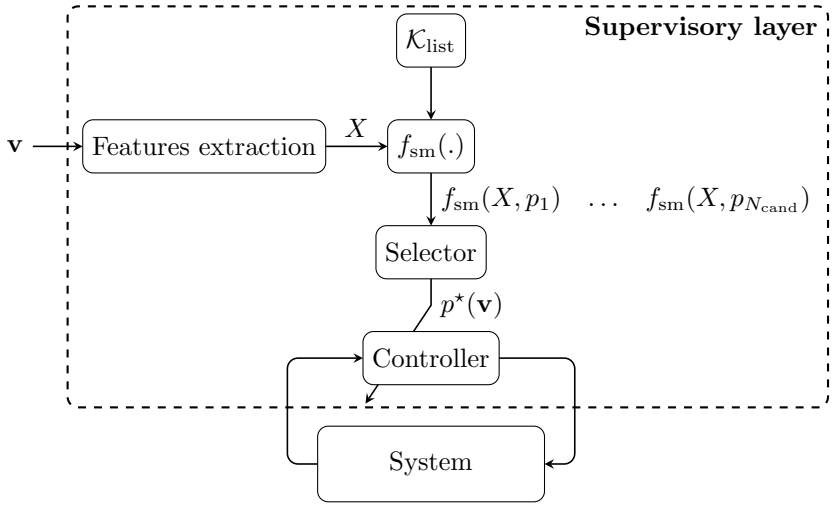

FIGURE 5.1 – Schematization of the supervisory layer online implementation with its various components, on a HAWT.

The three main blocks constituting the supervisory layer approach are thus :

— The features extraction block, which yields the vector of wind features X necessary for the surrogate model fatigue cost predictions.
— The surrogate model block, using f_{sm} to predict the fatigue cost or probability that a controller minimizes the fatigue cost from the vector of wind features X.
— The 'Selector' block which selects the controller parameterized by $p^\star$, which is the most likely to yield the lowest fatigue cost $\mathcal{J}$ among the candidates in $\mathcal{K}_{\text{list}}$.

Coupling between feature extraction and surrogate model regression

It can be noticed that, the feature extraction and the surrogate model blocks are intimately related, because the features strongly influence the regression and quality of a data-driven model. Moreover, feature extraction is an entire field of machine learning and the feature extraction block could be merged with the surrogate model block, depending on the technique employed. Nevertheless, in this chapter the features extraction block is considered to be limited to the information provided by WFR algorithms [5] or state estimation [9].

Data-driven surrogate model derivation

The data-driven surrogate model derivation is the most important part of this supervisory layer approach, as the quality of the Supervisor selection directly depends on the surrogate model predictions' accuracy. Note that it is not mandatory to directly predict the fatigue cost of the candidate controllers in case of regression. Indeed, having an idea

of the relative fatigue cost between candidates might be sufficient for controller selection.

Therefore, as the fatigue cost $\mathcal{J}$ is likely to follow a Rayleigh distribution, it might be preferable to apply a transformation allowing us to approximate its distribution as Gaussian. Having a Gaussian-like distribution could be more suitable than a Rayleigh distribution for classical regression strategies using a mean square error loss function.

Concerning the machine learning techniques to be employed, any available one could be used as long as it allows us to properly select the controllers. However, the surrogate model and hyperparameters must be chosen appropriately in order to be able to at least interpolate correctly the results. A classical issue in statistics and machine learning, called overfitting, is that a model can yield great results on the data it was fitted on, but very poor results on other data. This is why the data must always be split into three sets :

— A training set to fit the model
— A validation set to tune the model hyperparameters
— A test set to eventually evaluate the model accuracy

Moreover, several rules and tips can be found in machine learning or statistic books in order to avoid overfitting. Some of them are :

— Have a number of weights to fit in accordance with the amount of data available
— Apply regularizations on the model weights
— Preprocess the input features of the model appropriately
— Perform cross-validation for hyperparameter tuning

For model selection, there are other criteria than model accuracy which can be taken into account. For example, the ability of the model to keep learning online, which could become important in order to keep refining the surrogate model once new field data are coming in. For instance, neural network structures allowing us to easily learn online when new samples are coming in.

Once the surrogate model is designed, the Selector can be designed. Possible design options of the Selector are presented in the next Subsection.

5.1.2 Possible design settings of the 'Selector' function

The Selector function must meet the following objectives :

— Select appropriately the candidate controllers in order to efficiently reduce the fatigue cost function $\mathcal{J}$, based on the surrogate model predictions.
— Appropriately switch the controller in closed-loop, without excessively increasing the fatigue cost.

This subsection, describing possible Selector design settings, first presents strategies which can be used, in order to efficiently select the candidate controllers based on the surrogate model predictions. Secondly, a possible switching strategy is described.

Controller selection strategies

Fatigue cost prediction The surrogate model $f_{\text{sm}}^{\text{reg}}$ allows us to predict a deterministic fatigue cost for every candidate controller and a given vector of wind features. The surrogate model prediction for the vector of wind features X and the j^{th} candidate is denoted by $f_{\text{sm}}^{\text{reg}}(X, p_j)$. The straightforward strategy is to select the candidate whose fatigue cost prediction is the lowest :

$$p^{\star}(X, f_{\text{sm}}^{\text{reg}}, \mathcal{K}_{\text{list}}) = \underset{p_j \in \mathcal{K}_{\text{list}}}{\operatorname{argmin}} f_{\text{sm}}^{\text{reg}}(X, p_j) \tag{5.3}$$

Probability prediction The surrogate model $f_{\text{sm}}^{\text{clf}}$ allows us to predict the probability that a candidate controller minimizes the fatigue cost given a vector of wind features. In this case, the straightforward strategy is to select the candidate which maximizes the probability of minimizing the fatigue cost :

$$p^{\star}(X, f_{\text{sm}}^{\text{clf}}, \mathcal{K}_{\text{list}}) = \underset{p_j \in \mathcal{K}_{\text{list}}}{\text{argmax}} f_{\text{sm}}^{\text{clf}}(X, p_j) \tag{5.4}$$

It should be noted that these approaches are greedy and do not take into consideration that switching controller can add an unexpected fatigue cost or prediction uncertainty. Therefore, it might be possible to robustify this selection law by including such considerations. Such prospects should be explored in future works.

Duality between probability prediction and classification should be noted that if a surrogate model predicts the probability that a candidate minimizes the fatigue cost for given wind conditions, the selection task is exactly the same as the one of a classifier. Indeed, classifiers often fit the regression of log-likelihood function representing the probability that a sample is in a class. Therefore, the likelihood predictions of classifiers can be used for probability predictions, provided that the classifier was fitted in order to classify which controller minimizes the fatigue cost for given wind conditions.

Once the Selector is defined, it must avoid that the transition from one controller to another induces a considerable additional fatigue cost, which would cancel all the gains that could have been obtained. A relevant switching strategy implemented by the Selector can allow us to alleviate the undesirable transition effects on the closed loop.

Discussion on switching strategies

The problem discussed here can be stated as 'how to mitigate the transient on the system closed-loop, while switching at the time instant $t = t_{\text{switch}}$ from a controller parameterized by p_1 to another controller parameterized by p_2'. The most possible cause of transients when a switch between controller occurs is a discontinuity on the inputs of the HAWT or the controller internal state. An example of switching strategy which avoids such discontinuities consists in using a parameter α which evolves smoothly and weighs the outputs of p_1 and p_2 to ensure a smooth transition.

Let u_1 and u_2 be the outputs of the candidate controllers parameterized by p_1 and p_2 respectively. The output of the supervisory layer, which is also the control input of the HAWT denoted by u, is expressed as follows :

$$u(t) = \alpha(t)u_1(t) + (1 - \alpha(t))u_2(t) \tag{5.5}$$

where $\alpha(t)$ is defined as a filter of order n_{filt} with a step as input :

$$\alpha(t) = \mathcal{L}^{-1}\left(\frac{1}{(1+\tau s)^{n_{\text{filt}}}}\frac{1}{s}e^{-st_{\text{switch}}}\right) \tag{5.6}$$

where $\mathcal{L}$ is the Laplace transform, $s \in \mathbb{C}$ is the Laplace variable and $\tau > 0$ is the time constant of the i^{th} pole of the filter. This method provides that all the $n_{\text{filt}} - 1$ first derivatives of the switched controller output are continuous, which is expected to mitigate the transient effects in the input. Moreover, thanks to the integrator and the fact that the poles of the filter are all real negatives, α is monotonically increasing from 0 to 1.

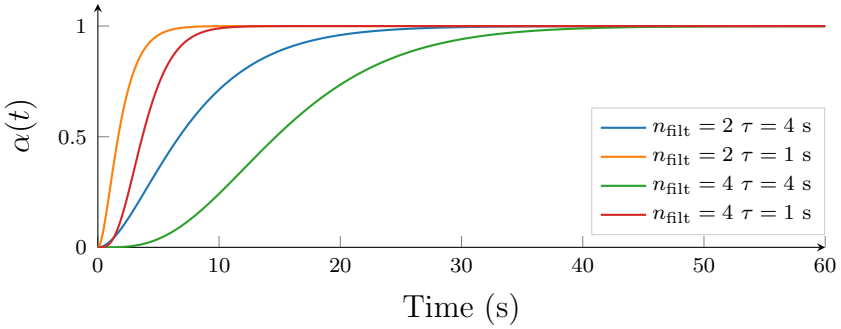

FIGURE 5.2 – Evolution of α with time for various parameters τ and n_{filt}.

Figure 5.2 plots the evolution of the α parameter for several combinations of the parameters n_{filt} and τ.

If α varies slowly compared to the closed-loop dynamics of each system, then α dynamic is negligible compared to the one of the closed-loop system. On the other hand, a slowly varying α is equivalent to using an infinity of candidate controllers in between p_1 and p_2 during the transient phase, which are not taken into account in the surrogate model predictions and whose performances are not taken into account in the supervisory layer.

These candidate controllers used during the transition could thus yield unexpected closed-loop behaviour and induce unexpected additional fatigue cost. Moreover, controllers featuring internal integrator states are initialized to 0, as no prior on their value is available, and need time to converge, i.e. the dynamics of α must be sufficiently slow to allow convergence. Therefore, the parameters of the filter which yields the α value must be optimized in order to minimize the additional fatigue cost during transients.

Besides, it should be noted that the above switching strategy requires running two controllers in parallel, in order to have the outputs of u_1 and u_2 at each time step. Therefore, this switching strategy would be more suited to control strategies needing a relatively low evaluation time, in order to be real time implementable.

The switching strategies implemented in the Selector function are not limited to the one presented above. The designer of the supervisory layer can design any switching strategies, as long as they efficiently mitigate the transient effects on the closed-loop system.

Summary

The point of this section was to introduce the four main blocks of the supervisory layer approach, which can be designed to achieve wind turbine control system expert requirements :

1. The set of candidate controllers which could efficiently reduce the fatigue cost.
2. The surrogate model which allows us to predict the fatigue cost $\mathcal{J}$, or the probability that a candidate controller minimizes the fatigue cost under given wind conditions, for all candidate controllers.
3. The controller selection strategy of the Selector function, which must select a controller likely to minimize the fatigue cost for given wind conditions, based on the surrogate model predictions.

4. The Selector switching strategy, which must mitigate the undesirable transient effects during switch between controllers.

In the next section, a first example using the supervisory layer approach described in this section is presented, which also constitutes a first proof of concept.

5.2 Preliminary results : a first proof of concept

In this section, a first application of the supervisory layer approach is presented. It should be noted that the candidate controllers, features extracted from wind time series, surrogate model, selection law and switching strategies presented in the sequel are not unique. These are only blocks of the framework, that the user can fill with any suitable method which can allow us to obtain the expected behaviour. There is thus a wide range of combination possibilities and finding the optimal one is outside the scope of this study.

The outline of this section follows the design steps of the framework presented in Section 5.1 :

— In Subsection 5.2.1, the simulation settings used for the data generation process and the proof of concept is described.
— In Subsection 5.2.2, the cost function evaluating the HAWT fatigue cost $\mathcal{J}$ is defined.
— In Subsection 5.2.3, the set of candidate controllers design is detailed.
— In Subsection 5.2.4, the wind features extracted from wind time series are presented.
— In Subsection 5.2.5, the surrogate model derivation is described
— In Subsection 5.2.6, the preliminary results obtained with this first application of the supervisory layer approach are presented.

5.2.1 Simulation settings

In this chapter, the simulation settings considered are much more realistic than those used in the previous chapters :

— The model is an aero-elastic HAWT simulator with all degrees of freedom relative to on-shore HAWT activated. The model corresponds to a Senvion MM82 HAWT, whose technical characteristics are summarized in Table 5.1 and simulated with the NREL aero-elastic HAWT simulator FAST [34].
— The winds considered are full-field turbulent winds, generated with the wind generator TurbSim [73]. The winds are generated such that their characteristics respect the wind distribution of the NWTC wind farm, based on the measurement campaign defined in Subsection 2.3.1.

Characteristics	Value
Rated Power (kW)	2050
Cut-in wind speed (m/s)	3.5
Rated wind speed (m/s)	14.5
Cut-off wind speed (m/s)	25
Rotor diameter (m)	82
Rotor speed (rpm)	8.5 — 17.1

TABLE 5.1 – Summary of the MM82 HAWT characteristics.

TurbSim wind generator allowing us to generate a three dimensional velocity wind field which evolves in the rotor plane and time, called full-field wind. The winds generated are supposed to model coherent turbulences in space and time, based on computational fluid dynamic simulation results. One thousand winds are generated for various combinations of the following parameters :

— The seed number used for the generation of random numbers in the TurbSim algorithm, which is a random integer.
— The mean wind speed, which is randomly drawn in a wind speed distribution obtained from the measurement campaign described in [83].
— The power law exponent, denoted by α_{PL}, which is a number accounting for the vertical variations of the wind speed. Let $V(z)$ be the wind speed at altitude z and z_{bottom} and z_{top} be respectively the altitudes of the top and bottom of the rotor. The power law exponent α_{PL} is defined as follows

$$\frac{V(z_{\mathrm{top}})}{V(z_{\mathrm{bottom}})} = \left(\frac{z_{\mathrm{top}}}{z_{\mathrm{bottom}}}\right)^{\alpha_{\mathrm{PL}}}$$

This value is randomly drawn in a power law exponent distribution, obtained from the measurement campaign described in [83].
— The turbulence intensity, which is also randomly drawn in a distribution obtained from the measurement campaign described in [83].

The HAWT aero-elastic model is simulated over 600 second-long time horizons, for a sampling time of 0.0125 seconds. It can be noticed that this kind of simulation is also used for controller validation or HAWT lifetime estimation. This simulation's settings are thus very close to reality compared to the other considered in the previous chapters, and much more demanding for the turbine.

In the next subsection, the definition of the fatigue cost function $\mathcal{J}$, which is also much more realistic than the one used in the previous chapters of this thesis, is given.

5.2.2 Fatigue cost function definition

In order to give an idea of the economic costs of fatigue damage in the various components of an HAWT, a fatigue cost function $\mathcal{J}$ is defined. So far, the fatigue cost function used only considered the fatigue damage related to the rotor of the HAWT. The fatigue cost function defined in this subsection considers fatigue damage related to various components of the HAWT, such as :

— The blade roots
— The blade pitch actuators
— The tower root
— The blade bearings
— The rotor shaft
— The gearbox

The fatigue damage of each component is estimated from the load time series obtained from the relevant output of the system. To get an economic estimation of the HAWT fatigue cost during a simulation, the prices of replacement of the considered components, denoted by π, are estimated as specified in [95], with additional assumptions on transportation and installation costs. It is assumed that the transportation and installation costs of each component are ratios of the entire turbine cost, and that the larger and heavier the components, the larger their ratio. A summary of the outputs used for each component and its price of replacement is given in Table 5.2.

Damage estimation without ultimate load

Nevertheless, the access to the ultimate load L^{ult} information, which is the load for which the component breaks instantly, defined in Subsection 1.3.1, is not accessible for all these components. Therefore a trick described below is used in order to have a relatively good estimation of the fatigue damage without the ultimate load information.

In page 24 of Subsection 1.3.1, it was mentioned that estimating the DEL value of a load time series does not need the ultimate load if the Goodman correction is neglected. It should be noted that the fatigue damage is nonlinearly related to the DEL value with equation (1.31) :

$$\mathcal{D}_k(\mathbf{y}_k) \propto (\text{DEL}_k(\mathbf{y}_k))^{m_k} \tag{5.7}$$

where DEL_k and m_k are respectively the DEL and Wöhler exponent of the k^{th} component. The current issue is thus to find the appropriate proportional coefficient between $\mathcal{D}_k$ and $(\text{DEL}_k)^{m_k}$.

In order to find the linear coefficient relating $\mathcal{D}_k$ and $(\text{DEL}_k)^{m_k}$, it is proposed to make a lifetime assessment of the HAWT in closed-loop with a baseline controller under a comprehensive set of winds. It is possible to assess the lifetime DEL of every component without Goodman correction, which does not need the ultimate load information. However, it is not possible to assess the fatigue damage using fatigue theory in this case. Nevertheless, by making assumptions on the components' lifetime, it is possible to associate an average damage rate to the DEL values obtained and derive the proportional coefficients between the $\mathcal{D}_k$ and $(\text{DEL}_k)^{m_k}$.

Let $\text{DEL}_k^{\text{base}}$ and $\dot{\mathcal{D}}_k^{\text{base}} = \frac{1}{\text{LT}_k}$ be respectively the baseline DEL and damage rate of the k^{th} component obtained from the lifetime assessment simulations, where LT_k is a guessed value of the k^{tm} component lifetime. A summary of the lifetimes used for every component can be found in Table 5.2. The fatigue damage $\mathcal{D}_k$ can be thus estimated as follows :

$$\mathcal{D}_k(\mathbf{y}_k) = \underbrace{\frac{\dot{\mathcal{D}}_k^{\text{base}} T_{\text{sim}}}{(\text{DEL}_k^{\text{base}})^{m_k}}}_{\text{constant}} (\text{DEL}_k(\mathbf{y}_k))^{m_k} \tag{5.8}$$

where T_{sim} is the time length of the simulation, multiplied by $\dot{\mathcal{D}}_k^{\text{base}}$ such that $\mathcal{D}_k$ becomes a damage instead of a damage rate.

k	Component	Corresponding output	π_k (k-euros)	LT_k (years)
1,2,3	Blades	Blade root bending moments	103	20
4,5,6	Blade pitch actuator	Blade root torsional moments	31	40
7	Tower	Fore-aft tower root bending moments	398	20
8,9,10	Blade bearings	Blade root bending moments	33	30
11	Rotor shaft	Rotor shaft bending moment	35	30
12	Gearbox	Rotor torque	316	20

TABLE 5.2 – Summary of the components considered in the fatigue cost function, with their corresponding system outputs used for their estimation, price of replacement and lifetime assumption.

It should be noted that the baseline controllers used for this lifetime assessment are :
— A PI controller gain-scheduled on blade pitch angle, described in [37], for CPC regulation.

— A static map for torque regulation.

The fatigue cost function $\mathcal{J}$ is thus defined as a weighted sum of the price of replacements and fatigue damages :

$$\mathcal{J}(\mathbf{y}) = \sum_{k=1}^{N_c=12} \pi_k \mathcal{D}_k(\mathbf{y}_k) \tag{5.9}$$

It should be stressed that this fatigue cost function does not claim to be the most realistic one, as some of the lifetimes guessed might not be realistic. 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. Therefore, the turbine owners / manufacturers can implement this method with their own data in order to assess the potential fatigue cost reduction that they could obtain.

Discussion on the inclusion of energy production in $\mathcal{J}$

It should be noted that there are no terms accounting for the quality of the CPC and torque regulation in $\mathcal{J}$. It would have been possible to add a term accounting for the energy produced, by multiplying the price of electricity with the energy produced during a simulation. However, the price of wind energy electricity on the electrical network is highly varying, depending on the demand and the energy produced by the other renewable energies on the network, which makes this information very uncertain.

Moreover, the objective of CPC regulation is to correctly regulate the power output to the HAWT's nominal power when wind speed is high. Therefore, rewarding the quantity of energy produced could encourage overshoots on the power output and deteriorate the quality of the CPC regulation. Deteriorating the CPC regulation has also an economic cost, as it could deteriorate the quality of the electrical network signal. The cost relative to the energy production is thus very complex to estimate and its estimation is outside the scope of this thesis.

Therefore, due to actuator limitations and the fact that IPC regulation allows us to ensure that the blade pitch angle average is equal to the collective blade pitch angle given by the CPC, it is assumed that the impact of IPC on the CPC regulation is negligible. It is a heavy assumption indeed, but to my best knowledge it is not possible to relevantly quantify the economical impact of a deteriorated CPC regulation yet. Therefore, they cannot be compared to the gain in fatigue cost obtained with the approaches described in this thesis. Nevertheless, it would be interesting to obtain a method allowing us to quantify the economic effects of the CPC and torque regulations, in order to include them in the fatigue cost function $\mathcal{J}$ and generalize the supervisory layer approach to CPC and torque regulations.

Therefore, the supervisory layer approach is limited in this proof of concept to the selection of IPC candidate controllers and fatigue cost reduction. The design of the various candidate controllers used in this section is described in the following subsection.

5.2.3 Candidate controllers design

In this application, IPC controllers in addition to a CPC controller are considered (Figure 5.3) as candidates in $\mathcal{K}_{\text{list}}$. The CPC controller corresponds to the PI controller, gain scheduled on blade pitch angle, designed as specified in [37]. This controller was used in the HAWT lifetime assessment necessary to the derivation of the fatigue cost function

$\mathcal{J}$, described in Subsection 5.2.2.

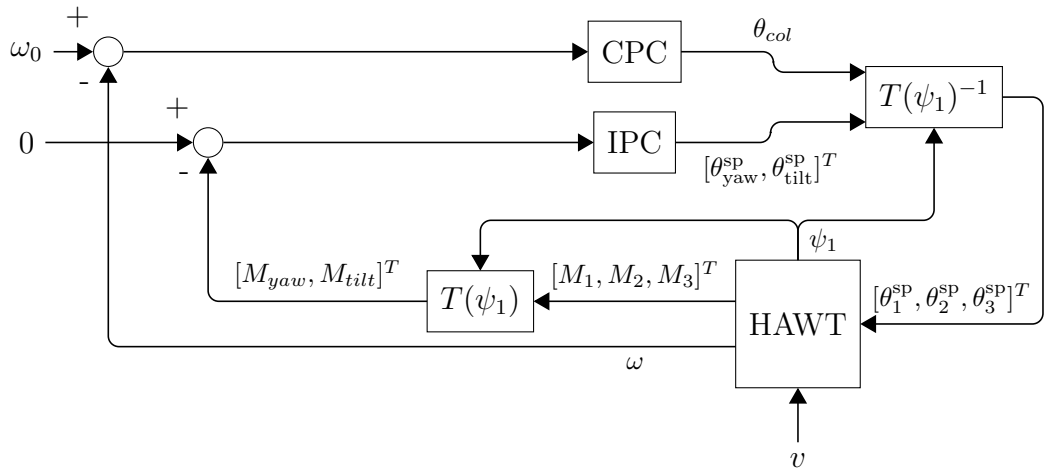

FIGURE 5.3 – Wind turbine blade pitch control system scheme, with independent IPC and CPC controllers. ω refers to the rotational speed and ω_0 to its set point. θ_{col} is the collective pitch angle $T(\psi)$ is the MBC transform defined by (1.1).

The IPC controllers considered for this proof of concept are double SISO PI controllers, similar to those defined by (1.6) :

$$[\theta^{\text{sp}}_{\text{yaw}}(t), \theta^{\text{sp}}_{\text{tilt}}(t)]^T = -K_{P,j}[M_{\text{yaw}}(t), M_{\text{tilt}}(t)]^T - \int_{t_0}^{t} K_{I,j}[M_{\text{yaw}}(\tau), M_{\text{tilt}}(\tau)]^T d\tau \qquad (5.10)$$

where $K_{P,j}$ and $K_{I,j}$ are the proportional and integral coefficient of the j^{th} PI controller in the set of candidates, and the parameter vector $p_j = \{K_{P,j}, K_{I,j}\}$. The values of the parameters $K_{P,j}$ and $K_{I,l}$ are thus varied between candidates. A summary of the four parameter vectors used in this proof of concept is given in Table 5.3. It should be noted that the 4^{th} candidate is the baseline CPC controller, without IPC regulation.

j	$K_{P,j}$	$K_{I,j}$
1	0.0186	0.0037
2	0.0186	0.0066
3	0.0186	0.0095
4	0	0

TABLE 5.3 – Summary of the $K_{P,j}$ and $K_{I,j}$ parameters used for the generation of the PI candidate controllers in $\mathcal{K}_{\text{list}}$.

The candidate controllers designed in this proof of concept might not optimally alleviate fatigue but were designed in order to yield convenient closed-loop behaviours. It would therefore be interesting to test this approach with more advanced controllers, in order to assess the performance which could be obtained. Considering the potential fatigue reduction which could be achieved with several MPCs, highlighted in the study of Subsection 3.3.4, there might be a great fatigue cost reduction potential for the supervisory layer approach compared to single MPC controllers.

Then, the candidate controllers described in this subsection can be simulated in closed loop with the HAWT under the simulation settings described in Subsection 5.2.1. The

next and last step of the fatigue cost mapping generation is to extract features of the wind time series which can help in predicting the fatigue cost of the closed-loop HAWT.

5.2.4 Wind features extraction

In order to predict accurately the fatigue cost $\mathcal{J}$ for given wind conditions, it is of primal importance to extract features from the wind which are likely to explain wind turbine fatigue cost $\mathcal{J}$. It was mentioned in Subsection 5.1.1 that feature extraction is an entire field of machine learning and thus the feature extraction block is limited to the information on wind field, which could be provided by WFR algorithms or state estimation.

The wind characteristics which can be provided by the WFR algorithms or state estimation, considered for fatigue cost prediction in this section are the following ones :

— The Rotor Averaged Wind Speed (RAWS), defined as the average of the wind speed over the rotor plane, defined by (5.11a).
— The horizontal (respectively vertical) wind shear, denoted by δ_y (respectively δ_z), which is the average of the horizontal (respectively vertical) wind speed variations over the rotor plane, defined by (5.11b) and (5.11c) respectively.
— The wind direction in the vertical (respectively horizontal) plane, denoted by θ_y (respectively θ_z), which is the wind direction in the vertical (respectively horizontal) plane, averaged over the rotor plane, defined by (5.11d) and (5.11e) respectively.
— The Rotor Averaged Turbulence Intensity (RATI), which is the turbulence intensity evaluated over a given time horizon, averaged over the rotor plane, defined by (5.11f).

To define mathematically these quantities let :

$$\overrightarrow{V}(t,y,z) = [u_V(t,y,z), v_V(t,y,z), w_V(t,y,z)]^T$$

be the three dimensional velocity vectors at time instant t, horizontal position y and vertical position z of the rotor plane. Let $V(t,y,z)$ be the Euclidean norm of $\overrightarrow{V}(t,y,z)$. The mathematical expressions of the wind features are the following :

$$\text{RAWS}(t) = \frac{1}{S}\int_S V ds \tag{5.11a}$$

$$\delta_y(t) = \frac{1}{S}\int_S \frac{\partial V}{\partial y} ds \tag{5.11b}$$

$$\delta_z(t) = \frac{1}{S}\int_S \frac{\partial V}{\partial z} ds \tag{5.11c}$$

$$\theta_y(t) = \frac{1}{S}\int_S \arctan\left(\frac{w_V}{u_V}\right) ds \tag{5.11d}$$

$$\theta_z(t) = \frac{1}{S}\int_S \arctan\left(\frac{v_V}{u_V}\right) ds \tag{5.11e}$$

$$\text{RATI}(t_0,t_f) = \frac{1}{S}\int_S \frac{\int_{t_0}^{t_f} V^2 dt - \left(\int_{t_0}^{t_f} V dt\right)^2}{\int_{t_0}^{t_f} V dt} ds \tag{5.11f}$$

where S is the rotor area, $ds = dydz$ is an infinitesimal area of the rotor, t_0 and t_f are respectively the initial and final time instants of the time horizon considered for turbulence intensity estimations in the RATI. It should be noted that all of these features except the

RATI are time varying. In order to give consistency to the features considered, only the average and standard deviation over $[t_0, t_f]$ of the time varying features are considered instead of the time series of the time varying features.

This eventually yields 11 features, which are defined over a time horizon $[t_0, t_f]$. For the 1000 simulations of 600 seconds are considered three intervals of 100 seconds starting at 300, 400 and 500 seconds, for both the wind features and fatigue cost evaluation. It should be noted that the 300 first seconds are ignored in order to avoid considering transients in the data, which could bias the surrogate model.

Now that the wind features are defined, it is time to derive the surrogate model which will allow us to select the candidate controller most likely to minimize the fatigue cost $\mathcal{J}$ under given wind conditions.

5.2.5 Surrogate model derivation

For the surrogate model derivation, prediction of the probability that a candidate minimizes a fatigue cost is preferred, i.e. classification, as the generated mapping $\mathcal{Y}$ allows this possibility. Several classification strategies have been tested, but only the one yielding the best performance is presented in this section.

The surrogate model eventually used is decomposed into several processing steps :

1. A standardization of the data, in order to ensure that all the features of the dataset have a zero mean and standard deviation equals to unity.
2. A Principal Component Analysis (PCA) allowing us to extract the linear combinations of features allowing us to explain the largest amount of variance.
3. A Gaussian process classifier, whose goal is to estimate the likelihood that a candidate minimizes the fatigue cost $\mathcal{J}$, for given wind conditions.

In the remainder of this subsection, the design of these various processing steps is described.

Standardization

Let X be the vector of wind features given by the feature extraction block of the supervisory layer approach (see Figure 5.1). The standardization consists in transforming the data such that the average and standard deviation of the X values contained in the dataset are equal to 0 and 1 respectively. This first standardization step allows all the features of X to have the same variance, which avoids that the magnitude of a feature becomes predominant relative to others in the regression.

Let μ_X and σ_X be respectively the average and standard deviation of the feature space $\mathcal{X} = \{X(\mathbf{v}_1), \ldots, X(\mathbf{v}_{N_{\text{wind}}})\}$. The standardized value of X, denoted by X_{std}, is obtained as follows :

$$X_{\text{std}}(\mathbf{v}) = f_{\text{std}}\left(X(\mathbf{v})\right) = \frac{X(\mathbf{v}) - \mu_X}{\sigma_X} \tag{5.12}$$

Let $\mathcal{X}_{\text{std}} = \{X_{\text{std}}(\mathbf{v}_1), \ldots, X_{\text{std}}(\mathbf{v}_{N_{\text{wind}}})$ be the input space $\mathcal{X}$ transformed by the standardization process.

Another issue which could prevent the good derivation of the classifier is the presence of redundant features. Therefore, a PCA can allow us to extract the meaningful features and mitigate this kind of issue.

Principal Component Analysis

A PCA allows us to extract the orthogonal axis of a space which explains the most variance with fewer components. The projection axes are obtained through a singular value decomposition of the input space. This singular value decomposition allows us to obtain :

— An orthonormal matrix P, which allows us to project the vectors expressed in the input space on the orthogonal principal components space.
— A square matrix S, which describes the variance of each principal component of the principal components space.

Therefore, after a PCA on $\mathcal{X}_{\text{std}}$, X_{std} can be projected on the principal components of $\mathcal{X}_{\text{std}}$ as follows :

$$X_{\text{pca}}(\mathbf{v}) = f_{\text{pca}}\left(X_{\text{std}}(\mathbf{v})\right) = PX_{\text{std}}(\mathbf{v}) \tag{5.13}$$

Let $\mathcal{X}_{\text{pca}} = \{X_{\text{pca}}(\mathbf{v}_1), \ldots, X_{\text{pca}}(\mathbf{v})\}$ be the dataset $\mathcal{X}_{\text{std}}$ projected on its principal components.

Moreover, the cumulative variance of the principal components of the space $\mathcal{X}_{\text{std}}$ can be analyzed in Figure 5.4. It can be seen that 6 components out of 11 are sufficient to explain the variance of the space $\mathcal{X}_{\text{std}}$. This means that the 5 last components are explaining less than 5% of the variance of $\mathcal{X}_{\text{std}}$. Therefore, the new orthogonal space $\mathcal{X}_{\text{pca}}$ can be limited to 6 components.

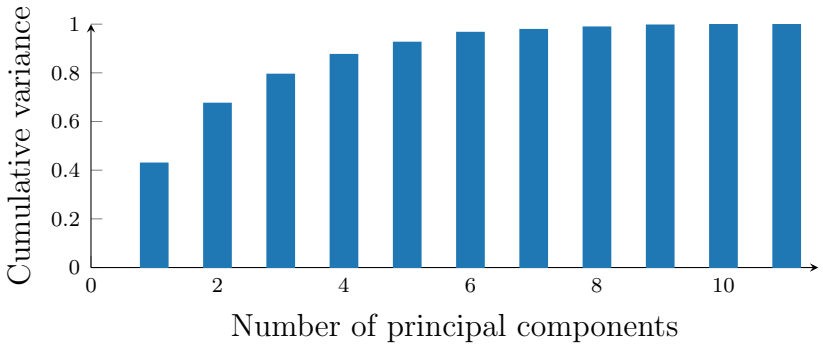

FIGURE 5.4 – Bar plot of the cumulative variance of the principal components obtained from the principal component analysis of the standardized space of wind features $\mathcal{X}_{\text{std}}$.

This dimensionality reduction allows us to have a problem which is likely to be better conditioned for the fit of a regression or a classification model.

Gaussian process classifier

There exists a vast collection of tools for classification tasks in machine learning, which allow us to predict the probability that a sample is in a class. Finding the right classification model is often a matter of trial and error, i.e. we need to test the model in order to find which model performs better for a given dataset.

In order to fit and test a machine learning model, the dataset is split into a training and a testing set of 2250 and 750 samples respectively. Several classification models were tested in the trial and error process, such as logistic regression, random forests or support vector machine, but the one which yields the best results so far is the Gaussian process classifier. The interested reader can find more information on the models mentioned above

in [96].

The essential information on Gaussian processes that the reader needs to understand in this section is that, for regression :

— A Gaussian process allows us to obtain a probabilistic distribution of the function f which can describe the process observed with the data.
— A Gaussian process is parameterized by a kernel used for the estimation of its covariance function, which is optimized with a specific algorithm detailed in [97], using cross-validation.
— A Gaussian process allows us not only to predict a deterministic value that a process would take for a given sample, but also its variance and thus the uncertainty of the prediction.

For classification, the Gaussian process performs actually the regression of a 'latent function' $f(x)$, which is squashed through the logit function :

$$\text{logit}\left(f(x)\right) = \frac{1}{1+e^{-f(x)}} \tag{5.14}$$

in order to obtain an estimation of the probability that the sample x is in a class. Therefore the Gaussian processes must fit one latent function by class. The interested reader can find more information on Gaussian processes in [97].

In our specific case, the Gaussian process classifier needs to classify the candidate controller which minimizes the fatigue cost for a given vector of wind features. A function f_{gp} is thus obtained, such as $f_{\text{gp}}(X_{\text{pca}}(\mathbf{v}), p_j)$ yields the probability that the candidate controller parameterized by p_j minimizes the fatigue cost for $X_{\text{pca}}(\mathbf{v})$. The surrogate model f_{sm} is thus derived from the following composition of functions :

$$f_{\text{sm}}^{\text{clf}}(X(\mathbf{v}), p_j) = f_{\text{gp}}\left(f_{\text{pca}} \circ f_{\text{std}} \circ X(\mathbf{v}), p_j\right) \tag{5.15}$$

It should be noted that for classification, the selection law defined by (5.4) is eventually used with the predictions of the surrogate model $f_{\text{sm}}^{\text{clf}}$.

Hyperparameters tuning

It should be noted that the resulting surrogate model has two hyperparameters to tune :

— The number of components kept after the PCA.
— The kernel used for the covariance function estimation in the Gaussian process classifier.

The optimization of the kernel and its parameters used in the Gaussian process classification is implemented automatically in the `sklearn` module under `Python`, using the algorithm described in [97]. Therefore, the remaining hyperparameter to tune is the number of components extracted in the PCA.

It can be observed in Figure 5.4, that 6 components allowed us to explain more than 95% of the space in input of the classifier. However, this does not tell us whether 6 components are appropriate or not for an efficient classification of the surrogate model with the selection law defined by (5.4). Therefore, the influence of the number of components is studied under two metrics, the accuracy and the potential fatigue reduction, denoted respectively by R_{acc} and R_{fat}.

Accuracy R_{acc} is a very classical metric for classification. It computes the fraction of correct predictions :

$$R_{\text{acc}}(f_{\text{sm}}^{\text{clf}}, \mathcal{X}, \mathcal{Y}) = \frac{1}{\text{card}(\mathbf{X})} \sum_{i=1}^{\text{card}(\mathcal{X})} \mathbf{1}\left(p^\star(X(\mathbf{v}_i), f_{\text{sm}}^{\text{clf}}, \mathcal{K}_{\text{list}}) = \underset{p_j \in \mathcal{K}_{\text{list}}}{\text{argmin}} \mathcal{Y}(X(\mathbf{v}_i), p_j) \right) \tag{5.16a}$$

$$\mathbf{1}(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{5.16b}$$

where card() is a function that gives the number of elements in a set.

The potential fatigue reduction metric R_{fat} consists in estimating the fatigue reduction which could be obtained by using the predictions of the surrogate model f_{sm} and the selection law defined by (5.4), compared to the candidate minimizing the fatigue cost expectancy individually :

$$R_{\text{fat}}(f_{\text{sm}}^{\text{clf}}, \mathcal{X}, \mathcal{Y}) = 1 - \frac{\sum_{i=1}^{\text{card}(\mathcal{X})} \mathcal{Y}\left(X(\mathbf{v}_i), p^\star\left(X(\mathbf{v}_i), f_{\text{sm}}^{\text{clf}}, \mathcal{K}_{\text{list}}\right)\right)}{\min_{p_j \in \mathcal{K}_{\text{list}}} \sum_{i=1}^{\text{card}(\mathcal{X})} \mathcal{Y}\left(X(\mathbf{w}_i), p_j\right)} \tag{5.17}$$

Figures 5.5 and 5.6 plot the evolution of respectively the metrics R_{acc} and R_{fat} with the number of components extracted in the PCA of $f_{\text{sm}}^{\text{clf}}$. When the surrogate model $f_{\text{sm}}^{\text{clf}}$ is trained on the training set and the metrics are evaluated on the testing set. It can be seen that both metrics stop improving above 5 components. Therefore, by keeping a 'simpler is better' rule, it is chosen to keep 5 components after the PCA in $f_{\text{sm}}^{\text{clf}}$.

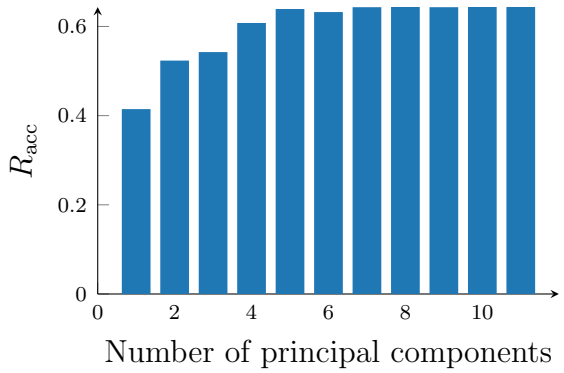

FIGURE 5.5 – Evolution of the metric R_{acc} evaluated on the test set with the number of components kept after the PCA.

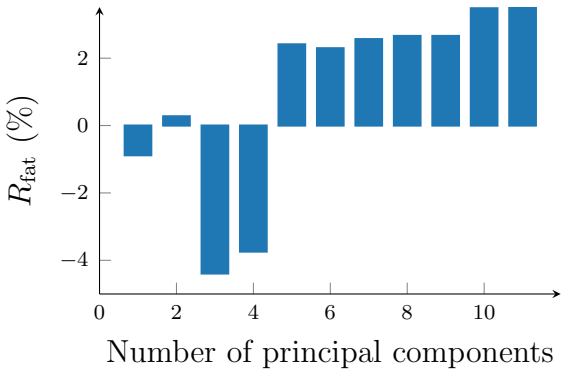

FIGURE 5.6 – Evolution of the metric R_{fat} evaluated on the test set with the number of components kept after the PCA.

The optimized kernel parameters, obtained when 5 components are extracted from the PCA, are radius basis function kernels with a length scale of 1.

Now that the surrogate model $f_{\text{sm}}^{\text{clf}}$ is fitted and its hyperparameters are optimized, it is possible to simulate the turbine in closed-loop with the controllers selected by the classifier, provided that an efficient switching strategy is implemented.

5.2.6 Preliminary results

This subsection presents the preliminary results obtained with the supervisory layer approach, which uses the surrogate model described in the previous subsection. The two important aspects to evaluate are :

— The ability of the surrogate model with the selection law to select the correct controller and above all, the ability of reducing the fatigue cost $\mathcal{J}$ expectancy.
— The ability of the switching strategy to mitigate the additional fatigue costs due to transients when a controller switches in closed-loop simulations.

Fatigue cost expectancy reduction

The selection law used in this proof of concept is the greedy one defined by (5.4), which selects the candidate controller maximizing the probability to minimize the fatigue cost, based on the surrogate model $f_{\text{sm}}^{\text{clf}}$ predictions. It can be first assumed that the potential additional fatigue cost due to controller switch transients is negligible. Therefore, the potential fatigue cost reduction which could be obtained with the surrogate model $f_{\text{sm}}^{\text{clf}}$ and the greedy selection law is equivalent to the R_{fat} score.

In order to assess the quality of the selection given by the Selector, a fictitious surrogate model denoted $f_{\text{sm}}^{\text{clf}^\star}$ is considered. This fictitious surrogate model coupled with the greedy selection law allows us to have perfect predictions and thus has an accuracy $R_{\text{acc}}(f_{\text{sm}}^{\text{clf}^\star}, \mathcal{X}, \mathcal{Y}) = 1$.

Table 5.4 gives a summary of the R_{fat} scores obtained with the surrogate models $f_{\text{sm}}^{\text{clf}}$ and $f_{\text{sm}}^{\text{clf}^\star}$, under the initial sets $\mathcal{X}$ and $\mathcal{Y}$, and the testing sets denoted by $\mathcal{X}_{\text{test}}$ and $\mathcal{Y}_{\text{test}}$. It can be observed that the surrogate model $f_{\text{sm}}^{\text{clf}}$ derived is far from perfect, as it reduces the fatigue cost expectancy by only 2.4% compared to $f_{\text{sm}}^{\text{clf}^\star}$ which managed to reduce the fatigue cost expectancy by 7.02%. As a matter of fact, the surrogate model might slightly overfit, as its R_{fat} value on the whole dataset almost reaches that of $f_{\text{sm}}^{\text{clf}^\star}$.

Surrogate model	$R_{\text{fat}}(., \mathcal{X}_{\text{test}}, \mathcal{Y}_{\text{test}})$	$R_{\text{fat}}(., \mathcal{X}, \mathcal{Y})$
$f_{\text{sm}}^{\text{clf}}$	2.41%	8.33%
$f_{\text{sm}}^{\text{clf}^\star}$	7.02%	9.63%

TABLE 5.4 – Summary of the R_{fat} scores obtained with the surrogate models $f_{\text{sm}}^{\text{clf}}$ and $f_{\text{sm}}^{\text{clf}^\star}$, under the initial and testing sets.

In my opinion, the derivation of this surrogate model is far from being optimal and might be highly improved by a machine learning specialist. Moreover, the process of taking only the mean and standard deviation of the wind features time series over 100 seconds, as described in Subsection 5.2.4, might lose a lot of information. Therefore, considering time series, a larger dataset and maybe additional features in X would allow us to select the controller more efficiently.

Nevertheless, it is already interesting to assess the potential fatigue reduction which could be obtained in closed-loop with the surrogate model $f_{\text{sm}}^{\text{clf}}$ and greedy selection law.

Mitigation of fatigue cost transients

For the closed-loop implementation of the supervisory layer, the switching strategy considered to smoothly switch from one controller to another is the one described in Sub-

section 5.1.2. The order n_{filt} of the filter is taken to be 2, while its time constants are $\tau_1 = \tau_2 = 4$ seconds. These parameters were chosen empirically in order to minimize the fatigue cost expectancy using the surrogate model controller selection. In Figure 5.7, the evolution of the parameter α with time, under the aforementioned parameters, is plotted. It can be seen that it takes approximately 20 seconds for α to reach its steady value, which is a significant amount of time compared to the time lapse between two switches (100 seconds).

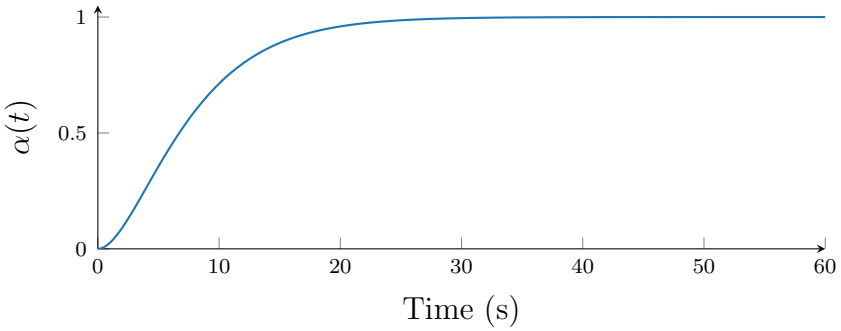

FIGURE 5.7 – Evolution of the parameter α with time for $n_{\text{filt}} = 2$ and $\tau_1 = \tau_2 = 4$ seconds.

For a 600 second simulation, the possibility of switching controller is evaluated at 300, 400 and 500 seconds, based on the estimation of wind features over respectively the intervals $[300, 400[$, $[400, 500[$ and $[500, 600[$. Moreover, switches are not considered among the 300 first seconds in order to avoid transients due to unexpected initial conditions in the turbine.

It should be noted that for the controller selection, it is assumed that the vector of wind features is supposed to be exactly known over the 100 next seconds, which is not realistic. However, it might be more realistic to predict characteristics which are statistics of the wind features over 100 seconds, which should have significant inertia and vary slowly, than predicting directly time series of the wind features over 100 seconds which would be highly uncertain.

The HAWT aero-elastic simulator is thus run in closed-loop with the supervisory layer under the 1000 full-field winds previously generated with TurbSim, described in Subsection 5.2.1. The fatigue cost $\mathcal{J}$ of every closed-loop simulation is evaluated on each of the three intervals where the supervisory layer is active.

In order to quantify the unexpected fatigue cost due to the switch transient, the fatigue cost of the supervisory layer in closed-loop can be compared with those of the individual candidates, obtained during the off-line simulations needed for the generation of the mapping $\mathcal{Y}$. Figure 5.8 plots the scatter of :

— The fatigue cost $\mathcal{J}$ of the supervisory layer in closed-loop, i.e. with switch transient, on the y-axis.

— The fatigue cost of the corresponding wind and controller obtained from the off-line simulations performed for the generation of the mapping $\mathcal{Y}$, i.e. without switch transient, on the x-axis.

It can be seen that the switch transient can both induce an unexpected increase or reduction of the fatigue cost, with almost equal probability. There is thus a slight uncertainty between the fatigue cost obtained from the mapping and the one obtained from the supervisory layer closed-loop simulations. It should be noted that this uncertainty is added

to the surrogate model prediction errors.

It is then considered that the fatigue cost expectancy is the sum of the fatigue costs over all the intervals considered. Even though it is not totally accurate due to unclosed cycles (as explained in Subsection 1.3.2), it can be considered to be a fair approximation. Figure 5.9 plots the evolution of the cumulative sum of the candidate controllers and supervisory layer fatigue cost, with not directly time but the index of the interval evaluated, depending on its temporal bound and simulation number. For example the index 1 corresponds to the first simulation, first time interval from 300 to 400 seconds, index 3 corresponds to the third interval from 500 to 600 seconds and index 4 to the second simulation first interval, and so on.

It can be seen in Figure 5.9 that throughout the evaluations, the supervisory layer approach allows us to efficiently select the candidate controller such that the fatigue cost is efficiently reduced. Note that the last value of the plot is equal to the sum of $\mathcal{J}$ over all the evaluations, which is thus the approximation of the fatigue cost expectancy. Therefore, it can be observed that the supervisory layer approach allows us to obtain a fatigue cost reduction of 6.1% compared to the candidate controller parameterized by p_2, minimizing the fatigue cost expectancy, while without switch transients, a fatigue cost reduction of 9.6% was expected, as detailed in Table 5.4.

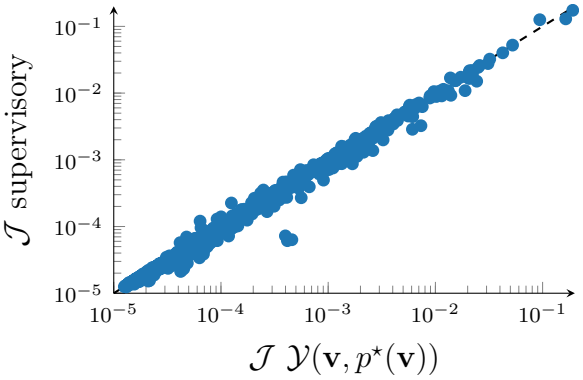

FIGURE 5.8 – Scatter plot of the fatigue costs obtained with the supervisory layer in closed-loop simulation against that of the mapping $\mathcal{Y}$, for every intervals.

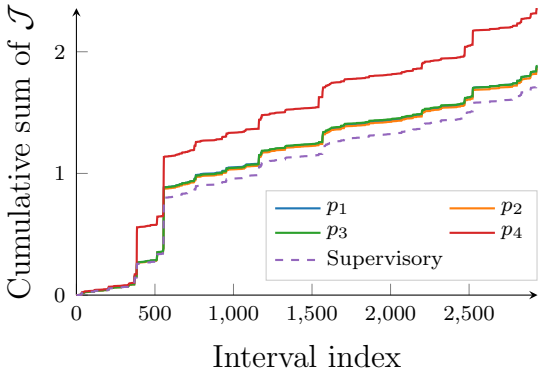

FIGURE 5.9 – Plot of the cumulative sum of fatigue costs evolution with the interval index, for the four candidate controllers and the supervisory layer in closed-loop simulation.

Summary

To summarize the example described in this section, the designed supervisory layer is probably sub-optimal. However, it proves to be able to efficiently reduce the fatigue cost expectancy compared to the best candidate controller. It is true that the candidate controllers selected might not be very efficient in reducing the fatigue cost individually. Nevertheless, the point of this section is to **highlight the possibility of using a data-driven prediction of the fatigue cost in order to efficiently select the candidate controllers**.

Moreover, as no universal design settings allow us to efficiently reduce the fatigue cost for all conditions, this additional layer could become particularly interesting for fatigue

reduction purposes, as it enables us to further reduce the fatigue cost by efficiently selecting candidate controllers. Besides, this additional layer can take into account plenty of other parameters which can hardly be taken into account in standard wind turbine control strategies, such as atmospheric conditions or quality / availability of measurements. Besides, there are also possibilities to refine the surrogate model online and turn this approach into a high-level reinforcement learning.

However, this method has still some weaknesses, as it relies on the assumption that the features considered should vary very slowly, although this might not be such a limiting assumption. Moreover, the switch between controllers should be relatively slow and smooth in order to alleviate the additional fatigue cost due to transients, which imposes a very slow rate of adaptability.

To conclude this section, it seems that several steps are remaining for this approach before being implemented on a commercial HAWT. Therefore, in the next section several prospects of improvements and possible solutions for the scalability of this approach on a commercial HAWT are discussed.

5.3 Prospects of improvements & scalability to commercial HAWT

The proof of concept described in the previous section highlighted that the supervisory layer can potentially reduce the fatigue cost $\mathcal{J}$ expectancy of a HAWT, compared to the best candidate controller considered. However, the proof of concept might not be flawless and could be improved. Moreover, several design steps are remaining before being able to implement this concept on a commercial HAWT.

This section describes prospects of improvements for the example detailed previously and possible issues which could be encountered in implementation prospects of the supervisory layer approach on a commercial HAWT. The outline of this section is organized as follows :

- In Subsection 5.3.1, the shortcomings of the proof of concept presented in Section 5.2 are detailed and possible solutions to improve the performance of this first example are described.
- In Subsection 5.3.2, discussions and guidelines on the scalability and implementation of this supervisory layer approach on commercial HAWT are given.

5.3.1 Possible improvements

The proof of concept presented in Section 5.2 contains some flaws in its design which are detailed in this subsection. Then, solutions to overcome these shortcomings are detailed, design block by design block.

Set of candidate controllers

The supervisory layer approach is one possible solution for an efficient fatigue cost reduction because there are no universal controllers which can minimize the fatigue cost $\mathcal{J}$. Moreover, the supervisory layer objective is to reduce the fatigue cost expectancy compared to the candidate yielding the lowest fatigue cost individually. Therefore the performance of the supervisory layer approach directly depends on the performance of

the best candidate controller considered. However, it is a possibility that the best candidate controller performs always better than the others, therefore no switches should be considered and this approach would have no interest. Nevertheless, as no universal controller minimizing the fatigue cost function $\mathcal{J}$ exists, if one candidate always performs better than the others, it means that the design of the other candidates should be revised.

The set of candidate controllers considered in the proof of concept was very simplistic, containing only four PI controllers. Therefore, as the fatigue reduction and adaptive potential of PI controllers is relatively poor compared to other controllers which can be found in the IPC literature, there is a great potential for improvement in the design of candidate controller. Two possible solutions are proposed, either considering more advanced controllers from the literature allowing efficient fatigue reduction, or considering a larger amount of simple controllers.

The first solution consists in considering several advanced controllers from the literature which are known to reduce efficiently the fatigue cost, possibly assisted by LiDAR for the anticipation of the disturbance rejection, such as [13, 14, 12, 42] for example. Nevertheless, there might be a computational limitation for this kind of set of candidates, as two controllers must be run in parallel with the considered switching strategy described in Subsection 5.1.2. However, it should be noted that more advanced control strategies might be less sensitive to switches than PI controllers.

The second solution consists in considering more PI controllers or controller parameters. A finer mesh of controller parameters could yield a finer adaptation of the parameters and thus further reductions, provided that the classifier can still classify accurately the controllers. Indeed, the region where a candidate minimizes the fatigue cost $\mathcal{J}$ might be shrunk if the number of candidates is increased. Therefore, it might be preferable to generate a denser dataset $\mathcal{X}$, in order to obtain finer borders and an acceptable classification accuracy.

Surrogate model

The performance of the surrogate model presented in Subsection 5.1.1 is sufficient for fatigue cost reduction but still relatively poor, as the classifier might slightly overfit. Therefore, it might be necessary to generate more samples in order to fit a more complex model while avoiding overfitting.

On the other hand, it was mentioned that the feature extraction considered in Subsection 5.2.4 might lose a consequent amount of information, and feature extraction methods from the machine learning literature could help in extracting more meaningful features for classification. Moreover, considering the internal state of the turbine or PI controller integrator as features in the surrogate model could also help in the controller selection process, because the fatigue cost on a horizon depends also on the initial conditions of the closed-loop system.

Eventually, to ensure an efficient design of the surrogate model, it might be preferable to contact a machine learning expert, who could design an accurate surrogate model free from overfitting.

Selector function

Eventually, it might also be interesting to consider a less greedy and more relevant selection law, for the selection of the candidate controllers based on the surrogate model predictions. For example, allowing us to switch to another candidate only if a fatigue reduction is ensured with a certain probability. This could allow us to undergo untimely controller switches which would only cause fatigue transients.

Concerning the switching strategies, the one implemented in the proof of concept might be relatively slow, resulting in a very slow adaptation rate. It is mentioned in Subsection 5.3.1 that, if more advanced control strategies were used in the candidates' design, the switching strategy used in the proof of concept might be limiting in terms of computational cost, as two controllers would be run in parallel until the smoothing coefficient α converges to 1. Moreover, more advanced control strategies could allow us to efficiently switch from one candidate to another without a smoothing coefficient, such as bumpless PID control [98].

On the other hand, a small gap in the controller parameters should limit the transient effects. Therefore, if more PI controllers or parameters were used and the switches were limited to the candidates in the vicinity of the current one, it might be possible to allow faster and more frequent switches, as the gap between controller parameters would be reduced.

In this subsection, only the shortcomings specific to the proof of concept design presented in Section 5.2 and possible solutions to mitigate them are highlighted. In the next subsection, issues relative to the scalability and implementation of the approach in commercial HAWT are described and prospects to overcome these issues are discussed.

5.3.2 Scalability to commercial HAWT

The main pitfalls which limit the implementation of the supervisory layer approach on a commercial HAWT are the following :

1. An estimation of the vector of wind features values over the 100 future seconds is needed for the surrogate model predictions. However, LiDAR and WFR algorithms can allow an accurate prediction of the wind features over an horizon of 2 to 4 seconds only. Therefore the estimation of the vector of wind features is delayed, which could deteriorate the closed-loop performance.

2. Moreover, LiDAR measurements could be deteriorated for some atmospheric conditions, e.g. when the fog is thick or the air is almost free of aerosols. Therefore, the LiDAR measurements and thus the estimations of the WFR algorithms become very uncertain, which could furthermore degrade the closed-loop performance.

3. The surrogate model is derived from data which are entirely obtained from simulation. Therefore, if the simulation model mismatches the real HAWT, the data and the surrogate model might be biased, which could once again worsen the performance of the supervisory layer.

In this subsection, the above issues are discussed and possible solutions to mitigate the latter are given.

Vector of wind features estimation delay

The proof of concept presented in Subsection 5.2 relied on the assumption that the features extracted from the time series of the WFR algorithm outputs could be exactly known over the 100 future seconds. The LiDAR technology and WFR algorithm of today do not allow us to have such an estimation.

However, it should be noted that the features extracted are statistics of the WFR algorithm time series outputs over 100 seconds (mean and standard deviation) and should thus vary relatively slowly. Therefore, if the variations are sufficiently slow, it might be possible to :

— Estimate these features from the last time instants, similarly to what is done in Section 4.3 with MPC_{filt}.

— It might be possible to have an accurate estimation of the current mean and standard deviation of the WFR algorithms outputs in a shorter period than 100 seconds.

Thus, it might be possible to reduce the delay and uncertainty of the vector of wind features estimation by estimating them on a shorter horizon and on previous data, similar to what is done with MPC_{filt} in Section 4.3.

Nevertheless, it is of primal importance to analyze the fatigue cost expectancy sensitivity of the supervisory layer approach, to uncertainty and delay on the vector of wind features estimation. This verification can allow us to finalize the proof of concept presented in Section 5.2.

LiDAR measurements uncertainties

Let us assume that the variations of the vector of wind features are sufficiently slow, such that the delay does not affect too importantly the closed-loop performance of the supervisory layer. LiDAR measurements use the movements of the aerosols floating in the air in order to estimate the wind speed of a point in space. The LiDAR measurements are accurate for only a certain range of aerosol density in the air. Therefore, in dusty or foggy conditions where the density of aerosols is high, or when the air is very clear and the density of aerosols is low, the LiDAR measurements can become inaccurate. Hence, the WFR algorithm outputs and vector of wind feature estimations become uncertain. Because of these uncertain estimations, the prediction of the surrogate model could be biased and the supervisory layer could become inefficient.

A solution to this kind of issue is that the surrogate model predictions not only rely on features obtained from LiDAR measurements. For example, information given by wind cup, wind vane, accelerators or strain gauges, which are already installed on commercial HAWT, could help in estimating the current wind conditions and thus efficiently selecting the candidates. Moreover, it would allow us to consider also the state of the turbine in the surrogate model prediction, which could also help in the candidate selection.

Besides, adding features such as the signal to noise ratio of the LiDAR measurements could help in discriminating the cases where the surrogate model can be confident in the LiDAR measurements and the cases where it cannot. Moreover, as candidate controllers using LiDAR measurements should also be deteriorated by atmospheric conditions, the signal to noise ratio could also help in efficiently selecting them over candidates which are not LiDAR-assisted. Obviously, this implies considering different levels of noise on the LiDAR measurements during the simulations necessary for the mapping generation.

Model mismatch

Finally, let us assume that the surrogate model selects perfectly the candidates for given wind conditions in simulation. It is known that simulation never reflects perfectly the reality, even with digital twins. Model mismatch is, by the way, a classical issue in model-based control theory. However, in model-based control theory, frequent actuation and feedback information help in mitigating the model mismatch's undesirable effects.

The supervisory layer approach can be seen as a high-level model-based control strategy, as it uses a data-driven model to select candidate controllers in order to minimize a fatigue cost. However, it has no feedback and its actuation frequency is dramatically slow. Therefore, there is a strong risk that model mismatch leads to irrelevant candidate selection and deteriorates supervisory layer performance.

Fortunately, in the supervisory layer approach, irrelevant candidate selection might not be as bad as it seems. Indeed, all the candidates are supposed to stabilize and regulate the system relatively well. The main undesirable effect being that an irrelevant controller selection might result in increasing the fatigue cost expectancy, compared to the candidate minimizing the fatigue cost expectancy individually.

Nevertheless, it might be possible to correct this model mismatch by keeping on gathering data online and learning from this new data. It should be noted that the proof of concept described in Subsection 5.2 does not allow this online learning feature, because the probability that a candidate minimizes the fatigue cost for given wind conditions is predicted by the surrogate model. Indeed, while learning online, the HAWT can only gather triplets of wind conditions, candidate controller and fatigue cost. Therefore, it does not allow us to compare the performance to other candidates under the same conditions.

This data does not allow us to use an off-the-shelf classifier to fit the probability that a candidate minimizes the fatigue cost. However, it does allow us to make a regression of the fatigue cost for given candidate controllers and wind conditions. Therefore, it is possible to predict the fatigue cost for every candidate under given wind conditions, which allows us to select the candidate minimizing the fatigue cost with a selection law such as the one described in Subsection 5.1.2. Moreover, it would allow us to keep on learning the fatigue from measurements and refine the regression surrogate model.

A summary of the issues, prospects and solutions proposed for the implementation of the supervisory layer approach on commercial HAWT is given in Table 5.5.

5.4 Conclusion and perspectives

In this chapter, a new supervisory layer approach for online selection of controller parameter vector p among a discrete set of candidates has been presented. This method consists in using a surrogate model to predict the fatigue cost $\mathcal{J}$, or the probability that a controller with given parameters minimizes the fatigue cost of each for a discrete set of candidates under given wind conditions. These predictions are thus used to select the parameter $p^\star(\mathbf{v})$ of the candidate controller most likely to minimize the fatigue cost $\mathcal{J}$ under given wind conditions $\mathbf{v}$. This developed framework has the advantage that its objective is to minimize a complex cost function that the turbine manufacturer actually wants to minimize (e.g. the operational fatigue cost of a HAWT), while using controllers with a classical architecture. Moreover, compared to the MPC approaches presented in

Issue	Prospects / Solutions
The vector of wind features estimation is delayed.	Reduce the delay by estimating the vector of wind features on a shorter period of time. Study the sensitivity of the supervisory layer approach to delayed estimation of the wind features vector.
The vector of wind features estimation only depends on LiDAR measurements	Consider more features which are provided by other sensors than LiDAR, i.e. accelerators, wind cup and so on. Use the signal to noise ratio of LiDAR measurements as a feature to discriminate the bad LiDAR measurements.
The surrogate model is fitted on data coming only from simulation time series.	Predict the fatigue cost instead of the probability that a candidate minimizes the fatigue cost and consider an online learning surrogate model.

TABLE 5.5 – Summary of the issues, solutions and prospects of the supervisory layer approach, in order to make it implementable on commercial HAWT.

Chapter 4, it allows us to yield good performance with more simple control strategies, more adapted to an industrial environment.

A first proof of concept using this approach was presented and implemented in closed-loop using several optimistic assumptions :

— The wind features extracted from the oncoming wind are perfectly estimated.
— There is no model mismatch between the one used to generate the data for the surrogate model and the one used in simulation.

On the other hand, very simplistic designs of candidate controllers and surrogate models allowed us to obtain a significant reduction of the fatigue cost expectancy compared to the best candidate individually.

The important results are that :

— It is possible to select the parameters of candidate controllers in order to minimize the fatigue cost expectancy based on a vector of wind features extracted from WFR algorithm outputs.
— This supervisory layer approach allows us to reduce the fatigue cost expectancy compared to the best candidate controller.
— The absolute fatigue cost expectancy of the supervisory layer approach highly depends on the best candidate controller fatigue cost expectancy.

The main advantages of this approach are to limit the sensitivities to controller tuning procedure and to provide an economically driven control strategy, that can be effectively adapted to different HAWT systems. By setting parameters of the controller showing the greatest potential in the current wind conditions, it could be possible to boost closed-loop performances with simple control strategies. Finally, the algorithm still has several shortcomings which prevent its implementation on commercial HAWT :

— The information on the vector of wind features is delayed by several tens of seconds. Solutions to reduce this delay are proposed in Subsection 5.3.2, but **the sensitivity of this method to delayed information on the vector of wind features must be assessed**.
— The controller parameter selection only relies on LiDAR measurements which in some conditions can be highly deteriorated. However, prospects are given in Subsection 5.3.2 in order to extend the supervisory layer approach to other features which do not rely on LiDAR measurements.

— The quality of the surrogate model relies completely on the quality of the simulations on which it was fitted. Therefore, if the simulations did not represent accurately the turbine dynamics, the surrogate model might be biased. Therefore, prospects of an **online learning surrogate model** are described in Subsection 5.3.2, which would **turn the supervisory layer approach in high level reinforcement learning for fatigue cost reduction**.

Finally, it should be noted that this approach is a high-level control framework and not a specific control technique. Moreover, its design is not only the concern of control experts but also :

— Mechanical experts for an efficient design of a digital twin for data generation.
— Material experts for the design of an accurate fatigue cost function.
— Control experts for the design of the set of candidate controllers, the selection law and the switching strategy.
— Machine learning experts for the surrogate model design.

Chapitre 6

Conclusion & Perspectives

This dissertation addresses the issue of efficient reduction of the economic fatigue cost expectancy with closed-loop individual pitch control of three-bladed horizontal axis wind turbines.

It is first shown that classical control strategies with fixed parameters are a constraint on the reduction of fatigue cost expectancy. However, combining several controllers, designed using standard control strategies and parameterized differently, could allow tremendous reductions in fatigue cost expectancy. There is thus a strong need for an efficient adaptation of standard control strategies parameters, in order to efficiently reduce the fatigue cost expectancy. In this thesis, two data-driven solutions are presented for an online adaptation of classical control strategies parameters under given wind conditions. The first one involves the optimization of a data-driven identification of the fatigue cost function in an optimal control problem. The second solution considers a data-driven identification of the controller parameters, allowing us to efficiently reduce the fatigue cost for given wind conditions, which can be seen as a highly-complex gain scheduling. Both solutions are real-time implementable and allow significant reduction of both the fatigue cost expectancy compared to standard control strategies with fixed parameters and the sensitivity to controller tuning procedures.

The main results presented in this thesis along with the possible perspectives of this work may be summarized as follows.

6.1 Main contributions & results

An IPC controller whose parameters are adapted to given wind conditions can allow significant reduction of the fatigue cost expectancy, compared to an individual controller with fixed parameters, optimized for reduction of the fatigue cost expectancy. This was shown through several studies in Chapters 2, 3 and 5, using different kinds of controllers, numbers of parameters, simulation settings and under disturbances with different levels of complexity. A summary of the results obtained, along with the corresponding conditions of the experiments, is given in Table 6.1.

A methodology based on a data-driven identification of the fatigue cost from quadratic forms is developed in order to adapt the parameters of a standard optimal control problem for efficient reduction of the fatigue cost. It is shown that this methodology, tested under a simplified environment, can allow a reduction of the fatigue cost expectancy of approximately 50%, compared to an optimal control problem with fixed parameters. At the moment, this method has only been tested on an LTI system. However, it is possible to apply this method to more general LPV or nonlinear systems.

From the above methodology, an adaptive MPC is designed, allowing us to efficiently adapt the MPC parameters based on the current level of vibration of the system. This method tested under a simplified environment allowed a reduction of the fatigue cost expectancy of approximately 30%. Nevertheless, it was stressed that too short simulations might avoid further reduction due to unexpected transients in the beginning of the closed-loop simulations. Similarly to the fatigue-oriented open-loop optimization, this adaptive MPC has only been tested on an LTI system and can be adapted to LPV or nonlinear systems.

Finally, a framework allowing us to select parameters defining a controller, from several candidate controllers, based on the current wind conditions, is presented. This method

Characteristic	Chapter 2	Chapter 3	Chapter 5
Simulation settings	HAWT aero-elastic simulator with only the blades flapwise DOF activated	LTI system representing the blades flapwise vibrations for a 12 m/s hub-height wind speed	HAWT aero-elastic simulator with all DOF activated
Controller	LiDAR-assisted nonlinear MPC	LiDAR-assisted linear MPC	PI controller without feedforward
Number of possible values for the controller parameters p	6	50	4
Disturbance	Realistic hub-height turbulent wind with speed, direction and shear variations	Realistic hub-height wind speed variations only	Realistic full-field turbulent wind
Cost function components	Fatigue of the blades and their actuators only	Fatigue of the blades and their actuators only	Fatigue of the blades and their actuators, tower, gearbox and rotor shaft
Level of reality	++	+	+++
$\mathcal{J}$ expectancy reduction	23%	50%	9%

TABLE 6.1 – Summary of the potential reduction of the fatigue cost expectancy that an adaptation of the controller parameters can bring, compared to a controller with fixed parameters, optimized for fatigue cost expectancy reduction.

can be seen as an identification of a function which yields the parameters of the controller minimizing the fatigue cost under given wind conditions. It is shown under a realistic simulation environment, but with significant assumptions on the wind estimation, that a 7% reduction of the fatigue cost expectancy can be expected, against potential 9% if the fatigue cost surrogate model predictions within the framework were perfect. This framework features several blocks, whose designs involve multiple engineering skills and can be improved for further reduction of the fatigue cost expectancy.

6.2 Discussions & Perspectives

The adaptive solutions presented above are all exploratory and the corresponding results preliminary. Therefore, the future directions concerning these methodologies would be to test them under more complex and realistic simulation environments, i.e. simulated with a HAWT aero-elastic simulator with a complete model under full-field turbulent winds. Moreover, no uncertainties are considered on the state nor the wind estimation throughout this thesis. Therefore, it is important to study the sensitivity of the presented approaches to uncertainties on state and wind estimation.

More particularly, concerning the supervisory layer approach, it is of primal importance to test its sensitivity to a significant delay on the wind characteristics estimation, which would be present in a realistic implementation. Moreover, another perspective concerning

the development of the supervisory layer approach is the online learning feature, which would allow us to turn this framework into a high-level reinforcement learning for fatigue cost reduction.

The approaches presented in this thesis all aimed at efficiently minimizing the fatigue cost of a HAWT for given wind conditions using IPC regulation. Efficiently reducing the economic fatigue cost of a HAWT might be important for a turbine owner, as it could allow him to reduce the cost of maintenance. However, a turbine owner also needs to make money by selling wind energy to the production grid, while ensuring that the power production signal matches particular specifications. These two objectives are handled by torque and CPC regulation. The impact of IPC, which modifies the blade pitch angle compared to CPC, on power production and regulation is often assumed to be negligible. Nevertheless, in practice a very active IPC could disrupt power production and regulation. Therefore, there is a more general trade-off which must be considered between fatigue cost minimization and power production maximization, while efficiently regulating the power produced, which can ultimately be summarized as a profit maximization problem.

Throughout the approaches developed in this dissertation, the consequences on power production and regulation of the IPC controllers implemented were not considered. However, if these approaches do not allow us to efficiently maximize the profit of a HAWT owner, it constitutes one of the fundamental blocks for the efficient maximization of profit. Moreover, the methodologies presented in this thesis, especially the supervisory layer approach, could be generalized to the profit maximization problem, provided that a satisfactory economic model of power production and regulation is available.

Chapitre 7

Bibliographie

[1] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind energy handbook*. John Wiley & Sons, 2011.

[2] GWEC, "Global wind report," tech. rep., Global Wind Energy Council, 2017.

[3] GWEC, "Global wind energy outlook," tech. rep., Global Wind Energy Council, 2016.

[4] E. J. Simley, "Wind speed preview measurement and estimation for feedforward control of wind turbines," 2015.

[5] F. Guillemin, H.-N. Nguyen, G. Sabiron, D. Di Domenico, and M. Boquet, "Real-time three dimensional wind field reconstruction from nacelle LiDAR measurements," in *Journal of Physics : Conference Series*, vol. 1037, p. 032037, IOP Publishing, 2018.

[6] A. Scholbrock, P. Fleming, L. Fingersh, A. Wright, D. Schlipf, F. Haizmann, and F. Belen, "Field testing LiDAR-based feed-forward controls on the NREL controls advanced research turbine," in *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, p. 818, 2013.

[7] R. S. Hunter, B. M. Pedersen, T. Pedersen, H. Klug, N. van der Borg, N. Kelley, and J. Dahlberg, "Recommended practices for wind turbine testing and evaluation. wind speed measurement and use of cup anemometry.," IEA, 1999.

[8] C. L. Bottasso and C. Riboldi, "Estimation of wind misalignment and vertical shear from blade loads," *Renewable energy*, vol. 62, pp. 293–302, 2014.

[9] M. Bertelè, C. Bottasso, S. Cacciola, *et al.*, "Simultaneous estimation of wind shears and misalignments from rotor loads : formulation for ipc-controlled wind turbines," in *J. Phys. Conf. Ser*, vol. 1037, 2018.

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

[11] J. Laks, L. Pao, A. Wright, N. Kelley, and B. Jonkman, "The use of preview wind measurements for blade pitch control," *Mechatronics*, vol. 21, no. 4, pp. 668–681, 2011.

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

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

[14] D. Ossmann, J. Theis, and P. Seiler, "Load reduction on a clipper liberty wind turbine with linear parameter-varying individual blade pitch control," *Wind Energy*, vol. 20, no. 10, pp. 1771–1786, 2017.

[15] K. Selvam, S. Kanev, J. W. van Wingerden, T. van Engelen, and M. Verhaegen, "Feedback–feedforward individual pitch control for wind turbine load reduction," *International Journal of Robust and Nonlinear Control : IFAC-Affiliated Journal*, vol. 19, no. 1, pp. 72–91, 2009.

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

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

[18] Y. Zhang, Z. Chen, and M. Cheng, "Proportional resonant individual pitch control for mitigation of wind turbines loads," *IET Renewable Power Generation*, vol. 7, no. 3, pp. 191–200, 2013.

[19] W. Leithead, V. Neilson, S. Dominguez, and A. Dutka, "A novel approach to structural load control using intelligent actuators," in *17th Mediterranean Conf. on Control and Automation*, pp. 1275–1262, 2009.

[20] R. P. Coleman and A. M. Feingold, "Theory of self-excited mechanical oscillations of helicopter rotors with hinged blades," *National Advisory Committee for Aerounautics*, 1957.

[21] S. Cacciola, C. E. Riboldi, and A. Croce, "A new decentralized pitch control scheme for wind turbines," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9908–9913, 2017.

[22] S. T. Navalkar, J.-W. van Wingerden, E. van Solingen, T. Oomen, E. Pasterkamp, and G. Van Kuik, "Subspace predictive repetitive control to mitigate periodic loads on large scale wind turbines," *Mechatronics*, vol. 24, no. 8, pp. 916–925, 2014.

[23] S. Wortmann, J. Geisler, and U. Konigorski, "Lidar-assisted feedforward individual pitch control to compensate wind shear and yawed inflow," in *Journal of Physics : Conference Series*, vol. 753, p. 052014, IOP Publishing, 2016.

[24] M. H. Do, J. G. Njiri, and D. Söffker, "Structural load mitigation control for nonlinear wind turbines with unmodeled dynamics," in *2018 Annual American Control Conference (ACC)*, pp. 3466–3471, IEEE, 2018.

[25] E. Bossanyi, B. Savini, M. Iribas, M. Hau, B. Fischer, D. Schlipf, T. van Engelen, M. Rossetti, and C. Carcangiu, "Advanced controller research for multi-mw wind turbines in the UPWIND project," *Wind Energy*, vol. 15, no. 1, pp. 119–145, 2012.

[26] W. Duesterhoeft, M. W. Schulz, and E. Clarke, "Determination of instantaneous currents and voltages by means of alpha, beta, and zero components," *Transactions of the American Institute of Electrical Engineers*, vol. 2, no. 70, pp. 1248–1255, 1951.

[27] Y. Zhang, M. Cheng, and Z. Chen, "Load mitigation of unbalanced wind turbines using PI-R individual pitch control," *IET Renewable Power Generation*, vol. 9, no. 3, pp. 262–271, 2014.

[28] W. E. Leithead, Y. Han, and J. Day, "Field tests of individual blade control and its impact on the wind turbine components lifetime," in *EWEA 2016*, 2016.

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

[30] W. H. Lio, B. L. Jones, Q. Lu, and J. A. Rossiter, "Fundamental performance similarities between individual pitch control strategies for wind turbines," *International Journal of Control*, vol. 90, no. 1, pp. 37–52, 2017.

[31] Z. Liu, F. Huo, S. Xiao, X. Zhang, S. Zhu, G. Ji, H. Dai, and J. Feng, "Individual pitch control of wind turbine based on RBF neural network," in *Control Conference (CCC), 2016 35th Chinese*, pp. 5769–5773, IEEE, 2016.

[32] Z. Civelek, M. Lüy, E. Çam, and H. Mamur, "A new fuzzy logic proportional controller approach applied to individual pitch angle for wind turbine load mitigation," *Renewable Energy*, vol. 111, pp. 708–717, 2017.

[33] P. A. Ioannou and J. Sun, *Robust adaptive control*. Courier Corporation, 2012.

[34] J. Jonkman and B. Jonkman, "NWTC information portal (FAST v8) https ://nwtc.nrel.gov," 2016.

[35] J. Chauvin and Y. Creff, "Nonlinear two stage control strategy of a wind turbine for mechanical load and extreme moment reduction," in *ASME 2011 30th International Conference on Ocean, Offshore and Arctic Engineering*, pp. 359–369, American Society of Mechanical Engineers, 2011.

[36] S. Gros, "An economic NMPC formulation for wind turbine control," in *Proc. 52nd Annual Conference on Decision and Control (CDC)*, pp. 1001–1006, IEEE, 2013.

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

[38] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW reference wind turbine for offshore system development," tech. rep., National Renewable Energy Laboratory, 2009.

[39] M. Harris, M. Hand, and A. Wright, "LiDAR for turbine control," *National Renewable Energy Laboratory, Golden, CO, Report No. NREL/TP-500-39154*, 2006.

[40] D. Schlipf, P. Fleming, F. Haizmann, A. Scholbrock, M. Hofsäß, A. Wright, and P. W. Cheng, "Field testing of feedforward collective pitch control on the CART2 using a nacelle-based LiDAR scanner," in *Journal of Physics : Conference Series*, vol. 555, p. 012090, IOP Publishing, 2014.

[41] E. J. N. Menezes, A. M. Araújo, and N. S. B. da Silva, "A review on wind turbine control and its associated methods," *Journal of cleaner production*, vol. 174, pp. 945–953, 2018.

[42] I. Houtzager, J. van Wingerden, and M. Verhaegen, "Wind turbine load reduction by rejecting the periodic load disturbances," *Wind Energy*, vol. 16, no. 2, pp. 235–256, 2013.

[43] S. Kanev and T. van Engelen, "Wind turbine extreme gust control," *Wind Energy*, vol. 13, no. 1, pp. 18–35, 2010.

[44] M. Vali, J.-W. van Wingerden, and M. Kühn, "Optimal multivariable individual pitch control for load reduction of large wind turbines," in *American Control Conference (ACC), 2016*, pp. 3163–3169, IEEE, 2016.

[45] R. Ungurán, V. Petrović, L. Y. Pao, and M. Kühn, "Performance evaluation of a blade-mounted LiDAR with dynamic versus fixed parameters through feedback-feedforward individual pitch and trailing edge flap control," in *Journal of Physics : Conference Series*, vol. 1037, p. 032004, IOP Publishing, 2018.

[46] Q. Lu, R. Bowyer, and B. L. Jones, "Analysis and design of coleman transform-based individual pitch controllers for wind-turbine load reduction," *Wind Energy*, vol. 18, no. 8, pp. 1451–1468, 2015.

[47] W. F. Arnold and A. J. Laub, "Generalized eigenproblem algorithms and software for algebraic riccati equations," *Proceedings of the IEEE*, vol. 72, no. 12, pp. 1746–1754, 1984.

[48] 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.

[49] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[50] J. Friis, E. Nielsen, J. Bonding, F. D. Adegas, J. Stoustrup, and P. F. Odgaard, "Repetitive model predictive approach to individual pitch control of wind turbines," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 3664–3670, IEEE, 2011.

[51] J. B. Rawlings, D. Bonné, J. B. Jorgensen, A. N. Venkat, and S. B. Jorgensen, "Unreachable setpoints in model predictive control," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2209–2215, 2008.

[52] M. Diehl, R. Amrit, and J. B. Rawlings, "A lyapunov function for economic optimizing model predictive control," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, 2010.

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

[54] G. Freebury and W. Musial, "Determining equivalent damage loading for full-scale wind turbine blade fatigue tests," in *2000 ASME Wind Energy Symposium*, p. 50, 2000.

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

[56] B. Biegel, "Distributed control of wind farm," *MSc Theses*, 2011.

[57] A. Peña, C. B. Hasager, J. Lange, J. Anger, M. Badger, and F. Bingöl, "Remote sensing for wind energy."

[58] T. Knudsen, T. Bak, and M. Svenstrup, "Survey of wind farm control—power and fatigue optimization," *Wind Energy*, vol. 18, pp. 1333–1351, 2015.

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

[60] 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.

[61] K. Hammerum, P. Brath, and N. K. Poulsen, "A fatigue approach to wind turbine control," *Journal of Physics : Conference Series*, vol. 75, p. 012081, 2007.

[62] J. de Jesus 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.

[63] M. Brokate and J. Sprekels, *Hysteresis and phase transitions*, vol. 121. Springer Science & Business Media, 2012.

[64] M. Brokate, K. Dressler, and P. Krejci, "Rainflow counting and energy dissipation for hysteresis models in elastoplasticity," *European journal of mechanics A/Solids*, vol. 15, no. 4, pp. 705–737, 1996.

[65] S. Belbas and I. Mayergoyz, "Dynamic programming for systems with hysteresis," *Physica B : Condensed Matter*, vol. 306, no. 1-4, pp. 200–205, 2001.

[66] S. Belbas and I. Mayergoyz, "Optimal control of dynamical systems with preisach hysteresis," *International journal of non-linear mechanics*, vol. 37, no. 8, pp. 1351–1361, 2002.

[67] M. Musallam and C. M. Johnson, "An efficient implementation of the rainflow counting algorithm for life consumption estimation," *IEEE Transactions on reliability*, vol. 61, no. 4, pp. 978–986, 2012.

[68] S. Loew, D. Obradovic, and C. Bottasso, "Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control," in *Proc. European Control Conference (ECC)*, pp. 3371–3376, 2019.

[69] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux meeting on systems and control*, vol. 11, pp. 119–141, Technische Universiteit Eindhoven Veldhoven Eindhoven, The Netherlands, 2002.

[70] Y. Liu, R. J. Patton, and S. Shi, "Wind turbine load mitigation using MPC with gaussian wind speed prediction," in *2018 UKACC 12th International Conference on Control (CONTROL)*, pp. 32–37, IEEE, 2018.

[71] M. N. Sinner and L. Y. Pao, "A comparison of individual and collective pitch model predictive controllers for wind turbines," in *2018 Annual American Control Conference (ACC)*, pp. 1509–1514, IEEE, 2018.

[72] M. N. Sinner and L. Y. Pao, "A study on horizon length for preview-enabled model predictive control of wind turbines," in *2019 American Control Conference (ACC)*, pp. 3488–3493, IEEE, 2019.

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

[74] D. Dougherty and D. Cooper, "A practical multiple model adaptive strategy for multivariable model predictive control," *Control Engineering Practice*, vol. 11, no. 6, pp. 649–664, 2003.

[75] M. Pisaturo, M. Cirrincione, and A. Senatore, "Multiple constrained MPC design for automotive dry clutch engagement," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 1, pp. 469–480, 2014.

[76] J. Pahasa and I. Ngamroo, "PHEVs bidirectional charging/discharging and SoC control for microgrid frequency stabilization using multiple MPC," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 526–533, 2014.

[77] L. Chisci, P. Falugi, and G. Zappa, "Gain-scheduling MPC of nonlinear systems," *International Journal of Robust and Nonlinear Control : IFAC-Affiliated Journal*, vol. 13, no. 3-4, pp. 295–308, 2003.

[78] M. Grant, S. Boyd, and Y. Ye, "CVX : Matlab software for disciplined convex programming," 2008.

[79] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[80] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit—an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[81] M. Alamir and G. Pannochia, "A new formulation of economic model predictive control without terminal constraint," *arXiv preprint arXiv :2005.04381*, 2020.

[82] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[83] 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.

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

[85] S.-C. Chang, *A hierarchical, temporal decomposition approach to long horizon optimal control problems*. PhD thesis, University of Connecticut, 1986.

[86] W. Xu and M. Anitescu, "Exponentially accurate temporal decomposition for long-horizon linear-quadratic dynamic optimization," *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 2541–2573, 2018.

[87] I. Van der Hoven, "Power spectrum of horizontal wind speed in the frequency range from 0.0007 to 900 cycles per hour," *Journal of meteorology*, vol. 14, no. 2, pp. 160–164, 1957.

[88] A. Clifton, S. Schreck, G. Scott, N. Kelley, and J. K. Lundquist, "Turbine inflow characterization at the national wind technology center," *Journal of Solar Energy Engineering*, vol. 135, no. 3, p. 031017, 2013.

[89] F. Mouzakis, E. Morfiadakis, and P. Dellaportas, "Fatigue loading parameter identification of a wind turbine operating in complex terrain," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 82, no. 1-3, pp. 69–88, 1999.

[90] H. S. Toft, L. Svenningsen, W. Moser, J. D. Sørensen, and M. L. Thøgersen, "Wind climate parameters for wind turbine fatigue load assessment," *Journal of Solar Energy Engineering*, vol. 138, no. 3, p. 031010, 2016.

[91] I. Abdallah, A. Natarajan, and J. D. Sørensen, "Influence of the control system on wind turbine loads during power production in extreme turbulence : Structural reliability," *Renewable Energy*, vol. 87, pp. 464–477, 2016.

[92] J. P. Murcia, P.-E. Réthoré, N. Dimitrov, A. Natarajan, J. D. Sørensen, P. Graf, and T. Kim, "Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates," *Renewable Energy*, vol. 119, pp. 910–922, 2018.

[93] N. Dimitrov, M. Kelly, A. Vignaroli, and J. Berg, "From wind to loads : wind turbine site-specific load estimation using databases with high-fidelity load simulations," *Wind Energ. Sci. Discuss*, 2018.

[94] L. Schröder, N. K. Dimitrov, D. R. Verelst, and J. A. Sørensen, "Wind turbine site-specific load estimation using artificial neural networks calibrated by means of high-fidelity load simulations," in *Journal of Physics : Conference Series*, vol. 1037, p. 062027, IOP Publishing, 2018.

[95] L. J. Fingersh, M. M. Hand, and A. S. Laxson, "Wind turbine design cost and scaling model," 2006.

[96] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics New York, NY, USA, 2001.

[97] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, pp. 63–71, Springer, 2003.

[98] R. Hanus, M. Kinnaert, and J.-L. Henrotte, "Conditioning technique, a general anti-windup and bumpless transfer method," *Automatica*, vol. 23, no. 6, pp. 729–739, 1987.

[99] M. O. Hansen, *Aerodynamics of wind turbines*. Routledge, 2015.

[100] L. H. Donnell, *Beams, plates and shells*. McGraw-Hill Companies, 1976.

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

Annexe A

Blade element momentum theory

The rotor is the most nonlinear subsystem of the turbine, and that is because of the aerodynamic considerations. Rotor modelling must be able to estimate the loads on the turbine blades, induced by the inflow wind field and blade vibrations. The theory which is used to relate the inflow wind to blade loads is called Blade Element Momentum (BEM) [99].

As said previously, the rotor can be subdivided in three subsystems which are the three individual blades. BEM theory allows to determine the load distribution along the blade from the inflow wind and blade spatial coordinates. Thus, by integrating this load distribution, it is possible to determine each blade root out-of-plane and in-plane axial forces and bending moments, denoted respectively by F^i_{Oop}, F^i_{Ip}, M^i_{Oop} and M^i_{Ip} for the i^{th} blade. Therefore :

$$T_a = \sum_{i=0}^{2} M^i_{Ip} \tag{A.1a}$$

$$F_t = \sum_{i=0}^{2} F^i_{Oop} \tag{A.1b}$$

$$M_{\text{yaw}} = \sum_{i=0}^{2} M^i_{Oop} \cos(\psi_1 + 2i\frac{\pi}{3}) \tag{A.1c}$$

$$M_{\text{tilt}} = \sum_{i=0}^{2} M^i_{Oop} \sin(\psi_1 + 2i\frac{\pi}{3}) \tag{A.1d}$$

$$\tag{A.1e}$$

where ψ_1 is the azimuth angle of the first blade. The transformation to obtain M_{yaw} and M_{tilt} is called the Coleman or MBC transform [20, 17].

Aerodynamic behavior To give an overview of BEM theory, a quick recall on aerodynamic basics is needed. As can be seen in Figure A.1, when a blade element is submitted to a wind velocity vector $\overrightarrow{V_0}$, an aerodynamic force is induced, which can be divided in two components named the lift and the drag, denoted respectively by $\overrightarrow{L}$ and $\overrightarrow{D}$. The drag direction is always collinear with the direction of $\overrightarrow{V_0}$, denoted by the dotted line in Figure A.1, while the direction of the lift is orthogonal to $\overrightarrow{V_0}$ direction. Their magnitudes are computed as follows :

$$L = \frac{1}{2}\rho V_0^2 c C_L(\alpha) \tag{A.2a}$$

$$D = \frac{1}{2}\rho V_0^2 c C_D(\alpha) \tag{A.2b}$$

where ρ is the air density, c is the chord length of the blade element, α is the angle of attack, i.e. the angle between the fluid and the chord directions as illustrated in Figure A.1, C_L and C_D are respectively the lift and drag coefficients, and V_0 is the wind speed associated to $\overrightarrow{V_0}$. The highly nonlinear lift and drag coefficients depend on the angle of attack and are usually obtained from look-up tables for the estimation of L and D, an example of the relation between these coefficients and the angle of attack is illustrated in Figure A.2.

In BEM theory, each blade is discretized in blade elements along its longitudinal axis. The rotor is thus discretized in annular elements named control volumes, as illustrated

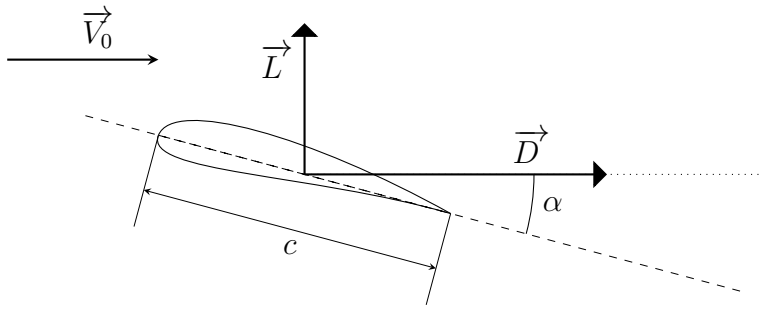

FIGURE A.1 – Resulting aerodynamic forces on a blade element submitted to an ahead wind of magnitude V_0.

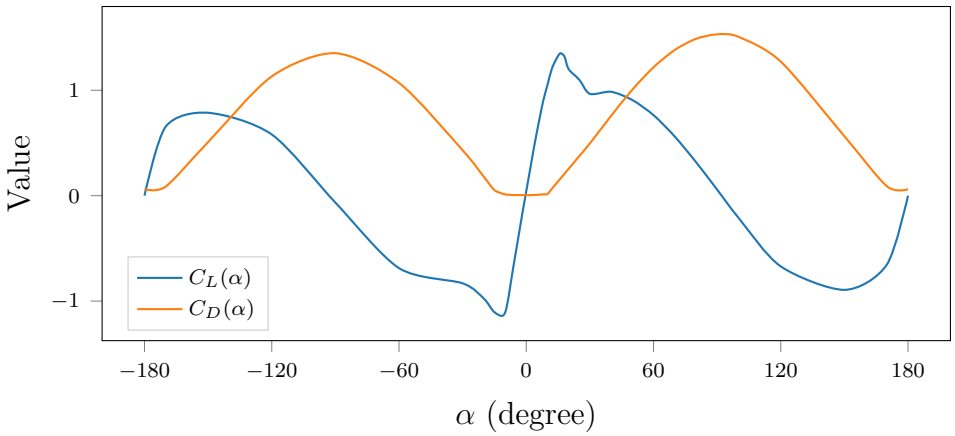

FIGURE A.2 – Lift and drag coefficients of a blade section as a function of the angle of attack α.

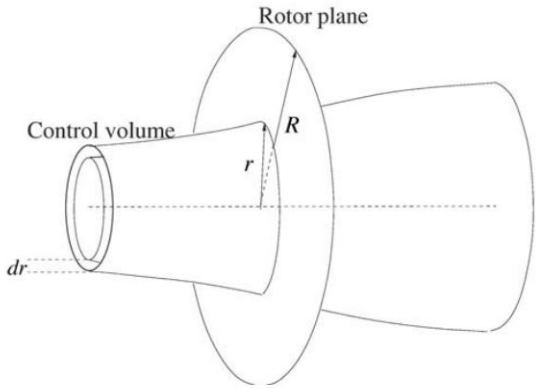

FIGURE A.3 – Illustration of the notions of the rotor discretization in annular elements and control volume, from [99].

in Figure A.3. BEM theory is based on the assumption that the air flowing through one annular element stays in the same control volume. When the air crosses the rotor, the air kinetic energy is transformed in rotational kinetic energy. Therefore, the fluid velocity decreases through the control volume and the energy transmitted to the turbine corresponds to the loss of air kinetic energy.

Let a be the axial induction factor of the turbine, which is the ratio of wind velocity loss along the rotor axis before the rotor plane, defined by :

$$V_{\text{plane,z}} = (1 - a)V_z$$

where $V_{\text{plane,z}}$ and V_z are the wind velocity components along the rotor axis at respectively the rotor plane and infinity. In order to induce their rotational movement, the turbine blades must exert a reactional load on the air, which produces a rotation of the air mass in the control volume in the opposite direction of the turbine rotor. Because of induction phenomenons and the air continuity, the air mass in front of the rotor plane also rotates and the local wind direction is again modified. The rotation of the air mass induced is also taken into account with the radial induction factor, denoted by a', such that the radial component of the wind velocity in the wake C_θ equals :

$$C_\theta = 2a'\omega_r r$$

where ω_r is the rotational velocity of the rotor and r the radial position of the blade element.

These two inductions factors result in the induction velocity $\overrightarrow{W} = (a'\omega_r r, aV_z)^T = (W_y, W_z)^T$ (Figure A.4). The blade relative local velocity $\overrightarrow{V_{rel}} = (V_{rel,y}, V_{rel,z})^T$ is defined as :

$$\overrightarrow{V_{rel}} = \overrightarrow{V_0} + \overrightarrow{V_{rot}} + \overrightarrow{W} \tag{A.3}$$

where $\overrightarrow{V_0} = (V_y, V_z)^T$ is the wind velocity ahead of the rotor (time and space varying), $\overrightarrow{V_{rot}} = (V_{rot,y}, 0)^T$ is the velocity due to blade rotation. Let ϕ be the angle between the blade and the direction of the local flow. In BEM theory, it is assumed that the lift force $\overrightarrow{L}$ is collinear to the induced velocity $\overrightarrow{W}$, thus $\overrightarrow{W} \perp \overrightarrow{V_{rel}}$. In BEM theory, at each time step, the quasi-steady value of the induced velocity denoted by $\overrightarrow{W_{qs}} = (||\overrightarrow{W_{qs}}|| \sin\phi, -||\overrightarrow{W_{qs}}|| \cos\phi)^T$

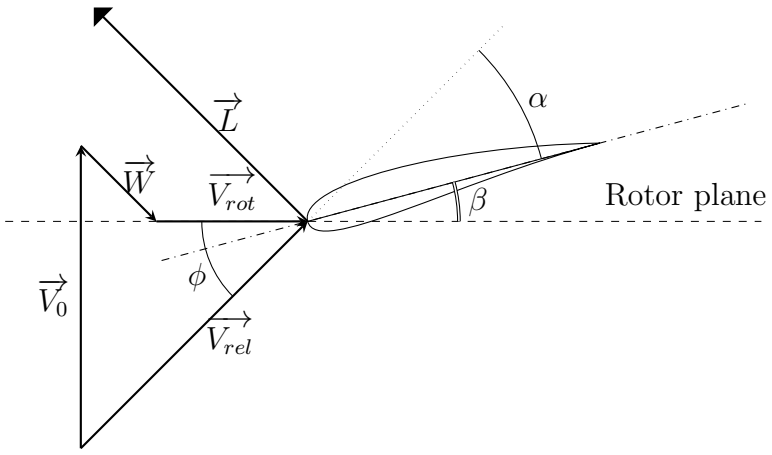

FIGURE A.4 – Illustration of the BEM problem of local speed calculation.

must be determined with the corresponding inflow angle ϕ by solving the following system :

$$\tan\phi = \frac{V_y + V_{rot,y} + ||\overrightarrow{W_{qs}}|| \sin\phi}{-V_z + ||\overrightarrow{W_{qs}}|| \cos\phi} \tag{A.4}$$

$$||\overrightarrow{W_{qs}}|| = -\frac{3c}{8\pi r} \frac{||\overrightarrow{V_{rel}}(W_{qs},\phi)||^2}{\sqrt{V_y^2 + (V_z + W_z)^2}} \frac{C_L(\phi,\beta)}{F(\phi)} \tag{A.5}$$

where β is an angle due to the addition of the blade pitch and twist angles, $F(\phi)$ is the Prandtl's tip loss factor to correct the assumption of an infinite number of blades, defined as follows :

$$F(\phi) = \frac{2}{\pi}\cos^{-1}(e^{-f(\phi)}) \tag{A.6}$$

$$f(\phi) = \frac{3}{2}\frac{R-r}{r\sin\phi} \tag{A.7}$$

where R is the radius of the blade. Once $\overrightarrow{W_{qs}}$ is estimated, the actual induced velocity at the current time instant $\overrightarrow{W}$ can be computed by integrating the following set of differential equation to account for the wake dynamic :

$$\overrightarrow{W_{int,i}} + \tau_1(a,V_0)\dot{\overrightarrow{W_{int,i}}} = \overrightarrow{W_{qs,i}} + \kappa\tau_1(a,V_0)\dot{\overrightarrow{W_{qs,i}}} \tag{A.8}$$

$$\dot{\overrightarrow{W_i}} + \tau_{2,i}(a,V_0) = \overrightarrow{W_{int,i}} \tag{A.9}$$

where $\overrightarrow{W_{int}}$ is an intermediate states of the dynamic model, the subscript i refers to the i^{th} annular elements, τ_1 and τ_2 are time constants of the dynamic model, defined as :

$$\tau_1(a,V_0) = \frac{1.1}{1-1.3a}\frac{R}{||\overrightarrow{V_0}||_2} \tag{A.10}$$

$$\tau_2(\tau_1(a,V_0)) = \left(0.39 - 0.26\left(\frac{r}{R}\right)^2\right)\tau_1(a,V_0) \tag{A.11}$$

To this can be added dynamics of the lift aerodynamic force (which does not react instantly to angle of attack variations), yaw and tilt misalignment corrections, which makes an additional state at each time. This results in an highly nonlinear dynamic system of at least 8 states per annular element per blade. For the rotor considered throughout

this thesis, the blade is discretized in 40 annular elements, which makes a $3 \times 8 \times 40 = 960$ states dynamic system. The outputs of the system are the loads applied to each blade element of each blade :

$$p_z = \frac{1}{2}\rho||\overrightarrow{V_{rel}}||^2 c\left(C_L(\phi,\beta)\cos\phi + C_D(\phi,\beta)\sin\phi\right)dr \quad (A.12)$$

$$p_y = \frac{1}{2}\rho||\overrightarrow{V_{rel}}||^2 c\left(C_L(\phi,\beta)\sin\phi - C_D(\phi,\beta)\cos\phi\right)dr \quad (A.13)$$

where dr is the length of the blade element. This model eventually yield an approximation of the load distribution on each blade. From these load distributions it is possible to compute the M_{Ip}, M_{Oop} and F_{Oop}, defined in (A.1), as follows :

$$M_{Ip} = \sum_j p_y(r_j) r_j \quad (A.14)$$

$$M_{Oop} = \sum_j p_z(r_j) r_j \quad (A.15)$$

$$F_{Oop} = \sum_j p_z(r_j) \quad (A.16)$$

where r_j is radius of the blade element j.

This model stands only for the aerodynamic efforts. To have an aero-elastic dynamic system, the coupling between blade deflection and aerodynamic efforts must be taken into account. An example of this coupling is that the aerodynamic efforts deflect the blade, modifying the local flow speed and angle and eventually the aerodynamic efforts.

Elastic behavior To model the elastic behavior of the blade, beam theory is used [100]. For the problem statement, the beam is considered as a technical beam with a distributed loading $\vec{p}(x)$ along it (Figure A.5). The distributed loading components are p_z and p_y, obtained from the BEM aerodynamic calculations and equations (A.12) and (A.13).
Using Newton's second law of motion, it is possible to compute from the distributed loading, the shear forces T_z and T_y, and the bending moments M_z and M_y with integration. The computations are made as follows :

$$\frac{dT_z}{dx} = -p_z(x) + m(x)\ddot{u}_z(x) \quad (A.17)$$

$$\frac{dT_y}{dx} = -p_y(x) + m(x)\ddot{u}_y(x) \quad (A.18)$$

$$\frac{dM_y}{dx} = T_z \quad (A.19)$$

$$\frac{dM_z}{dx} = -T_y \quad (A.20)$$

where $m(x)$ is the mass of an infinitesimal element of blade at radius x, $\ddot{u}_z(x)$ and $\ddot{u}_y(x)$ are the accelerations of the blade at radius x along respectively the vertical and transversal axis.

To simplify the problem, the principle of superposition is used and the bending moments are projected on the principal axis of the blades, named flapwise and edgewise, which yields M_1 and M_2 :

$$M_1 = M_y\cos(\beta+\nu) - M_z\sin(\beta+\nu) \quad (A.21)$$

$$M_2 = M_y\sin(\beta+\nu) + M_z\cos(\beta+\nu) \quad (A.22)$$

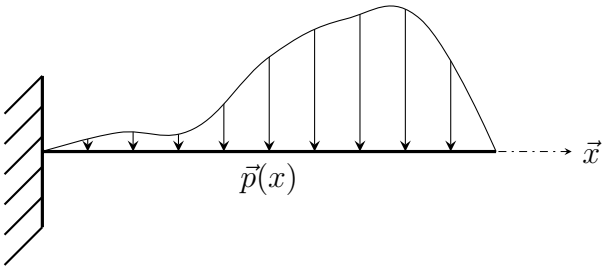

FIGURE A.5 – Technical beam with distributed loading $\vec{p}(x)$

where β is the blade pitch angle, and $\beta + \nu$ is the angle between the y axis and the first principal axis. The moments of stiffness inertia EI are well known over each of the principal axis from blade structural data. From beam theory, curvatures κ are obtained :

$$\kappa_1 = \frac{M_1}{EI_1} \tag{A.23}$$

$$\kappa_2 = \frac{M_2}{EI_2} \tag{A.24}$$

Then, with a double integration of the curvatures along x, the blade deformations over the principal axis are obtained and can be projected back on the y and z axis to close the differential equation. For computations, the blade is discretized in blade elements, whose each of them are considered as beams welded to each other.

Aero-elastic behavior Once the blade deflections are known, the deflection velocity $\overrightarrow{U} = (u_y, u_z)^T$ can be added in the velocity triangle from equation (A.3) :

$$\overrightarrow{V_{rel}} = \overrightarrow{V} + \overrightarrow{V_{rot}} + \overrightarrow{W} - \overrightarrow{U} \tag{A.25}$$

Thus, this modifies equations (A.4) and (A.5), and the system of elastic differential equations is an additional constraint for the induced velocity problem, which must be solved at each time step.

This description of the aero-elastic modelling of a rotor is very shallow, and more details can be found in [99], where all these informations have been taken from. One can now understand that the raw aero-elastic model of the rotor, which is very complex and highly nonlinear, is not directly suited for control purposes and must be simplified.

Annexe B

Rainflow counting algorithm

This appendix describes the Rainflow Counting (RFC) algorithm defined in [55], used to count the number of hysteresis cycles in a time series for fatigue damage estimation.

A preprocessing step to RFC algorithm is to only consider the reversals (maxima and minima, or peaks and valleys) of the time series. A vector $S = [s_1, \ldots, s_n]$ is thus obtained from a time series, where n is the number of reversals, and s_i is the i^{th} reversal. An illustration of vector S is available in Figure 1.17. It should be pointed out that the vector S respects the following conditions :

— s_1 is the first value of the time series.
— s_n is the last value of the times series.
— $\text{sign}(s_{i-1} - s_i) \neq \text{sign}(s_i - s_{i+1}) \quad \forall i \in \{2, \ldots, n-1\}$

RFC holds its name from an analogy with rain drops falling off a pagoda roof. In Figure 1.17, it is possible to visualize how the RFC algorithm works. To see the analogy, the pagoda roof is the blue line that represents the vector S reversals on the figure. Then, one must place the x-axis vertically with the positive end towards the bottom, and imagine the rain dropping from the top, which corresponds to the orange streaks on the figure. The full cycles correspond to the pair of streaks that are surrounded by another pair of streaks like $\{EF, HI, JK, LM\}$ and the half cycles are the remaining pairs $\{AB, BC, CD, DG, GN, NO\}$. The RFC algorithm is detailed in Algorithm 3 where v_i is the i^{th} term of the V array.

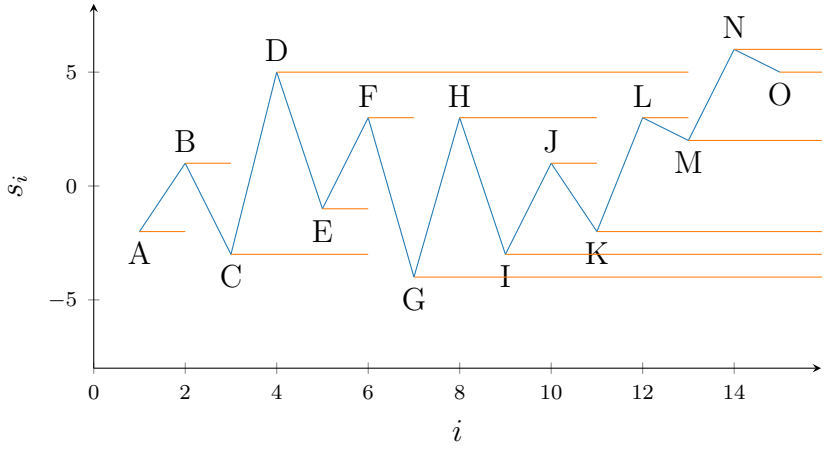

FIGURE B.1 – Illustration of the vector S and the RFC algorithm. The full cycles correspond to the pair of orange streaks that are surrounded by another pair of streaks like $\{EF, HI, JK, LM\}$ and the half cycles are the remaining pairs $\{AB, BC, CD, DG, GN, NO\}$.

Algorithm 3: RainFlow Counting (RFC) algorithm [55]

Result: Compute the number of hysteresis cycles in the vector S. N is an array containing the number of hysteresis cycles corresponding to the mean and amplitude in $\bar{L}$ and L^R respectively.

$Z \leftarrow 1$;
$i \leftarrow 0$;
$V \leftarrow \{\}$;
$N \leftarrow \{\}$;
$\bar{L} \leftarrow \{\}$;
$L^R \leftarrow \{\}$;
while $i < n$ **do**
  $m \leftarrow$ length of V;
  if $m \geq 3$ **then**
    $X \leftarrow \{v_{m-1}, v_m\}$;
    $Y \leftarrow \{v_{m-2}, v_{m-1}\}$;
    if range$(X) \geq$ range(Y) **then**
      if $m - 2 \leq Z \leq m - 1$ **then**
        Count Y ***as a half cycle :***
        $N \leftarrow \{N, \frac{1}{2}\}$;
        $\bar{L} \leftarrow \{\bar{L}, \text{mean}(Y)\}$;
        $L^R \leftarrow \{L^R, \text{range}(Y)\}$;
        Discard V of v_{m-2};
      else
        Count Y ***as a full cycle :***
        $N \leftarrow \{N, 1\}$;
        $\bar{L} \leftarrow \{\bar{L}, \text{mean}(Y)\}$;
        $L^R \leftarrow \{L^R, \text{range}(Y)\}$;
        Discard V of v_{m-2} and v_{m-1};
      end
    end
  end
  $i \leftarrow i + 1$;
  $V \leftarrow \{V, s_i\}$;
end
Count the remaining reversals as half cycles :
$m \leftarrow$ length of V;
for $i \in \{1, \ldots, m - 1\}$ **do**
  $Y \leftarrow \{v_i, v_{i+1}\}$;
  $N \leftarrow \{N, \frac{1}{2}\}$;
  $\bar{L} \leftarrow \{\bar{L}, \text{mean}(Y)\}$;
  $L^R \leftarrow \{L^R, \text{range}(Y)\}$;
end

Annexe C

Derivation of a QP analytical solution

Let be a dynamic system in discrete time :

$$x_{l+1} = Ax_l + Bu_l + B_d v_l \tag{C.1a}$$
$$y_l = Cx_l + Du_l + D_d v_l \tag{C.1b}$$

where x_l, u_l, y_l and v_l are respectively the state, input, output and disturbance of the system at the l^{th} time instant. A, B, B_d, C, D and D_d are respectively the dynamic, input to state, disturbance to state, state to output, input to output and disturbance to output matrices of the dynamic system.

Let be a discrete time optimal control problem free of inequality constraints, with an objective function expressed as the integral of a quadratic stage cost over time, such as :

$$\min_{u_0,\dots,u_N} \quad J_d = \sum_{l=1}^{N} y_l^T Q y_l + \sum_{l=0}^{N} u_l^T R u_l \tag{C.2a}$$
$$\text{s.t.} \quad x_{l+1} = Ax_l + Bu_l + B_d v_l \tag{C.2b}$$
$$y_l = Cx_l + Du_l + D_d v_l \tag{C.2c}$$
$$x_0 = X_0 \tag{C.2d}$$
$$E_i = \sum_{l=1}^{N} F_{i,l} y_l + G_{i,l} u_l \quad \forall i \in \{1, \dots, N_{\text{cons}}\} \tag{C.2e}$$

where N is the optimization horizon length, Q and R are semi positive definite weighting matrices of the stage cost, X_0 is the given initial state of the system. The matrices E_i, $F_{i,k}$ and $G_{i,k}$ are matrices which define the i^{th} equality constraint on the output and input. The number of equality constraints is denoted by N_{cons}.

Let $\mathbf{u} = [u_0^T, \dots, u_N^T]^T$, $\mathbf{v} = [v_0^T, \dots, v_N^T]^T$ and $\mathbf{y} = [y_0^T, \dots, y_N^T]^T$ be the discrete input and output trajectory. An optimal control problem such as (C.2) can be written more compactly :

$$\min_{\mathbf{u}} \quad J_d = \mathbf{y}^T \mathcal{Q} \mathbf{y} + \mathbf{u}^T \mathcal{R} \mathbf{u} \tag{C.3a}$$
$$\text{s.t.} \quad \mathbf{y} = \Phi X_0 + \Psi \mathbf{u} + \Gamma \mathbf{v} \tag{C.3b}$$
$$E = F\mathbf{y} + G\mathbf{u} \tag{C.3c}$$

where :

$$\Phi = \begin{pmatrix} CA \\ \vdots \\ CA^N \end{pmatrix} \qquad \Psi = \begin{pmatrix} CB & D & 0 & \dots & 0 \\ CAB & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ CA^{N-1}B & \dots & CAB & CB & D \end{pmatrix} \tag{C.4a}$$

$$E = \begin{pmatrix} E_1 \\ \vdots \\ E_{N_{\text{cons}}} \end{pmatrix} \qquad \Gamma = \begin{pmatrix} CB_d & Dd & 0 & \dots & 0 \\ CAB_d & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ CA^{N-1}B_d & \dots & CAB_d & CB_d & D_d \end{pmatrix} \tag{C.4b}$$

$$F = \begin{pmatrix} F_{1,1} & \dots & F_{1,N} \\ \vdots & \ddots & \vdots \\ F_{N_{\text{cons}},1} & \dots & F_{\text{N}_{\text{cons}},\text{N}} \end{pmatrix} \quad G = \begin{pmatrix} G_{1,1} & \dots & G_{1,N} \\ \vdots & \ddots & \vdots \\ G_{N_{\text{cons}},1} & \dots & G_{\text{N}_{\text{cons}},\text{N}} \end{pmatrix} \tag{C.4c}$$

$$\mathcal{Q} = \begin{pmatrix} Q & & 0 \\ & \ddots & \\ 0 & & Q \end{pmatrix} \qquad \mathcal{R} = \begin{pmatrix} R_2 & & 0 \\ & \ddots & \\ 0 & & R_2 \end{pmatrix} \tag{C.4d}$$

Injecting equation (C.3b) into (C.3a) and (C.3c), the optimal control problem defined by (C.3) becomes :

$$\min_{\mathbf{u}} \quad J_{\mathrm{d}} = \mathbf{u}^T \left(\Psi^T \mathcal{Q} \Psi + \mathcal{R}\right) \mathbf{u} + 2\mathbf{u}^T \Psi^T \mathcal{Q} \left(\Phi X_0 + \Gamma \mathbf{v}\right) + C \tag{C.5a}$$

$$\text{s.t.} \quad E = F\Phi X_0 + (F\Psi + G)\,\mathbf{u} + F\Gamma \mathbf{v} \tag{C.5b}$$

where $C \in \mathbb{R}$ is a constant.

Therefore, from the Lagrangian of (C.5), the optimality conditions can be derived and summarized by the following linear equations :

$$\underbrace{\begin{pmatrix} 2\left(\Psi^T \mathcal{Q} \Psi + \mathcal{R}\right) & (F\Psi + G)^T \\ F\Psi + G & 0 \end{pmatrix}}_{\mathcal{H}} \begin{pmatrix} \mathbf{u} \\ q \end{pmatrix} + \begin{pmatrix} 2\Psi^T \mathcal{Q} \Phi \\ F\Phi \end{pmatrix} X_0 + \begin{pmatrix} 2\Psi^T \mathcal{Q} \Gamma \\ F\Gamma \end{pmatrix} \mathbf{v} - \begin{pmatrix} 0 \\ E \end{pmatrix} = 0 \tag{C.6}$$

where q is the Lagrangian costate. Eventually, by solving off-line the above equation, which needs the inversion of the matrix $\mathcal{H}$, matrices K_1, K_2 and K_3 can be retrieved such that the optimal solution $\mathbf{u}^\star$ is expressed as follows :

$$\mathbf{u}_{\mathrm{z}}^\star = -K_1 X_0 - K_2 \mathbf{v} - K_3 \tag{C.7}$$

Therefore, the solution of the optimal control problem defined by (C.2) is summarized by a linear algebraic equation, which is less computationally expensive than solving directly the optimization problem with a QP.

Annexe D

Examples of fatigue-oriented cost function derivation

This appendix aims at highlighting examples of relationships between fatigue damage and common quadratic forms used in optimal control.

Data generation

In order to derive the relationships between fatigue damage and quadratic features, time series are needed. The time series selected for the relationships derivations are all Gaussian noise filtered through the following system :

$$\dot{x} = -\frac{1}{\tau(n)}x + \frac{K(n)}{\tau(n)}u \tag{D.1a}$$

$$y = x + \varepsilon(n) \tag{D.1b}$$

where $x, u, y \in \mathbb{R}$ are respectively the system state, input and output, $n \in \mathbb{N}$ is the simulation number. The parameters of the system τ, K and ε are respectively the time constant, static gain and offset, whose values are randomly drawn between every single generation of time series. In the sequel, for all simulations time series are sampled at 0.1 second, 200 seconds long and are starting from the system steady-state. Several cases of the parameters τ, K and ε variations are considered, which are summarized in Table D.1, where $\mathcal{N}(\mu, \sigma)$ refers to a Gaussian distribution of mean μ and standard deviation σ, and $[a, b]$ refers to an uniform distribution between the values a and b.

Case number	Variation of τ	Variation of ε	Variation of K
1	$\tau = 1$	$\varepsilon = 0$	$K \sim [1, 11]$
2	$\tau \sim [1, 11]$	$\varepsilon = 0$	$K = 1$
3	$\tau = 1$	$\varepsilon \sim \mathcal{N}(0, 10)$	$K = 1$
4	$\tau \sim [1, 11]$	$\varepsilon \sim \mathcal{N}(0, 10)$	$K \sim [1, 11]$

TABLE D.1 – Summary of the parameters τ, K and ε variations in function of the cases considered.

Fatigue estimation

The damage estimation are performed using the Palmgrem-Miner rule with the Goodman correction and the hysteresis cycles are counted using the RFC algorithm. The ultimate load selected is $L^{\text{ult}} = 20$ and the time series where the maximal absolute load goes past L^{ult} are not considered in the regression. Two Wöhler coefficients are considered for the fatigue damages estimation, $m = 4$ and 10, which corresponds respectively to steel and glass fiber.

Quadratic forms

The quadratic forms considered in the examples presented below are the variance J_{Var} defined by (3.1b), the mean square $J_{\text{MS},0}$, the mean square of the first derivative $J_{\text{MS},1}$ and the mean square of the second derivative $J_{\text{MS},2}$ defined by (3.1c, and the average $J_{\text{L},0}$ defined by (3.1a). From $J_{\text{L},0}$ is derived a new feature

$$J_{\mu}(\mathbf{y}) = L^{\text{ult}} - |J_{\text{L},0}(\mathbf{y})|$$

which is inspired from the spectral expression of fatigue detailed in Subsection 1.3.3 and where $\mathbf{y}$ is the output trajectory.

Regression	$\mathbf{J}(\mathbf{y})$
Reg1	$J_{\text{Var}}(\mathbf{y})$
Reg2	$[J_{\text{Var}}(\mathbf{y}), J_{\text{MS,0}}(\mathbf{y})]^T$
Reg3	$[J_{\text{Var}}(\mathbf{y}), J_{\text{MS,0}}(\mathbf{y}), J_{\mu}(\mathbf{y})]^T$
Reg4	$[J_{\text{Var}}(\mathbf{y}), J_{\text{MS,0}}(\mathbf{y}), J_{\mu}(\mathbf{y}), J_{\text{MS,1}}(\mathbf{y})]^T$

TABLE D.2 – Summary of the quadratic features used in the regressions.

Regressions considered

The regressions between the quadratic features and fatigue damage need the definition of a vector of quadratic features $\mathbf{J}(\mathbf{y})$. Several regressions were considered with different vector of quadratic features, which are summarized in Table D.2. Apart from the vector of quadratic features, all the regressions are using the same structure, i.e. a linear regression between the logarithms of the vector of wind features and fatigue damage. Therefore the estimated fatigue damage has the following expression :

$$\hat{\mathcal{D}}(\mathbf{y}) = e^{b} e^{w^T \mathbf{J}(\mathbf{y})} \tag{D.2}$$

where w and b are respectively the vector of weights and bias obtained from the regression.

Regression evaluation metrics

In order to test the accuracy of the resulting regression on the testing set, let us define the R^2 and R^2_{exp} scores :

$$R^2 = 1 - \frac{||\log \mathcal{D} - \log \hat{\mathcal{D}}||_2^2}{||\log \mathcal{D}||_2^2} \qquad R^2_{\text{exp}} = 1 - \frac{||\mathcal{D} - \hat{\mathcal{D}}||_2^2}{||\mathcal{D}||_2^2} \tag{D.3a}$$

The R^2 score can be seen as a general quality of the regression, in the sense that it assess the prediction accuracy with the same importance for very low values and very high values. On the other hand, R^2_{exp} also evaluates the quality of predictions, but more importance will be given to high fatigue damage values.

Now that all the necessary information is defined for the fatigue damage regressions from quadratic features derivation and evaluation, regressions are performed for every individual cases in the following.

D.1 Case 1 : Varying static gain

In the first case, it is considered that $\tau = 1$, $\varepsilon = 0$ and K is randomly drawn for every simulations, which corresponds to an LTI system. In Figures D.1, D.2 and D.3, scatter of respectively J_{var}, $J_{\text{MS,0}}$ and J_{μ} are plotted in the logarithmic scale. On this scale, clear trends can be seen for J_{var} and $J_{\text{MS,0}}$, for both Wöhler coefficients, where the relation between these latter and fatigue damage is affine. Concerning J_{μ}, as it varies very little, no clear trend can be seen, which was expected as the offset on the system output is fixed to zero.

The regressions Reg1 and Reg2 are performed as the information on J_{μ} and $J_{\text{MS,1}}$ are not necessary on this data. Summary of the values of w, b, R^2 and R^2_{exp} obtained from the regressions Reg1 and Reg2 can be found in Table D.3. It can be noticed that the performance achieved by the two regressions are very similar and it seems that the

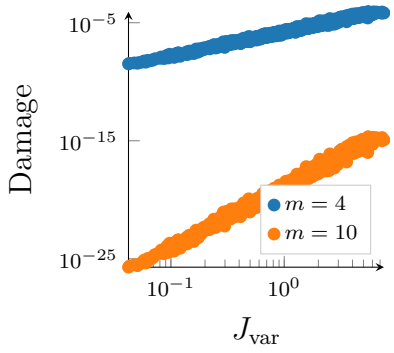

FIGURE D.1 – Scatter of J_{var} and fatigue damage, for $\tau = 1$, $\varepsilon = 0$ and random K in the logarithmic scale. (Case 1)

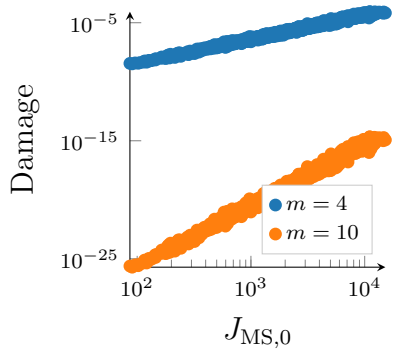

FIGURE D.2 – Scatter of $J_{\text{MS,0}}$ and fatigue damage, for $\tau = 1$, $\varepsilon = 0$ and random K in the logarithmic scale. (Case 1)

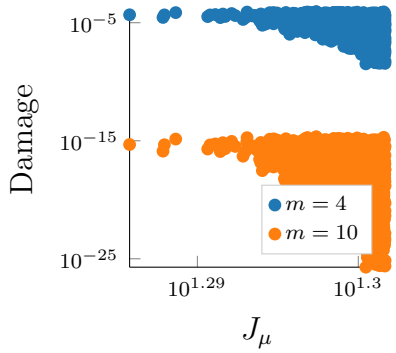

FIGURE D.3 – Scatter of J_{μ} and fatigue damage, for $\tau = 1$, $\varepsilon = 0$ and random K in the logarithmic scale. (Case 1)

addition of $\log J_{\text{MS,0}}$ does not bring any improvement to the regression. It even worsens the R^2_{exp} score for $m = 10$, which could be expected as J_{var} and $J_{\text{MS,0}}$ are almost equal in this case.

Regression	m	w^T	b	R^2	R^2_{exp}
Reg1	4	$[2.07]$	-13.26	0.99	0.94
	10	$[5.16]$	-43.51	0.99	0.82
Reg2	4	$[1.84, 0.23]$	-15.01	0.99	0.94
	10	$[4.59, 0.57]$	-47.89	0.99	0.79

TABLE D.3 – Summary of regression Reg1 and Reg2 on data generated under Case 1.

D.2 Case 2 : Varying time constant

For the second case, the time constant τ is randomly drawn between simulations, $\varepsilon = 0$ and $K = 1$, which corresponds to an LPV system. In Figures D.4, D.5 and D.6, the scatter of respectively J_{var}, $J_{\text{MS,0}}$ and J_{μ} against their corresponding fatigue damage are plotted in the logarithmic scale for 100 simulations. An affine relation can be seen for J_{var} and $J_{\text{MS,0}}$, however it is less obvious than in case 1. Concerning J_{μ}, as $\varepsilon = 0$, J_{μ} varies even less than in case 1 and its variation are difficult to perceive.

The regressions Reg1 and Reg2 are then performed on the generated data and the results are summarized in Table D.4. It can be seen that Reg1 and Reg2 performance are similar and greatly decreased compared to case 1.

Regression	m	w^T	b	R^2	R^2_{exp}
Reg1	4	$[3.15]$	-10.85	0.88	0.70
	10	$[7.87]$	-37.48	0.88	0.21
Reg2	4	$[2.41, 0.79]$	-16.65	0.88	0.69
	10	$[6.03, 1.97]$	-51.98	0.88	0.22

TABLE D.4 – Summary of regression Reg1 and Reg2 on data generated under Case 2.

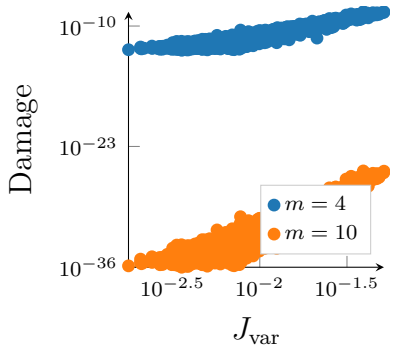

FIGURE D.4 – Scatter of J_{var} and fatigue damage, for random τ, $\varepsilon = 0$ and $K = 1$ in the logarithmic scale. (Case 2)

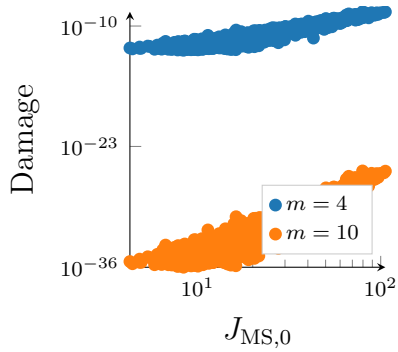

FIGURE D.5 – Scatter of $J_{\text{MS},0}$ and fatigue damage, for random τ, $\varepsilon = 0$ and $K = 1$ in the logarithmic scale. (Case 2)

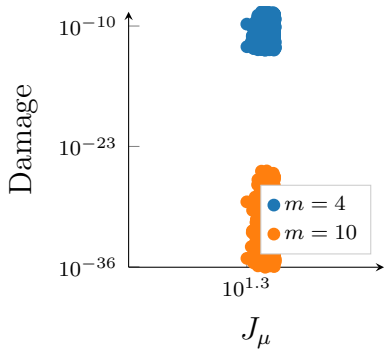

FIGURE D.6 – Scatter of J_{μ} and fatigue damage, for random τ, $\varepsilon = 0$ and $K = 1$ in the logarithmic scale. (Case 2)

D.3 Case 3 : Varying offset

The third case has fixed time constant and static gain to one, with randomly drawn offset ε. In Figures D.7, D.8 and D.9, the scatter of respectively J_{var}, $J_{\text{MS},0}$ and J_{μ} against their corresponding fatigue damages are plotted in the logarithmic scale for 100 simulations. As τ and K are fixed, the variations of J_{var} are very weak and no trend can be seen for J_{var}. As ε varies on a relatively wide range, so is J_{μ}, therefore J_{var} is not proportional anymore to $J_{\text{MS},0}$ and a polynomial relation can be seen between the logarithms of $J_{\text{MS},0}$ and fatigue damage. Concerning J_{μ}, an obvious affine relationship between its logarithm and the one of fatigue damage can be observed.

Therefore, the regression Reg3 is performed in addition to Reg1 and Reg2. In Table D.5, a summary of the regressions Reg1, Reg2 and Reg3 performed on Case 3 data is given. It can be seen that as expected Reg1 can hardly predict anything in these conditions, yielding very low R^2_{exp} scores, since it lacks the information on the varying offset. Reg2 has somehow the information on the offset, since it has the variance and the sum of squares. However it is nonlinearly related, as can be observed in Figure D.9, therefore Reg2 is not tailored to predict fatigue damage in this third case. As opposite, Reg3 which takes into account the information on J_{μ} achieves much better regression performances, predicting accurately fatigue damage. It is interesting to notice that in analyzing the w value, Reg3 ignores $J_{\text{MS},0}$.

Regression	m	w^T	b	R^2	R^2_{exp}
Reg1	4	$[0.78]$	-15.10	0.43	-0.04
	10	$[1.95]$	-48.11	0.427	0
Reg2	4	$[0.60, 0.78]$	-24.01	0.60	0.01
	10	$[1.51, 1.95]$	$-70, 40$	0.60	0
Reg3	4	$[0.28, 0, -3.98]$	-6.75	0.99	0.99
	10	$[0.69, 0, -9.96]$	-9.96	0.99	0.98

TABLE D.5 – Summary of regression Reg1, Reg2 and Reg3 on data generated under Case 3.

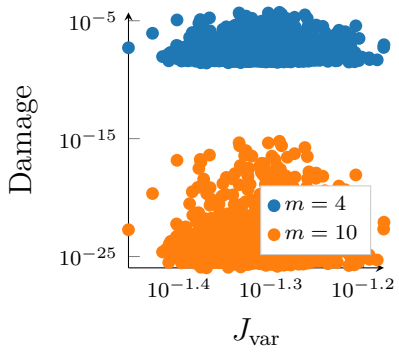

FIGURE D.7 – Scatter of J_{var} and fatigue damage, for $\tau = 1$, random ε and $K = 1$ in the logarithmic scale. (Case 3)

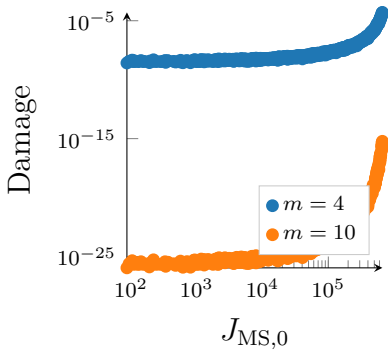

FIGURE D.8 – Scatter of $J_{\text{MS},0}$ and fatigue damage, for $\tau = 1$, random ε and $K = 1$ in the logarithmic scale. (Case 3)

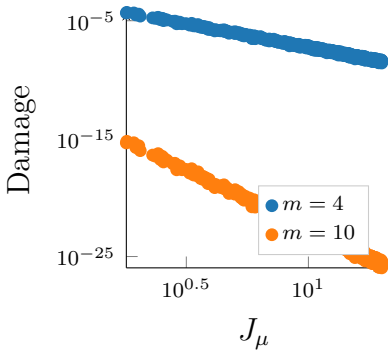

FIGURE D.9 – Scatter of J_{μ} and fatigue damage, for $\tau = 1$, random ε and $K = 1$ in the logarithmic scale. (Case 3)

D.4 Case 4 : All parameters varying

For the last case, τ, ε and K are all randomly drawn. In Figures D.10, D.11 and D.12, the scatter of respectively J_{var}, $J_{\text{MS},0}$ and J_{μ} against their corresponding fatigue damages are plotted in the logarithmic scale for 100 simulations. An affine relation can be seen between the logarithms of J_{var} and fatigue damage, which is less obvious than in the first and second cases. Concerning the relationship of the logarithms of J_{var} and J_{μ}, with the one of the fatigue damage, no clear trend can be spotted. In Figure D.13 is plotted the scatter of $J_{\text{SS}\dot{y}}$ and corresponding fatigue damages in the logarithmic scale. It can be clearly seen that there is an affine relationship between the logarithms of $J_{\text{MS},1}$ and fatigue damage.

Hence, the regression Reg4 is performed in addition to Reg1, Reg2 and Reg3. In Table D.6, the summary of the regressions Reg1, Reg2, Reg3 and Reg4 can be found. It can be seen that Reg1, Reg2 and Reg3 achieves to roughly predict fatigue damage, with some very bad R^2_{exp} scores which are mainly driven by the quality of prediction of the high fatigue damage values. On the other hand, Reg4 manages to obtain regression of great quality. Once again, as the information of the signal sum of squares and average is present twice due to the linear dependencies between J_{var}, $J_{\text{MS},0}$ and J_{μ}. Hence $J_{\text{MS},0}$ is neglected by the regression.

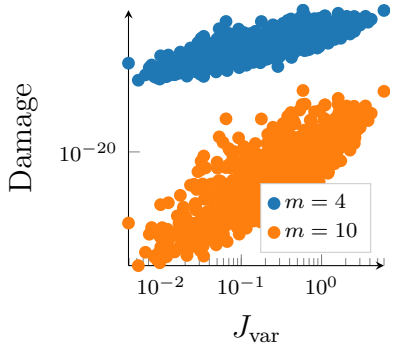

FIGURE D.10 – Scatter of J_{var} and fatigue damage, for random τ, ε and K in the logarithmic scale. (Case 4)

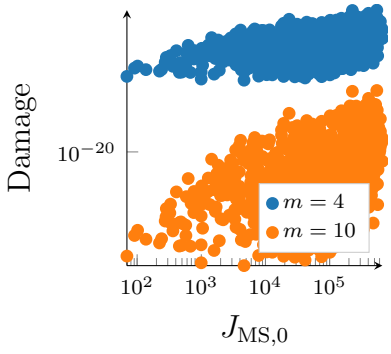

FIGURE D.11 – Scatter of $J_{\text{MS},0}$ and fatigue damage, for random τ, ε and K in the logarithmic scale. (Case 4)

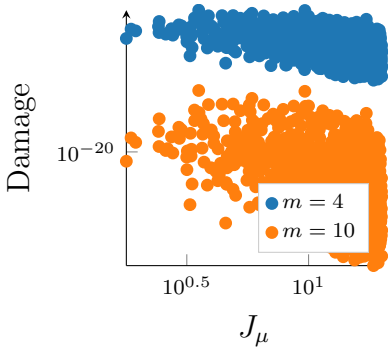

FIGURE D.12 – Scatter of J_{μ} and fatigue damage, for random τ, ε and K in the logarithmic scale. (Case 4)

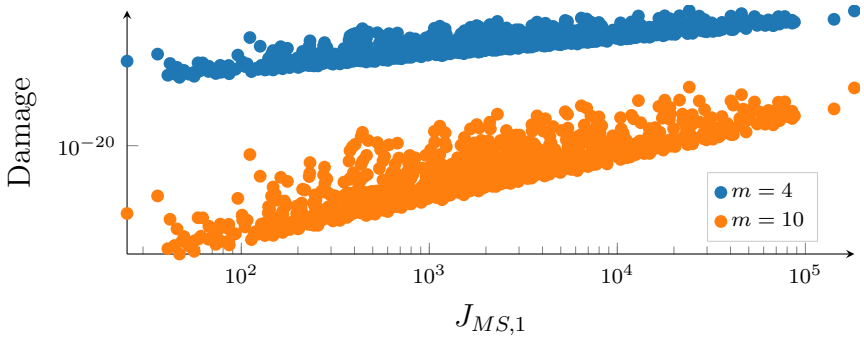

FIGURE D.13 – Scatter of $J_{\text{MS},1}$ and fatigue damage, for random τ, ε and K in the logarithmic scale. (Case 4)

Regression	m	w^T	b	R^2	R^2_{exp}
Reg1	4	$[2.20]$	-14.06	0.68	0.04
	10	$[5.50]$	-45.50	0.68	0.10
Reg2	4	$[2.19, 0.86]$	-23.42	0.84	0.13
	10	$[5.47, 2.14]$	-68.90	0.84	0
Reg3	4	$[2.29, 0.05, 3.79]$	-5.00	0.91	0.37
	10	$[5.73, 0.12, -9.48]$	-22.87	0.91	-2.06
Reg4	4	$[0.11, 0, 1.89, -3.94]$	-21.61	0.99	0.99
	10	$[0.27, 0, 47, -9.86]$	-64.38	0.99	0.98

TABLE D.6 – Summary of regression Reg1, Reg2, Reg3 and Reg4 on data generated under Case 4.

Annexe E

Decomposition of the optimization horizon

This appendix presents the decomposition of the optimization horizon of an optimal control problem using the integral of a quadratic stage cost over time as objective function, free of constraints :

$$\min_{u_0,\dots,u_N} \quad J_d = \sum_{l=1}^{N} (y_l - \mu)^T Q (y_l - \mu) + \sum_{l=0}^{N} u_l^T R u_l \tag{E.1a}$$

$$\text{s.t.} \quad x_{l+1} = Ax_l + Bu_l + B_d v_l \tag{E.1b}$$

$$y_l = Cx_l + Du_l + D_d v_l \tag{E.1c}$$

$$x_0 = X_0 \tag{E.1d}$$

where x_l, u_l, y_l and v_l are respectively the state, input, output and disturbance of the system at the k^{th} time instant. A, B, B_d, C, D and D_d are respectively the dynamic, input to state, disturbance to state, state to output, input to output and disturbance to output matrices of the dynamic system. N is the optimization horizon length, Q and R are semi positive definite weighting matrices of the stage cost, X_0 is the given initial state of the system and μ is an offset on the output.

The solution to the QP defined by (E.1), denoted by $u^\star$ is a linear combination of X_0, $\mathbf{v}$ and μ :

$$u^\star = K_1 X_0 + K_2 \mathbf{v} + K_3 \mu \tag{E.2}$$

where K_1, K_2 and K_3 are given by the resolution of the QP defined by (E.1). The resolution of the latter QP requires a matrix inversion of dimension $[N \times n_{\text{u}}, N \times n_{\text{u}}]$, where n_{u} is the dimension of the system input. Therefore, the longer the prediction horizon is, the more RAM and CPU time are required.

The optimization horizon of a QP can be temporally decomposed as in [85, 86], allowing to significantly alleviate the computational cost. The temporal decomposition of a 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 (E.1) is recovered by solving another QP managing the initial states of the smaller QPs. However, if the cost function J_d is actually variance, the cost function on the smaller intervals is not independent from the others, as μ becomes the average of the outputs over the whole horizon :

$$\mu = \frac{1}{N} \sum_{k=1}^{N} y_l \tag{E.3}$$

Therefore, μ 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 E.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^{th} interval, for $j \in \{1, \dots, M\}$, denoted by QPj, is parameterized by the initial condition and disturbance over 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 matrices Q and R in their stage cost, moreover μ which is

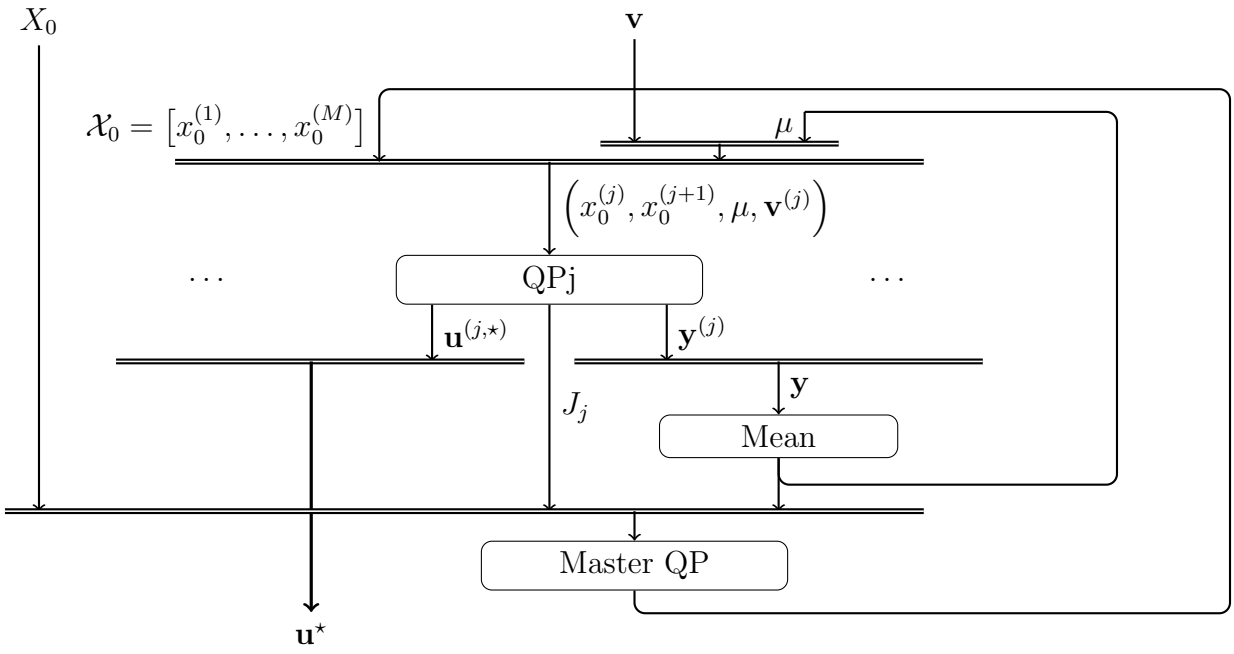

FIGURE E.1 – Schematization of the temporal decomposition scheme.

considered as an external parameter :

$$\min_{\mathbf{u}^{(j)}} \quad J_j = \sum_{l=l_0(j)}^{l_f(j)} \left(y_l^{(j)}\left(\mathbf{u}^{(j)}, \mathbf{v}^{(j)}, x_0^{(j)}\right) - \mu\right)^T Q \left(y_l^{(j)}\left(\mathbf{u}^{(j)}, \mathbf{v}^{(j)}, x_0^{(j)}\right) - \mu\right) + u_l^{(j)^T} R u_l^{(j)} \tag{E.4a}$$

$$\text{s.t.} \quad x_0^{(j+1)} = x_f\left(\mathbf{u}^{(j)}, \mathbf{v}^{(j)}, x_0^{(j)}\right) \tag{E.4b}$$

where $\mathbf{u}^{(j)}$ is the input trajectory on the j^{th} interval, $u_l^{(j)}$ is the input value at the l^{th} time instant in $\mathbf{u}^{(j)}$, $l_0(j) = (j-1)\frac{N}{M} + 1$ and $l_f(j) = j\frac{N}{M}$ are respectively the indices of the first and last time instants of the j^{th} interval. $y_l^{(j)}$ and x_f are affine functions giving respectively the output value at the l^{th} time instant and the j^{th} interval final state for given $x_0^{(j)}$, $\mathbf{v}^{(j)}$ and $\mathbf{u}^{(j)}$. It should be noticed that that $x_0^{(1)} = X_0$, which is the initial state of the QP defined by (E.1).

The solution to (E.4), denoted by $\mathbf{u}^{(j,\star)}$, is an affine function of $x_0^{(j)}$, $x_0^{(j+1)}$, $\mathbf{v}^{(j)}$ and μ :

$$u^{(j,\star)} = \tilde{k}_1^{(j)} x_0^{(j)} + \tilde{k}_2^{(j)} \mathbf{v}^{(j)} + \tilde{k}_3^{(j)} \mu + \tilde{k}_4^{(j)} x_0^{(j+1)} \tag{E.5}$$

where $\tilde{k}_1^{(j)}$, $\tilde{k}_2^{(j)}$, $\tilde{k}_3^{(j)}$ and $\tilde{k}_4^{(j)}$ are given by the resolution of the QP defined by (E.4). Besides each resolution of a QPj needs a matrix inversion of dimension $[\frac{N}{M} \times n_{\text{u}}, \frac{N}{M} \times n_{\text{u}}]$.

The output trajectory, denoted by $\mathbf{y}^{(j)}$, which is a concatenation of the

$$y_l^{(j)}\left(\mathbf{u}^{(j,\star)}, \mathbf{v}^{(j)}, x_0^{(j)}\right) \quad \forall l \in \{l_0(j), \ldots, l_f(j)\}$$

is an affine function of $x_0^{(j)}$, $\mathbf{u}^{(j,\star)}$ and $\mathbf{v}^{(j)}$, because the system is linear. Moreover, by composition of affine functions $\mathbf{y}^{(j)}$ is also an affine function of $x_0^{(j)}$, $x_0^{(j+1)}$, $\mathbf{v}^{(j)}$ and μ. Hence, J_j, which is defined by (E.4a), can be expressed as a quadratic form of $x_0^{(j)}$, $x_0^{(j+1)}$,

$\mathbf{v}^{(j)}$ and μ.

Let $\mathcal{X}_0 = [x_0^{(1)}, \dots, x_0^{(M+1)}]$ be a vector concatenating the initial conditions of every intervals. The cost function J_j can thus be expressed as a quadratic form of $\mathcal{X}_0$ and μ, given the disturbance trajectory $\mathbf{v}$ and X_0. Therefore the cost function $J_d(\mathbf{y})$ of the QP defined by (E.1) can be expressed as the sum of the $J_j \quad \forall j \in \{1, \dots, M\}$. On the other hand, μ 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 μ an affine function, denoted by $\mathcal{F}$, of $\mathcal{X}_0$ and μ, given X_0 and $\mathbf{v}$. The master QP that manages the vector of initial conditions $\mathcal{X}_0$, such that the concatenation of $\mathbf{u}^{(j,\star)} \quad \forall j \in \{1, \dots, M\}$ matches the solution of (E.1) is the following :

$$\min_{\mathcal{X}_0} \qquad J_d = \sum_{j=1}^{M} J_j \left(\mathcal{X}_0, \mu | \mathbf{v}^{(j)}, x_0 \right) \tag{E.6a}$$

$$\text{s.t.} \qquad \mu = \mathcal{F} \left(\mathcal{X}_0, \mu | \mathbf{v}^{(j)}, x_0 \right) \tag{E.6b}$$

where the implicit equation defined by (E.6b) is actually a linear equality constraint allowing to retrieve μ. The solution of (E.6) is thus a linear combination of X_0 and $\mathbf{v}$, requiring a matrix inversion of dimension $[n_\mathrm{x} \times (M+1), n_\mathrm{x} \times (M+1)]$ where n_x is the state dimension. Eventually, the optimal input trajectory $\mathbf{u}^{(\star)}$ can be recovered by concatenating the solutions of the QPj defined by (E.4), denoted by $\mathbf{u}^{(j,\star)}$ with the $x_0^{(j)}$ and $x_0^{(j+1)}$ obtained from the solution of (E.6).

Résumé étendu en français

Introduction

Les éoliennes à axe horizontal, qui sont la figure de proue de l'énergie éolienne sont devenues une technologie mature. Bien que l'éolien ait une tendance mondiale à la hausse, l'accroissement de la capacité installée faiblit dans certains pays, à cause des réductions des subventions. Dans un contexte de réchauffement climatique et de transition énergétique, il est de première importance d'optimiser le coût de l'éolien, de manière à rendre cette énergie compétitive face aux énergies fossiles, et ainsi permettre un développement de l'éolien moins dépendant des subventions économiques. Le contrôle des éoliennes peut fortement contribuer à la réponse à cette problématique [1].

Le contrôle individuel des pâles (IPC) d'éoliennes peut permettre de modifier les propriétés aérodynamiques du rotor, et ainsi réguler les charges déséquilibrées induites par des vents asymétriques [10]. Réguler ces déséquilibres peut aider à réduire la fatigue des pièces de l'éoliennes en rotation, telle que les pâles. Par ailleurs, le contrôle IPC est connu pour accroître l'activité des actionneurs de pâles, induisant une fatigue additionnelle sur les actionneurs ainsi que les roulements portant les pâles. Par conséquent, le contrôle IPC peut avoir des effets positifs sur la fatigue de certaines pièces, mais négatifs sur d'autres [12, 13, 29]. Pour optimiser le coût de l'énergie éolienne, il est nécessaire qu'un contrôleur IPC soit optimisé de manière à pondérer efficacement le compromis entre les fatigues des divers composants de l'éolienne.

Une manière d'évaluer le coût économique d'une éolienne dû à la fatigue mécanique est de créer une fonction de coût fatigue $\mathcal{J}$, exprimée comme la somme des fatigues de composants individuels, pondérées par leur prix de remplacement :

$$\mathcal{J}(\mathbf{y}) = \sum_{k=1}^{N_\mathrm{c}} \pi_k \mathcal{D}_k(\mathbf{y})$$

où k correspond au composant k de l'éolienne, N_c est le nombre de composants considérés dans la fonction de coût, $\mathcal{D}_k$ et π_k sont respectivement les dommages et prix de remplacement du composant k et $\mathbf{y}$ est une trajectoire des sorties de l'éolienne. Notons que les dommages $\mathcal{D}_k$ sont compris entre 0 et 1, et prennent la valeur 0 lorsque le composant est neuf et supérieur à 1 lorsque le composant est rompu. L'objectif d'un contrôleur IPC est donc de minimiser cette fonction de coût, en considérant a minima les dommages sur les pâles et actionneurs de pâles dans $\mathcal{J}$. Cependant, les modèles standards de fatigue mécanique ne permettent pas d'inclure les dommages comme objectif dans un problème de contrôle optimal de manière triviale et sans faire d'approximations grossières [61, 62, 101, 68].

Cela implique que les stratégies de contrôles standards ne sont pas conçues de manière à réduire efficacement le coût $\mathcal{J}$ quand elles sont appliquées en boucle fermée d'un système, pour un large ensemble de conditions de vent auxquelles l'éolienne pourrait être soumise. Par la suite, nous considérons un vecteur ou objet, noté p, contenant les paramètres d'un contrôleur, e.g. une commande prédictive (MPC) avec ses matrices de pondérations et son horizon de prédiction, ou un contrôleur Proportionnel Intégral (PI) et ses gains proportionnel et intégral. Communément, lorsqu'un contrôleur IPC d'éolienne est conçu, son concepteur va essayer de trouver la combinaison de paramètre $\bar{p}^\star$ minimisant l'espérance de $\mathcal{J}$ pour un ensemble de conditions de vent $\mathbb{V}$:

$$\bar{p}^\star = \arg\min_p \mathbb{E}_\mathbb{V}\Big[\mathcal{J}(\mathbf{y}(\mathbf{v}, p))\Big]$$

où la trajectoire $\mathbf{y}$ dépend des conditions de vent $\mathbf{v} \in \mathbb{V}$ et des paramètres du contrôleur mis en boucle fermée p. Le paramètre $\bar{p}^\star$ peut être considéré comme finement réglé et restera fixe pendant le fonctionnement de l'éolienne dans a minima l'ensemble de conditions de vent considérées.

Il serait aussi possible de trouver le paramètre p minimisant $\mathcal{J}$ pour une condition de vent donnée $\mathbf{v} \in \mathbb{V}$ et obtenir le paramètre $p^\star(\mathbf{v})$:

$$p^\star(\mathbf{v}) = \arg\min_p \mathcal{J}(\mathbf{y}(\mathbf{v}), p)$$

Notons que le paramètre $p^\star(\mathbf{v})$ peut varier avec les conditions de vent et il est trivial de montrer que :

$$\mathbb{E}_\mathbb{V}\Big[\mathcal{J}(\mathbf{y}(\mathbf{v}), p^\star(\mathbf{v}))\Big] \leq \mathbb{E}_\mathbb{V}\Big[\mathcal{J}(\mathbf{y}(\mathbf{v}), \bar{p}^\star)\Big]$$

et

$$\mathbb{E}_\mathbb{V}\Big[\mathcal{J}(\mathbf{y}(\mathbf{v}), p^\star(\mathbf{v}))\Big] < \mathbb{E}_\mathbb{V}\Big[\mathcal{J}(\mathbf{y}(\mathbf{v}), \bar{p}^\star)\Big]$$

si $\exists \mathbf{v} \in \mathbb{V} \backslash p^\star(\mathbf{v}) \neq \bar{p}^\star$. Les propriétés ci-dessus sont illustrées dans les Figures 1 et 2.

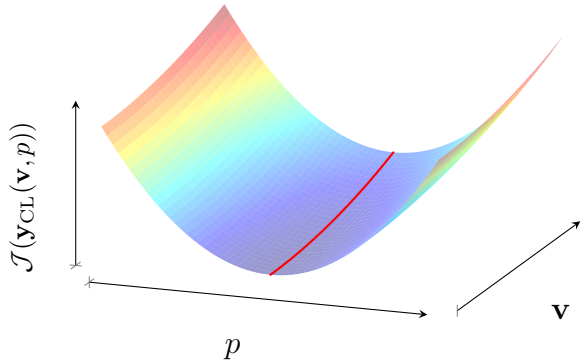

FIGURE 1 – Surface de $\mathcal{J}(\mathbf{y}(\mathbf{v}, p))$ dans le cas où $p^\star(\mathbf{v}) = \bar{p}^\star \ \forall \mathbf{v} \in \mathbb{V}$. La courbe rouge montre l'évolution de $p^\star(\mathbf{v})$ avec les conditions de vent.

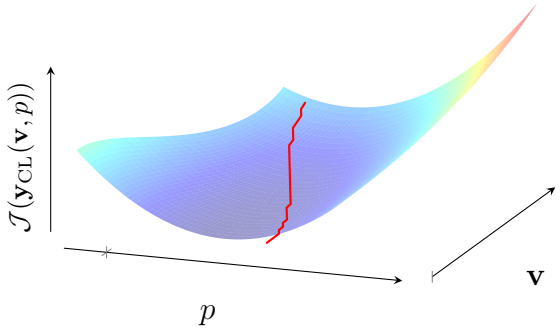

FIGURE 2 – Surface de $\mathcal{J}(\mathbf{y}_{\mathrm{CL}}(\mathbf{v}, p))$ dans le cas où $p^\star(\mathbf{v})$ varie. La courbe rouge montre l'évolution de $p^\star(\mathbf{v})$ avec les conditions de vent.

Les deux problèmes auxquels cette thèse va essayer de répondre sont les suivants :

1. Y a-t-il un avantage à adapter la combinaison de paramètres p pour une meilleure réduction de l'espérance du coût fatigue $\mathcal{J}$?
2. Si oui, la réduction de l'espérance du coût fatigue est-elle significative ? Et de quelle manière p pourrait-il être adapté efficacement ? Sachant que la considération de la fatigue comme objectif dans un problème de contrôle optimal n'est pas chose triviale.

Pour répondre à cette problématique, deux approches sont développées par la suite :

1. La première consiste à approcher le coût de la fatigue par une fonction de coût orientée fatigue, grâce à une identification basée sur des données. Cette fonction de coût orientée fatigue utilisée dans un problème de contrôle optimal permet de paramétrer un problème d'optimisation standard, qui approche le problème d'optimisation de la fatigue autour de son optimum. Ensuite, une MPC adptative basée sur cette optimisation orientée fatigue est développée puis testée en boucle fermée.
2. La deuxième approche consiste à approximer hors-ligne une cartographie de $p^\star(\mathbf{v})$ à partir d'expériences sur un simulateur. Cette cartographie sera ensuite utilisée en-ligne pour retrouver le paramètre $p^\star(\mathbf{v})$ correspondant à la condition de vent courante $\mathbf{v}$.

Fonction de coût orientée fatigue à partir de données

Il est usuel en contrôle optimal de considérer des fonction de coût quadratique comme objectif, or l'approche spectrale de la fatigue montre que la relation entre le taux de dommage et, la variance de la sortie d'un processus et de ses premières dérivées temporelles, est non linéaire [59, 60]. La variance étant une fonction quadratique, cela suggère que la relation entre la fatigue et les trajectoires de sortie d'un système est non quadratique.

Ce sont pour toutes ces raisons qu'il est proposé dans cette thèse d'observer la relation entre la valeur de fonctions quadratiques et de dommages calculés sur un ensemble de trajectoires de sorties d'une éolienne en boucle fermée. Il est ensuite possible de réaliser une régression des dommages sur les valeurs de fonctions quadratiques et obtenir une approximation de $\mathcal{J}$, notée $\hat{\mathcal{J}}$, à partir de fonction quadratique, basée sur les données. Il est ensuite possible d'utiliser l'expression de $\hat{\mathcal{J}}$ comme objectif dans un problème de contrôle optimal.

Dans ce manuscrit, l'exemple suivant est développé :

1. Un ensemble de trajectoires est obtenu en simulant des MPC paramétrées par un paramètre scalaire noté ν, pour différentes valeurs de ν, sous différents vents $\mathbf{v}$.
2. Les dommages associés à chacune des sorties et pour chacune des simulations $\mathcal{D}_k$ sont calculés.
3. Les variance $J_{\text{var}}(\mathbf{y}_k)$ sont calculées pour chaque sortie de l'éolienne et chaque simulation, où $\mathbf{y}_k$ est la trajectoire de la sortie k de l'éolienne.
4. D'après la relation observée dans la Figure 3, la relation entre les logarithmes de $J_{\text{var}}(\mathbf{y}_k)$ et $\mathcal{D}_k(\mathbf{y}_k)$ est linéaire. Et $\mathcal{D}_k$ a pu être approximé de la manière suivante :

   $$\log \mathcal{D}_k(\mathbf{y}_k) \simeq \log \hat{\mathcal{D}}_k(\mathbf{y}_k) = w_k \log(J_{\text{var}}(\mathbf{y}_k)) + b_k$$

   où w_k et b_k sont respectivement les coefficients linéaire et ordonnée à l'origine de la régression linéaire k.
5. Il est alors possible d'obtenir l'expression de $\hat{\mathcal{J}}$ de la manière suivante :

   $$\hat{\mathcal{J}}(\mathbf{y}) = \sum_{k=1}^{N_c} \pi_k \underbrace{e^{b_k} J_{\text{Var}}(\mathbf{y}_k)^{w_k}}_{\hat{\mathcal{D}}_k(\mathbf{y}_k)}$$

   dont sa structure fait qu'elle est plus adaptée à être utilisée comme fonction objectif dans un problème de contrôle optimal que $\mathcal{J}$.

Notons que grâce à la génération des signaux en boucle fermée, il est possible d'observer que la valeur de ν qui minimise $\mathcal{J}$ n'est pas tout le temps la même (Figure 4). Nous avons donc :

$$\nu_{\text{best}} = \arg\min_{\nu} \mathbb{E}_{\mathbb{V}}\Big[\mathcal{J}(\mathbf{y}(\mathbf{v}, \nu))\Big]$$

où $\mathbf{y}(\mathbf{y}, \nu)$ est la trajectoire de sortie du système en boucle fermée quand il est contrôlé par la MPC paramétrée par la valeur ν, et :

$$\nu_{\text{adapt}}(\mathbf{v}) = \arg\min_{\nu} \mathcal{J}(\mathbf{y}(\mathbf{v}, \nu))$$

Notons aussi que $\nu_{\text{adapt}}(\mathbf{v})$ est un paramètre qui est trouvé après avoir réalisé des simulations coûteuses. Par conséquent, cette méthode n'est pas utilisable pour calculer en

ligne la valeur du paramètre $\nu_{\text{adapt}}(\mathbf{v})$ à donner à la MPC. Cependant, cette MPC adaptative paramétrée par ν_{adapt} servira par la suite de "contrôleur parfait" pour comparaison.

Cela permet de répondre au premier problème soulevé par cette thèse, i.e. y a-t-il un intérêt à adapter le paramètre ν ?. En effet, l'utilisation de $\nu_{\text{adapt}}(\mathbf{v})$ comme paramètre de la MPC permet d'apporter une réduction de l'espérance de $\mathcal{J}$ de 50% par rapport à l'utilisation de ν_{best} qui est fixe.

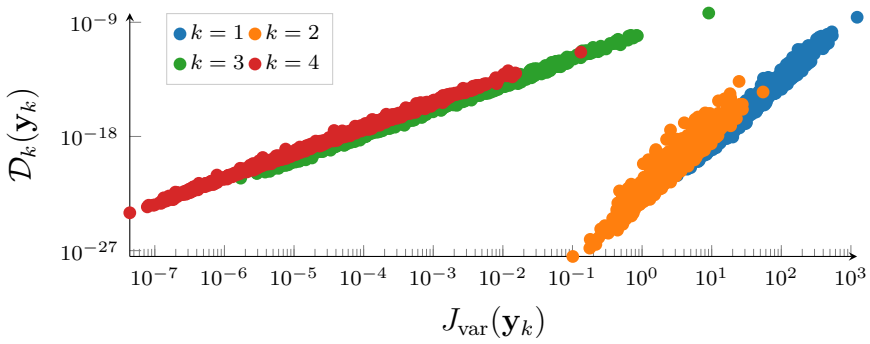

FIGURE 3 – Relation entre la variance $J_{\text{var}}(\mathbf{y}_k)$ et les dommages $\mathcal{D}_k(\mathbf{y}_k)$ des sorties du système, obtenus à partir de la méthodologie décrite ci-dessus.

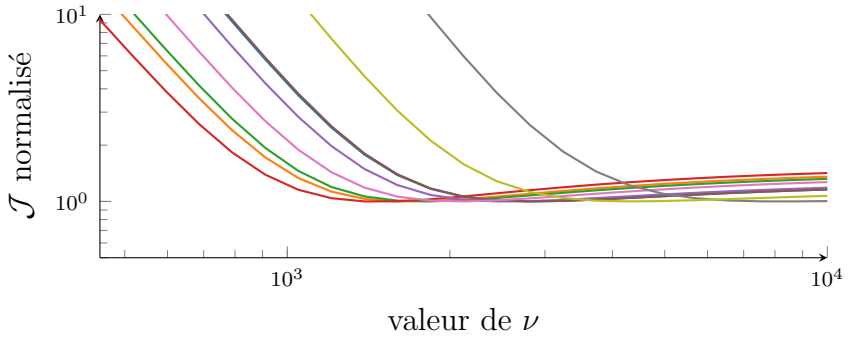

FIGURE 4 – Évolution du coût fatigue $\mathcal{J}$ avec le paramètre de la MPC quadratique ν pour 9 vents différents, normalisé par le minimum de fatigue obtenu pour chaque vent, chaque couleur représentant une réalisation du vent.

Il est possible d'utiliser la fonction de coût orientée fatigue $\hat{\mathcal{J}}$ comme objectif dans un problème de contrôle optimal en boucle ouverte, qui va s'appeler FO-OCP (Fatigue-Oriented Optimal Control Problem). FO-OCP est un problème convexe et lisse qui peut être résolu par un programme non linéaire, cependant cette résolution est coûteuse en calcul. Une résolution alternative est proposée, consistant à approximer $\hat{\mathcal{J}}$ avec une approximation de Taylor de premier ordre qui donne une fonction de coût quadratique dépendant de la variance des sorties considérées. De cette approximation, s'en suit un problème de point fixe consistant à trouver la variance permettant d'approximer correctement FO-OCP autour de l'optimum. Cette résolution alternative permet d'obtenir l'optimum global et de diminuer le coût de calcul par un à deux ordres de grandeurs selon l'horizon de prédiction considéré.

L'optimisation de FO-OCP permet d'obtenir des performances en termes de réduction du coût fatigue $\mathcal{J}$ comparable à l'optimisation quadratique paramétrée avec le paramètre $\nu_{\text{adapt}}(\mathbf{v})$.

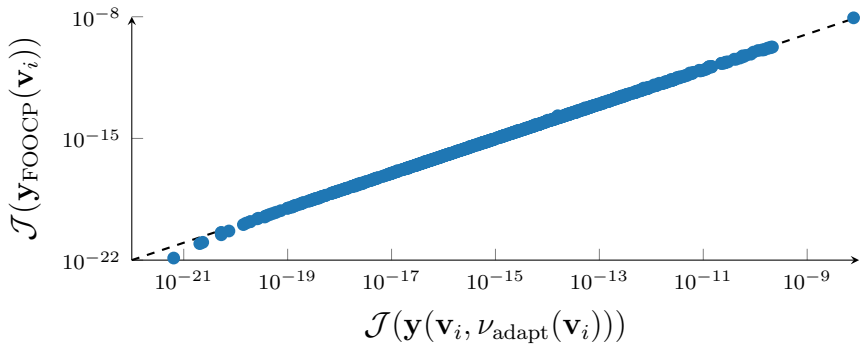

FIGURE 5 – Relation entre les coûts fatigue des solutions obtenues avec FO-OCP $\mathcal{J}(\mathbf{y}_{\text{FOOCP}}(\mathbf{v}_i))$ et l'optimisation quadratique paramétrée avec le paramètre $\nu_{\text{adapt}}(\mathbf{v})$ $\mathcal{J}(\mathbf{y}(\mathbf{v}_i, \nu_{\text{adapt}}(\mathbf{v}_i)))$, pour $i \in \{1, \ldots, 1000\}$.

Mise en œuvre en boucle fermée

La commande prédictive (MPC) est une stratégie de contrôle qui permet d'optimiser une fonction de coût spécifique, en résolvant en ligne un problème de contrôle optimal. Il semblerait naturel d'utiliser FO-OCP comme problème de contrôle optimal d'une MPC, cependant le coût de calcul nécessaire à la résolution de FO-OCP semble prohibitif pour embarquer ce contrôleur sur une éolienne.

Si le temps de résolution de FO-OCP a été réduit de manière significative grâce à la résolution alternative utilisant l'algorithme de point fixe, la résolution est toujours trop coûteuse en calcul pour une résolution en temps réel. Notons que l'algorithme de point fixe consiste à trouver les variances appropriées pour approximer FO-OCP autour de son optimum. Pour réduire encore le temps de calcul, il est proposé d'estimer en ligne ces variances à partir de mesures réalisées en lignes sur les sorties de l'éolienne. Cela permet une estimation de la variance très rapide et le problème d'optimisation de la MPC à résoudre en ligne peut être résolu par de la programmation quadratique très rapide.

Cette MPC adaptée sur le niveau de vibration de l'éolienne, notée MPC_{filt}, est mise en œuvre en boucle fermée sur un modèle d'éolienne simplifié, et simulée sous un millier de vents générés d'après une campagne de mesure. Pour comparaison, l'éolienne est aussi simulée avec la MPC quadratique paramétrée par ν_{best} (MPC_{best}) et $\nu_{\text{adapt}}(\mathbf{v})$ ($\text{MPC}_{\text{adapt}}$). Il est possible d'observer dans les Figures 6 et 7 que MPC_{filt} permet d'avoir des performances équivalentes à celles de $\text{MPC}_{\text{adapt}}$ dans 85% des cas, et meilleures que MPC_{best} dans 95% des cas, ce qui permet une réduction du coût fatigue $\mathcal{J}$ de 30% par rapport à l'éolienne contrôlée par MPC_{best}. Il est néanmoins possible d'observer qu'il subsiste quelque cas où les performances ne sont pas satisfaisantes. Ceux-ci sont principalement dûs à de mauvaises initialisations du niveau de vibration dans MPC_{filt}, et au fait que des simulations de 50 secondes sont trop courtes par rapport à la période de régime transitoire induite par ses mauvaises initialisations.

Sur-couche dîte « Supervisory »

Nous avons pu constater précédemment qu'une MPC quadratique paramétrée adaptée par un paramètre $\nu_{\text{adapt}}(\mathbf{v})$ en fonction des conditions de vent $\mathbf{v}$ pouvait permettre une réduction significative de la fatigue, par rapport à une MPC à paramètres fixes finement réglée. L'inconvénient est que le paramètre ν_{adapt} a été trouvé après avoir simuler des MPC

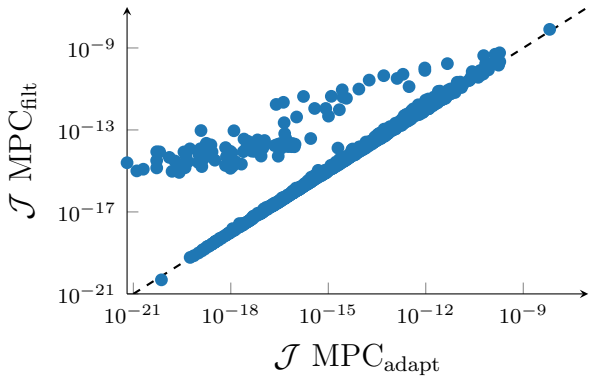

FIGURE 6 – Comparaison des coûts fatigue $\mathcal{J}$ de l'éolienne en boucle fermée avec MPC$_{\text{filt}}$ et MPC$_{\text{adapt}}$, sous 1000 vents.

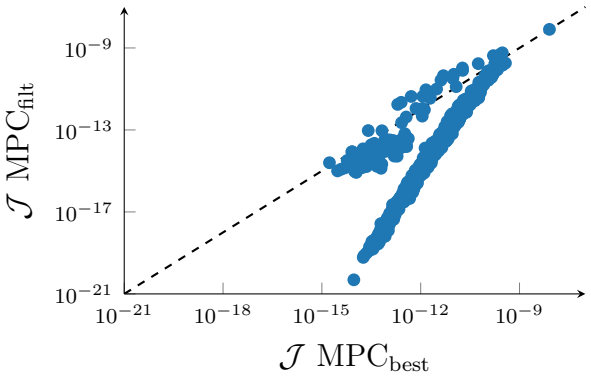

FIGURE 7 – Comparaison des coûts fatigue $\mathcal{J}$ de l'éolienne en boucle fermée avec MPC$_{\text{filt}}$ et MPC$_{\text{best}}$, sous 1000 vents.

paramétrées avec différents ν sous une même condition de vent $\mathbf{v}$, pour enfin trouver le ν minimisant $\mathcal{J}$ pour la condition de vent courante, ce qui est très coûteux en calcul. Pour pallier cela, une méthode consistant à approximer une cartographie associant une condition de vent $\mathbf{v}$ aux paramètres d'un contrôleur minimisant le coût fatigue $p^\star(\mathbf{v})$, basée sur des expériences. Ce qui dans le cas de la MPC paramétrée, consisterait à approximer la fonction $\nu_{\text{adapt}}(\mathbf{v})$.

Pour cela, une éolienne doit être simulée en boucle fermée avec un ensemble de contrôleurs candidats, qui ont potentiellement de bonnes performances, sous un ensemble de vents réalistes. Le coût fatigue $\mathcal{J}$ de chaque simulation doit être évalué, de manière à obtenir une cartographie $\mathcal{Y}$, associant un vent et un contrôleur candidat à un coût fatigue. Enfin, à partir de cette cartographie, il est possible de faire apprendre à un modèle de substitution :

— La cartographie $\mathcal{Y}$, permettant d'avoir pour une condition de vent courante, une estimation du coût fatigue en boucle fermée avec chacun des contrôleurs candidats, de manière à choisir le contrôleur qui minimiserait le coût fatigue en boucle fermée (avec une régresseur).
— La probabilité qu'un contrôleur candidat minimise le coût fatigue pour une condition de vent donnée (avec une classifieur).

Une preuve de concept est proposée dans cette thèse, dans un environnement de simulation très réaliste, avec quatre PI comme contrôleurs candidats, où le modèle de substitution est un classifieur à processus gaussiens. Dans cette preuve de concept, une réduction du coût fatigue $\mathcal{J}$ de 6.1% est obtenue.

Conclusions & Perspectives

Dans cette thèse, deux stratégies de contrôle originales, visant à réduire l'espérance du coût fatigue, sont proposées. La première consiste à réaliser une approximation du coût fatigue basée sur les données, pouvant être plus facilement utilisée dans un problème d'optimisation, pour ensuite concevoir une MPC adaptative basée sur la formulation de cette fonction de coût fatigue approximée, qui permet d'adaptée les paramètres de la MPC sur le niveau de vibration de l'éolienne. La deuxième consiste à concevoir un modèle de substitution liant des paramètres de contrôleurs et des conditions de vent à un coût fatigue basée sur l'expérience, de manière à utiliser ce modèle de substitution en-ligne pour choisir le contrôleur le plus approprié à minimiser le coût fatigue pour une condition de vent donnée.

Les deux stratégies proposées ont été simulées en boucle fermée dans des conditions différentes. La première a été simulée dans un environnement de simulation simplifié, où le système était linéaire et le vent représenté par sa vitesse au moyeu, ce qui a permis d'observer une potentielle réduction de fatigue de 30%. La seconde stratégie a été simulée dans un environnement de simulation plus réaliste, où le système est non linéaire et le vent est aussi turbulent dans le temps de que dans l'espace, ce qui a permis d'observer une réduction de fatigue de 6.1%.

Les perspectives pour les deux stratégies présentées sont les suivantes :

— Pour la première, continuer de valider cette stratégie dans un environnement de simulation plus réaliste et généraliser cette MPC adaptative au cas non linéaire.

— Pour la seconde, réitérer l'expérience avec des contrôleurs candidats plus avancés et considérer non seulement les conditions de vent en entrée du modèle de substitutions, mais aussi d'autres paramètres environnementaux de l'éolienne.