Iterative feedback tuning of feed-forward IPC for two-bladed wind turbines
A comparison with conventional IPC

S.P. Mulders
Iterative feedback tuning of feedforward IPC for two-bladed wind turbines
A comparison with conventional IPC

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft University of Technology

S.P. Mulders

September 14, 2015

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology
The work in this thesis is supported by 2-B Energy. Their cooperation is hereby gratefully acknowledged.
The development of sustainable energy production methods is an important aspect in lowering the emission of greenhouse gases and exhaustion of fossil fuels. Wind energy is recognized globally as one of the most promising sustainable forms of electricity generation, but the cost of offshore wind energy does not meet the level of fossil-based energy sources. An opportunity for wind energy cost reduction is the deployment of two-bladed wind turbines at offshore locations. Two-bladed turbines save the mass and cost of one rotor blade, which allows the entire wind turbine construction to be designed lighter, and in effect leads to lower initial costs. Moreover, reduction of harmonic fatigue loads on blades and other turbine parts using Individual Pitch Control (IPC) is a way to extend the turbine lifetime. This type of control, using a feedback control structure incorporating the Multi-Blade Coordinate (MBC) transformation, is capable of mitigating the most dominant periodic loads. It is generally known that significant turbine load reductions can be achieved using IPC, however, it is unclear to what extent the MBC pitch signal is optimal in terms of load alleviations. The main goal of this thesis is to develop a self-learning feedforward IPC strategy for a state-of-the-art two-bladed wind turbine. This IPC strategy will be compared to the conventional feedback IPC implementation.

By making use of properties of the MBC transformation, implementations of yaw control by IPC in different configurations are evaluated in terms of performance and stability. As a preparation for the comparison between conventional feedback and self-learning feedforward IPC strategies, a linear control-oriented model from blade pitch angles to harmonic blade loads is identified and used throughout this work for two main purposes. The first purpose is to reveal the level of interaction between both blades, which turns out to be negligible. On the basis of this reasoning, an appropriate cost-function is implemented for optimization of the feedforward controller. The second purpose of the linear model is the simplified and faster development of the Iterative Feedback Tuning (IFT) algorithm, which is later implemented in high-fidelity non-linear wind turbine simulation software. IFT is a self-learning model-free algorithm, and is used to optimize the rotor-position dependent feedforward IPC implementation. It is shown that the self-learning algorithm succeeds in optimization of the feedforward controller at all constant wind speeds, but also in more realistic turbulent wind conditions in the above-rated region. As the feedforward controller generates a constant amplitude pitch
signal for each wind speed, the amplitude of the feedforward pitch signal is gain-scheduled on exogenous load signals, in an effort to improve feedforward performance.

Results show that the conventional IPC strategy is optimal in terms of load reductions in steady state wind conditions, as the IFT algorithm optimizes to the exact same pitch signal at various constant wind speeds. In turbulent wind conditions, performance results indicate that the constant amplitude feedforward controller is able to attain significant load reductions, but that the performance of the conventional feedback control method is still superior. Comparing the pitch signals of both controllers in turbulent wind conditions, reveals that the conventional method continuously changes the phase of the implemented pitch signal, which is not driven by the varying rotor speed. To see how these changing pitch periodics have an effect on the load reduction capabilities, the feedforward (rotor speed dependent) pitch signal amplitude is scheduled on exogenous signals in various ways. This scheduling shows only minor performance improvements, and it can be concluded that the frequency changes in the pitch signal imposed by MBC, help to actively mitigate periodic blade loads. Using both the azimuth and blade load measurements, the conventional IPC strategy seems to actively track and mitigate the current present blade load harmonic, and it appears to be a serious challenge to develop control strategies that can improve performance already attained by MBC.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>v</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Description of the 2-B Energy wind turbine</td>
<td>5</td>
</tr>
<tr>
<td>3 Individual Pitch Control using the MBC transformation</td>
<td>9</td>
</tr>
<tr>
<td>3-1 The MBC transformation for IPC</td>
<td>10</td>
</tr>
<tr>
<td>3-2 Influence of the MBC reverse transformation phase-offset</td>
<td>12</td>
</tr>
<tr>
<td>3-3 Influence of the MBC integrator gains</td>
<td>17</td>
</tr>
<tr>
<td>4 IPC for yaw control</td>
<td>23</td>
</tr>
<tr>
<td>4-1 Implementations of yaw-by-IPC</td>
<td>23</td>
</tr>
<tr>
<td>4-2 Results - Stability of both configurations</td>
<td>25</td>
</tr>
<tr>
<td>4-3 Results - Performance of both configurations</td>
<td>25</td>
</tr>
<tr>
<td>5 Identification of a linear wind turbine model</td>
<td>31</td>
</tr>
<tr>
<td>5-1 Identification of the linear model</td>
<td>31</td>
</tr>
<tr>
<td>5-2 Interactions between in- and outputs</td>
<td>33</td>
</tr>
<tr>
<td>6 Implementation of Iterative Feedback Tuning on feedforward IPC</td>
<td>35</td>
</tr>
<tr>
<td>6-1 Theory on Iterative Feedback Tuning</td>
<td>36</td>
</tr>
<tr>
<td>6-1-1 Minimizing the cost function</td>
<td>38</td>
</tr>
<tr>
<td>6-1-2 Gradient estimation using output related signals</td>
<td>39</td>
</tr>
<tr>
<td>6-1-3 Gradient estimation using input related signals</td>
<td>40</td>
</tr>
<tr>
<td>6-2 Application of IFT on feedforward IPC</td>
<td>42</td>
</tr>
<tr>
<td>6-2-1 IFT SISO application on feedforward IPC</td>
<td>42</td>
</tr>
<tr>
<td>6-2-2 IFT MIMO application on feedforward IPC</td>
<td>44</td>
</tr>
</tbody>
</table>
# Table of Contents

6-3 Implementation on the linear and non-linear model .............................................. 46
  6-3-1 IFT implementation on the linear model .......................................................... 46
  6-3-2 IFT implementation on the non-linear model ...................................................... 47

7 Results on feedforward IPC and IFT ........................................................................... 53
  7-1 Linear model MIMO optimization results ................................................................. 53
  7-2 Non-linear model MIMO optimization results ......................................................... 54

8 Gain-scheduling the feedforward controller ............................................................... 63
  8-1 Determining the effect of only changing the pitch amplitude .................................... 64
  8-2 Scheduling on blade and tower loads ..................................................................... 66
  8-3 Investigating the MBC-IPC pitch signal .................................................................. 70

9 Conclusions and recommendations ............................................................................. 73
  9-1 Conclusion and discussion ....................................................................................... 73
  9-2 Recommendations ................................................................................................... 75

A Appendices ................................................................................................................... 77
  A-1 Appendix A - filtfiltC Simulink implementation ...................................................... 77
  A-2 Appendix B - IIR and FIR filters ............................................................................. 78

Bibliography ..................................................................................................................... 81

Glossary ............................................................................................................................ 85
  List of Acronyms ............................................................................................................. 85
  List of Symbols ............................................................................................................... 86
Acknowledgements

I would like to thank my supervisors dr. ir. J.W. van Wingerden and J. Beerens (2-B Energy) for their assistance during the course of this graduation project. Special thanks and appreciation go to ir. E. van Solingen for his motivation and dedication in helping to make this project a success.

I want to thank my mother, father and sister for their endless support during the whole period of my bachelor and master study.

Delft, University of Technology

September 14, 2015

S.P. Mulders
The development of sustainable energy production methods is an important aspect in lowering the emission of greenhouse gases and exhaustion of fossil fuels. Sustainable solutions like solar energy and hydropower are already being deployed and deliver to the existing electrical energy network. Another important contributing sustainable energy source is wind energy, which is seen as an opportunity because of its high power generation capabilities, scalability and possibility for deployment at offshore locations. During Europe’s Wind Energy Event (EWEA) 2013, it was announced that wind energy will be the cheapest electricity technology after 2020, while meeting 50% of electricity demand in Europe by 2050 [1].

Offshore wind turbine exploitation is interesting because of the large available space and less social resistance. At these sites, the wind tends to be stronger and more consistent, with a smaller wind shear and less turbulence compared to onshore locations [2]. Currently, wind turbines with three blades are dominant at offshore locations. A look back in history shows that in the eighties and nineties, a large amount of two-bladed turbine concepts were proposed through national research programs [3]. However, disadvantages such as excessive noise emission and the restless view due to the continuously changing geometry, led to its declination in favor of three-bladed models. The mentioned disadvantages are in particular applicable at onshore locations, but have little influence at offshore locations. Recent signs of interest are shown by different manufacturers for development of new offshore two-bladed turbines. One big advantage is the saving of the mass and cost of one rotor blade. Due to the lower weight of the rotor, the entire wind turbine construction can be designed lighter, which in effect leads to lower initial costs. Another advantage is that two-bladed turbines can be accessed safely by helicopter when the turbine is not operational and the blades are positioned horizontally.

A manufacturer that is currently working on its first two-bladed offshore wind turbine is 2-B Energy in Hengelo, the Netherlands [4]. The turbine has a rotor diameter of approximately 140 meters and a rated power of 6 MW. Combined with the use of proven technologies, 2-B Energy focuses on a simplified design, which leads to less material use and an extended life span of up to 40 years [5]. As a result of the downwind configuration (i.e., seen from upwind the rotor is positioned behind the tower), the blades can be designed more flexible as
they bend away from the tower support structure. The tower is a full jacket truss structure which reduces the effect of cyclic loads due to tower shadow. Calculations of 2-B Energy show an overall offshore wind energy cost reduction of 30–40%, including costs for maintenance and removal [4].

In order to realize the earlier mentioned lifetime, active control methods are required. Turbine loads can be substantially lowered by the application of Individual Pitch Control (IPC). The continuous adjustment of individual blade pitch angles by IPC, is a method for periodic wind turbine load reduction. The conventional implementation of IPC incorporates the Multi-Blade Coordinate (MBC) transformation, which has proven itself able to reduce dominant periodic turbine loads. With IPC, the rotor angular position and blade load measurement signals are used to calculate the IPC contribution to the blade pitch signals. An important property of MBC is the decoupling of rotor loads in a tilt and yaw component, which makes control of the yaw-alignment angle by IPC possible. By making use of properties of the MBC transformation, different configurations of yaw-by-IPC are investigated and results on performance, ease of implementation and stability will be discussed.

The focus of this thesis will be on the development of a self-learning feedforward IPC control system. In this feedforward control strategy, the signal frequency is only dependent on the current rotor position (azimuth angle), which in literature is generally called cyclic-pitch control. The optimal cyclic-pitch control signal for turbine load reductions is subsequently found for each wind speed by the implementation of Iterative Feedback Tuning (IFT). The found pitch signal is compared to the pitch signal generated by the conventional MBC control method for IPC. It is generally known that significant turbine load reductions can be achieved using conventional IPC, however, it is unclear to what extent the MBC pitch signal is optimal in terms of load alleviations. The main goal of this thesis is to develop a self-learning feedforward IPC strategy for a state-of-the-art two-bladed wind turbine. This IPC strategy will be compared to the conventional feedback IPC implementation.

Because the cyclic-pitch signal in its basic form has a constant amplitude and is only dependent on the azimuth angle, it is investigated whether changing the amplitude of the pitch signal improves the feedforward control load reduction capabilities. Adjustment of the cyclic-pitch amplitude can be gain-scheduled on different signals, like blade load, tower load or future wind measurements/predictions. In this thesis, scheduling of the signal on blade loads is implemented, and methods for implementation on tower loads and wind predictions are discussed.

In Chapter 2, a description of the 2B6 wind turbine of 2-B Energy is given. A high-fidelity model of this turbine will be used throughout this work to evaluate new control concepts. Then, in Chapter 3, the general applied method for IPC called the MBC transformation is presented and the effect of including a phase-offset to the azimuth angle in the reverse MBC transformation is investigated. Chapter 4 describes an addition to the MBC control scheme, which makes yaw-by-IPC possible. It is possible to implement this addition in different ways, and advantages/disadvantages of these configurations in terms of performance, stability and ease of implementation will be outlined. In Chapter 5, system identification is performed on input/output data to obtain a control-oriented model. The interactions between the various in- and outputs are investigated, as a preparation for the main subject of this thesis: application of IFT on feedforward IPC. This topic is split in three different, but subsequent chapters: Chapter 6 gives an introduction of IFT and implements the algorithm for finding the
optimal cyclic-pitch controller on the identified linear and high-fidelity non-linear model. The method uses blade load measurement data to generate the optimal pitch signal for fatigue load reductions, in an iterative manner. Results are given in Chapter 7, and Chapter 8 presents a gain-scheduling for the optimized cyclic-pitch controller, which is carried out for possible further performance improvements in terms of blade and other turbine loads.
Chapter 2

Description of the 2-B Energy wind turbine

Wind turbine design and deployment is currently dominated by three-bladed wind turbines. Despite that this type of wind turbine has, in comparison with its two-bladed opponent, advantages in terms of noise emission, visual impact and load distribution, opportunities can be found in the two-bladed rotor configuration. Because of the increasing interest in the deployment of offshore wind farms, most of the earlier mentioned advantages of three-bladed rotors over the two-bladed configuration do no longer hold. Moreover, due to the absence of one additional blade, the wind turbine structure can be designed lighter, which leads to the use of less material and a reduction in overall turbine costs. An advantage of deployment at offshore compared to onshore locations, is the wind that is less disturbed and tends to be more consistently with a smaller wind shear and less turbulence [2].

Several companies are currently developing a two-bladed wind turbine, of which some have already deployed their design. A recent overview paper with more information on current developments of two-blades turbines can be found in [6]. The work that is performed in this thesis is based on a computational model of the 2B6 turbine of 2-B Energy, a company situated in Hengelo, The Netherlands [4]. The 2B6 turbine is a two-bladed offshore wind turbine with a rated power output of 6 MW, which is reached around a wind speed of 13 m/s, and the rotor has a span of approximately 140 m. 2-B Energy focuses on simplified turbine design and this, combined with the use of proven technologies, leads to less material use and an extended lifetime of up to 40 years. A table with the most relevant turbine properties is given in Table 2-1 and an illustration of the turbine is given in Figure 2-1.

The 2B6 turbines are designed to operate in a so called PowerBlock of nine wind turbines in a 3-by-3 topology, where the eight outer turbines are connected to the center turbine. The benefit of this topology is that a group transformer located in the truss tower of the center turbine can increase the grid voltage to High Voltage (HV) (see Figure 2-2). Therefore, there is no need for a separate offshore transformation platform [4].

The 2B6 turbine possesses several distinguishing properties. The first is the implementation of a damped free-yaw system, enabled by the downwind rotor configuration. As a result of
Table 2-1: 2-B Energy 2B6 wind turbine specifications [4]

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated power</td>
<td>6 MW</td>
</tr>
<tr>
<td>Rotor diameter</td>
<td>140.6 m</td>
</tr>
<tr>
<td>Hub height</td>
<td>95 – 100 m</td>
</tr>
<tr>
<td>Orientation</td>
<td>Downwind</td>
</tr>
<tr>
<td>Support structure</td>
<td>Full 3 leg jacket</td>
</tr>
<tr>
<td>Yaw system</td>
<td>Active - soft/damped</td>
</tr>
</tbody>
</table>

the downwind configuration, the blades can be designed rather flexible because the blades bend away from the tower support structure. A recently published study [7] shows that a damped free-yaw system has the potential of 40% tower fatigue load reductions, and 19% ultimate load alleviations compared to fixed-yaw configurations. Even higher load reduction capabilities are possible when overload protection in the hydraulic yaw system is introduced. Most large and commercially deployed wind turbines are upwind and use a fixed, but adjustable yaw system, which changes the nacelle orientation when the yaw-alignment error reaches certain thresholds. A damped-free yaw configuration makes yaw-by-IPC control possible, for improved and continuous alignment with the current wind direction. As effect of the continuous alignment, the overall energy production might increase, while only slightly alleviating the blade and tower load reduction capabilities of Individual Pitch Control (IPC) [8].

A second feature for overall turbine load reductions, is the implementation of IPC. As the
2B6 turbine does not incorporate a teeter hinge and IPC is used to compensate for cyclic bending moments [9].

Another property is the three-legged jacked support structure. This feature reduces periodic blade loads caused by tower shadow, reduces the use of material (costs) and is easier transportable than a conventional monopile by the tower stacking possibilities [5]. A disadvantage can be found in the higher complexity and the labor intensive welding of the tubular joints.

The last feature worth mentioning is the helicopter deck on top of the turbine nacelle, which allows ease of access when maintenance needs to be performed. A first prototype of the 2B6 turbine is expected to be deployed and operational by the end of 2015.

A computational model describing the dynamics of the 2B6 turbine is supplied by 2-B Energy, and is developed in the high-fidelity and state-of-the-art wind turbine simulation software DNV GL Bladed 4.20 [10]. This software package is certified and widely used in industry to simulate turbine responses subject to three-dimensional turbulent wind fields. The simulation software allows implementation of new and alternative control algorithms, by giving the ability to load a Dynamic-link library (DLL) [9] generated from the internal MATLAB Simulink [11] wind turbine control template by Delft Center for Systems and Control (DCSC). It is stressed that the control strategies developed in this thesis do by no means represent the 2-B Energy control system implementation. It should also be noted that some information cannot be given due to confidentiality, and this is also the reason why certain data will be obscured throughout the course of this thesis.
Individual Pitch Control using the MBC transformation

All work in this thesis will be performed on a two-bladed Variable-Speed Variable-Pitch (VS-VP) wind turbine. The operating regions for this type of wind turbine can be roughly divided in two regions: the below-rated and above-rated region. In the below-rated region, the torque controller is active for speed regulation, while the blade pitch angle is kept constant at fine pitch, i.e. the optimum pitch angle for maximum aerodynamic efficiency of the rotor [12, 13]. When rated power output is reached, the torque set-point of the torque controller is kept constant at rated and Collective Pitch Control (CPC) is used to control the rotor speed by collectively pitching all blades [13]. Individual Pitch Control (IPC) is a periodic addition to the CPC pitch signal, and is primarily used for turbine load reductions. A schematic overview of the considered wind turbine control system is given in Figure 3-1.

At present, IPC is generally implemented on blade load signals and azimuth angle using the so called Multi-Blade Coordinate (MBC) transformation. This transformation transforms the blade load signals from a rotating into a non-rotating reference frame and decouples the signals in a tilt and yaw component, which makes Single-Input Single-Output (SISO) controller design possible [14, 15]. This is done since the turbine rotor reacts as a whole to changes in blade loads caused by e.g. wind shear, tower shadow and turbulence.

As stated in [16], the implementation of the MBC transformation on two-bladed turbines is mathematically not sound and therefore a linear coordinate transformation is proposed in [14]. This transformation has the advantage that only one control loop is needed to gain the ability to reduce all load harmonics $n\cdot P$, while for MBC one transformation is needed for each load harmonic that is desired to be reduced. The linear transformation transforms the blade load signals in a differential and collective mode, where odd and even denoted harmonics appear in the former and latter mode respectively. With only one transformation, this approach gives the possibility to form the complete control system using well-known filter types, such as low-pass, high-pass and notch filters. However, in this work the conventional MBC transformation is used, as it incorporates some properties which cannot be exploited in
In this chapter, the MBC control implementation for load reductions is explained in Section 3-1. Section 3-2 investigates the influence of including a constant lead phase-offset to the azimuth angle in the MBC transformation. Section 3-3 examines whether the effect of the phase-offset can be matched by a combination of MBC integrator gains.

### 3-1 The MBC transformation for IPC

This section gives a short explanation of the working principles of the MBC transformation for IPC applied to a two-bladed wind turbine; more general and elaborate descriptions can be found in [14, 15]. An overview of the MBC control implementation is given in Figure 3-2. Considering a certain blade load harmonic $n\Omega$, the MBC transformation transforms the specified blade load harmonic to a $0\Omega$ frequency in the non-rotating frame of reference. Most often, only integral control is applied to this load signal in the non-rotating reference frame, but it is also possible to include proportional control. The integrator gain $K_I$ has to be chosen small, to account for the magnitude difference between blade load and pitch angle signals.
Figure 3-2: Schematic overview of a forward and reverse $nP$ MBC transformation, using PI-controllers in the non-rotating reference frame.

The signals after the (P)I-controllers represent the decoupled and non-rotating pitch angles, which are fed through the reverse MBC transformation to obtain the implementable pitch signals in the rotating frame of reference. Pitch signals are in general filtered before or after control is applied, to prevent high frequency content being sent directly to the pitch actuator.

First the blade load signals of both blades are passed through the forward MBC transformation

$$
\begin{bmatrix}
M_{\text{tilt}}^* \\
M_{\text{yaw}}^*
\end{bmatrix} = 
\begin{bmatrix}
\cos (n\psi) & \cos n(\psi + \pi) \\
\sin (n\psi) & \sin n(\psi + \pi)
\end{bmatrix}
\begin{bmatrix}
M_{y,1} \\
M_{y,2}
\end{bmatrix},
$$

where the measured blade loads are denoted by $M_{y,1}$ and $M_{y,2}$, the tilt and yaw components of the blade loads in the non-rotating frame by $M_{\text{tilt}}^*$ and $M_{\text{yaw}}^*$ respectively, the harmonic under consideration by $n$ and $\psi$ represents the rotor azimuth angle. Note that in the sequel of this thesis, quantities expressed in the non-rotating reference frame are supplied with a $^*$ in the superscript. The decoupled non-rotating blade load signals $M_{\text{tilt}}^*$ and $M_{\text{yaw}}^*$ are fed through two separate (P)I-controllers resulting in the non-rotating pitch angles $\theta_{\text{tilt}}^*$ and $\theta_{\text{yaw}}^*$. These signals are transformed to the pitch angles $\theta_1$ and $\theta_2$ in the rotating reference frame by the reverse MBC transformation, which is given by

$$
\begin{bmatrix}
\theta_1 \\
\theta_2
\end{bmatrix} = 
\begin{bmatrix}
\cos (n\psi) & \sin n(\psi) \\
\cos (n\psi + \pi) & \sin n(\psi + \pi)
\end{bmatrix}
\begin{bmatrix}
\theta_{\text{tilt}}^* \\
\theta_{\text{yaw}}^*
\end{bmatrix}.
$$

For illustration purposes, consider the 1P periodic blade load, which originates particularly from wind shear and tower shadow and its frequency is determined by the rotor speed. This 1P blade load measurement signal enters the 1P forward transformation, in which the frequency content of the signal is moved to 0P and 2P in the non-rotating reference frame. The property of the signal being moved to a 0P signal, makes integral (and optionally proportional) control possible. After integration of $M_{\text{tilt}}^*$ and $M_{\text{yaw}}^*$ in the non-rotating frame of reference, the non-rotating pitch signals $\theta_{\text{tilt}}^*$ and $\theta_{\text{yaw}}^*$ are converted to the actual rotating pitch signals by the reverse transformation. Other harmonic blade load frequencies $nP$ can be included to further improve load reduction capabilities, but this requires the set-up of additional MBC transformations.

An additional azimuth phase-offset $\psi_{\text{offset}}$ can be included in the reverse transformation [17], which can give additional load reductions on different parts of the turbine in turbulent wind, by changing the characteristics of the pitch signal. Results and conclusions on load reductions attained with MBC, and the effect of setting different phase-offs $\psi_{\text{offset},i}$, will be discussed in Section 3-2.
Individual Pitch Control using the MBC transformation

Forward transformation
Harmonic nP

Reverse transformation
Harmonic nP

PI-controller

θ

*tilt

θ

*yaw

ψ

offset

Figure 3-3: Schematic overview of a forward and reverse nP MBC transformation, using PI-controllers in the non-rotating reference frame. A phase-offset ψ\textsubscript{offset} can be incorporated in the reverse MBC transformation for additional tuning capabilities of wind turbine load reductions.

3-2 Influence of the MBC reverse transformation phase-offset

As already mentioned in the previous section, a phase-offset can be added to the reverse MBC transformation. Multiple sources [18, 19] state that a phase-offset can be used to compensate for time lags caused by the feedback control system. Although this statement appears to be valid, the effect on turbine (fatigue) loads has not yet been validated. In this section, IPC using the MBC transformation in its most basic form will be incorporated in the high-fidelity wind turbine simulation software Bladed for constant and turbulent wind conditions, with wind speeds of 8, 14 and 20 m/s. Simulations will be performed in the rigid-yaw configuration. The phase offset-values are represented by ψ\textsubscript{offset,i}, where i represents integers for increasing offset values. A phase-offset can be incorporated in the reverse MBC transformation as follows

\[
\begin{bmatrix}
\theta_1 \\
\theta_2
\end{bmatrix} =
\begin{bmatrix}
\cos(n(\psi + \psi_{\text{offset},i})) & \sin(n(\psi + \psi_{\text{offset},i})) \\
\cos(n(\psi + \pi + \psi_{\text{offset},i})) & \sin(n(\psi + \pi + \psi_{\text{offset},i}))
\end{bmatrix}
\begin{bmatrix}
\theta_{\text{tilt}}^* \\
\theta_{\text{yaw}}^*
\end{bmatrix}.
\]

(3-3)

The forward transformation, as given in Equation (3-1) remains unchanged. A schematic representation of the MBC feedback control implementation, including the reverse offset is given in Figure 3-3.

First the effect of including lead phase-offsets ψ\textsubscript{offset,i} for constant wind is investigated, where i = 1 represents no offset (0 degrees) and i = 5 represents the highest phase-offset (60 degrees), with increments of 15 degrees between different values of i. Time-domain simulation plots, representing the pitch and blade load response of all different offset values including transient behavior, are presented in Figure 3-4. For convenience reasons, the 14 m/s wind speed case is left out of the figure.

As can be observed, the initialization of the pitch signal is different for all phase-offsets, but after convergence all signals appear to represent the same periodic behavior. This effect can be explained by the MBC feedback structure which directly acts on the to be mitigated blade load harmonic nP, and optimizes for specific sine and cosine gains θ\textsubscript{tilt} and θ\textsubscript{yaw}, which minimizes this periodic load component in the blade load signal. Thus, in the constant wind case, adding a phase-offset to the reverse transformation has no influence on the steady-state result of the pitch signal, as this effect is compensated for by the sine and cosine gains θ\textsubscript{tilt} and θ\textsubscript{yaw}, as clarified in Figure 3-5. For purpose of illustration, a sinusoidal wave with a

S.P. Mulders
Master of Science Thesis
Different phase-offset values are implemented in the reverse transformation. For constant wind conditions (no turbulence), the phase-offset shows to have little effect on the pitch signal. It only appears that the initial transient towards the final is different, but the signals at all wind speeds converge to the same periodic signal. Blade loads characteristics are also unaffected after convergence in constant wind conditions.

certain amplitude $c$, frequency $x = \omega t$ and phase-offset $\phi$ can be decomposed into a weighted linear combination of geometrical functions with the same frequency, but without the phase offset $\phi$. To clarify:
Individual Pitch Control using the MBC transformation

\[ k_1 \sin(x) + k_2 \cos(x) = c \sin(x + \phi), \]

where \( c = \sqrt{k_1^2 + k_2^2}, \quad \phi = \text{atan2}(k_2, k_1). \) \hspace{1cm} (3-4)

For example, when a phase-offset value of \( \psi_{\text{offset,5}} \) is included in the reverse MBC transformation, the gains \( \theta^*_{\text{tilt}} \) and \( \theta^*_{\text{yaw}} \) converge in such a way that \( \text{atan2} (\theta^*_{\text{yaw},\psi_1}, \theta^*_{\text{tilt},\psi_1}) - \text{atan2} (\theta^*_{\text{yaw},\psi_5}, \theta^*_{\text{tilt},\psi_5}) = \psi_{\text{offset,5}}, \) and the included phase-offset is counteracted.

Due to the varying nature of turbulent wind in intensity and direction, the periodic blade load characteristics will continuously change, to which the pitch signal has to adapt. Now it is known that at constant wind speeds the pitch signals generated by the MBC individual pitch controller converge to the exact same signals for distinct phase-offsets, but with different transient behavior, it is interesting to see the influence of the phase-offset during turbulent wind. In the next series of simulations, the same phase-offset values will be implemented as used in previous simulations, but subjected to turbulent wind fields with an intensity of 15%. Turbulence intensity is defined as the ratio of turbulence power to the mean wind speed [20]. Time-domain results of the obtained pitch signals and blade loads are presented in Figure 3-6. It can be observed that the pitch and blade load characteristics change with increasing phase-offsets.

Wind turbine loads can be quantified by different measures and one important measure is the Damage Equivalent Load (DEL) [9, 21]. Load measurements can be converted into the DEL as a measure of fatigue loading. The DEL represents the amplitude of a certain harmonic
load variation that would cause the same damage level when it is repeated for a given amount of cycles. A DEL blade load performance comparison between different phase-offsets at the predefined wind speeds for constant wind is ignored, since the time domain pitch and blade load measurement signals are similar. However, pitch and blade load signals in turbulent wind show different behavior, so it is interesting to explore if the performance is optimal at a particular offset value. A graphical representation of the DELs at different wind speeds is

Figure 3-6: Different phase-offset values are implemented in the reverse transformation. For turbulent wind conditions (intensity 15%), the implementation of various phase-offset values seems to dynamically change the course of the signal, which has its effect on load reduction capabilities.
Including a phase-offset in the reverse MBC transformation during turbulent wind shows limited effect on blade loads reductions, but tower and rotating hub loads seem to benefit. A clear optimum can be observed between $\psi_{\text{offset},3}$ and $\psi_{\text{offset},4}$. Turbulence intensity is 15% for all wind speeds.

Figure 3-7: including a phase-offset in the reverse MBC transformation during turbulent wind shows limited effect on blade loads reductions, but tower and rotating hub loads seem to benefit. A clear optimum can be observed between $\psi_{\text{offset},3}$ and $\psi_{\text{offset},4}$. Turbulence intensity is 15% for all wind speeds.
ψ_{offset,i} for all wind speeds. Another remarkable property is that performance improvements become more apparent at higher wind speeds.

### 3-3 Influence of the MBC integrator gains

As concluded from the previous section, including a phase-offset ψ_{offset,i} in the reverse MBC transformation changes the behavior of the resulting IPC pitch signal, which can have a positive influence on load reductions of various turbine parts. However, an interesting hypothesis is that changing the integrator gains (separately) can result in the exact same pitch signal behavior, as varying the phase-offset with fixed integrator gains. In this section, the influence of the MBC integrator gains is examined on the resulting IPC pitch signal.

In the simulations of the previous section, the integrator gains $K_I$ for the tilt and yaw components were chosen equal, so $K_{I,1} = K_{I,2}$. The baseline value will from now on be denoted as $K_{I,0}$. By setting the phase-offset value to $ψ_{offset,3}$ and varying the integrator gains both symmetrically and asymmetrically, it is investigated whether the response with $K_{I,0}$ and $ψ_{offset,0}$ can be obtained. To make a comparison on how the integrator gains change the behavior of the pitch signal, the following gain adjustment cases are made with respect to the baseline integrator gain $K_{I,0}$:

1. $K_{I,1} = 0.5K_{I,0}$; $K_{I,2} = 0.5K_{I,0}$;
2. $K_{I,1} = 2K_{I,0}$; $K_{I,2} = 2K_{I,0}$;
3. $K_{I,1} = 2K_{I,0}$; $K_{I,2} = 0.5K_{I,0}$;
4. $K_{I,1} = 0.5K_{I,0}$; $K_{I,2} = 2K_{I,0}$.

Cases (1, 2) where the integrator gains are multiplied by the same factor, as cases (3, 4) where the gains are multiplied separately by different factors will be investigated. Like in the previous section, plots on the resulting constant wind IPC pitch signal transients and non-rotating sine/cosine gains are given in Figures 3-8 and 3-9 respectively. In Figure 3-8, it can be observed that primarily the convergence speed of the pitch signal improves with increasing integrator gains $K_I$. This is unlike the case discussed in the previous section, where particularly the behavior in time (phase) towards the steady-state IPC pitch signal changes (see Figure 3-4). The effect is also confirmed by Figure 3-9, where distinct integrator gains converge to the same weighted linear combination of geometrical functions, but with a different convergence rate.

A time-domain comparison of the implemented IPC pitch signal during turbulent wind conditions, at the same time-interval used in Figure 3-6, is presented in Figure 3-10. Comparing the graphs to the ones given in Figure 3-6, it can be stated that the convergence rate of the pitch signal increases with higher integrator gains. For completeness, a DEL comparison at the various integrator gains and wind speeds is given in Figure 3-11. The integrator gains have a great influence on blade and fixed-structure load reduction capabilities, while blade load DELs were merely affected by the distinct phase-offsets values.

The hypothesis made in the introduction of this section, does not seem to hold as the phase-offset changes the behavior in time (phase) towards the steady-state IPC pitch signal, while the integrator gains primarily change the convergence speed towards the desired IPC pitch signal for $nP$ harmonic load minimization.
Figure 3-8: Implementation of different integrator gains \( K_i \) in the MBC control structure. It can be seen that, unlike the different behavior in time (phase) towards the steady-state pitch signal for changing phase-offsets, only the convergence speed of the pitch signal alters.
Figure 3-9: The 1P MBC transformation converges to the same sine and cosine gains $\theta_{\text{tilt}}^*$ and $\theta_{\text{yaw}}^*$ for all integrator gains $K_I$, but with different rates of convergence. Gains are plotted for a constant wind speed of 14 m/s.
IPC pitch signals, varying $K_I$ gains, turbulent wind speed: 8 m/s

Blade load signals, varying $K_I$ gains, turbulent wind speed: 8 m/s

IPC pitch signals, varying $K_I$ gains, turbulent wind speed: 20 m/s

Blade load signals, varying $K_I$ gains, turbulent wind speed: 20 m/s

Figure 3-10: Different integrator gains are implemented for turbulent wind conditions (intensity 15%). The rate of convergence is higher with increasing integrator gains.
Figure 3-11: DEL plot for different integrator gains under turbulent wind conditions (intensity 15%). The integrator gains at different values show a great impact on all turbine loads.
To maximize energy extraction from the wind, the rotor axis of a wind turbine needs to be aligned with the dominating wind direction. Because the wind direction changes over time, a yaw system is required to keep the orientation of a wind turbine aligned with the wind direction to capture as much wind energy as possible [20]. A yaw system can be incorporated into the wind turbine nacelle in various ways. One way is by the use of an active yaw configuration, where electrical drives actively rotate the nacelle and rotor into the current wind direction, and brakes on a circular brake disk prevent unwanted yaw motion [12]. An alternative way of keeping the nacelle of a downwind free-yaw turbine aligned with the current wind direction, is by adding a yaw component to the Individual Pitch Control (IPC) signal. This method, which will be called yaw-by-IPC in the course of this thesis, is a way to improve the self-alignment property of downwind turbines and to reduce loads on the support structure and the yaw system itself [6].

The proof of concept of yaw-by-IPC is already given in [8]. Conclusions that can be drawn from this work are that the yaw-by-IPC component reduces yawing moments transferred from the nacelle to the yaw system, resulting in significant load reductions on the support structure. The negative effect of the yaw-by-IPC addition is very limited: the trade-off between yaw-alignment control and load reduction capabilities can be tuned by changing the bandwidth of the yaw controller. The contribution of this chapter is to introduce another yaw-by-IPC implementation, and to compare both configurations in terms of stability and load reduction capabilities.

The two possible methods for implementation of yaw-by-IPC will be discussed in Section 4-1. Results and advantages/disadvantages on stability and load reduction capabilities of both set-ups will be discussed in Sections 4-2 and 4-3 respectively.

4-1 Implementations of yaw-by-IPC

As described in the introduction of this chapter, different implementations of yaw-by-IPC are possible. Both use the property of the Multi-Blade Coordinate (MBC) transformation that
decouples the blade load signal into a tilt and yaw component. The yaw-misalignment error $\chi$ is low-pass filtered in $\chi^\prime$, fed through a PI-controller and added to the yaw component in the non-rotating frame of reference. The step of adding the yaw-setpoint control signal can be accomplished in two ways: (1) by adding the signal $M^\chi\yaw$ to the yaw component of the blade load signal $M^\yaw$ in the non-rotating frame (configuration #1), or (2) by adding the signal $\theta^\chi\yaw$ to the yaw component of the pitch angle $\theta^\yaw$ in the non-rotating reference frame (configuration #2) [22]. Both configurations #1 and #2 are graphically presented in Figures 4-1 and 4-2 respectively.

As can be seen in Figure 4-2, the yaw component of the non-rotating pitch signal $\theta^\yaw$ is high-pass filtered before the yaw-by-IPC control signal is added. The yaw motion of the nacelle has much slower dynamics than the loads acting on the rotor blades, and for this reason both signals can be separated in frequency. A higher bandwidth of the yaw controller means that the wind direction can be tracked more accurately and power extraction from the wind is maximized [23]. However, tracking the wind more strictly will cause higher blade loads, so a clear trade-off has to be made during control design. If one would disregard the frequency separation, both signals would interfere at lower frequencies and deteriorate yaw-alignment performance. The high-pass filter on the non-rotating pitch angle $\theta^\yaw$ is designed to have a cut-off frequency of 0.01 Hz, while the cut-in frequency of low-pass filter on the yaw-error signal $\chi^\prime$ is set to a frequency of 0.1 Hz. It is noted that configuration #1 (Figure 4-1) does not include the frequency separation, as the summed signal is a contribution to the blade load signal, which is fed through the integrator of the IPC PI-controller (part of the MBC...
4-2 Results - Stability of both configurations

In the first part of the results, the stability of both configurations is evaluated. A simulation for yaw-by-IPC configurations #1 and #2 is performed under constant wind conditions (no turbulence) and the blade load measurement signal is disrupted after 500 s. The results by means of time-series plots of the yaw-misalignment angle of both configurations are given in Figures 4-3 and 4-4.

It is observed that, while configuration #2 remains stable after the blade load signal interruption, configuration #1 gets unstable for most wind speeds after signal loss. A schematic overview of the situation of configuration #1 is given in Figure 4-5. It is observed that when the blade load signals $M_y$ disappear, the overall closed-loop configuration changes to a set-up with two integrators in series (yaw-misalignment and MBC integrator). This results in a total phase shift of the low-pass filtered yaw-by-IPC signal $\chi'$ of 180 degrees, and explains the unstable behavior in closed-loop. The described effect shows the advantage of implementation of configuration #2 in terms of stability.

4-3 Results - Performance of both configurations

Both configurations have their own characteristics, which will be illustrated by two types of simulation experiments at wind speeds of 8, 14 and 20 m/s. The first experiment involves a yaw-angle set-point change (step response) in constant wind conditions. Next, the performance of both set-ups will be compared in turbulent wind conditions by the Damage Equivalent Load (DEL) on various turbine parts. The controllers in both configurations are
tuned in such a way that a reasonable yaw-response is obtained and minimize the impact on the overall blade, hub and tower loads. One great advantage of configuration #1 is the ease of tuning: once the proportional $K_{P,1}$ and integral $K_{I,1}$ gains are obtained for one specified wind speed, these gains appear to be optimal for all wind speeds in terms of the before mentioned criteria. Configuration #2 however requires more tuning as a gain-scheduling is needed, based on the rotor speed and power output of the turbine, when IPC is only used for yaw control [22].

The step response of the yaw system to a sudden change in the yaw-alignment set-point from 0 to 15 degrees is presented for configurations #1 and #2 in Figures 4-6 and 4-7 respectively. It can be observed that the first configuration reacts with more overshoot to this change of reference than the second configuration.

In contrast to the previously found results, where configuration #1 reacts with more overshoot to a change in reference, time-series plots of the yaw-misalignment angle at different wind speeds show in both cases similar responses around the 0 degrees reference angle. Graphs representing the comparison between both configurations at the already mentioned wind speeds are given in Figures 4-8, 4-9 and 4-10.
A performance comparison between both configurations is made with the use of the DEL as a measure in Figures 4-11 and 4-12, where all DEL values are normalized with respect to configuration #1. The overall conclusion that can be drawn from these graphs is that the first configuration performs slightly better than the second, but none of the two clearly outperforms. It is expected that tuning the gain-schedule of the second configuration can improve performance, such that the level of the first controller is met. Clearly, a trade-off between stability and ease of implementation is needed for selection of the most appropriate control strategy.

**Figure 4-8:** Yaw-misalignment response of the wind turbine nacelle at a turbulent wind speed of 8 m/s, comparison of configuration #1 (blue, solid) and #2 (green, dashed). Turbulence intensity is 15%.

**Figure 4-9:** Yaw-misalignment response of the wind turbine nacelle at a turbulent wind speed of 14 m/s, comparison of configuration #1 (blue, solid) and #2 (green, dashed). Turbulence intensity is 15%.

**Figure 4-10:** Yaw-misalignment response of the wind turbine nacelle at a turbulent wind speed of 20 m/s, comparison of configuration #1 (blue, solid) and #2 (green, dashed). Turbulence intensity is 15%.
Figure 4-11: DELs comparison for configuration #1 (blue) and #2 (red) for blade, tower and hub loads. The DELs are normalized with respect to configuration #1.
Figure 4-12: Comparison for configuration #1 (blue) and #2 (red) for yaw misalignment, velocities and energy production. Overall, configuration #1 seems to perform better than configuration #2.
Chapter 5

Identification of a linear wind turbine model

The well known control design methods based on Linear Time-Invariant (LTI) models are developed by extensive research in the last few decades and have proven their effectiveness since then [17]. The most popular and frequently applied control design methods are based on linear system models at a certain operating point of interest. LTI models are also easily implementable in numerical software packages, which makes control design convenient.

For the above mentioned reasons, this chapter presents the process of identification of a linear model. The identification yields a control-oriented model from the pitch angles $\theta_i$ to the blade root moments $M_{y,j}$, used for development and implementation of the Iterative Feedback Tuning (IFT) algorithm. The identification of this Multiple-Input Multiple-Output (MIMO) model is done using the Predictor-Based Subspace IDentification (PBSIDopt) method by [24] in Section 5-1. Section 5-2 examines the level of interaction between the in- and outputs of the identified model using the Relative Gain Array (RGA).

5-1 Identification of the linear model

A linear wind turbine model will be identified on the basis of open loop input/output data, which is generated by the non-linear wind turbine simulation software Bladed, using a modified identification script by [25]. Although Bladed has the ability to create linear wind turbine models by the use of a module called Model Linearization, the PBSIDopt identification method will be used as this gives more insight and flexibility in the identification process, because arbitrary input and output data can be used for system identification of control-oriented models. It should be stressed that the model that is created in this section, will only be used for implementation in Simulink and for the development of the IFT algorithm (Chapter 6), and not for LTI control design.

The pitch angle is excited persistently, by means of a Random Binary Signal (RBS) with a bandwidth of $5\pi$ rad/s (2.5 Hz) and a maximum amplitude of 0.035 rad/s (2.0 degrees).
After data collection, the signals are detrended, resampled and split into a identification and validation data set. Resampling is performed to obtain input and output data sets of equal sampling frequency, but also to meet the standard rule of thumb where the sampling frequency is about ten times larger than the model frequency region of interest [26]. The dominant 1P periodic disturbance is added to the input signal, which is done to only identify the rotor dynamics with periodic effects [27]. After the data-preprocessing, the function `dordvarx` from the PBSID$_{opt}$ toolbox [24] is used to compute the singular values, which are useful to determine the appropriate model order. The result is presented in a semi-logarithmic plot in Figure 5-1. It can be observed that the last clear singular value drop is observed after a model order of twenty-four. The validity of the identified model is checked by the computation of the Variance-Accounted-For (VAF) values. The higher the VAF value, the lower the prediction error and the better the model is [26]. The choice of the model order from the singular values is also confirmed as the model order with the highest attainable accuracy in Figure 5-2, where VAF values are plotted against increasing model order. At the order of twenty-four, the accuracy seems to have reached its limit. The model order is chosen at its highest accuracy level (24), because the model is only used for excitation at specific $nP$ harmonic frequencies, and will not be directly used for control design. The identified model by the PBSID$_{opt}$ method yields satisfactory VAF values around 85% for both outputs. The resulting model is averaged over all rotor azimuth positions.

Finally, Bode magnitude and phase plots of the identified linear model from inputs [1, 2] to outputs [1, 2] are given in Figure 5-3. The direct (diagonal) magnitude plots seem to have a higher gain at the important frequencies 1P/2P than the off-diagonal plots, and this supports the claim of the model validity.
In this section, the RGA will be used to quantify the level of interactions in MIMO systems between the in- and outputs [28]. As in the previous section, the averaged model over all azimuth positions is used. The identified discrete time state-space model is first converted to an array of continuous time transfer functions, such that evaluation over all frequencies of the RGA is possible. A frequency-dependent plot of the RGA is generated and presented in Figure 5-4. From this graph it can be concluded that interactions between in- and outputs are limited and the application of decentralized control, where \( \theta_i \) controls \( M_{y,i} \), is justified. This motivation is used for the development and implementation of the IFT algorithm for optimization of the feedforward individual pitch controller in a MIMO configuration, which will be outlined in Chapter 6.
**Figure 5-4:** Frequency dependent RGA plotted for all four channels: diagonal direct terms (dashed blue [1,1], blue circles [2,2]) and cross terms (green solid [1,2], red crosses [2,1]). According to the RGA based on the identified linear model, it appears that in the frequency range of interest (1P - 4P), none or very little interactions are present between different in- and outputs.
Chapter 6

Implementation of Iterative Feedback Tuning on feedforward IPC

More advanced controllers are designed on the basis of a linear model and aim to reduce a cost function on the control input and/or states of the system. In most cases, however, not all plant dynamics and external disturbances are perfectly known and it is most often aimed to achieve the best possible performance with a Linear Time-Invariant (LTI) fixed-structure controller. It would be appealing to optimize the performance of a control system directly with respect to its individual control parameters, without the need for a plant model. However, such a method would require the gradient of the earlier specified cost function with respect to the control parameters, which has always been a stumbling block to come up with [29].

A commonly employed and model-free PID-tuning method includes the Ziegler-Nichols method to achieve a certain level of stability and performance in closed-loop [17]. Though, because this method requires the engineer to determine the ultimate proportional gain $K_u$, which brings the closed-loop system near the edge of instability, practical execution of the Ziegler-Nichols tuning method might be undesirable [30]. Other methods include the online identification of system models on which, after enough 'rich' data is collected, the controller is constructed or updated. When separate disturbance and plant models are identified, one could take into account changing environmental effects and plant dynamics, yielding the best possible performance throughout the life cycle of the system. One such approach applied for wind turbine load control is described in [31], where Repetitive Control (RC) is combined with subspace identification to form an adaptive control law for periodic disturbances, called Subspace Predictive Repetitive Control (SPRC). In turbulent wind fields, the SPRC method shows equivalent performance in terms of load reductions to that of conventional Individual Pitch Control (IPC), with a smoother control input, and reduced input energy.

Although the latter discussed method uses online identification, the identified model is still needed for construction of the controller. A method that abandons the system identification step, and where optimization is carried out directly on fixed-structure controller parameters, is Iterative Feedback Tuning (IFT). IFT computes an unbiased gradient of the predefined
cost function with respect to the control system parameters, by using closed-loop experiment data with the current controller operating on the actual system, in an iterative way [29]. It has been shown that IFT is capable of optimizing control parameters for disturbance rejection [32], Multiple-Input Multiple-Output (MIMO) [33], periodic time varying [34] and non-linear systems [35].

This chapter will elaborate the implementation of IFT on the feedforward individual pitch controller to find the optimal pitch signals for turbine load reductions, using an iterative open-loop procedure on linear and non-linear wind turbine models. Optimizing the azimuth angle dependent feedforward IPC by IFT, can help to provide IPC as a form of Fault-Tolerant Control (FTC) when blade loads measurements become unavailable, but also as a replacement of existing feedback control systems which depend on blade load measurements. By omitting or simplifying the blade load sensor system, overall wind turbine costs can be reduced.

The choice for optimization of a feedforward controller, instead of optimizing the integrator gains in the feedback Multi-Blade Coordinate (MBC) set-up has different reasons. The main reason is to investigate the optimality of the current conventional IPC strategy by the implementation of self-learning feedforward control. It could be possible that the MBC set-up induces delays in the pitch signal by its feedback configuration and filtering actions, which can be omitted using a controller scheduled only on the rotor azimuth angle. As will become clear in the course of this work, investigating and comparing the pitch signal of both controller types, yields a better understanding on how turbine load reductions can be further improved.

Section 6-1 describes the general IFT framework and Section 6-2 applies this framework to optimize the feedforward individual pitch controller in a SISO and MIMO set-up. Section 6-3 outlines the self-learning feedforward IPC implementation on both linear and non-linear wind turbine models. Results on the feedforward IPC controllers are presented in Chapter 7, and Chapter 8 describes the results of an additional gain-scheduling to vary the amplitude of the feedforward IPC signal.

### 6-1 Theory on Iterative Feedback Tuning

In this section, the general IFT framework by Hjalmarsson et al. [29] will be outlined. This framework is in the next section applied to the feedforward individual pitch controller for turbine load reductions. An unknown system is described by the discrete time model

\[ y(t) = G_0 u(t) + v(t), \]

where \( G_0 \) is an LTI operator. According to [34, 35], the operator can also be replaced by a non-linear periodic time-varying operator \( G_0(t) \), which is the case when IFT is implemented on the non-linear wind turbine model. The term \( v(t) \) is an unmeasurable disturbance, and \( u(t) \) and \( y(t) \) represent the control input and system output, to and from \( G_0 \) respectively. In the general framework, a two-degrees-of-freedom controller is defined by

\[ u(t, \rho_i) = C_r(\rho_i) r(t) - C_y(\rho_i) y(t, \rho_i), \]
where $C_r(\rho_i)$ and $C_y(\rho_i)$ are fixed-structure LTI transfer functions parameterized by parameter vector $\rho_i$ and $r(t)$ is the external deterministic reference signal. A schematic overview of the described set-up is given in Figure 6-1. To simplify notation, the time argument will be omitted in the rest of this description whenever possible, and an added argument $\rho_i$ indicates that data is gathered from the closed-loop system, with the current controllers parameterized by $\rho_i$ acting on the system. The desired output of the system is defined by $y_d$ and the error $\tilde{y}(\rho_i)$ between the actual and desired output is given by

$$\tilde{y}(\rho_i) = y(\rho_i) - y_d = \left(\frac{C_r(\rho_i)G_0}{1 + C_y(\rho_i)G_0}r + \frac{1}{1 + C_y(\rho_i)G_0}v\right) - y_d = (T_0(\rho_i)r + S_0(\rho_i)v) - y_d,$$

(6-3)

where $S_0(\rho_i)$ and $T_0(\rho_i)$ represent the sensitivity and complementary sensitivity function respectively. It is desirable to minimize some norm of this error quantity $\tilde{y}(\rho_i)$ with respect to the control parameter vector $\rho$. In general, a cost function $J(\rho_i)$ of the following form is implemented

$$J(\rho_i) = \frac{1}{2N}E\left[\sum_{t=1}^{N}(L_y\tilde{y}(t, \rho_i))^2 + \lambda\sum_{t=1}^{N}(L_uu(t, \rho_i))^2\right],$$

(6-4)

where $L_y$ and $L_u$ represent filters on the output and input signals, and can be used to put emphasis on certain frequencies and/or filter noise. In this section, the filters are chosen equal to 1, but this feature will be used and explained in greater detail in Section 6-2. The resulting cost function is given by

$$J(\rho_i) = \frac{1}{2N}E\left[\sum_{t=1}^{N}(\tilde{y}(t, \rho_i))^2 + \lambda\sum_{t=1}^{N}(u(t, \rho_i))^2\right],$$

(6-5)
and the optimal set of control parameters \( \rho^* \) minimizing the described cost function \( J(\rho) \) is defined by

\[
\rho^* = \arg \min_{\rho} J(\rho). \tag{6-6}
\]

The next sections will elaborate on the working principles of the IFT algorithm for minimization of the cost function.

### 6.1.1 Minimizing the cost function

To minimize the cost function given in the previous section, one might simply take the partial derivative of \( J(\rho_i) \) with respect to the control parameter vector \( \rho \)

\[
\frac{\delta J}{\delta \rho}(\rho_i) = \frac{1}{N} E \left[ \sum_{t=1}^{N} \tilde{y}(t, \rho_i) \frac{\delta \tilde{y}}{\delta \rho}(t, \rho_i) + \lambda \sum_{t=1}^{N} u(t, \rho_i) \frac{\delta u}{\delta \rho}(t, \rho_i) \right], \tag{6-7}
\]

and use the obtained gradient vector in the control parameter update equation

\[
\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \frac{\delta J}{\delta \rho}(\rho_i). \tag{6-8}
\]

In the latter presented equation, the quantity \( \gamma_i \) is the step size of the algorithm which is a positive scalar and may be altered after each iteration to improve speed of convergence. \( R_i \) is usually taken as a Gauss-Newton approximation of the Hessian of \( \tilde{y}(\rho_i) \), so

\[
R_i = \text{est} \left( \frac{\delta \tilde{y}}{\delta \rho}(\rho_i) \right)_1 \text{est} \left( \frac{\delta \tilde{y}}{\delta \rho}(\rho_i) \right)^T_1, \tag{6-9}
\]

where the subscript indicates the experiment number. The Hessian could also be simply chosen as an identity matrix \( I \), but this would in general greatly deteriorate the convergence speed of the algorithm [32]. Another option to reduce the amount of iterations is to include one additional experiment to obtain an extra set of the gradient \( \text{est} \left( \frac{\delta \tilde{y}}{\delta \rho}(\rho_i) \right)_2 \), which can be used to generate an unbiased estimate of the Hessian \( R_i \) [32].

Going back to Equation (6-7), it can be seen that the quantities \( \tilde{y}(\rho_i), u(\rho_i), \frac{\delta \tilde{y}}{\delta \rho}(\rho_i), \frac{\delta u}{\delta \rho}(\rho_i) \), and the unbiased estimates of the products \( \tilde{y}(\rho_i) \frac{\delta \tilde{y}}{\delta \rho}(\rho_i) \) and \( u(\rho_i) \frac{\delta u}{\delta \rho}(\rho_i) \) need to be obtained. The main problem is found in the computation of the latter two unbiased estimates. Hjalmarsson et al. show in [36] that additional experiments on the system produce unbiased estimates of these quantities, by re-injecting closed-loop data into the system while the controllers \( C_r(\rho_i) \) and \( C_y(\rho_i) \) are operational.

S.P. Mulders Master of Science Thesis
6-1 Theory on Iterative Feedback Tuning

6-1-2 Gradient estimation using output related signals

To come up with an estimate of the gradient \( \delta J / \delta \rho(\rho_i) \), the quantity \( \delta \tilde{y} / \delta \rho(\rho_i) \) needs to be obtained first. This is done by taking the partial derivative of \( \tilde{y}(\rho_i) \) in Equation (6-3) to \( \rho \)

\[
\frac{\delta \tilde{y}(\rho_i)}{\delta \rho} = \frac{G_0}{1 + C_y(\rho_i)G_0} \frac{\delta C_r(\rho_i)}{\delta \rho} r - \frac{C_r(\rho_i)G_0^2}{(1 + C_y(\rho_i)G_0)^2} \frac{\delta C_y(\rho_i)}{\delta \rho} r - \frac{G_0}{(1 + C_y(\rho_i)G_0)^2} \frac{\delta C_y(\rho_i)}{\delta \rho} v, \quad (6-10)
\]

which can be rewritten as

\[
\frac{\delta \tilde{y}(\rho_i)}{\delta \rho} = \frac{1}{C_r(\rho_i)} \frac{\delta C_r(\rho_i)}{\delta \rho} T_0(\rho_i) r - \frac{1}{C_r(\rho_i)} \frac{\delta C_y(\rho_i)}{\delta \rho} (T_0^2(\rho_i)r + T_0(\rho_i)S_0(\rho_i)v), \quad (6-11)
\]

now note that

\[
T_0^2(\rho_i)r + T_0(\rho_i)S_0(\rho_i)v = T_0(\rho_i)y, \quad (6-12)
\]

and therefore Equation (6-11) can be written as

\[
\frac{\delta \tilde{y}(\rho_i)}{\delta \rho} = \frac{1}{C_r(\rho_i)} \left[ \left( \frac{\delta C_r(\rho_i)}{\delta \rho} - \frac{\delta C_y(\rho_i)}{\delta \rho} \right) T_0(\rho_i) r + \frac{\delta C_y(\rho_i)}{\delta \rho} T_0(\rho_i) (r - y) \right]. \quad (6-13)
\]

The partial derivative of both controllers with respect to \( \rho \) can be computed, since the fixed-structure is known on beforehand. However, the sensitivity and complementary sensitivity function \( S_0(\rho_i) \) and \( T_0(\rho_i) \) are not, and experiments on the actual closed-loop system need to be performed to obtain an estimate for \( \delta \tilde{y} / \delta \rho(\rho_i) \). For this two-degrees-of-freedom set-up, the IFT tuning algorithm needs to collect data from three experiments for each controller update iteration. The first and third experiment are data-collection experiments of closed-loop data under normal operating conditions of the system, while the second experiment (referred to as the gradient experiment) re-injects the difference between data from the first experiment and the reference \( r \) to the system.

Summarizing, considering the output related signals, the following three experiments for data collection need to be performed

\[
y_1(\rho_i) = T_0(\rho_i)r + S_0(\rho_i)v_{i,1}, \quad (6-14)
\]
\[
y_2(\rho_i) = T_0(\rho_i)(r - y_1(\rho_i)) + S_0(\rho_i)v_{i,2}, \quad (6-15)
\]
\[
y_3(\rho_i) = T_0(\rho_i)r + S_0(\rho_i)v_{i,3}. \quad (6-16)
\]

Now, \( \tilde{y}(\rho_i) \) can be obtained by means of the first closed-loop experiment
\[ \hat{y}(\rho_i) = y_1(\rho_i) - y_d, \]  
(6-17)

and by rewriting (6-13) and substituting the data found in (6-14)-(6-16), an estimate of \( \delta \hat{y}/\delta \rho \) can be computed by

\[
\text{est} \left[ \frac{\delta \hat{y}}{\delta \rho} (\rho_i) \right] = \frac{1}{C_r(\rho_i)} \left[ \left( \frac{\delta C_r}{\delta \rho} (\rho_i) - \frac{\delta C_y}{\delta \rho} (\rho_i) \right) y_3(\rho_i) + \frac{\delta C_y}{\delta \rho} (\rho_i) y_2(\rho_i) \right] 
\]

(6-18)

\[
= \frac{\delta \hat{y}}{\delta \rho} (\rho_i) + \frac{S_0(\rho_i)}{C_r(\rho_i)} \left[ \left( \frac{\delta C_r}{\delta \rho} (\rho_i) - \frac{\delta C_y}{\delta \rho} (\rho_i) \right) u_{i,3} + \frac{\delta C_y}{\delta \rho} (\rho_i) u_{i,2} \right]. 
\]

(6-19)

It can be seen that the obtained quantity is a perturbed estimate of \( \delta \hat{y}/\delta \rho(\rho_i) \), which is from now on denoted by \( \text{est} [\delta \hat{y}/\delta \rho(\rho_i)] \).

### 6-1-3 Gradient estimation using input related signals

Now that an estimate of \( \delta \hat{y}/\delta \rho(\rho_i) \) is obtained, the only quantity that still needs to be determined is the gradient of input \( u(\rho_i) \) with respect to the control parameters \( \rho \). In a similar fashion as the derivation described above, an estimate of \( \delta u/\delta \rho(\rho_i) \) will be generated using closed-loop input data. The input signal and its partial derivative with respect to \( \rho \) are respectively described by

\[
u(\rho_i) = \frac{C_r(\rho_i)}{1 + C_y(\rho_i) G_0} r - \frac{C_y(\rho_i)}{1 + C_y(\rho_i) G_0} v = S_0(\rho_i) (C_r(\rho_i) r - C_y(\rho_i) v), \]

(6-20)

\[
\frac{\delta u}{\delta \rho} (\rho_i) = S_0(\rho_i) \left( \frac{\delta C_r}{\delta \rho} r - \frac{\delta C_y}{\delta \rho} v \right) + \frac{\delta S_0}{\delta \rho} (\rho_i) (C_r(\rho_i) r - C_y(\rho_i) v), \]

(6-21)

where \( \delta S_0/\delta \rho(\rho_i) \) is found by taking the partial derivative of \( S_0(\rho_i) \) from Equation (6-3) to \( \rho \)

\[
\frac{\delta S_0}{\delta \rho} (\rho_i) = -\frac{\delta C_y}{\delta \rho} (\rho_i) G_0 (1 + G_0 C_y(\rho_i))^2. \]

(6-22)

Further expansion of (6-21) and substitution of (6-22) yields the following result
Using the data collected in (6-30)-(6-32), the following estimate of the gradient can be generated.

\[
\frac{\delta u}{\delta \rho}(p_i) = S_0(p_i) \left[ \frac{\delta C_r}{\delta \rho}(p_i) r - \frac{\delta C_y}{\delta \rho}(p_i) v - \frac{\delta C_y}{\delta \rho}(p_i) T_0(p_i) r + \frac{\delta C_y}{\delta \rho}(p_i) C_y(p_i) T_0(p_i) v \right] \tag{6-23}
\]

Taking the expectation with respect to the disturbance cost function gradient, which is needed for the stochastic approximation algorithm to work properly. The motivation for the third experiment is to end up with an unbiased estimate of the cost function gradient, which is needed for the stochastic approximation algorithm to work properly. Taking the expectation with respect to the disturbance \( v \)
one ends up with an unbiased estimate of the gradient of the cost function and can be implemented in the
control parameter update equation in (6-8).

\[ E \left( \text{est} \left[ \frac{\delta J}{\delta \rho} (\rho_i) \right] \right) = \frac{\delta J}{\delta \rho} (\rho_i), \quad (6-36) \]

6-2 Application of IFT on feedforward IPC

In the previous section, the general IFT framework is given for a two-degrees-of-freedom controller. The IFT algorithm performs closed-loop controller optimizations, without the need for a model and by only using in-/output data and a special gradient experiment. In this section, the mathematical implementation of IFT on feedforward IPC is given in a SISO and MIMO set-up. The implementation of IFT on feedforward IPC for 1P and 2P fatigue blade load reductions is a contribution of this work. For optimization of feedforward IPC, an open-loop set-up will be used, where the controller is only scheduled on the current rotor azimuth angle \( \psi \). This method for IPC will be referred to as cyclic-pitch control in the sequel of this work. Performing open-loop experiments is possible, because the wind turbine is stable without the addition of IPC to the pitch signal, while the only aim is to reduce the 1P and 2P periodic loads. A schematic overview of the set-up for IFT optimization of cyclic-pitch IPC is given in Figure 6-2. Compared to Figure 3-1, where blade load measurements \( M_y \) are directly fed to the feedback controller, the blade load signals are collected in open-loop and used to optimize the rotor position dependent cyclic-pitch controller.

First, the mathematical SISO and MIMO implementation of the IFT algorithm will be discussed in Sections 6-2-1 and 6-2-2, respectively. This theory will be applied to the identified control-oriented linear model (see Chapter 5) in MATLAB Simulink \[11\] in Section 6-3-1, and the implementation in high-fidelity wind turbine simulation software Bladed \[10\] is described in Section 6-3-2.

6-2-1 IFT SISO application on feedforward IPC

In the Single-Input Single-Output (SISO) case, the IFT algorithm aims to optimize the cyclic-pitch controller for minimization of 1P and 2P harmonic loads, under the assumption that both blade properties are identical. Therefore, the controller is only optimized with respect to the load measurements of blade 1. The optimized pitch signal \( \theta_1(\psi, \rho_i) \) is applied to blade 2 with a phase shift of 180 degrees, \( \theta_2(\psi, \rho_i) = \theta_1(\psi + \pi, \rho_i) \). The fixed-structure cyclic-pitch controller is defined as

\[
C_{\text{SISO}}(\psi, \rho_i) = \theta_1(\psi, \rho_i) = \rho_1 \sin \psi + \rho_2 \cos \psi + \rho_3 \sin 2\psi + \rho_4 \cos 2\psi, \quad (6-37)
\]

\[
\theta_2(\psi, \rho_i) = \theta_1(\psi + \pi, \rho_i). \quad (6-38)
\]

All the control parameters are collected in a single vector \( \rho \). For optimization of the cyclic-pitch controller by IFT, a cost function \( J_{\text{SISO}}(\rho_i) \) on the blade 1 load measurement signal \( y_1 \) is defined as follows.
Figure 6-2: Implementation of IPC based on the current rotor azimuth position \( \psi \). This method of IPC is named cyclic-pitch. Blade load measurements \( M_{y,1} \) and \( M_{y,2} \) are collected in open-loop and are used to update the cyclic-pitch controller.

\[
J_{SISO}(\rho_i) = \frac{1}{2N} \sum_{t=1}^{N} \left( y_1(t, \rho_i)y_1^T(t, \rho_i) \right).
\]  
(6-39)

As can be seen, the cost function \( J_{SISO}(\rho_i) \) is minimized when the magnitude of the periodic load is minimized. The gradient of the cost function with respect to all the control parameters is defined by

\[
\frac{\delta J_{SISO}}{\delta \rho}(\rho_i) = \frac{1}{N} \sum_{t=1}^{N} \left( y_1(t, \rho_i) \left[ \frac{\delta y_1}{\delta \rho}(t, \rho_i) \right]^T \right),
\]  
(6-40)

where \( y_1(t, \rho_i) \) and is given by

\[
y_1(t, \rho_i) = G_0(t)C_{SISO}(\psi, \rho_i) + v_1(t),
\]  
(6-41)

and \( \delta y_1 / \delta \rho(t, \rho) \) follows from Equation (6-41) and is defined as
\[
\frac{\delta y_1}{\delta \rho}(t, \rho_i) = G_0(t) \frac{\delta C_{SISO}}{\delta \rho}(\psi, \rho_i) + v_1(t),
\]

(6-42)

where \( G_0(t) \) is the (non-)linear plant model. In the notation of \( G_0(t) \), the time argument is included to indicate that the IFT algorithm can also be applied to Linear Time-Variant (LTV) types of systems in the non-linear case. However, when the time argument is omitted from \( G_0 \), the plant turns into an LTI model, which will be used during the implementation of IFT to the identified linear model.

To obtain the gradient in Equation (6-42), gradient experiments need to be performed. Hjalmarsson et al. [29] describes a method where only one gradient experiment is needed to come up with the gradients with respect to all control parameters. This method is employed for IFT optimization of the cyclic-pitch controller: during the gradient experiment, the blade load signal output of the model is filtered through an array of partial derivatives of the controller in Equation (6-37) with respect to parameter vector \( \rho \).

Another option is to perform one additional gradient experiment, which results in the ability to obtain an unbiased estimate of the Hessian \( R_i \) (see Equation (6-8)), and leads to faster convergence of the algorithm [32]. However, comparing results using a biased and unbiased estimate of the Hessian, shows that the unbiased estimate has a very limited effect on convergence speed iteration wise, and because of the extra experiment, the overall optimization takes more time. For this reason, IFT is implemented with a biased estimate of the Hessian, resulting in the need for one normal and one gradient experiment. In the next section, the mathematical implementation of IFT in the MIMO set-up is elaborated.

**6-2-2 IFT MIMO application on feedforward IPC**

The MIMO implementation of the IFT algorithm is quite similar to the SISO case, however some important assumptions are made which need clarification and justification. To begin with, a different cost function \( J_{MIMO}(\rho_i) \) will be considered, which takes into account the minimization of both blade loads, by generating individual parameterized cyclic-pitch controllers for both blades. For this reason, the parameter vector \( \rho_i \) will be split in two parts

\[
\rho_i = \begin{bmatrix} \rho_{i,14} & \rho_{i,58} \end{bmatrix} = [\rho_1 \rho_2 \rho_3 \rho_4 | \rho_5 \rho_6 \rho_7 \rho_8].
\]

(6-43)

This leads to a decentralized controller \( C_{MIMO}(\psi, \rho_i) \), in which both pitch angles are parameterized individually

\[
C_{MIMO}(\psi, \rho_i) = \begin{cases} 
\theta_1(\psi, \rho_{i,14}) = \rho_1 \sin \psi + \rho_2 \cos \psi + \rho_3 \sin 2\psi + \rho_4 \cos 2\psi \\
\theta_2(\psi, \rho_{i,58}) = \rho_5 \sin \psi + \rho_6 \cos \psi + \rho_7 \sin 2\psi + \rho_8 \cos 2\psi.
\end{cases}
\]

As can be seen, a total of eight parameters need to be optimized by performing IFT in the MIMO case for 1P and 2P blade load reductions. The cost function \( J^*_i \) is given by

S.P. Mulders  
Master of Science Thesis
\[ J_{\text{MIMO}}^{\star}(\mathbf{p}_i) = \frac{1}{2N} \sum_{t=1}^{N} \left( \mathbf{y}_1(t, \mathbf{p}_{i,14}) \mathbf{y}_1^T(t, \mathbf{p}_{i,14}) + \lambda \left\| \mathbf{y}_1(t, \mathbf{p}_{i,14}) - \mathbf{y}_2(t, \mathbf{p}_{i,58}) \right\| + \mathbf{y}_2(t, \mathbf{p}_{i,58}) \mathbf{y}_2^T(t, \mathbf{p}_{i,58}) \right). \] (6-44)

A cross term between \( y_1 \) and \( y_2 \) with a weighing factor \( \lambda \) is included in \( J_{\text{MIMO}}^{\star}(\mathbf{p}_i) \) in Equation (6-44). However, as concluded during the linear model identification in Chapter 5, the interactions between both blade loads at the considered harmonic frequencies are negligible, which follows from the Relative Gain Array (RGA). For this reason, the cross term will be neglected yielding the following simplified cost function

\[ J_{\text{MIMO}}(\mathbf{p}_i) = \frac{1}{2N} \sum_{t=1}^{N} \left( \mathbf{y}_1(t, \mathbf{p}_{i,14}) \mathbf{y}_1^T(t, \mathbf{p}_{i,14}) + \mathbf{y}_2(t, \mathbf{p}_{i,58}) \mathbf{y}_2^T(t, \mathbf{p}_{i,58}) \right). \] (6-45)

The ideal cyclic-pitch signal is optimized for each blade individually, which makes the control set-up more suitable for practical implementation. In practice, blades are not completely identical causing a rotor imbalance. Also systematic measurement errors [37] are better dealt with by performing IFT on the individual blade loads. The partial derivative of the cost function is now taken with respect to all control parameters, yielding the expression

\[ \frac{\delta J_{\text{MIMO}}}{\delta \mathbf{p}}(\mathbf{p}_i) = \frac{1}{N} \sum_{t=1}^{N} \left( \mathbf{y}_1(t, \mathbf{p}_{i,14}) \left[ \frac{\delta \mathbf{y}_1}{\delta \mathbf{p}}(t, \mathbf{p}_{i,14}) \right]^T + \mathbf{y}_2(t, \mathbf{p}_{i,58}) \left[ \frac{\delta \mathbf{y}_2}{\delta \mathbf{p}}(t, \mathbf{p}_{i,58}) \right]^T \right). \] (6-46)

Because the blade root moment \( y_1 \) is only dependent on \( \mathbf{p}_{i,14} \), and \( y_2 \) only on \( \mathbf{p}_{i,58} \), the derivatives of the cost function \( J_{\text{MIMO}}(\mathbf{p}_i) \) end up separate for either \( y_1 \) or \( y_2 \). This simplifies the problem to two times evaluating a set of four independent parameters

\[ \frac{\delta J_{\text{MIMO}}}{\delta \mathbf{p}}(\mathbf{p}_i) = \left[ \begin{array}{c} \frac{\delta J_{\text{MIMO}}}{\delta \mathbf{p}_{i,14}}(\mathbf{p}_{i,14}) \\ \frac{\delta J_{\text{MIMO}}}{\delta \mathbf{p}_{i,58}}(\mathbf{p}_{i,58}) \end{array} \right] = \left[ \begin{array}{c} \frac{1}{N} \sum_{t=1}^{N} \mathbf{y}_1^T(t, \mathbf{p}_{i,14}) \frac{\delta \mathbf{y}_1}{\delta \mathbf{p}_{i,14}}(t, \mathbf{p}_{i,14}) \\ \frac{1}{N} \sum_{t=1}^{N} \mathbf{y}_2^T(t, \mathbf{p}_{i,58}) \frac{\delta \mathbf{y}_2}{\delta \mathbf{p}_{i,58}}(t, \mathbf{p}_{i,58}) \end{array} \right]. \] (6-47)

The fact that IFT optimizations for both blades are now separate, also means that (like in the SISO case) only one normal and one gradient experiment is needed for optimization of the complete control parameter vector \( \mathbf{p} \). The output equations for the blade load signals \( y_1(t, \mathbf{p}_{i,14}) \) and \( y_2(t, \mathbf{p}_{i,58}) \) are defined as

\[ y_1(t, \mathbf{p}_{i,14}) = G_0(t) C_{\text{MIMO},1}(\psi, \mathbf{p}_{i,14}) + v_1(t), \] (6-48)

\[ y_2(t, \mathbf{p}_{i,58}) = G_0(t) C_{\text{MIMO},2}(\psi, \mathbf{p}_{i,58}) + v_2(t), \] (6-49)

with the partial derivatives given by
\[
\delta y_1(t, \rho_{i,14}) = G_0(t) \frac{\delta C_{\text{SISO}}}{\delta \rho_{14}}(\psi, \rho_{14}) + v_1(t), \quad (6-50)
\]
\[
\delta y_2(t, \rho_{i,58}) = G_0(t) \frac{\delta C_{\text{SISO}}}{\delta \rho_{58}}(\psi, \rho_{58}) + v_2(t). \quad (6-51)
\]

A last important remark is made on the construction of the Hessian \( R_i \) using the gradients in Equations (6-50) and (6-51). As stated earlier in this section, blade loads for both blades are minimized separately because interactions are negligible (see Section 5-2), and the Hessian \( R_i \) is chosen to have the following form

\[
R_i = \begin{bmatrix} R_{14} & 0 \\ 0 & R_{58} \end{bmatrix}, \text{ where}
\]

\[
R_{14} = \left( \frac{\text{est} \, \delta y_1(t, \rho_{i,14})}{\delta \rho_{14}} \right) \left( \frac{\text{est} \, \delta y_1(t, \rho_{i,14})}{\delta \rho_{14}} \right)^T,
\]

\[
R_{58} = \left( \frac{\text{est} \, \delta y_2(t, \rho_{i,58})}{\delta \rho_{58}} \right) \left( \frac{\text{est} \, \delta y_2(t, \rho_{i,58})}{\delta \rho_{58}} \right)^T.
\] (6-52)

In the next section, general ideas of the described IFT framework will be used for optimization of cyclic-pitch control. Application of IFT on the identified linear model with zero DC-offset is fairly straightforward and is described in Section 6-3-1. However, the implementation on the non-linear model, which has a realistic blade load signal as output, requires additional filter design and implementation and will be outlined in Section 6-3-2.

### 6-3 Implementation on the linear and non-linear model

The IFT algorithm implementation on the linear and non-linear wind turbine models in \textit{Simulink} and \textit{Bladed}, are described in this section. As the algorithm can be tested conveniently and fast on the linear model, development of IFT for feedforward IPC will be performed in \textit{Simulink}. When the IFT optimization code is implemented correctly, with some mild changes, the code can be compiled by the \textit{MATLAB} compiler into a file which can be linked to \textit{Bladed} as an external controller. For the linear case, the algorithm is implemented for the SISO and MIMO set-up, while for the non-linear case IFT is only implemented in the MIMO configuration.

First the linear \textit{Simulink} implementation of the algorithm will be elaborated in Section 6-3-1, after which an explanation of the non-linear \textit{Bladed} implementation is given in Section 6-3-2. In the latter section, a pseudocode of IFT in the MIMO case is given for clarification purposes of the full algorithm.

#### 6-3-1 IFT implementation on the linear model

This section will elaborate the implementation of IFT on the linearized model in \textit{MATLAB Simulink}, which is obtained in Chapter 5. The model is implemented in a SISO and MIMO...
set-up, of which the mathematical implementation details are given in Section 6-2. The inputs on the linearized model are the pitch signals $\theta_i$ and the outputs represent the blade load measurements $M_{y,i}$. Arbitrary periodic and sinusoidal disturbance load signals $v_i(t)$ are added to both outputs, where both disturbance signals have an opposite sign. Schematic overviews of the SISO and MIMO Simulink implementations are given in Figures 6-3 and 6-4 respectively. The former figure presents the IFT Simulink implementation, only including the linear model of the first blade, while the latter figure demonstrates the set-up incorporating linear models of both blades with cross terms.

### 6-3-2 IFT implementation on the non-linear model

Implementation of the IFT algorithm on the non-linear Bladed model involves additional implementation obstacles which need to be solved. Because the Bladed wind turbine model is able to simulate wind turbine effects throughout all operating regions, control systems such as torque control and Collective Pitch Control (CPC) need to be active. As an addition to this, IFT uses realistic blade load measurements to optimize for a periodic (rotor position dependent) addition to the pitch angle, which minimizes blade fatigue loads. As explained in Section 6-2, the cyclic-pitch controller is implemented in a feedforward open-loop setting, and it will not be needed for the gradient experiment to perturb the system by re-injecting a previously recorded signal, as is the case in closed-loop configurations (see Section 6-1). A schematic overview of the implementation of IFT on the non-linear model is given in Figure 6-2, where blade load measurements $M_y$ are stored and filtered to iteratively update the cyclic-pitch controller.

The IFT algorithm uses a cost function $J(\rho)$ based on minimization of blade loads, which is minimized with respect to a certain set of control parameters $\rho$, to attain the most optimal performance in terms of load reductions. An example time-series of an unfiltered blade load signal during high turbulence is presented in Figure 6-6 (blue line). Using this unfiltered signal would yield undesired results, as it incorporates a DC-offset and additional (high-frequency) turbulent effects which will not be reduced by the IFT-IPC control scheme. The DC-offset cannot be reduced as a whole by IPC, as this would interfere with CPC used for
rotor speed regulation. The main objective is to reduce the periodic $nP$ blade loads, which are multiples of the rotational speed of the wind turbine rotor. Some sort of filtering around the $nP$ frequencies is needed to propose an effective cost function $J(\rho)$.

Two main types of filters are available for implementation, namely Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters. While IIR filters are easily implementable and generally of low order, FIR filters are of higher order and can filter offline stored signals with zero-phase distortion. An elaborate description of IIR and FIR filters can be found in Appendix A-2. Because the IFT algorithm requires to have pre-recorded data sets for performing the controller update, this condition is met and various advantages of the FIR filter type can be exploited. A bandpass filter passing the $1P$ and $2P$ frequencies in both below- and above-rated conditions is designed by the MATLAB filterbuilder, which allows for GUI-based filter design according to user-defined filter specifications [38]. The order of the obtained filter is 988 and a magnitude plot of the digital filter is presented in Figure 6-5. The FIR filter passing frequencies between $1P$ and $2P$ will be incorporated in the non-linear IFT implementation. An example of a $1P/2P$ filtered blade load with zero-phase distortion is presented by the green dashed line in Figure 6-6.

The function \texttt{filtfilt}, which is part of the MATLAB Signal Processing Toolbox, provides a comprehensive way of FIR filtering an already stored signal [39]. Because the \texttt{filtfilt} function cannot be incorporated in the Simulink template designed for Bladed controller design, a C-implementation of the \texttt{filtfilt} function by [40] is modified and implemented.

\textbf{Figure 6-5:} Magnitude frequency response of the digital FIR filter. This filter is used to obtain a zero-phase filtered signal of the original blade load signal, at $1P$ and $2P$ frequencies. The order of the filter equals 988.
Figure 6-6: Blade load measurement signals, filtered by a non-causal FIR filter resulting in zero-phase distortion. The original signal is represented by the solid blue line (turbulence intensity 15%), while the filtered 1P - 2P signal is represented by the dashed green line.

in Simulink. The implementation uses forward and reverse filtering to end up with a filtered signal, without any phase distortion (zero-phase filtering). The modified version of the `filtfilt` C-implementation, called `filtfiltC`, is given in Appendix A-1, and is part of the non-linear IFT algorithm. A pseudocode of the MIMO IFT algorithm implementation is given in Algorithm 1.
Algorithm 1 IFT MIMO implementation for cyclic-pitch IPC optimization for wind turbine load reductions

procedure IFT–IPC–MIMO(t, \( \psi \), \( y_1 \), \( y_2 \), \( b_k \))

Set amount of samples to be captured \( N \) (dependent on constant or turbulent wind);
Set \( N_{\text{mask}} \) for mask on filtered signal to remove begin/end effects;
Set amount of full rotations \( N_{\text{rot}} = 0 \);
Set initial transient time \( t_{\text{trans}} \);
Set control parameter vector \( \rho_0 = 0 \);
\( \text{phase}_1 = \text{true}; \ \text{phase}_2 = \text{false}; \ \text{phase}_3 = \text{false}; \ \text{normalexp}_1 = \text{false}; \ \text{gradexp}_1 = \text{false}; \)

for \( i = 1 \) to amount of IFT iterations do

\[ \theta_1(\rho_{14}) = \rho_1 \sin(\psi) + \rho_2 \cos(\psi) + \rho_3 \sin(2\psi) + \rho_4 \cos(2\psi) \]
\[ \theta_2(\rho_{58}) = \rho_5 \sin(\psi) + \rho_6 \cos(\psi) + \rho_7 \sin(2\psi) + \rho_8 \cos(2\psi) \]

if rotor is in upright position then

\( \text{RotorUp} = \text{true}; \)
end if

if \((\text{phase}_1 \ \text{AND} \ \text{RotorUp}) \ | \ | \ \text{normalexp}_1\) then

Set \( \text{normalexp}_1 = \text{true}; \)
Record data 1 \( \text{y}_1,\text{record}_1 (\text{cnt}_1) = y_1(t); \)
Record data 2 \( \text{y}_2,\text{record}_1 (\text{cnt}_1) = y_2(t); \)
Increment \( \text{cnt}_1 \) by 1;
if \( \text{RotorUp} == \text{true} \) then

Increment \( N_{\text{rot}} \) by 1;
if amount of rotations \( N_{\text{rot}} == 4 \) then

Set \( N_{\text{mask}} \) to the amount of recorded samples \( \text{cnt}_1; \)
end if
end if
end if

if \( \text{cnt}_1 \) equals \( N \) then

Filter \( \text{y}_1,\text{record}_1 \) with \( \text{filtfilt}(b_k) \) 1P/2P zero-phase filter;
Go to phase 2: \( \text{phase}_1 = \text{false}; \ \text{phase}_2 = \text{true}; \)
end if

Go to next time step, \( t = t + T_s; \)
end while
while phase2 do
  \( \theta_1 = 0; \)
  \( \theta_2 = 0; \)

  if Rotor is in upright position then
    RotorUp = true;
  end if

  if (phase2 AND RotorUp) || gradexp1 then
    Set gradexp = true;
    Record data \( y_{1,\text{record2}}(1 : 4, \text{cnt}_2) = [\sin(\psi) \cos(\psi) \sin(2\psi) \cos(2\psi)]^T y_1(t); \)
    Record data \( y_{2,\text{record2}}(1 : 4, \text{cnt}_2) = [\sin(\psi) \cos(\psi) \sin(2\psi) \cos(2\psi)]^T y_2(t); \)
    Increment \( \text{cnt}_2 \) by 1;
  end if

  if \( \text{cnt}_2 \) equals \( N \) then
    Filter \( y_{1,\text{record2}} \) with \text{filtfiltC}(b_k) 1P/2P zero-phase filter;
    Go to phase 3: phase2 = false; phase3 = true;
  end if

  Go to next time step, \( t = t + T_s \);
end while

while phase3 do
  \( \theta_1(\rho_{14}) = \rho_1 \sin(\psi) + \rho_2 \cos(\psi) + \rho_3 \sin(2\psi) + \rho_4 \cos(2\psi) \)
  \( \theta_2(\rho_{58}) = \rho_5 \sin(\psi) + \rho_6 \cos(\psi) + \rho_7 \sin(2\psi) + \rho_8 \cos(2\psi) \)

  Crop \( N_{\text{mask}} \) sampled from start and end of \( y_{12,\text{record1}} \) and \( y_{12,\text{record2}} \);
  Compute cost function value \( J(\rho_i); \)
  Compute cost function gradient \( \delta J/\delta \rho_i(\rho_i); \)
  Compute Hessian \( R_i(\rho_i) \), only including the two diagonal block matrices;

  Update controller parameter vector: \( \rho_{i+1} = \rho_i - \gamma R_i^{-1}(\rho_i) \delta J/\delta \rho_i(\rho_i); \)

  Go to next iteration \( i \), reset arrays and phase3 = false, phase1 = true;
end while

Go to next time step, \( t = t + T_s \);
end for
end procedure
Chapter 7

Results on feedforward IPC and IFT

In the previous chapter, the Iterative Feedback Tuning (IFT) framework and the application of the algorithm on feedforward Individual Pitch Control (IPC) are outlined. In this section, results on the linear and non-linear implementations of IFT will be discussed. During non-linear simulations in Bladed, constant and turbulent wind conditions will be used to optimize the cyclic-pitch controller in ideal and realistic scenarios. Similarly, a noise is added to the outputs of the linear model to mimic the effect of turbulence.

Section 7-1 discusses the results of IFT on the linear model, and Section 7-2 gives results on the non-linear implementation during ideal constant and realistic turbulent wind conditions. Only the results of the Multiple-Input Multiple-Output (MIMO) set-up will be considered for both models, as the Single-Input Single-Output (SISO) configuration is only used for the development of the IFT MIMO implementation.

7-1 Linear model MIMO optimization results

In this section, the results of the implementation of the IFT algorithm on the linear model are discussed. First graphs of the progress of optimization of the parameter vector $\rho$ for constant and turbulent wind conditions are presented in Figure 7-1, where zero-mean Gaussian noise is added to the disturbance channels $v_i$ to mimic the effect of turbulence. It can be observed that in the case of constant wind, the control parameters in $\rho$ converge to optimum values, which minimize the predefined cost function $J(\rho)$. In the case where Gaussian noise is included, the algorithm converges (on average) to equal control parameters in the same period of time, while following a more variable path.

The obtained periodic pitch signals and the disturbance rejection capabilities are presented in Figure 7-2. It appears that the found pitch signal is a non-smooth periodic pitch signal, but this was expected as two arbitrary chosen 1P and 2P disturbance signals $v_i$ were implemented on the model output. The optimized pitch signals for disturbance attenuation found by the IFT algorithm are implemented after 10 s, and shows its capability of near complete reduction of the imposed disturbances $v_i$. 

Master of Science Thesis

S.P. Mulders
54 Results on feedforward IPC and IFT

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{IFT_MIMO_optimization.png}
\caption{Convergence of the elements in the control parameter vector $\rho$ for constant and turbulent wind conditions. In both cases, the algorithm converges on average to the same optimal gains for the periodic pitch signal, while taking the same amount of iterations. The solid lines represent the gains related to the pitch signal for blade 1, while the dashed lines show the convergence progress for the gains related to blade 2.}
\end{figure}

The goal of implementation on the linear model is to show the capability of the algorithm to work in the desired set-up, as a preparation for implementation on the non-linear model, which will be discussed in Section 7-2.

### 7-2 Non-linear model MIMO optimization results

During simulations on the non-linear model at different wind speeds, it appeared that the step size $\gamma$ of the IFT algorithm had to be modified for every single wind speed, to get a descent convergence speed. Because the magnitudes of the periodic loads at different wind speeds differ significantly from each other, and the fact that the algorithm uses these signals to optimize the cyclic-pitch controller, causes the algorithm to have different convergence rates with a fixed $\gamma$. To circumvent this issue, the step size is expressed after the first iteration in a fixed and empirically found norm of the parameter vector and the 2-norm of the cost function gradient in the following way

$$\gamma = \frac{\eta}{\|\frac{\delta J}{\delta \rho(\rho_1)}\|_2}.$$  \hspace{1cm} (7-1)

The fixed and empirically found parameter $\eta$ is found being optimal when it is set to 0.017 in constant wind cases, and to 0.010 during simulations involving turbulent wind conditions. As stated earlier, wind speeds that will be investigated are 8, 14 and 20 m/s; where the lowest wind speed causes the wind turbine to operate in the below-rated region, while the latter two result in an above-rated angular rotor speed. Also, for the three earlier mentioned wind speeds, turbulent wind files with a maximum length of 5000 s are generated to prevent correlation when multiplying stored load signal matrices, which may cause the algorithm to diverge. Therefore, the step size was empirically found to be adequate to ensure descent convergence for the above-mentioned wind speeds. The results for the optimization process are visualized in Figure 7-1.
to fail. Graphs of the optimization progress of the parameter vectors $\rho$, during predefined constant and turbulent wind speeds are presented in Figure 7-3.

The IFT optimization algorithm succeeds in the optimization of the parameter vector $\rho$ for the controller $C_{\text{MIMO}}(\psi, \rho)$ in the above-rated wind conditions. The optimization for the below-rated wind speed of 8 m/s shows good convergence in the case of a constant wind speed, but shows poor results for turbulent wind conditions. Blade load data sets of $N$ rotations are used by the IFT algorithm for the cyclic-pitch controller update, and the continuous changing rotor speed in the below-rated region causes the iterations to show a much more erratic behavior. A blade load power spectrum at the various wind speeds is given in Figure 7-4, where the broader frequency range of the 1P peak at 8 m/s can also be observed. Zooming in to the convergence progress of the constant below-rated 8 m/s wind speed shows higher oscillation transients, before reaching the final and optimal controller gains. This is most probably caused by the lower dominance of the 2P peak in the blade load spectrum.

The minimization progress of the predefined cost function $J(\rho)$ is for both constant and turbulent wind conditions presented in Figure 7-5. Here, it again becomes clear that for constant wind the algorithm has no problem finding the optimal controller gains. In turbulent and more realistic conditions, IFT is still able to find the optimum values, but follows a more vari-

Figure 7-2: IFT optimized pitch signals and load reduction results on the linear model. Upper left a plot with 1P and 2P components separated, in the upper right graph both frequency components are added to one pitch signal for each blade. In the lower plots, the IFT optimized pitch signal is activated after 10s and show near complete mitigation of the imposed disturbances, with and without added noise.
Figure 7-3: IFT convergence progress of the control parameter vector $\rho$ using different wind speeds and conditions. All constant wind optimizations show good and fast convergence; the same holds for optimizations during turbulent wind conditions. Only the optimization of the lowest wind speed (8 m/s) for turbulent wind shows erratic behavior, which is caused by the continuously changing rotor speed in the below-rated region. Turbulence intensity is 15% for all wind speeds.

A time-domain representation of the blade loads with and without the optimized cyclic-pitch controller is given in Figure 7-6, at the predefined wind speeds of 8, 14 and 20 m/s. In constant wind conditions, the harmonic fatigue load is reduced almost completely, while a significant reduction is seen in turbulent wind conditions. A more extensive comparison including power spectra en Damage Equivalent Load (DEL) comparisons is given in Chapter 8.

The optimized IFT-IPC pitch signals are graphically presented in Figure 7-7 for the predefined wind speeds. The signals are plotted with their corresponding scaled sine of the azimuth angle in the same color. It can be observed that all cyclic-pitch signals have a non-zero mean, while
Figure 7-4: Power spectrum of the blade load measurement signals during the three predefined wind speeds in turbulent conditions (15% intensity). The two highest wind speeds show clear 1P and 2P peaks at the same frequency. The lowest wind speed shows a less dominant 1P and 2P peak, and especially the 1P peak is broader, caused by the variable rotor speed in the below-rated region.

Figure 7-5: Convergence trajectories of the cost functions during predefined wind speeds and conditions. The cost for constant wind speeds show continuous convergence, while for turbulent wind a more variable behavior during controller optimizations is observed. Turbulence intensity is 15% for all wind speeds.
Figure 7-6: Blade loads with and without optimized IFT-IPC feedforward controller, represented by the red and blue line respectively, for wind speeds of 8, 14 and 20 m/s. For constant wind conditions, the fatigue harmonic load is reduced almost completely, while a significant reduction is seen in turbulent wind conditions. Turbulence intensity is 15% for all wind speeds.

The IPC pitch signal amplitude shows an alternating behavior, while in the above-rated region the amplitude scaling is linear. Interesting behavior is observed at a wind speed of 12 m/s, where the amplitude shows a large amplitude peak. A verification of the IFT optimized pitch signal with the signal obtained by Multi-Blade Coordinate (MBC) control during constant wind conditions shows an exact match, so an error in the optimization process is excluded. Because the blade pitch angles are kept at fine pitch for the considered wind speed, this results in increased static blade loads and actuator torques, as the wind speed of 12 m/s is
Figure 7-7: Optimized 1P/2P cyclic-pitch signals for wind speeds 8, 14 and 20 m/s. A scaled sine of the azimuth angle is plotted together with the corresponding IPC signal. All signals have a certain phase-shift with respect to the current azimuth angle. It is notable that the pitch signals all have a non-zero mean by the addition of the 2P component, a wider top and a sharper bottom.

Figure 7-8: Amplitude and phase of the 1P/2P cyclic-pitch signal with respect to a sine of the azimuth angle, plotted for various wind speeds (circles) and interpolated between results. It is clear that the pitch signal amplitude (left plot) shows random behavior in the below-rated region (8-13 m/s), and a linear behavior at above-rated wind speeds (13-22 m/s). The phase-shift with respect to the azimuth angle (right plot) stays approximately constant over all wind speeds.

the upper limit of the below-rated region. This effect probably explains the increased pitch signal amplitude.

It is important to stress again that the controller $C_{MIMO}(\psi, \rho)$ is a constant amplitude feedforward controller, scheduled on the rotor azimuth angle $\psi$ and is for each wind speed optimized to reduce periodic 1P and 2P blade loads. It would be interesting to see how this controller performs compared to the conventional MBC implementation. A time-domain plot in Figure 7-9 includes a comparison between the 1P pitch signals generated by feedforward and

Master of Science Thesis

S.P. Mulders
conventional IPC for constant wind conditions, and reveals that the IFT optimized signals exactly match the ones obtained by conventional IPC. This is an interesting result, as it indicates that in the ideal scenario, conventional MBC-IPC indeed implements the optimal signal to minimize the 1P blade load component and does not suffer from delays introduced by feedback control system; the turbulent case is discussed in Chapter 8.

Load and performance comparisons are given in Chapter 8, where the case of No-IPC is compared to feedforward IPC (IFT-IPC) and conventional feedback IPC (MBC-IPC). Additionally, gain-scheduling of the amplitude of the cyclic-pitch signal based on exogenous signals (GS-IFT-IPC), is compared IFT-IPC and MBC-IPC. For convenience reasons, only the 1P component is included in performance comparisons; including the 2P harmonic in the IPC signal does not significantly help to improve blade load reductions and introduces extra tuning parameters in the MBC-IPC implementation.
Figure 7-9: Time-domain simulation plot of the IFT optimized cyclic-pitch (blue circles) and conventional IPC (green line) 1P pitch signals at wind constant speeds of 8, 14 and 20 m/s. It appears that, after the transient of the conventional controller, signals at all wind speeds are identical.
Chapter 8

Gain-scheduling the feedforward controller

As described in Chapter 7, the obtained feedforward controller generates a constant amplitude pitch signal and is optimal for blade load reductions while the rotor is subjected to constant wind conditions. As presented in Figure 7-9, the obtained pitch signals with Iterative Feedback Tuning (IFT) match the Multi-Blade Coordinate (MBC) pitch signals for constant wind. In case of higher turbulence levels, the IFT algorithm also optimizes to a constant amplitude pitch signal, while feedback Individual Pitch Control (IPC) reacts to the varying intensity of the periodic blade loads. Inspecting the pitch signal generated by conventional IPC, a continuously changing phase of the implemented pitch signal is observed. This is presented in Figure 8-1, and because the feedforward IPC is dependent on a matching rotor azimuth signal, the effect is not driven by the varying rotor speed. For this reason, it is assumed that these phase changes are caused by the feedback configuration and filtering actions, present when using conventional IPC for load reductions.

To explore the above mentioned effect, a gain-schedule is added to the feedforward controller, changing the amplitude of the IFT-IPC pitch signal (GS-IFT-IPC). The purpose is to investigate whether the phase changes between periods of the pitch signal by conventional feedback IPC, improve or deteriorate load reduction performance. The amplitude of the cyclic-pitch signal is first scheduled on prior knowledge of the pitch signal amplitude from an already performed MBC-IPC simulation, under the same turbulent wind conditions. Next, an online gain-scheduling on the pitch signal amplitude is carried out, where controller scheduling parameter $\beta$ is obtained by a mapping on a filtered blade or tower load signal. The set-up is schematically presented in Figure 8-2.

In this chapter, a comparative analysis is made between GS-IFT-IPC and cases without IPC (No-IPC), fixed amplitude feedforward IPC (IFT-IPC) and conventional IPC (MBC-IPC). To make a fair comparison, the same turbulent wind conditions are used for all cases. Section 8-1 investigates the case where a priori pitch amplitude information is available from an earlier performed MBC-IPC simulation, while Section 8-2 extracts amplitude information online.
Gain-scheduling the feedforward controller

Figure 8-1: A comparison between the pitch signals of IFT-MBC and MBC-IPC. The IFT signal is the (optimal) mean of the MBC signal, both in terms of amplitude and phase. It is interesting to observe that the IFT-IPC signal has a constant, but rotor position dependent frequency, while MBC-IPC changes the phase of the generated pitch signal.

Figure 8-2: Scheduling the pitch signal amplitude of the cyclic-pitch controller on blade or tower load measurements, which in transformed to the scheduling parameter $\beta_{\text{blade/tower}}$ through a (non-)linear mapping.

from blade or tower loads. Section 8-3 gives a more in-depth analysis on how the MBC-IPC pitch signal is generated by the conventional IPC implementation.

8-1 Determining the effect of only changing the pitch amplitude

As already discussed in the introduction of this chapter, the influence of only changing the amplitude of the optimized IFT-IPC signal is investigated, yielding the GS-IFT-IPC pitch signal. The pitch signal amplitude information is extracted from an already performed simulation by MBC-IPC under the same conditions. Inspecting the MBC-IPC pitch signal in more detail, reveals that the MBC-IPC pitch signal leads or lags in time compared to the feedforward controller which is dependent on the same rotor azimuth angle (Figure 8-1). As the lead/lag phenomenon is not driven by the varying rotor speed, the effect of omitting these
phase changes, possibly caused by feedback control and filtering actions, is evaluated in this section.

To extract amplitude information from the MBC-IPC pitch signal, local maxima of the absolute pitch signal are detected by the *findpeaks* [41] function of the MATLAB Signal Processing Toolbox. The result is presented in Figure 8-3. The found maxima are interpolated and converted into factors for multiplication of IFT-IPC, yielding the GS-IFT-IPC signal (see Figure 8-4). The resulting GS-IFT-IPC signal is presented in Figure 8-5. It is immediately obvious that the (GS-)IFT-IPC signals have constant periodicities based on the current rotor azimuth position, while the MBC-IPC signal alternately leads and lags in time.

The now a priori amplitude scheduled GS-IFT-IPC pitch signal is implemented in Bladed under the exact same conditions as used during the MBC-IPC simulation. Additionally, a simulation without IPC is run to make a comparison of the attained performance with the various types of IPC. A performance comparison is made using the Damage Equivalent Load (DEL) and is given in Figure 8-7, where the No-IPC case is normalized to 100%. It is very clear that in general the performance of the GS-IFT-IPC strategy does not improve compared to IFT-IPC and MBC-IPC control. Considering the blade 1 IPC pitch and load spectra in Figure 8-6, it is shown that MBC-IPC is able to completely mitigate the 1P harmonic load, while (GS-)IFT-IPC is only able to partly reduce the periodic 1P load. However, the performance attained with MBC-IPC comes at expense of a higher pitch activity and increased actuator stresses and wear. It can be concluded that the changing phase of the MBC-IPC pitch signal (Figure 8-5) is needed to effectively reduce harmonic blade fatigue loads. As the phase changes are not driven by the varying rotor speed, a more in depth analysis of the MBC-IPC pitch signal is given in Section 8-3.
66 Gain-scheduling the feedforward controller

Figure 8-5: The MBC-IPC, IFT-IPC and GS-IFT-IPC signal compared. The (GS-)IFT-IPC signals show the same periodicities based on the rotor azimuth angle, while the MBC-IPC signal alternately leads and lags in time.

Figure 8-6: IPC pitch and blade load spectrum of the cases No-IPC, IFT-IPC, GS-IFT-IPC and MBC-IPC at a wind speed of 20 m/s. It is clear that all control implementations are able to reduce the blade load, but the MBC-IPC option still has a large advantage over (GS-)IFT-IPC. This comes at expense of higher pitch activity.

8-2 Scheduling on blade and tower loads

In the previous section, the MBC pitch signal was known on beforehand, which gave the possibility to schedule the amplitude of the IFT-IPC signal and to compare the results in terms of blade load reduction capabilities. In practical implementations, dynamical wind variations are not a priori known. A possibility to circumvent this issue is by using blade or tower load measurements for online adjustment of the feedforward IPC signal amplitude. Filtered load measurements can be used as an exogenous signal to schedule the cyclic-pitch controller, which allows the continuous modification of the pitch signal amplitude. In this
section, performance on scheduling the amplitude of the IPC signal on blade and tower loads is discussed, followed by a performance comparison. The implementation and performance results are given based on blade load scheduling, while only the implementation method for scheduling on tower loads is outlined.

The blade root moments \( M_y \) are filtered with a discretized inverted notch filter, scheduled on the angular rotor speed \([14]\). The continuous time transfer function of the parameterized inverted notch \([42]\) \( H_c(\omega_c, s) \) is given by

\[
H_c(\omega_c, s) = \frac{\omega_c^2}{s^2 + 2\omega_c s + \omega_c^2}
\]

Figure 8-7: A comparison using the Damage Equivalent Load shows that GS-IFT-IPC strategy does in general not improve the performance attained by IFT-IPC. The blade load reduction capabilities of MBC-IPC are superior to the other forms of IPC.
Gain-scheduling the feedforward controller

\[ H_c(\omega_c, s) = K \frac{s}{(Q/\omega_c s^2 + s + Q \omega_c)}, \quad (8-1) \]

where \( Q \) determines the sharpness and preciseness of the notch peak, \( \omega_c \) resembles the filtering frequency and is taken as the rotor speed (in this case the 1P frequency) and \( K \) represents a gain to increase or decrease the magnitude of the filtered signal. For ease of implementation, this continuous time transfer function is converted into its respective state-space representation, which is given by

\[
A_c(\omega_c) = \begin{bmatrix} -\omega_c/Q & -\omega_c^2 \\ 1 & 0 \end{bmatrix}, \\
B_c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\
C_c(\omega_c) = \begin{bmatrix} K\omega_c/Q \\ 0 \end{bmatrix}, \\
D_c = 0, \quad (8-2) \]

and this state-space representation will be discretized using the bilinear transformation (Tustin’s method) \([43]\), where

\[
A_d(\omega_c) = \left( I + \frac{1}{2}A_c(\omega_c)T_s \right) \left( I - \frac{1}{2}A_c(\omega_c)T_s \right)^{-1}, \\
B_d(\omega_c) = \sqrt{T_s} \left( I - \frac{1}{2}A_c(\omega_c)T_s \right)^{-1} B_c, \\
C_d(\omega_c) = \sqrt{T_s} C_c(\omega_c) \left( I - \frac{1}{2}A_c(\omega_c)T_s \right)^{-1}, \\
D_d(\omega_c) = \frac{T_s}{2} C_c(\omega_c) \left( I - \frac{1}{2}A_c(\omega_c)T_s \right)^{-1} B_c + D_c. \quad (8-3) \]

Taking into consideration the convergence speed and preciseness of the filter, the tuning parameter \( Q \) is set to a value 4 and the sampling time \( T_s \) is 0.02 s. A Bode magnitude and phase plot of the constructed inverted notch filter is given in Figure 8-8.

The absolute value of the output of the inverted notch filter signal is taken, and is twice per revolution used to determine the amplitude (peak) of the load signal (lower and upper peak of the original load signal). The mean value of these peaks is assumed to be corresponding to the optimal pitch signal found by the IFT optimization. This cyclic-pitch control signal is scaled according to a linear mapping between the current and mean peak value. The most optimal mapping is empirically found by comparing the effectiveness in reducing the 1P periodic blade load. The following linear relation is used for scaling of the IFT signal

\[
\text{scaling factor} = \frac{\text{current peak value}}{\text{mean peak value}} \alpha_i + (1 - \alpha_i), \quad (8-4) \]
where \( \alpha \) gives the linear scaling a certain slope, which will be taken in the range \([0, 0.3]\) with increments of 0.1, until the ideal slope is found. Because the amplitude changes are not continuous, a rate limiter is implemented after calculation of the pitch signal and before sending to the pitch actuator.

Simulations using the above described method are performed. The total simulation time is set to 2000 s, of which the first 1000 s are used for mean load peak calculations with the IFT-IPC controller active; the gain-schedule is activated in the last 1000 s (GS-IFT-IPC) and the obtained results from this part of the simulation are used for load reduction performance calculations.

The performance is evaluated by the blade pitch and load power spectra and the DEL on various parts of the turbine; results are presented in Figures 8-9 and 8-10 respectively. The power spectrum shows that all IPC control methods are able to significantly reduce the 1P periodic load, while MBC-IPC completely mitigates the 1P load. In the blade load spectrum it is observed that blade load reduction at 1P for GS-IFT-IPC is slightly better with amplitude scaling \((\alpha_3)\), than without \((\alpha_1)\). This comes at expense of increased pitch activity. In DEL results it is noticeable that in some cases, GS-IFT-IPC control outperforms MBC-IPC, while in other cases the contrary is true. Blade loads show, especially at lower wind speeds, that little improvement in terms of load reductions are attained. In the case of \( M_x \) and \( M_y \) tower load reductions, GS-IFT-IPC outperforms MBC-IPC, which is probably a result of the (nearly) constant pitch signal frequency in combination with the fixed timing where tower shadow and wind shear have an effect on tower loads. The case of the higher tower \( M_z \) component in the GS-IFT-IPC case is a direct result of the higher blade load component, which also has an immediate effect on the rotating hub loads.

The above obtained results are all performed on the basis of filtered blade load measurement
signals. Because blade load sensors are expensive and generally carried out redundantly, load measurement devices equipped at the turbine fixed structure, in for example the tower top, can also be used for GS-IFT-IPC. Imagining tower loads of a two-bladed turbine, reveals that especially the twice-per-revolution 2P periodic load is present due the amount of blades. Fixed-structure loads are caused by effects which occur from wind shear and tower shadow twice per revolution. For this reason, the tower moment in the y-direction \( M_{t,y} \) should be filtered with a discretized inverted notch filter on the 2P frequency to scale the amplitude and to obtain the GS-IFT-IPC pitch signal.

8-3 Investigating the MBC-IPC pitch signal

As shown in Sections 8-1 and 8-2, performance improvements in terms of fatigue load reductions, by changing the amplitude of the IPC signal are attained. However, the performance of the feedforward IPC method does not surpass the load reductions attained with MBC-IPC. Clearly, the timing of the IPC pitch signal is important for improvement of load reduction capabilities. As shown in Figure 8-1, the MBC-IPC pitch signal shows phase lead and lag, which is not driven by variations in rotor speed, as the IFT-IPC pitch signal (plotted in the same figure) is scheduled on the same azimuth angle. At first, it was assumed that these phase effects were caused by the feedback loop. The hypothesis made in this section is that phase advances and delays in the MBC pitch signal are caused by the varying periodic 1P load component, present on the blades subjected to turbulence.

To make a reasonable case for the above described hypothesis, an Finite Impulse Response (FIR) filtered blade load signal which is not subjected to any form of IPC is plotted together with an MBC-IPC pitch signal in Figure 8-11. Both results are obtained under the same turbulent wind conditions. The FIR filter used for this case is a bandpass filter with a large range around the 1P frequency, but not including the 2P harmonic. To make a clear
8-3 Investigating the MBC-IPC pitch signal

Figure 8-10: Simulation results in terms of DEL comparing load performance results on different wind turbine components while implementing different slope values $\alpha_i$. As can be observed, the effectiveness on blade loads is small and only present at the lower wind speed. Tower loads are improved by adding an amplitude scheduling to the feedforward pitch signal, and (GS-)IFT-IPC even outperforms MBC-IPC in the tower $M_x$ and $M_y$ cases. The DEL on the rotating hub improve with increasing $\alpha_i$, but MBC-IPC still outperforms GS-IFT-IPC.

comparison possible, both signals are normalized with respect to the maximum value in the time-series. It is very clear that the blade load signal shows phase leads and lags in time, which the pitch signal generated by MBC-IPC tries to follow. Thus, to account for changing blade load periodics caused by turbulence, control on blade load harmonics is required. An interesting option is to construct a blade load harmonic predictor, based on wind speed measurements ahead of the turbine, by also taking into account effects such as wind shear and tower shadow. With these predictions, the pitch signal can be sent to the pitch actuators at exactly the right timing and may improve performance compared to conventional feedback IPC.
Figure 8-11: Comparison between an FIR filtered blade load signal around the 1P frequency without IPC, and its corresponding MBC-IPC pitch signal at integrator gain $K_{I,0}$. All signals are normalized to 1 with respect to the maximum value in the time-series of each signal. It is clear that the MBC-IPC signal tries to follow the 1P periodic blade load frequency, which explains its phase delays and advances in time.

From Figures 8-1 and 8-11 it is concluded that IPC signal generated by conventional feedback IPC, change in amplitude and phase according to the actual present harmonic $nP$ blade load signal. Disturbances in the azimuth input of the MBC transformation are to a certain extend counteracted by the changing the coefficients in the weighted linear combination of sine and cosine in the reverse transformation. This statement is supported by the results discussed in Section 3-2.
Chapter 9

Conclusions and recommendations

9-1 Conclusion and discussion

In this thesis, Individual Pitch Control (IPC) is employed in a conventional feedback and in a self-learning feedforward set-up to investigate various objectives. In the first part, the effect of including a phase-offset to the azimuth angle in the reverse Multi-Blade Coordinate (MBC) transformation is compared to varying integrator gains. Constant wind simulations show that in all cases the pitch signals converge to the same steady-state signal in terms of amplitude and phase. However, the transient towards the final pitch signal is changed in time (phase) for different phase-offsets in the reverse MBC transformation, while only the convergence rate of the pitch signal changes by incorporating various integrator gains $K_I$. The hypothesis that the same pitch signal can be attained by incorporating different integrator gains, as obtained by including phase-offsets in the reverse MBC transformation, does not hold. It is shown that an optimal phase-offset exists for tower load reductions, while the Damage Equivalent Load (DEL) of blade loads are not much affected by changes in phase-offset.

Secondly, the property of the MBC transformation of decoupling the blade load measurements in a non-rotating tilt and yaw component is exploited in different configurations, with the aim to make yaw-by-IPC possible and to investigate performance and stability. The first configuration involves the addition of the PI-controlled yaw-misalignment error to the yaw-moment in the non-rotating frame of reference, while the second configuration sums the PI-controlled yaw-misalignment error directly to the non-rotating pitch signal. As is concluded from simulations, both configurations show approximately the same level of performance in terms of wind turbine load reductions using the DEL, but interesting effects can be observed concerning stability. The blade load measurement signals to the IPC controller might become unavailable, which causes a closed-loop configuration with two integrators in series for the first configuration. As a result, the yaw-misalignment control signal gets a 180 degrees phase-shift, which causes instability in closed-loop. The problem can however be solved by the second configuration, where yaw-by-IPC and harmonic load reduction signals are frequency separated: low frequency dynamics account to yaw-control, and higher frequencies for load...
Conclusions and recommendations

reduction control, where a trade-off has to be made between yaw-alignment and load reduction performance.

Next, a feedforward implementation of IPC, scheduled on the rotor azimuth position (cyclic-pitch) is employed and optimized at various wind speeds for 1P/2P harmonic load reductions using the Iterative Feedback Tuning (IFT) algorithm. The current implementation of the cyclic-pitch controller, makes the pitch signal to have a constant amplitude of the periodic pitch signal. Optimizations can be performed on both blades individually, as the Relative Gain Array (RGA) of the linear identified model shows that no interactions between both blades are present in terms of out-of-plane moments. The main reason for optimization of this controller type is to develop a self-learning feedforward IPC strategy for performance comparison to conventional feedback IPC. The first conclusion that can drawn from the comparison between the self-learning feedforward and conventional feedback pitch signals, is that in the ideal case with constant wind, the pitch signals and load reduction performance match at all wind speeds. This confirms the ability of both control strategies to mitigate harmonic blade load under ideal conditions. In a more realistic situation where turbulence is present, the IFT algorithm implementation succeeds in the above-rated operating region to optimize to approximately the same pitch signal as in the non-turbulent case. However, due to highly fluctuating rotor speeds in the below-rated region, the IFT algorithm is unable to converge to the optimal pitch angle during turbulence. Compared to the MBC set-up, the cyclic-pitch controller is able to reduce wind turbine loads considerably, but is not able to meet the performance of the MBC control configuration. An advantage of the constant amplitude pitch signal is the reduced wear and stresses on the pitch actuator, which can extend its life span.

To improve blade load reduction capabilities of the feedforward cyclic-pitch controller, the MBC pitch signal is compared to the cyclic-pitch signal, and it appears that the latter is a mean of the MBC signal in terms of amplitude and phase. As the IFT optimized cyclic-pitch controller is only frequency dependent on the current rotor azimuth angle, the amplitude of the pitch signal is adjusted by a gain-schedule on filtered blade load signals at the 1P frequency. Two types of implementations are considered: one where the constant amplitude of the IFT-IPC signal is scheduled on the pitch amplitude of an already existing MBC-IPC series, and second where filtered 1P blade load intensities are mapped to pitch signal amplitude adjustments. The reasoning behind these experiments is that the changing phase of the MBC-IPC pitch signal might be caused by integrator transients and filtering actions in the feedback loop, and these can be omitted by gain-scheduling the amplitude of the IFT-IPC signal. DEL comparisons however show that amplitude scaling can improve performance, but generally not to the level of MBC control.

In the last section, a comparison is made between a (broad) bandpass filtered 1P blade load signal without IPC, and a pitch signal generated by feedback IPC under the same simulation conditions. The combined effect of tower shadow, wind shear and turbulence results in unexpected harmonic blade load behavior, as the load is expanded or compressed in a certain time frame. Comparing the timing and amplitude of the filtered load and pitch signal reveals that the MBC transformation tracks the harmonic load in terms of phase and amplitude, using both the rotor azimuth and blade load measurements. To account for changing blade load periodics, control on blade load measurements or predictions is required.
9-2 Recommendations

Based on the results and conclusions made in this thesis, recommendations can be made for further research. The IFT optimization algorithm is implemented to reduce harmonic blade loads of both blades as much as possible, by optimizing a feedforward cyclic-pitch controller scheduled on the rotor azimuth angle. Instead of optimization of the cyclic-pitch controller, the IFT algorithm can be applied to the integrator gains in conventional IPC or another fixed-structure parameterizable closed-loop controller for fatigue (blade) load reductions.

Application of IFT to other control loops present in a wind turbine, would also be an interesting opportunity to check the validity of the current setting. Because the IFT is a gradient based optimization method, it would be interesting to see whether optimization constraints can be incorporated, since little to no literature is available on this topic.

As found in the course of this thesis, improving the performance of the conventional MBC implementation for IPC turns out to be a serious challenge. Due to the ever increasing rotor diameter, performance in terms of fatigue load reductions are limited by the span-wide control authority of IPC and spatially varying blade loads. MBC-IPC is able to directly act on and mitigate the $n P$ load harmonic it is configured to: phase and amplitude changes in the load signal are directly taken into account by the feedback set-up. A blade load harmonic predictor based on wind speed measurements ahead of the turbine, by also taking into account effects such as wind shear and tower shadow, has potential to further reduce blade loads. With predictions on blade load harmonics, the pitch signal can be implemented to the pitch actuators at exactly the right moment and may improve performance in the current set-up. Moreover, new control concepts, such as smart rotors where the blades are equipped with a number of devices that locally change the lift profile on the blade [44], might take the load alleviation capabilities to a higher level than possible with conventional IPC. IFT can play an important role in tuning such kind of multivariable controllers.
A-1 Appendix A - filtfiltC Simulink implementation

```matlab
function y = filtfiltC(x,bk)

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
ORDER = length(bk);
N = length(x);

% % FILTER FUNCTION - First filter action
% Filling y to i == ORDER of the filter - forward
y(1) = bk(1)*x(1);
for i = 2:ORDER
    y(i) = 0;
    for j = 1:i
        y(i) = y(i) + bk(j)*x(i-j+1);
    end
end

% Filling the rest of y - forward
for i = ORDER+1:N
    y(i) = 0;
    for j = 1:ORDER
        y(i) = y(i) + bk(j)*x(i-j+1);
    end
end

% reverse the series for second filter run (filtfilt)
for i = 1:N
    x(i) = y(N-i+1);
end

% % % FILTER FUNCTION - second filter action
```

Master of Science Thesis  
S.P. Mulders
% Filling y to i == ORDER of the filter - reverse
y(1) = bk(1) * x(1);
for i = 2:ORDER
    y(i) = 0;
    for j = 1:i
        y(i) = y(i) + bk(j) * x(i-j+1);
    end
end

% Filling the rest of y - reverse
for i = ORDER+1:N
    y(i) = 0;
    for j = 1:ORDER
        y(i) = y(i) + bk(j) * x(i-j+1);
    end
end

% reverse the series
for i = 1:N
    x(i) = y(N-i+1);
end
for i = 1:N
    y(i) = x(i);
end

**A-2 Appendix B - IIR and FIR filters**

Two main types of filters are available for implementation, namely Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. The difference equations that describe the outputs of both filters are given by Equations (A-1) and (A-2) for an FIR and IIR filter respectively. Figures A-1 and A-2 present the block diagrams of both types of filters.

\[ y_{FIR}(n) = \sum_{k=0}^{M} b_k x_{FIR}(n-k); \]  
\[ y_{IIR}(n) = \sum_{k=0}^{M} b_k x_{IIR}(n-k) - \sum_{k=1}^{N} a_k y_{IIR}(n-k), \]  

It can be seen from the equations and figures that the output of the FIR-filter is simply a weighted sum of the past input values, while the IIR-filter also takes into account a weighted combination of the past output [47]. So, the IIR filter has recursion and this is where the infinite nature of the impulse response originates from. Writing Equations (A-1) and (A-2) as transfer functions yields the following results:
It appears from the FIR transfer function that only zeros are present, while in the IIR case both zeros and poles can be used to obtain the desired frequency response. For this reason, the coefficients $b_k$ are determined by optimization routines to meet a desired frequency response $H_d(e^{j\omega})$. It is challenging to optimize IIR filters with respect to both coefficients in the numerator and denominator, and for this reason analog filters or standard forms of continuous time transfer functions are used to develop the discrete time equivalent. Because FIR filter design is optimization based, an arbitrary frequency and/or phase response can be obtained, while in the IIR case the gain and phase is dependent on frequency. As a result, FIR filters have a linear phase shift among all frequencies, and by passing the signal in forward and reverse order through the FIR filter, a filtered signal without any phase distortion can be obtained (non-causal zero-phase filtering). This is a huge advantage of FIR filters in comparison with IIR filters, which always have a non-linear phase shift among frequencies. As a consequence of the all-zero nature of the FIR filter, the order of the filter can be way larger than its IIR equivalent.


Master of Science Thesis

S.P. Mulders


Master of Science Thesis

S.P. Mulders


Glossary

List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPC</td>
<td>Collective Pitch Control</td>
</tr>
<tr>
<td>DCSC</td>
<td>Delft Center for Systems and Control</td>
</tr>
<tr>
<td>DLL</td>
<td>Dynamic-link library</td>
</tr>
<tr>
<td>DEL</td>
<td>Damage Equivalent Load</td>
</tr>
<tr>
<td>EWEA</td>
<td>Europe’s Wind Energy Event</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response</td>
</tr>
<tr>
<td>FTC</td>
<td>Fault-Tolerant Control</td>
</tr>
<tr>
<td>IFT</td>
<td>Iterative Feedback Tuning</td>
</tr>
<tr>
<td>IIR</td>
<td>Infinite Impulse Response</td>
</tr>
<tr>
<td>IPC</td>
<td>Individual Pitch Control</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear Time-Invariant</td>
</tr>
<tr>
<td>LTV</td>
<td>Linear Time-Variant</td>
</tr>
<tr>
<td>MBC</td>
<td>Multi-Blade Coordinate</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple-Input Multiple-Output</td>
</tr>
<tr>
<td>PBSID\textsubscript{opt}</td>
<td>Predictor-Based Subspace IDentification</td>
</tr>
<tr>
<td>RBS</td>
<td>Random Binary Signal</td>
</tr>
<tr>
<td>RC</td>
<td>Repetitive Control</td>
</tr>
<tr>
<td>RGA</td>
<td>Relative Gain Array</td>
</tr>
<tr>
<td>SISO</td>
<td>Single-Input Single-Output</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SPRC</td>
<td>Subspace Predictive Repetitive Control</td>
</tr>
<tr>
<td>VAF</td>
<td>Variance-Accounted-For</td>
</tr>
<tr>
<td>VS-VP</td>
<td>Variable-Speed Variable-Pitch</td>
</tr>
</tbody>
</table>