Optimal Feedforward Control for Offshore Wind Turbines During Grid Faults

D. Spijkerman

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Optimal Feedforward Control for Offshore Wind Turbines During Grid Faults

Master of Science Thesis

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

D. Spijkerman

July 18, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Siemens Gamesa Renewable Energy. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Optimal Feedforward Control for Offshore Wind Turbines During Grid Faults

by

D. Spijkerman

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: July 18, 2023

Supervisor(s):

dr.ir. S.P. Mulders, Supervisor

dr.ir. S.T. Navalkar, Second Supervisor

B. Solberg, PhD, Third Supervisor

Reader(s):

dr.ir. M. Khosravi, First Reader

Abstract

The share of electricity generated by (offshore) wind turbines has increased considerably in the previous decade(s). Due to this increase in market share, more stringent power requirements have been established to ensure a stable electrical grid. An important requirement is the low-voltage ride-through requirement that states that the wind turbine must remain connected to the electrical grid after a short intermittent grid fault. This requirement calls for new (optimal) grid fault controllers that minimise the impact of the requirement on the turbine structure. During this thesis, an optimal grid fault controller has been synthesised.

Grid fault controllers found in industry and academia are mostly aimed at reducing the imbalance between the rotor torque and electrical torque output and do not explicitly consider other objectives such as mechanical loading. Additionally, current state of the art controllers often require large amounts of system knowledge or are only tested on low-fidelity models. To address these issues, a new type of grid fault controller has been presented and tested, an Iterative Learning Control (ILC) controller. One of the major contributions of this work lies in applying an ILC controller for grid fault control on wind turbines.

The ILC controller is a feedforward controller that employs model-free learning based on iterations, where here PID-type ILC and norm-optimal ILC variants have been used. PID-type ILC applies learning by combining the previous input with a weighted proportional, integral, and derivative terms of the output error to calculate the input for the next iteration. The norm-optimal ILC looks for the feedforward input that minimises a weighted quadratic cost function of the error and input. It is shown that both of these controllers can reduce the output error to (near) zero if no input constraints are considered. For a more realistic scenario with input constraints, both ILC controllers manage a reduction in output error compared to the baseline feedback controller. Here specifically the optimisation based norm-optimal ILC controller achieves the highest performance which can largely be credited to the controllers ability to fully saturate the actuator for longer with accurate placement of the switching times between full positive and negative actuator saturation. The optimisation based norm-optimal ILC controller is an extension over the state of the art and as such the second major contribution of this work.

Both ILC controllers have been tested on a high-fidelity wind turbine model. These tests show that the optimisation based constrained norm-optimal ILC with the objective to reduce the rotor speed error works best, yielding a 30% improvement in the 2-norm of the output error compared to the baseline controller. For the objective of reducing mechanical loading,

the constrained PID-type ILC performs best with a 70% improvement in the 2-norm of the output error compared to the baseline controller. Recommendations for future work include investigating why dependent on the objective function a different ILC controller yields the best performance, and investigating the effects of stochastic grid faults on the ILC controllers performance.

Acknowledgements

The here presented works represents the culmination of my academic career at Delft University of Technology and will mark the beginning of my career as a control engineer. This work would not have been possible without the many support of my supervisors and colleagues at Siemens Gamesa Renewable Energy (SGRE), my supervisor at the university, friends, and family.

Before my internship at SGRE my knowledge about the vast field of wind turbines was slim, but has grown ever since. I will even go so far and say that fascination for the technical achievements of the whole (offshore) wind turbine industry has been created. Never had I known how much goes into building ever greater wind turbines that will lead us to a greener and cleaner future for the next decades.

Special thanks will have to be made to Brian Solberg and Sachin Tejwant Navalkar, my supervisors from SGRE. The meetings we had have brought me many insights and revelations from both your sharp comments and your vast knowledge of the field. Our meetings have often pointed me towards new exciting parts to be explored or expanded. Often times I was amazed by how well you understood the new material and were able to provide sharp feedback beyond the obvious points.

Last but not least, my thanks to my supervisor Sebastiaan Mulders from the TU Delft. He has guided me not only in finding direction for this thesis, but also in the reporting of it. Many times we have discussed the structure and substance of the report after which many things were changed or scrapped altogether. Sometimes I was presented with true head-scratchers on how to shorten ten pages of information into two, but in the end it was always for the better. In addition to reporting, Sebastiaan has also given me great leads and suggestions in ways to further conduct my thesis work from start to end.

Delft, University of Technology — D. Spijkerman
July 18, 2023

Table of Contents

Chapter 1

Introduction to wind turbines

In recent years, the urgency to pursue more renewable energy sources has become increasingly clear. There is not only a climate crisis, but also a growing realisation that the European Union (EU) needs more independence from the global energy market. The energy market has been heavily affected by the ongoing war in Ukraine and the aftermath of the global pandemic. As a result, more energy is imported from outside the EU, often in the form of liquefied natural gas [1, 2]. Importing energy is not without consequences, as not all of the major exporters conform to the same values as the EU. This, among other things, leads to the EU goal of 45% renewable energy use in 2030 compared to 22% in 2020 [3, 4]. The achievement of this goal can be done in part by increasing the use of (offshore) wind turbine (WT). Growth in WTs is not only limited to the EU, as other markets have also increased wind energy production in recent years [5] and are committed to continue to do so in the following years [6, 7].

The main goal of this thesis is to further increase the knowledge base on the control of (offshore) wind turbines during grid faults. This chapter starts with a short summary of nominal WT control, followed by an introduction to the thesis topic and research question.

1-1 Actuators available for control

Wind turbines are controlled using the following actuators, the (individual) blade pitch, the generator converter, and a yaw motor [8]. The yaw motor is typically only used to align the nacelle with the predominant wind direction due to its limited bandwidth of less than 1°/s [9, 8].

Blade pitch control is used to alter the lift and drag forces of the blades, resulting in a change in aerodynamic efficiency/torque. Depending on the angle of the blade pitch β and the tip speed ratio $\lambda = \omega_{\mathrm{r}} R/v$, there is an optimal value λ^* that maximises the aerodynamic efficiency $C_{\mathrm{p}} = P/P_{\mathrm{wind}}$ for a given wind speed, and thus yielding the highest power output P for that wind speed, with ω_{r} the rotor angular velocity, R the rotor radius, v the wind speed, P the turbine power, and P_{wind} the power available in the wind [8, 10]. The blade pith actuator

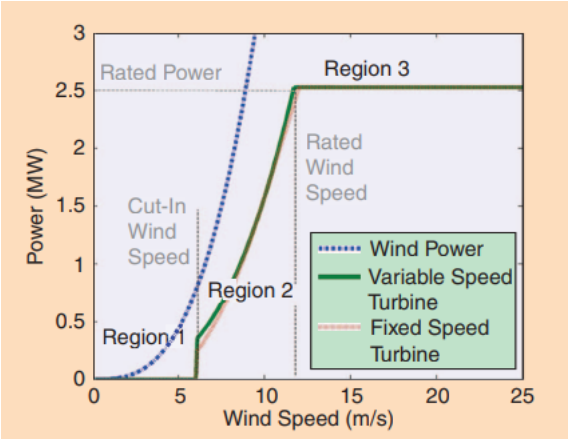


Figure 1-1: Depiction of available wind power and the WT's nominal output power as a function of wind speed, separated in four distinct regions. In region 1 the available power is insufficient to start the WT. In region 2, below-rated, the power output is increased with the wind power, up to the rated power. In region 3, above-rated, the WT's power output is at rated power. Region 4 is not shown, but starts right of region 3, at the cut-out wind speed. In region 4 there is zero power output to prevent damage to the WT in the high wind speeds [8].

has a speed of around 8°/s for 5-MW WTs [8], but as turbines get larger the speed usually decreases.

The generator is controlled through a power electronics converter [11] which can control the generator torque (power output) and the output frequency and phase, all with a bandwidth of 0.5-1 ms [9, 11]. The power converter decouples the generator speed from the frequency of the electrical grid by converting the generators output from Alternating current (AC) to Direct current (DC) and back to AC [12]. The input and output of the power converter have separate controllers [13], where here the focus is on the input side that controls the generator torque. Setting the generator torque higher or lower than the aerodynamic torque results in a respective deceleration or acceleration of the rotor [9].

1-2 Nominal control of wind turbines

Under nominal, non faulty, operating conditions, WT control is divided into four distinct operating regions [9, 8, 14], as shown in Figure 1-1. In region one the mean wind velocity is below a cut-in velocity, and the WT is kept idling until higher wind speeds arise. In region two, below-rated, the mean wind velocity is between the cut-in and rated velocity, where the power production increases in proportion with the wind speed. In region 3, above-rated, the mean wind velocity is between the rated and cut-out velocity, where a constant power output is maintained. In region 4 the mean wind velocity is above the cut-out velocity, and the WT is brought to a standstill for safety reasons.

Control in region 2 usually employs a fixed blade pitch such that the most energy is extracted from the available wind by regulating the rotor speed. The rotor speed is regulated using the generator torque, to operate near λ^* for maximum aerodynamic efficiency. The regulation in classic control is done using a PI or PID controller with as input the generator speed error

and as output the desired generator torque. In [14] speed exclusion zones are added in which they will increase the tip speed ratio above the optimal λ^* at specific wind speeds to increase tower damping and decreased fatigue loading. An alternative approach can be seen in the work of [15] that claims to give a more insightful trade-off between power maximisation and fatigue load minimisation by employing a model predictive control based scheme.

In region 3, the control objective is to keep the power output and the rotor speed constant by regulating the aerodynamic toque. In classical designs, this is done using a PI or PID controller. The input of this controller is the error of the desired rotor speed, with the output being the desired pitch angle which can alter the aerodynamic torque.

1-3 Transmission System Operator requirements

The control schemes as presented above are primarily focused on maximising energy production. However, in reality also objectives such as improving safety, decreasing mechanical loads or increasing fatigue life are considered [9]. Control objectives can also originate from external parties such as the Transmission System Operator (TSO). The TSO is the link between the WT (park) and the high-voltage electrical grid, where most nations have a single TSO in charge of maintaining said electrical grid [16]. In this section there is an elaboration on one of the TSO requirements that is motivating the research in this thesis.

To ensure stable operation of the network i.e., the electrical grid, the TSOs will set requirements on the operations of WT (parks). For the EU these requirements are summarised in [17], wherein the local TSOs can still enforce some input within the limits of the document. One challenging type of requirement for WT operators is low-voltage ride-through (LVRT) requirements, also known as fault ride-through in the literature. The LVRT requirements define how the WT must operate during and after intermittent drop(s) of the grid voltage. Such an event is denoted as an LVRT event or a grid fault during this report.

During an LVRT event the maximum generator torque output is reduced proportional to the reduction in grid voltage. If the output torque, and in turn power, were not to be limited, the output current could exceed current ratings of components such as the generator and transmission cables. Due to the sudden decrease in generator torque during LVRT events, the generator and rotor torque are not in balance causing the rotor to speed up. Without intervention this could lead to the rotor speed exceeding an overspeed limit, triggering a disconnect from the grid. If this were to happen for many turbines at the same time, this could lead to a cascade in power generating components disconnecting, leading to a network wide blackout [18]. To prevent such an event from happening, the LVRT requirements are in place.

The LVRT requirements specify, for example, what are tolerable power outputs after an LVRT event, and under which conditions a WT must stay connected to the grid [19, 20]. To comply with these LVRT requirements, special LVRT control schemes are in place, as is discussed in more detail further in the report. A summary of the most important LVRT requirements as specified by the TSOs can be seen below, with P_{max} the maximum rated power output and Q the reactive power output [17, 21].

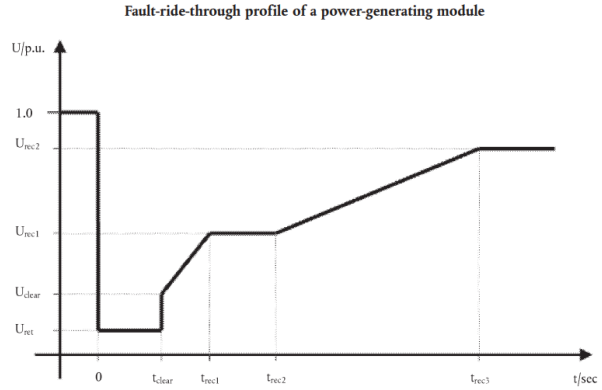


Figure 1-2: A lower limit for the grid voltage over time during which the WT's grid connection is required. If the grid voltage is below the specified level, the WT may disconnect from the grid, otherwise it has to stay connected to prevent cascading failures on the grid. The voltage levels and time points are TSO/country specific. Image by [17].

Grid requirements related to LVRT:

1. Voltage-against-time diagram depicting a lower limit for actual network voltages during which the connection must be maintained; see Figure 1-2 for an example by [17].
2. The system operator has the right to specify the reactive power capability for below maximum power capacity (P-$Q/P_{\max}$ -profile).
3. After a step change in voltage the (reactive) power has to settle within specified times.
4. It must be possible to ride through an arbitrary sequence of grid faults (Germany).

1-4 State of the art in grid fault control

With an increasing number of WTs connected to the grid, TSOs have had to expand the LVRT requirements to ensure stable grid operation during grid events [22], with the relevant grid LVRT requirements discussed in the previous section. The main causes of these grid events are investigated in [23]. To facilitate grid requirements, new LVRT control strategies have been developed. In this section the state of art in grid fault control, focused on LVRT, is discussed. For this, first patents are investigated, as most of the industry is not keen on publishing much information, followed by a study of academic papers.

1-4-1 State of art in patents

In this section, a number of patents related to control of the WT during LVRT events is discussed. The focus is on novelty and implications rather than technical implementation.

In the patents of [24, 25] a controller is described that calculates a corrective term $f^*(x)$ that is added to the nominal blade pitch controller $f(x)$ during a LVRT event, with x the WT's state. This results in the following blade pitch set point during LVRT events $\beta_{\text{set}} = f(x)+f^*(x)$. In [24] the corrective term is added to $d\beta/dt$, and in [25] to β itself. The function in the corrective terms $f^*(x)$ is a P, PI, or PID controller with as input the difference in the expected and real power output in [24] and the generator torque in [25].

An important novelty of these patents lies in the simple integration with existing blade pitch controllers by introducing the correction term $f^*(x)$. This correction term is only added during a grid fault, keeping the nominal controller intact. Other faults, such as a defect in the main converter, are also claimed to be reacted to with this method in [25]. Missing from the patent is if additional objectives such as reducing mechanical loads is also addressed by the scheme, or whether the controller is only tuned to minimize the power error.

In the patent of [20] a method is described to disconnect noncritical components during LVRT, and to power selected critical components using an Uninterruptible power supply (UPS). The UPS powered components, turbine controller, pitch motor, and rotor system, can monitor and prevent overspeed conditions. After switching power, the patent describes two steps of overspeed mitigation: (1) pitch the blades to feather to limit aerodynamic torque and, (2) use a current-limiting circuit connected to the rotor side of the inverter to dissipate part of the excess power if the current rises above a threshold.

The LVRT scheme presented in this patent is simple in implementation, as it contains only two steps and a single additional controller for the current-limiting circuit. The additional current-limiting circuit has the advantage of being faster than the blade pitch, allowing better control during an abrupt LVRT event [26]. This comes with the extra cost of more hardware, which will have to increase as the WTs get bigger. The need for additional hardware seems to be the biggest disadvantage, even though it does add an additional control input to the system giving more freedom in designing a controller.

In the patents of [27, 28] a zero-voltage ride-through (ZVRT) scheme is described, which is effectively an extension of the LVRT to facilitate a drop to zero volt. The patent describes a method in which, based on measurement of the voltages of the three individual phases, a phase error is calculated. The phase error is fed into an PI-controller with anti-windup scheme. The output of the PI-controller is integrated and saturated before being sent to the converter controller which sets the output current, aiming to set the highest back EMF to keep the rotor speed in an acceptable range. This scheme is in the art also known as a Phase-Locked Loop (PLL) regulator [29].

The novelty of the patent lies in the inclusion of a state machine that sets the PI-controller gains and saturation limits. The state machine switches depending on parameters such as grid voltage level and frequency, and can switch between either one of the four following states: (0) startup, (1) phase synchronisation, (2) nominal conditions, (3) ZVRT conditions. In the patent, it is claimed that using this scheme ZVRT is possible. However, in the author's opinion, using only this scheme will most likely be insufficient to act on ZVRT events, and requires additional control for the blade pitch to prevent overspeeding.

In the patent of [19] a method of reducing mechanical loading during a LVRT is described. In the patent the blades are pitched to feather once a grid fault occurs, where after recovery of the grid voltage a pitch reference signal is calculated using the nominal control scheme. The converter output power reference signal is also calculated using the nominal control scheme, but gets limited to allow some headroom such that an additional a drive train damper signal can be added to the power reference. This drive train damper signal has the objective to minimises the transient torque overshoot by enhancing the damping of the oscillations.

The patent claims that the inclusion of the drive train damper signal reduces the power overshoot with more than 50%. The calculation of the drive train damper signal is however

not described in the patent. There are however papers such as [30] describing drive train damping in non faulty conditions indicating that this method can indeed yield positive results.

1-4-2 State of art in papers

In this section, a quick overview of papers describing solutions for LVRT events is presented. An interesting, but less relevant, field has also been found during this survey in which additional hardware is used to improve LVRT, with the following being some interesting surveys [31, 32, 33]. The additional hardware used in these papers include a crowbar, a method to dissipate current to generate a back EMF on the rotor, and an energy storage system to store some of the excess DC power in the generator converter. Though these are interesting techniques, the focus during this paper is to make control schemes in which no additional hardware is required.

The remainder of this section is based on the literature survey of [34]. In this survey, multiple hardware free solutions for LVRT are referenced, a few of which are highlighted in the following paragraphs. More information on LVRT controllers can be seen in the work of [34]. The first reference to be made is to the work of [35] as this is the only paper that includes a feedforward control strategy. However, it is designed for a hybrid system with a fuel cell and battery storage, decreasing its relevance. The feedforward control is used to calculate a reference current output of the grid side converter for the individual electrical phases. Since there are energy storage elements in this system, controllers can direct the power that would normally go to the grid to these storage systems during LVRT events. However, this is not possible in the scenario considered in this literature survey. Additionally, the work only considers maximising power output and does not have any objective of reducing structural loading on the WT.

In the work of [36] a grid fault controller is presented that acts on the converters output current using multiple PI-controllers. The PI-controller parameters are tuned using particle swarm optimisation, with the goal of maximising power output, based on the work of [37]. The particle swarm optimization tries to minimise the following cost function $J = 1/|\max_{i=1..13}\{Re(\lambda_i)\}| + f_{\text{p}}$, where λ_i are the eigenvalues of matrix A of the linearised system and f_p a barrier function to keep the eigenvalues strictly negative. This cost function effectively tries to find the controller parameters that result in the fastest stable system dynamics. A simulation shows that the controller can successfully perform LVRT for a 50% drop in the grid voltage. The paper further presents a computationally light model useful for faster learning of data driven controllers. Since the presented model only outputs the converter's grid current, no analysis on turbine loading is included in this work. Additionally, the cost function used in this paper puts emphasis on controller speed, which could come at the expense of large overshoots of output current and or other variables such as tower loading.

In the work of [38] a Model Predictive Control (MPC) based controller is presented to be used on the generator side of the converter. The grid side of the converter is controlled using vector control based on the grid voltage. The blade pitch, controlled by a PI-controller, is reduced once a voltage drop is detected to reduce the rotor speed. The interesting part of the paper is the MPC-controller on the generator side of the converter. This MPC based controller uses a quadratic cost function on the difference between a reference output voltage and predicted next output voltage, aimed to provide fast current tracking. Using the cost

function, an optimal output current is calculated and used to control the generator side of the converter. Simulations show that successful LVRT to approximately 0.2 V/pu is possible, and only small fluctuations in the grid current and converter DC-voltage are present. The controller presented in this paper does not explicitly include tower dynamics, but does try to reduce the deviation of the output current. This in term could lead to reduced tower loading, as no large power oscillations are visible in the presented simulations.

In the work of [39] a sliding mode controller is implemented with the sliding surface being the error of real and reactive power. The controller will try to steer the systems state to a manifold with zero power error, so with perfect tracking of the reference of the real and reactive powers. In the paper, tests were performed using a small test setup, which showed that LVRT with a grid voltage drop of up to 0.35 pu is possible. However, the results showed quite high amounts of oscillations or chattering in the reactive power output. Chattering is then also a known disadvantage of the sliding mode controller [40]. The claimed advantages of this type of controller are the high robustness for disturbances and the relative simplicity of the controller. However, high amounts of chatter could result in increased mechanical loading of the WT, making this approach less attractive compared to previously mentioned controllers.

1-4-3 Conclusion of state of the art survey

In the previous two subsections the state of the art in LVRT control for WTs has been investigated both for the industry by patents and for academia by papers. This survey has shown that the industry uses mainly PI(D)-based controllers on both the blade pitch to reduce the incoming generator torque and on the generator converter to control the current of the system. These controllers usually include some switching behaviour that reconfigures the PI(D)-controller gains based on the state of the LVRT event. However, most patents lack actual or simulated results of the proposed controllers, making judgement of their effectiveness difficult. In addition, no details of how the controller gains are tuned are present in the patents, making it difficult to see if any mechanical load reduction objectives are considered or whether only rotor overspeed prevention and output power tracking requirements as defined by the TSOs were considered.

The state of the art in papers has revealed more diverse control solutions for LVRT events compared to the patent survey. Work found includes a switching mode controller and a controller with multiple PI-controllers tuned by an optimisation algorithm. One of the more promising controllers found is an MPC based controller that is used to control the current of the generator side converter. This MPC controller provided an excellent reduction of power fluctuations compared to other works. All of these works, however, only have objectives to track a reference power output by some means, and did not consider turbine load reduction objectives. The works found did not even model turbine loading. Although some of the works managed to perform LVRT with a small amount of fluctuation in power, it was not clear what the effects of the mechanical load of the WT are.

1-5 Research question

The state of the art survey from the previous section has shown that there is room for further research in the field of optimal LVRT controllers. The survey has shown that there are numerous works on optimal controllers for LVRT events, but the objective functions of these controllers only aim at reducing the power tracking error. Work to reduce the mechanical loads of the WT during LVRT events has not been found. Furthermore, the works found do not include the dynamics of the mechanical loads in their models, making these models unsuitable to properly evaluate load reduction strategies during LVRT events. This has lead to the formulation of the following research question and sub-questions.

How can data-driven algorithm(s) be used to learn the optimal (feedforward) control signal to resume regular operation after a predefined low-voltage ride-through scenario?

Which data-driven optimal algorithm(s) can be used for low-voltage ride-through? Following a review of the state of the art, multiple potential control algorithms have been identified. For this question it will have to be identified which algorithm is most suitable for the application of LVRT in wind turbines.

Which costs or signals should be taken into account in the design of an cost or objective function for the optimal controller? In the state of the art often only rotor speed objectives are considered. Here it remains however unclear what the effects are on different objectives, and if perhaps completely different (multi-objective) cost functions will have to be designed.

What type of unconstrained algorithm or configuration achieves the best performance? What is the best performing configuration of the found controllers of sub-question one e.g. how should constraints be handled or which algorithm is easiest to tune.

What is the performance impact of including input constraints to the optimisation problem for adhering to actuation limits? Do the control algorithms need or benefit from inclusion of constraints, or is this not needed?

During this thesis the WTs considered are (direct drive) pitch-regulated multi-MW turbines which are variable-speed with a Permanent Magnet Synchronous Generator (PMSG). This turbines type is currently used by the industry due to the superior performance over older WT types [9, 8]. Additionally, these WTs offer superior control in high winds [9] enabling compliance with stringent standards of power quality [41]. It is assumed that the WTs have no supplemental actuators such as an active crowbar or energy storage system as in [26]. The focus of this report is on individual turbine control, and not on wind farm control.

1-6 Organisation

The structure of this thesis is as following. In Chapter 2 the control methodologies used throughout this study are presented. In Chapter 3 the grid fault scenario, control objectives and models that are used in this study are presented. In Chapter 4 results of the aforementioned methodologies and grid fault are presented. Lastly in Chapter 5 the conclusions of the thesis work and is made, including a reflection back on the research questions.

Chapter 2

Methodology

In the introductory chapter the state of the art in LVRT control for WTs has been investigated. This survey concluded that the optimal controllers found focus exclusively on tracking a reference power output and do not include any objectives for load reduction. The most promising work was based on an MPC controller that also showed a reduction in power output fluctuations. That said, there are still control methods for which no work has been found. One such control method is the Iterative Learning Control (ILC) algorithm. In the ILC algorithm, an optimal feedforward signal is learnt iteratively with the goal of minimising a (multi-)objective function.

A valid question at this point would be why are ILC algorithms attractive for the application of LVRT control and how does it compare to other control methods. One of the main differentiators ILC has from other widely used controllers is that ILC is an algorithm that finds the optimal feedforward signal according to a user-specified objective function, with emphasis here on feedforward. Feedforward controllers do not base their input actions on the currently measured state of the system, but instead compute this based on known system dynamics. Such feedforward controllers are generally known to have a higher bandwidth than feedback controllers as they do not first need to wait on measurements but instead use prior or system knowledge.

An important implicit assumption made in the previous paragraph is that the feedforward controller operates in a deterministic setting with known dynamics. If there would be any deviation from the assumed system dynamics, then the feedforward controller would not be able to take appropriate actions since no data from this region are used or available. For this reason, a feedforward controller can be combined with a less aggressive feedback controller that can take care of any stochasticity or model uncertainty. In such a setting, the feedforward controller can provide the speed, and the feedback controller the robustness [42, 43, 44].

Thus, to use a feedforward ILC algorithm robustly, a combination with a feedback controller might have to be made. This could be done in multiple ways, but as an illustration of how this could be done, one possible control scheme is presented here. Consider a known grid fault scenario in which the ILC algorithm is used to calculate the optimal feedforward signal.

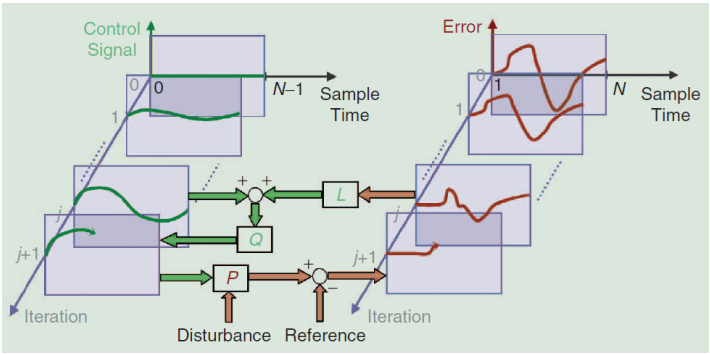


Figure 2-1: An exemplary learning of a first-order ILC system. For each iteration a control signal is applied and an output error w.r.t. a desired output calculated. The error is filtered by L and added to the previous control input, and than multiplied by Q to calculate a new control signal. This signal is than fed into the plant P, and the process gets repeated for the next iteration. Figure by [45].

Now take the difference between the optimally calculated output of the ILC algorithm and the output the original/nominal feedback controller during said grid event. In theory, if this exact grid event would now occur at a later stage, the previously calculated difference in output could be added to the feedback controller as a feedforward term, and again an optimal controller would be recreated. Of course, if the conditions during the actual grid fault do not exactly match the conditions used while synthesising the feedforward signal with the ILC algorithm, the results can be different. Now that the use of ILC is explained, a more in depth evaluation of the ILC algorithm is made in the rest of this chapter.

2-1 Iterative Learning Control introduction

ILC is a technique that is used mainly to control the transient response of a system that operates repetitively. Or, in the words of Ahn et al., "an approach for improving the transient performance of systems that operate repetitively over a fixed time interval" [46]. This technique could be used, for example, to control a manufacturing robot or an autonomous vehicle that performs a repeating task. There are also ILC algorithms made for when the task does not have perfect repetition due to a changing starting position or a nonequal task length [47, 48].

In the ILC framework, the goal is to learn an optimal feedforward signal that can direct the system to a user-defined reference output signal. The creation of this optimal feedforward signal is done on the basis of the knowledge of previous attempts, which are more formally known as iterations. After each iteration j, an error $e_j[n]$ is calculated between the measured output $y_j[n]$ and the desired reference output $y_\mathrm{d}[n]$ for each point in the iteration. This is formulated as $e_j[n] = y_j[n] - y_\mathrm{d}[n] \,\forall\, n \in [0, N]$, with j being the iteration number, n the nth sample in the iteration, and N the number of samples in a single iteration.

Based on the error e_j of the previous iteration, a new control signal is calculated using a learning rule. There are multiple types of learning rules, and even some ILC variants that rewrite the problem to a general optimisation problem, but this is discussed in the following subsections. To illustrate the working of ILC, one can see a commonly used update rule

in Figure 2-1. The learning rule shown in this figure has the form $\mathbf{u}_{j+1} = Q(\mathbf{u}_j + L\,\mathbf{e}_j)$ and can be used for both Multiple Input Multiple Output (MIMO) and Single Input Single Output (SISO) systems, with Q and L being appropriately sized matrices for MIMO or scalars for SISO systems [45]. The Q-matrix is also called the Q-filter, and the L-matrix is called the learning algorithm. What this learning rule does is use a linear combination of the input previously used u_j and the additional input for the next iteration $L\,\mathbf{e}_j$, and filter this with the Q-filter to calculate the input signal for the next iteration. Since the Q and L matrices are synthesised model-free, as is elaborated upon later, this method is a model-free learning algorithm. For many learning rules, there are proofs of asymptotic and or monotonic convergence of the error to zero under some basic conditions [45].

2-2 Iterative Learning Control algorithms

As was mentioned in the previous section, there are different types of ILC algorithms that all use different updating rules, and are designed for different types of systems. In this thesis, the focus is on two types of learning rules, namely a classical PID-type learning rule and a norm-optimal learning rule. The PID-type learning rule is often used in the literature and can almost be viewed as an equivalent to a time-domain PID controller but in two dimensions, namely the iteration- and time-domains. The norm-optimal ILC has a different approach to the PID-type learning rule in that it embodies the use of a (multi-objective) cost function for which, under specific conditions, an analytical solution exists. This approach can roughly be considered equivalent to a data-driven LQR control [49], but then again in a two-dimensional domain.

An overview of different ILC algorithms and a discussion on why the PID-type and norm-optimal ILC have been chosen can be found in Appendix A. In the remainder of this section, the aforementioned ILC algorithms are introduced including how constraints can be handled.

2-2-1 PID-type Iterative Learning Control

The first ILC algorithms that is presented is the PID-type ILC [45, 46, 50]. This learning rule exists for continuous-time and discrete-time systems, but the focus here is on the discrete time case. Consider the following linear time-invariant discrete-time system with state dynamics $x_j[n+1] = Ax_j[n] + Bu_j[n]$ and output dynamics $y_j[n] = Cx_j[n]$ where j is the iteration index and n the discrete-time index.

For this learning rule, it is assumed that; (1) every iteration has a fixed duration time, (2) the initial conditions of each iteration are equal, and (3) the system dynamics is invariant over the iterations [46]. The first assumption is reasonable for grid fault controllers, as the aim is to only help reduce negative transient effects after a grid fault. This means that we only need to assist the nominal controller for a fixed period after the fault, after which the nominal controller can again take over. The second assumption, equal initial conditions, is considered to be true for now since here simulations are used. If however real-world data is used, some further investigation into this assumption will have to be made. The third condition, invariance of the system over iterations, is also assumed to be adhered to, since wind turbines during normal operation are not expected to change dynamics for the same wind speed.

Now that we have made plausible that the assumptions needed for the PID-type ILC hold, the actual learning rule can be introduced, and states as following:

$$u_{j+1}[n] = u_j[n] + k_{\mathrm{p}} e_j[n] + k_{\mathrm{i}} \sum_{i=0}^{i=n} e_j[i] + k_{\mathrm{d}}(e_j[n+1] - e_j[n]) \tag{2-1}$$

with k_{p}, k_{i} and k_{d} the respective proportional, integral and derivative gains. Using this learning rule in a system as previously described, it is possible to guarantee the following convergence $\lim_{j\to\infty} y_j(n) \to y_{\mathrm{d}}(n) \, \forall \, n \in [0, N]$ for specific values of k_{p}, k_{i} and k_{d}. For the interested reader on how this can be proven, see Appendix A. This proof can also, for linear systems, help in tuning the controller gains and has also been used to determine optimal gains that guarantee the fastest convergence as can be seen in [50].

The PID-learning rule is often used on linear systems, but it is also indicated to work for many nonlinear systems [45]. It is, however, not clear which types of nonlinear systems suitable to be used in concurrence with the PID-type learning rule. For certain nonlinear systems, there do, however, exist proofs of convergence, as can be seen in the works of [51, 52, 53].

2-2-2 Norm-optimal Iterative Learning Control

The second ILC algorithm to be presented and used throughout the thesis is the norm-optimal ILC by [54]. The approach of this ILC learning rule is different from that of the PID-type ILC presented above. Where the PID-type ILC is a pure feedforward controller, the norm-optimal ILC uses a combination of a feedback and feedforward control loop, see Figure 2-2. The combination of both of these controller types should increase robustness of the controller, whilst still having the benefits of the increased speed that is usually associated with ILC algorithms.

In contrast to the PID-type controller mentioned above, the norm-optimal ILC does not start with a learning rule but instead with a cost function that gets minimised. Again, a linear time-invariant discrete-time system is assumed, with extension of the algorithm to linear time-varying systems also existing in the works of [55]. The cost function proposed to be minimised is as follows:

$$\mathcal{J}(\mathbf{f}_{j+1}) = \frac{1}{2}||W_{\mathrm{e}}\mathbf{e}_{j+1}||_2^2 + \frac{1}{2}||W_{\mathrm{f}}\mathbf{f}_{j+1}||_2^2 + \frac{1}{2}||W_{\Delta\mathrm{f}}(\mathbf{f}_{j+1} - \mathbf{f}_j)||_2^2 \tag{2-2}$$

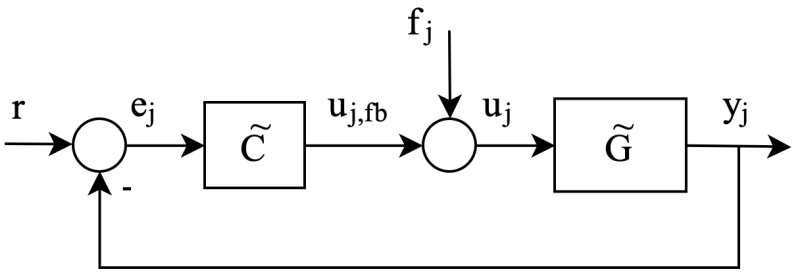


Figure 2-2: Block diagram of the norm-optimal ILC where $\tilde{C}$ is a feedback controller, $\tilde{G}$ a systems input-output description, $u_{j,\mathrm{fb}}$ the feedback input, f_j the learnt feedforward input and u_j the combined feedback and feedforward input signal.

with for a SISO system $\mathbf{e}_j \in \mathbb{R}^N$ a vector with the output error at every time step of iteration j, $\mathbf{f}_j \in \mathbb{R}^N$ a vector with the feedforward input at every time step of iteration j, W_e a matrix penalising the output error, W_f a matrix penalising the feedforward input signal and $W_{\Delta\mathrm{f}}$ a matrix penalising the learning speed over the iteration domain aimed at reducing the iteration-varying disturbances. For MIMO systems the respective sizes of $\mathbf{e}_j$ and $\mathbf{f}_j$ will have to be changed accordingly. For this cost function, an analytical solution of the learning rule can be obtained. This analytical solution is of the form $\mathbf{f}_{j+1} = Q\mathbf{f}_j + L\mathbf{e}_j$ with

$$Q = (J^T W_\mathrm{e} J + W_\mathrm{f} + W_{\Delta\mathrm{f}})^{-1}(J^T W_\mathrm{e} J + W_{\Delta\mathrm{f}}) \in \mathbb{R}^{N\mathrm{x}N},$$
$$L = (J^T W_\mathrm{e} J + W_\mathrm{f} + W_{\Delta\mathrm{f}})^{-1} J^T W_\mathrm{e} \in \mathbb{R}^{N\mathrm{x}N},$$

as per [54, p.4], where J is a convolution matrix of the impulse response of the closed-loop system as collected by experiments. The inclusion of this impulse-response leads to an up to one-step convergence of the learning rule. This one-step convergence can, for example, be realised on linear systems with $W_{\Delta\mathrm{f}} = 0$. The fast convergence is also one of main advantages claimed by the author of the norm-optimal ILC.

In the same way as for the PID-type ILC, the norm-optimal ILC is constructed for linear systems. However, during direct conversations with the author of [54], it was pointed out that the norm-optimal ILC framework should also work for nonlinear systems [56]. In such a case a slower convergence might be expected, but most of the results should remain equal.

2-3 Handling constraints

Up to this point, two ILC algorithms have been introduced, PID-type ILC and norm-optimal ILC. For both of these methods, no way of including time-domain constraints has been discussed. During this thesis, mainly input constraints are concerned. With the main input being the blade pitch, there are two types of constraints of concern, a constraint on the direct value of the input, and a constraint on the rate of change of the input. Both are needed to keep the requested blade pitch in a feasible region. In this section, the inclusion of constraints in the PID-type and norm-optimal learning controllers are discussed.

2-3-1 Constrained PID-type Iterative Learning Control

Handling constraints for the PID-type ILC can be done in quite a rudimentary way, namely by using input clipping. With this method, the input is basically processed before entering the controllable system, as can be seen in Figure 2-3. As the systems here considered are discrete-time, the clipping will also be done in discrete time. The clipping is done in two stages, in the first stage the zeroth order of the input is constrained by the inequality constraint $u_\mathrm{min} \leq u_j[n] \leq u_\mathrm{max}\ \forall n \in [0, N]$ with $u_j[n]$ the input at time step n and u_min and u_max the respective lower and upper limit. Using this constrained input, the rate of change, or the inputs' first-order difference, is constrained in the second block. This is done using $u_{\Delta\mathrm{min}} \leq u_j[n] - u_j[n-1] \leq u_{\Delta\mathrm{max}}\ \forall n \in [0, N]$ with $u_{\Delta\mathrm{min}}$ and $u_{\Delta\mathrm{max}}$ the respective lower and upper bounds on the rate of change. In practise, the minimum and maximum rates of change are often equally spaced around zero.

A problem that can occur with this setup is windup on the input, similar to integral windup in regular PID-controllers. This windup can occur when the PID-type ILC keeps requesting more input from the system whilst the input constraints are already violated. This leads to an increasing demand for the input signal, with no actual change in the used or realised input and resulting output signal. This windup is a result of the discrete-time integrating action that is inherent to the PID-type ILC learning rule. A solution for this problem is to feed back the constrained input signal to the learning rule in favour of the previously requested input signal. This will prevent windup on the input and makes reaching some steady-state value of the requested input possible, given that the learning rule itself is stable. An example of this anti-windup scheme can be seen in Figure 2-4 which demonstrates what the output does with and without this scheme whilst controlling a wind turbine rotor speed using the blade pitch.

2-3-2 Constrained Norm-optimal Iterative Learning Control

Constraints for the norm-optimal ILC work slightly different to those in the PID-type ILC. In principle, one could use the same scheme as presented above that clips the requested input signal, but this is a bit more involved due to the split feedback and feedforward parts of the norm-optimal ILC. Where in the PID-type learning rule the clipping could be done beforehand for all time steps, the norm-optimal ILC has to apply clipping to the combination of the feedback controller input together with the learnt feedforward input signal. This would require the need to read the requested input signal in real-time, after which the clipping can be done, and the requested input has to be fed back to the system. There are however also alternative strategies as described below.

The first, and perhaps most obvious, way of limiting the input value is to adjust the weight on the feedforward input by changing matrix W_{f} in cost function $\mathcal{J}(\mathbf{f}_{j+1})$ of Equation 2-2. By increasing the magnitude of the matrix elements, the feedforward input should decrease. This does however not guarantee any bounds on the input, but should in general decrease the magnitude of the feedforward input. Besides changing the magnitude of the matrix elements, also the form of the matrix W_{f} can be changed. For normal use, the choice of $W_{\mathrm{f}} = w_{\mathrm{f}} I$ is recommended with I an identity matrix of appropriate dimensions and $w_{\mathrm{f}} \in \mathbb{R}$ [55], but other forms can also be used. One could use, for example, the matrices as can be seen in Equation 2-3 where $W_{\mathrm{f,roc}}$ aims to penalise the rate of change of the input and $W_{\mathrm{f,tvar}}$ is a time-varying

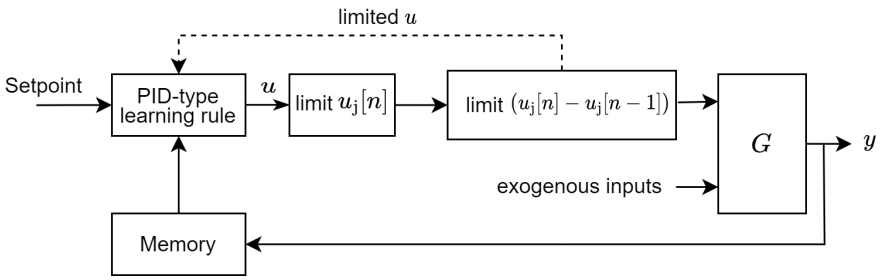
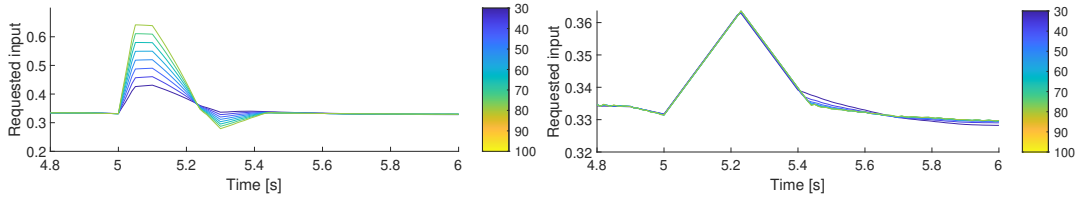


Figure 2-3: Input constraints for PID-type ILC using clipping on the zeroth and first-order difference of the requested input signal. The limited input signal is fed back to the PID-type learning rule to prevent integral windup of the input. Here G is the controlled system, u the requested system-input by the PID-type learning rule and y the output(s) of the system.



(a) Without anti-windup scheme **(b)** With anti-windup scheme

Figure 2-4: Requested input of the PID-type ILC after convergence of the output error to a non-zero value due to input constraints. In (a) the requested system input without an anti-windup scheme keeps on growing for higher iterations due to windup. In (b) the anti-windup scheme makes sure the requested input matches the input constraints, the maximum power the system can deliver, and prevents windup in the requested input of the controller.

penalty where the first inputs are penalised more than the final inputs aimed at limiting the initial peak of input usage with $\delta = 1/N$ with N the number of time steps in the iteration.

$$W_{\text{f,roc}} = w_{\text{e}} \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}, \quad W_{\text{f,tvar}} = w_{\text{e}} \begin{bmatrix} 1 & & & & \\ & 1-\delta & & & \\ & & \ddots & & \\ & & & 2\delta & \\ & & & & \delta \end{bmatrix} \tag{2-3}$$

Two ways of dealing with (input) constraints have been presented, however both methods still have some limitations. In the first method, the constraints are handled outside of the controller which leads to extra non-linear dynamics in the system that the controller has to deal with. This could, in the worst-case scenario, lead to unstable nonconverging solutions. The second method can be seen as a type of soft-constraints where the use of high inputs is penalised. There are no guarantees on how well these constraints work. This lack of guarantee could lead to the use of an overly aggressive weighting matrix W_{e} for which the input constraints, under testing, are still met. The use of such aggressive penalising of the input signal could lead to an under-performing system.

Therefore a third method is presented in which the ILC formulation is considered as an optimisation problem enabling the inclusion of hard constraints on the input and or the output. This method is effectively an extension of the vanilla method presented in Section 2-2-2, where the cost function $\mathcal{J}(\mathbf{f}_{j+1})$ as defined in Equation 2-2 is still minimised but now with additional constraints. An embodiment of the proposed method can be formulated as follows:

$$\begin{aligned} \mathbf{f}^*_{j+1} &= \arg\min_{\mathbf{f}_{j+1}} \mathcal{J}(\mathbf{f}_{j+1}) \\ &\text{s.t. } g(\mathbf{f}_{j+1}) \leq 0 \end{aligned} \tag{2-4}$$

with $\mathbf{f}^*_{j+1}$ the optimal feedforward input and $g(\cdot)$ some function. In this formulation, the same objective function is minimised, but now some constraints on the feedforward control signal

can be imposed. These constraints could, for example, be limiting the value of $\mathbf{f}_{j+1}$ itself or the rate of change of $\mathbf{f}_{j+1}$. In principle, since the norm-optimal framework uses a linear approximation of the output error, limits could also be imposed on the estimated output error $\mathbf{e}_{j+1} = \mathbf{e}_j - J(\mathbf{f}_{j+1} - \mathbf{f}_j)$ with J the convolution matrix of the closed-loop impulse response [57]. Note that for nonlinear systems the convolution matrix J is only an approximation, and as such only an estimate of $\mathbf{e}_{j+1}$ is available. Note that this approximation is also one of the main reasons that convergence for non-linear systems is expected to be slower. As a last step, the minimisation problem of Equation 2-4 using the cost function of Equation 2-2 can be rewritten into all known parts with only $\mathbf{f}_{j+1}$ as an unknown that has to be found as follows:

$$\begin{aligned}\mathbf{f}^*_{j+1} &= \arg\min_{\mathbf{f}_{j+1}} \mathcal{J}(\mathbf{f}_{j+1}) \\ &= \arg\min_{\mathbf{f}_{j+1}} \frac{1}{2}||W_{\mathrm{e}}(\mathbf{e}_j - J(\mathbf{f}_{j+1} - \mathbf{f}_j))||_2^2 \\ &\qquad + \frac{1}{2}||W_{\mathrm{f}}\mathbf{f}_{j+1}||_2^2 + \frac{1}{2}||W_{\Delta\mathrm{f}}(\mathbf{f}_{j+1} - \mathbf{f}_j)||_2^2\end{aligned} \tag{2-5}$$

For this last method, the proposed inequality constraint $g(\mathbf{f}_{j+1}) \leq 0$ can achieve different results depending on the exact form of $g(\cdot)$. An example of limiting the absolute value by function g_1 and the input rate of change by function g_2 can be seen in Equation 2-6. Of course one could also implement different constraint functions, for example a time-varying constraint on the input.

$$g_1(\mathbf{f}_{j+1}) = \begin{bmatrix} f_{j+1}[0] - f_{\max} \\ \vdots \\ f_{j+1}[N] - f_{\max} \\ f_{\min} - f_{j+1}[0] \\ \vdots \\ f_{\min} - f_{j+1}[N] \end{bmatrix} \leq 0, \quad g_2(\mathbf{f}_{j+1}) = \begin{bmatrix} (f_{j+1}[0] - f_{j+1}[1]) - f_{\Delta\max} \\ \vdots \\ (f_{j+1}[N-1] - f_{j+1}[N]) - f_{\max} \\ f_{\Delta\max} - (f_{j+1}[0] - f_{j+1}[1]) \\ \vdots \\ f_{\Delta\max} - (f_{j+1}[N-1] - f_{j+1}[N]) \end{bmatrix} \leq 0 \tag{2-6}$$

2-4 Control objective for Iterative Learning Control

In this chapter two ILC methods have been presented, the PID-type ILC and the norm-optimal ILC. For both of these methods, the aim is to reduce some objective or cost function. For the PID-type ILC this is an output error, and for the norm-optimal ILC this is a quadratic cost function that includes a term for the output error. So, in general, the aim of these ILC is to minimise an output error function. Thus resides the question; how should this output error be defined? Or, an alternative formulation; what is a suitable reference signal to define the error function against? The answer to this question is highly dependent on the specific aim of the controller. One could aim to tune the controller so that the TSO requirements are satisfied, or perhaps one would like to make lighter and cheaper wind turbines by reducing

certain mechanical loadings on the wind turbine. For each of these situations, a different reference or objective might have to be defined, but more on this is presented in the following chapters. For now, it is important to know that the methodologies presented in this chapter could have a wide applicability, depending on the specific goal and choice of reference signal.

2-5 Conclusion methodology

In this section it has been reasoned why the use of ILC is valid for the application of optimal grid fault control of wind turbines. It has been concluded that two ILC methods are of particular interest, the PID-type ILC and the norm-optimal ILC. Both of these ILC aim to follow a reference output signal and minimise the associated output error, but the way they do this is different. The PID-type ILC is model-free and rather simple in its setup. The norm-optimal adds a bit more complexity, but has the claimed benefit of faster convergence. For both methods there are ways to add system input constraints, where again the norm-optimal ILC has more advanced and complex solutions also as an option. Due to the increased complexity of the norm-optimal ILC, the results are expected to be superior to those of the more simplistic PID-type ILC. This also provides a choice between two ILC methods, a relatively simple to use PID-type ILC or a more advanced norm-optimal ILC. In the following chapters, implementation and testing using both ILC methods is presented, where one can also see if the claimed difference between both methods holds true in practise.

Chapter 3

Grid fault control introduction

In the previous chapter, the methodology used for controllers during this thesis has been presented. In this chapter that work is used to move from a general methodology, to a specific methodology for grid fault controllers. The chapter will start with an introduction of what the grid fault is and how it is used throughout the rest of the thesis starting with a short section on controller objectives and finishing with the introduction of the wind turbine models used.

3-1 Grid fault scenario definition

Before testing any grid fault controller, a definition of this grid fault is stated. As already discussed in Section 1-3, the TSO requirements form the primary foundation of the need for a grid fault controller. However, so as to not dilute the focus of thesis to much into regulatory requirements, it has been chosen to define a single grid fault and benchmark the controller based on that. As already concluded, the low-voltage ride-through (LVRT) requirement is one with high impact on the wind turbine and, as such, is the basis of the grid fault scenario.

The LVRT requirement defines that a wind turbine must remain connected to the grid during abrupt intermittent drops in line voltage. For the actual TSO requirements, these grid faults can lie in an envelope of shapes, but for now a single shape has been chosen. The shape of the grid fault can be seen in Figure 3-1 and is used for all the results in Chapter 4. As one can see in the figure, the grid fault starts with a quick linear decrease of the line voltage from 1 v/p.u. to 0.2 v/p.u. followed by a short time at 0.2 v/p.u. and finally a linear increase to 1v/p.u.. The fault time, the time at 0.2 v/p.u., has been chosen to be 0.3 seconds, as this is still short enough to qualify within the LVRT requirements and long enough to properly see the negative effects on the wind turbine. The fault slopes have been matched to the nominal LVRT tests done internally by Siemens Gamesa and are 1700% / sec for the downward slope and 400% / sec for the upward slope.

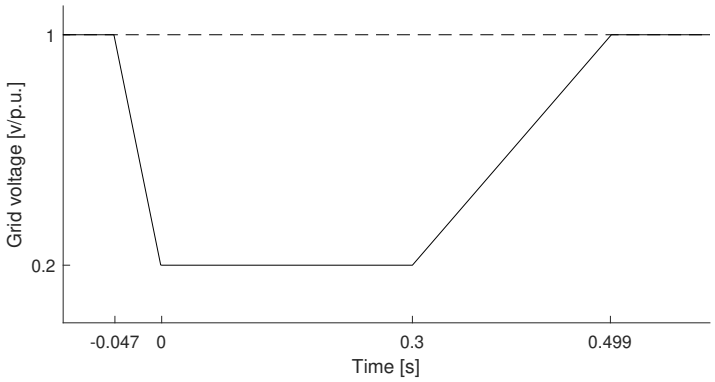


Figure 3-1: The low-voltage ride-through profile, or grid fault profile, during which the wind turbine needs to stay operational and connected to the electrical grid, that is used during the thesis. The voltage level is measured relative to the nominal voltage. The ramp-down and ramp-up rates are compliant with internal Siemens Gamesa testing with a fault time of 0.3 seconds.

3-2 Grid fault controller objectives

The subject of this thesis is creating optimal grid fault controllers, but before we can talk about optimality, an objective must be defined. As discussed in the previous section, the scenario considered during this thesis is a short intermittent fault on the voltage of the grid. Due to this grid fault, the wind turbine will start overspeeding and certain turbine loading channels might increase. After the fault is over, these unwanted effects still influence dynamic turbine behaviour until the system reaches normal operation again.

If one wants to minimise the total impact of the grid fault on the wind turbine, it would be logical to look at the minimisation of the l^2 norm of some output errors. Minimising an l^2 norm is also the basis of the PID-type ILC and the norm-optimal ILC. However, one could also define an objective function to reduce the l^∞ norm of a channel, which corresponds to the largest absolute error. This extension with the l^∞ norm could be done with the constrained norm-optimal ILC framework that was presented in Chapter 2-3-2. However, this extension will increase complexity and necessitate the need for an optimisation framework. Therefore, it is also recommended to only use the l^∞ norm if there is a valid reason to do so.

The definition of a relevant output channel or reference signal is still missing. This reference signal is the baseline towards which the controller tries to steer the system. If one thinks about what could happen during a grid fault, then two reference channels come to mind. The first being a reference signal on the rotor speed, since it is expected that the rotor will start overspeeding during the fault and show oscillations in the transient after the fault. The second reference signal is a turbine loading channel, as one would expect that the abrupt change in rotor speed would also translate into mechanical loading on the WT structure. A suitable mechanical WT loading channel is the tower-bottom moment, the bending moment that is in line with the dominant wind direction. It is expected that the tower-bottom moment will increase after the grid fault, and undergo oscillatory transients that decay back to nominal behaviour after the grid fault.

3-3 Wind turbine model

To demonstrate and test the controllers presented in this thesis, simulations are used. Simulations are used in preference to using real WT's since testing using real turbines can be slow, expensive, or in the worst case, impose risks to the (expensive) turbine. During this thesis, two models are used. The first model is a simplified WT model used to develop and test the proposed models, and the second model is a higher fidelity model that can also be used to validate the results. Both models are discussed in more detail below.

3-3-1 Low-fidelity wind turbine model

The first model mentioned here is the NREL 5-MW reference WT widely used in the literature, as described in [58]. The model is based on the following first-order differential equation,

$$J_{\mathrm{r}}\,\dot{\omega}_{\mathrm{r}}(t) = T_{\mathrm{r}}(t) - T_{\mathrm{g}}(t), \tag{3-1}$$

where ω_{r} is the speed of the rotor, T_{r} the torque of the rotor, T_{g} the generator torque, J_{r} the rotational inertia of the rotor. This equation is then converted to discrete time using zero-order hold that results in the following formulation as used in the model.

$$\begin{aligned}\omega_{\mathrm{r}}[n] &= \omega_{\mathrm{r}}[0] + \sum_{i=1}^{i=n}(T_{\mathrm{r}}[i] - T_{\mathrm{g}}[i])t_{\mathrm{s}}\,J_{\mathrm{r}}^{-1},\\ T_{\mathrm{r}}[n] &= \frac{1}{2}\rho AR\frac{C_{\mathrm{p}}(\lambda,\beta)}{\lambda[n]}V[n]^2.\end{aligned} \tag{3-2}$$

Where t_{s} is a sufficiently small sampling time, n the discrete time index, $\omega_{\mathrm{r}}[0]$ the initial rotor speed, ρ the air density, A the area swept by the rotor, β the pitch angle of the rotor blades, R the rotor radius, $C_p(\lambda,\beta)$ the nonlinear power coefficient, $\lambda = \omega_{\mathrm{r}}R/V$ the tip speed ratio and V the wind speed. Using the rotor speed and the generator torque, the output power P of the WT can be calculated simply by taking the product as follows $P_{\mathrm{g}}(t) = \omega_{\mathrm{r}}T_{\mathrm{g}}$. In the following sections, this model is denoted by G, a transfer function that outputs the rotor speed, generator power output and tip speed ratio based on the incoming wind speed , generator torque and reference blade pitch.

The model of equation 3-2 does not have any blade pitch actuator dynamics, thus an extension of the model with the blade pitch actuator dynamics is considered. Such pitch actuator dynamics are, in some literature, modelled by a first-order low-pass filter of the form $G_{\mathrm{a}}(s) = (\tau_{\mathrm{a}}s + 1)^{-1}$ with $\tau_{\mathrm{a}} \in \mathbb{R}^+$ the actuator time constant [59, 60]. Such a model is easy to implement, but finding the appropriate time constant for the NREL 5-MW reference WT has proven to be difficult. However, what is known is the equivalent spring and damping constant of the pitch actuator as defined by the NREL itself [58]. This leads to the following model of the blade pitch actuator;

$$\beta = G_{\mathrm{a}}(s)\beta_{set}, \qquad G_{\mathrm{a}}(s) = \frac{1}{m_{\mathrm{a}}s^2 + b_{\mathrm{a}}s + k_{\mathrm{a}}}, \tag{3-3}$$

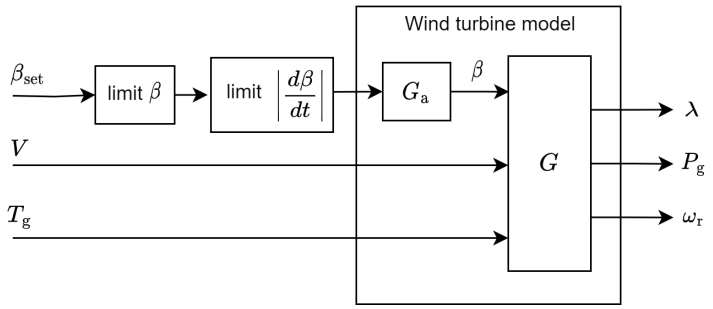


Figure 3-2: Block diagram of the WT model used during testing based on [58]. The transfer function G is a low-fidelity nonlinear wind turbine model which outputs the rotor speed, power output and tip-speed ratio based on the incoming wind speed, generator torque output and reference blade pitch. The blade pitch β is calculated by firs limiting β and its first derivative, and then calculate the response of the blade pitch actuator by transfer function G_{a}.

where β_{set} is the pitch setpoint as indicated by a controller and $m_{\text{a}}, b_{\text{a}}, k_{\text{a}}$ are the blade mass, the equivalent linear damping constant, and the equivalent linear spring constant, respectively. For the implementation of Equation 3-3 with the previous equations of motion, the transfer function will need to be converted from the Laplace domain to the Z domain to convert it to discrete time. The last step in modelling the blade pitch actuator is to include some physical constraints on the motion. Constraints are as per the WT definition in [58] and are as follows:

$$0^\circ \leq \beta \leq 90^\circ, \qquad \left|\frac{d\beta}{dt}\right| \leq 8^\circ/s. \tag{3-4}$$

These two constraints limit the absolute angle of the blade pitch and the blade pitch speed, respectively. A block diagram of the resulting model can be seen in Figure 3-2.

3-3-2 High fidelity wind turbine model

During this thesis, the model described in the previous subsection is used to develop and test controllers. The advantage of that model is the relative simplicity, allowing for fast simulations, and thus faster evolution of the controller during the initial testing phases. The model does, however, assume a completely fixed and rigid WT structure, while one of the objectives of this thesis is to reduce such structural loads. In addition, a more accurate higher-fidelity model could be more challenging to control due to higher-order models and disturbances, giving a more accurate insight into the performance of the proposed models.

This high-fidelity wind turbine model is based on a 15 MW commercially-designed turbine, and is modeled in Bonus Horizontal axis wind turbine simulation Code (BHawC), a nonlinear aeroelastic software tool developed by Siemens Gamesa. Studies of the BHawC software and models show that the simulation results agree with the widely used and validated FAST v8 software and with the experimental data collected from field tests [61, 62]. The BHawC software also includes field-deployed controllers, which could serve as a performance benchmark for the controllers developed here.

In the following chapter the here presented low-fidelity and high-fidelity wind turbine models is used to test and implement the ILC algorithms. In these test, the wind turbine models will experience a grid fault as also specified also in this chapter.

Chapter 4

Grid fault controller results

In the previous chapters, the control methodology, the grid fault scenario, and the wind turbine models are introduced. In this chapter, results are presented from the controllers tested on the two different wind turbine models, the low-fidelity and the high-fidelity. For this chapter, it is assumed that the shape of the fault is known and equal throughout learning. The eventual goal of this learning is to find an optimal feedforward signal that minimises a particular cost function. Once this signal is found, it could, in theory, be used on a real wind turbine if such a fault were to occur.

In this chapter, learned input signals as generated by PID-type ILC and the norm-optimal ILC are evaluated. For these two controllers multiple configurations have be implemented and tested, such as an unconstrained version, multiple constrained versions, and a causal versions. The unconstrained ILC algorithm show the theoretical maximum performance of the ILC but does not provide a realistic scenario, as actual wind turbines usually have (rate-)limited actuator dynamics. The final results is a causal constrained controller, which provides the most realistic indication of real-life performance gains that are possible. In this chapter all tests are performed with the objective of reducing rotor speed error, where in Appendix 4-7 the effects of this objective function on mechanical loading are investigated. Additionally, the direct objective to reduce the mechanical loading has also been documented in this appendix.

4-1 Baseline controller

Before any results are made using an ILC algorithm, it is useful to define some baseline controller to compare the results against. This baseline controller provides comparison on how well the ILC performs. Since there are two models used during this thesis, a low-fidelity and high-fidelity model, two baseline controllers will also have to be used. For the high-fidelity model, a Siemens Gamesa field-deployed controller already exists and is available for this thesis. This is then also the absolute performance of the currently field-deployed wind turbines and, as such, is the perfect candidate as a baseline controller.

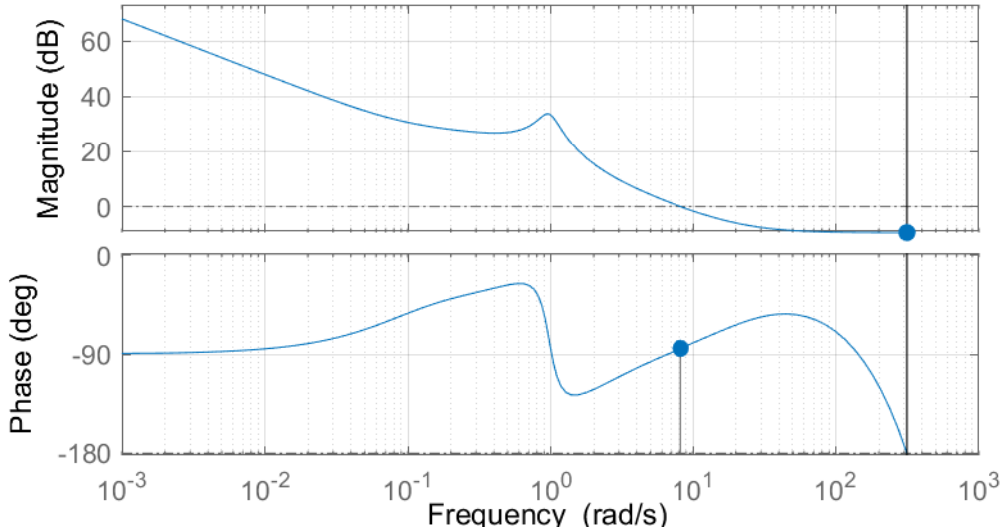


Figure 4-1: Bode Diagram of the baseline PID-controller applied on the low-fidelity wind turbine model, constructed around a linearisation point matching the conditions used during testing. So here visible is the bode diagram of the transfer function of the loop gain $L = GK$ with G defined in Chapter 3 and K the PID-controller.

For the low-fidelity model, no baseline controller has been provided. As such, this controller is designed as part of this thesis. As was already concluded during the literature review, many wind turbines make use of PID controllers, so this is the controller of choice that will serve as a baseline controller for the low-fidelity model. Since the creation of a baseline controller serves a comparative function during this thesis, the option of more complex and possibly better performing controllers has been dismissed in favour of the speed and ease of use of the PID controller.

The baseline PID-controller has been tuned by making use of the Bode plot of the open-loop system. To do this, a linearisation point has been employed since the low-fidelity model is also non-linear. This linearisation is done with the wind speed in steady state of 20 m/s with maximum power point tracking, and the blade also in steady state for this situation. The resulting PID controller has a gain margin of 9.44 dB and a phase margin of 95.2 degrees with a proportional gain of 10, integral gain of 1 and derivative gain of 0.4, as can be seen in Figure 4-1. This controller has a control bandwidth of 6 rad/s, which is on the high side and may cause problems in actuating high-frequency noise, where a more desirable is a control frequency around 1 rad/s. To do this properly, however, first the resonance frequency around 1 rad/s is decreased by including a notch filter around this frequency. The resulting bode plot can be seen in Figure 4-2, where the proportional gain is also reduced to two. This gives the final values of $K_\mathrm{p} = 2$, $K_\mathrm{i} = 1$, and $K_\mathrm{d} = 0.4$. This gives a gain-margin of 12.9 dB and a phase-margin of 74.3 degrees.

The actual results from the baseline controllers can be seen in Appendix C. Here, the time-domain performance of both controllers is tested on a grid fault.

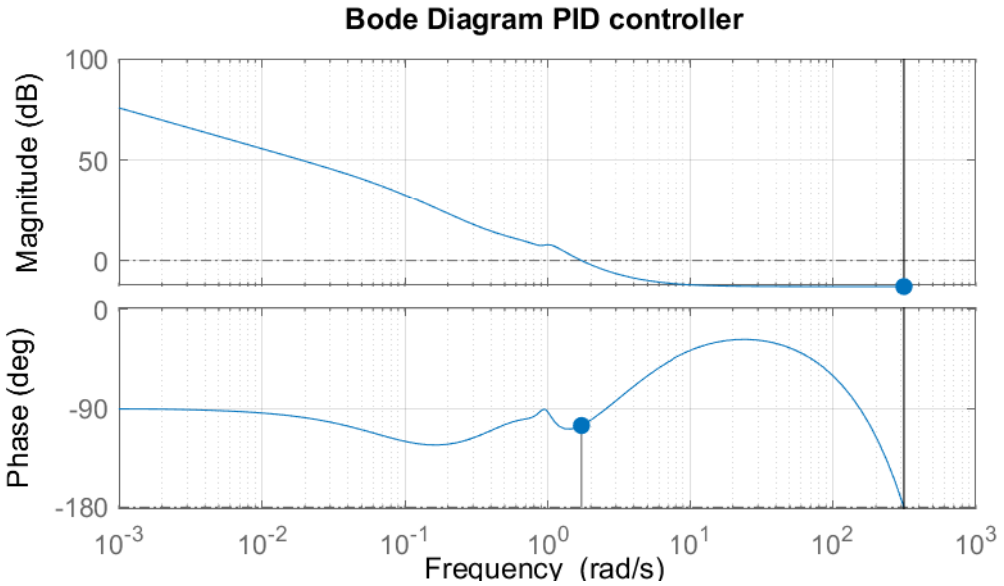


Figure 4-2: Bode Diagram of the baseline PID-controller with notch filer applied on the low-fidelity wind turbine model, constructed around a linearisation point matching the conditions used during testing. So here visible is the bode diagram of the transfer function GK with G defined in Chapter 3 and K the PID-controller. This controller will serve as a baseline controller.

4-2 Unconstrained controllers results

In this section, the unconstrained controller performance is presented. These results show a hypothetical case in which actuator constraints do not impede the controller design, and correspond to the block diagram of Figure 3-3-1 with the limits on β and limits on $|d\beta/dt|$ blocks removed. The main purpose of this unconstrained controller structure is to show the best-case scenario without any input constraints. This unconstrained controller result is presented for the low-fidelity model as a demonstration.

4-2-1 Unconstrained PID-type ILC on low-fidelity model

The first ILC algorithm that is used is the PID-type ILC. Since the low-fidelity model only has a limited number of outputs, the objective function is also limited. Of the available outputs, power, rotor speed and tip speed ratio, the rotor speed is minimised. Since no input constraints are used, the anti-windup scheme of Section 2-3-1 will not be used.

The aim of the PID-type ILC is to reduce the 2-norm of the output error, where this is preferably done in the least number of iterations. Also, instability of the learning rule is undesirable, where stability is defined as boundedness of the 2-norm of the output error. The tuning of the PID-type controller could be done using an analytical model, but looking towards implementation on the high-fidelity model, this is undesirable as its analytical model is not available. Additionally, this method would require linearising the models which can, especially during the grid faults, give unrealistic results. Instead, the tuning of the controllers is done by looking at the time-domain results of the controller parameters used.

K_p	K_i	K_d	1st iter < 0.01	mean 2-norm	max 2-norm	final iter 2-norm
0.1	0	0	n.a.	0.8575	1.0000	0.7928
0.1	0	1	7	0.0346	1.0000	0.0000
0.1	0.01	1	n.a.	$9 \cdot 10^{26}$	$3 \cdot 10^{28}$	$3 \cdot 10^{28}$
0.1	0.001	1	15	0.6301	7.0740	0.0000
0.1	0.0001	1	7	0.0337	1.0000	0.0000

Table 4-1: Tuning of the non-causal unconstrained PID-type ILC. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a controller tuning that has a zero 2-norm in the final iteration with a maximum 2-norm equal to 1, indicating no decrease in performance w.r.t. the baseline controller after any iteration. The mean 2-norm can give a secondary indication to the convergence speed.

The tuning of the PID-type ILC needs to find three values, a value for the proportional gain K_p, the integral gain K_i, and the derivative gain K_d. First, the process has started by finding values for which the learning is stable, so where the solution converges to some value lower than that of the baseline PID controller, where convergence is defined as boundedness of the 2-norm of the output error. The tuning process can be seen in Table 4-1, with the final values $K_\mathrm{p} = 0.1$, $K_\mathrm{i} = 1 \cdot 10^{-4}$ and $K_\mathrm{d} = 1$, as this has a fast convergence in seven steps, has no diminished performance for any intermediate iteration, max 2-norm $= 1$, and has the lowest mean 2-norm.

Since no input constraints are applied in this scenario, the output error decreases to zero as expected. The results can be seen in Figure 4-3. The starting blade pitch input that the PID-type ILC will supply is equal to the one of the baseline PID controller. This then also causes the 2-norm of the output to be equal for the first iteration. After each iteration, the 2-norm of the output error decreases, where after seven iterations the 2-norm has decreased to only 1% of the original value.

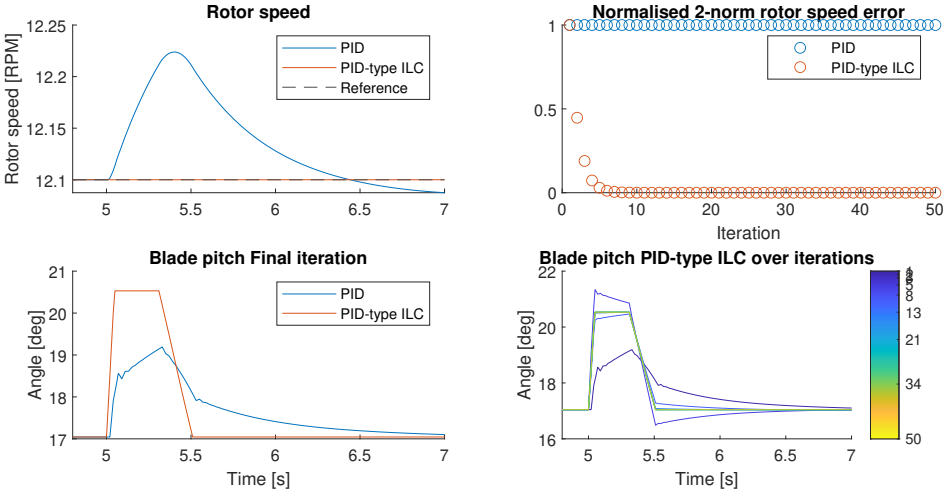


Figure 4-3: Non-causal unconstrained PID-type ILC compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1.

In the bottom right of Figure 4-3, one can see the blade pitch input requested from the PID-type ILC over the iterations. One can see that the shape of the blade pitch signal changes drastically over the iterations. In the first iteration the blade pitch is quite smooth, but for the final iteration there appear to be distinct transitions in the blade pitch signal. Up to the time of the fault, the blade pitch is constant and equal to the baseline PID controller. Once the fault starts, the PID-type ILC starts pitching the blades with a constant speed up to a steady-state value. After the fault, the blades are pitched with a constant speed back to the initial steady state blade pitch value. Close inspection of the shape of the blade pitch in the final iteration shows that it is, in fact, a linear copy of the grid fault even though the model is nonlinear. Since the controller manages to keep the rotor speed constant, the system is evidently 'pushed' into a linear region by the ILC algorithms. This then also explains why the shape of the blade pitch is a linear copy of that of the grid fault itself.

4-2-2 Unconstrained norm-optimal ILC on low-fidelity model

A norm-optimal ILC is also employed on the low-fidelity model without input constraints for reasons of comparison to the performance of the PID-type ILC. The tuning of the norm-optimal ILC is a bit different from the PID-type ILC. There are three parameters that need tuning, the weight on the output error W_{e}, the weight on the input signal W_{f} and a weight on the learning speed $W_{\Delta \mathrm{f}}$. Inside of the norm-optimal ILC the J matrix as described in Section 2-3-2 is used, where in Appendix D the calculation of the J matrix is shown in more detail. It has already been clearly indicated in the literature that $W_{\Delta \mathrm{f}}$ is only needed when using noisy data, which is not the case here, so this value is chosen to be zero. The weight on the input W_{f} has to be nonzero to prevent near singular matrix inversion in Matlab. Since the aim is to minimise a cost function, the exact values of the weight do not matter as much as the ratio between the two weights.

To start the tuning process, both the weight on the output error and feedforward input are set to identity. The results of this process can be seen in Table 4-2. Note that depending

W_{e}	W_{f}	$W_{\Delta \mathrm{f}}$	1st iter < 0.01	mean 2-norm	max 2-norm	final iter 2-norm
1	1	0	n.a.	0.9840	1.0000	0.9836
1	0.1	0	n.a.	0.8617	1.0000	0.8584
1	0.01	0	n.a.	0.4082	1.0000	0.3878
10	0.01	0	n.a.	0.1027	1.0000	0.0655
100	0.01	0	10	0.0486	1.0000	0.0077
1000	0.01	0	n.a.	0.2909	1.0000	0.5706
120	0.01	0	10	0.0474	1.0000	0.0064
140	0.01	0	9	0.0468	1.0000	0.0063
160	0.01	0	9	0.0484	1.0000	0.0156

Table 4-2: Tuning of the non-causal unconstrained norm-optimal ILC. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a controller tuning which has a zero 2-norm in the final iteration with a maximum 2-norm equal to 1, indicating no decrease in performance w.r.t. the baseline controller after any iteration. The mean 2-norm can give a secondary indication to the convergence speed.

on the controller tuning, the resulting 2-norm of the output error can change vastly. Take, for example, $W_e = 1$ and $W_f = 0.01$, where in the 2-norm it only decreases to 39% of the original value. This does not indicate that the norm-optimal ILC is not working properly, but instead that given the weighting matrices, the internally used cost function $\mathcal{J}(\mathbf{f}_{j+1})$ of Equation 2-2 is minimised. Evidently, a minimised cost function cost function here does not indicate a near-zero output error. If however $W_f = 0$, and $W_{\Delta f} = 0$ would be chosen, a zero output error could be expected [54, 56]. However, for the formulation of the learning rule used throughout this thesis, as defined in [54], the matrix L is singular to working precision when run in Matlab. To counteract this matrix singularity W_f will have to be chosen unequal to zero. That said, the best performing controller gains here are $W_e = 140$, $W_f = 0.01$, and $W_{\Delta f} = 0$ as this controller setting yields the lowest final 2-norm 0.6% of the original value with fast convergence.

A more detailed plot of the results from the tuned norm-optimal ILC can be seen in Figure 4-4. In this plot one can see that the norm-optimal ILC also manages to get the output error to near enough zero, but not completely zero. For linear systems it is claimed that the norm-optimal ILC manages to get convergence to zero, but evidently this does not hold extend for this nonlinear system. The maximum absolute output error is however very small at only $4.5 \cdot 10^{-4}$ rpm, but still noteworthy none the less.

Comparing the evolution of the blade pitch signal over iterations with that of the PID-type ILC of Figure 4-3, one can see that the input signal changes in a more gradual way with no 'overshoot' in the evolution of the input signal. Instead, it creeps more towards the final solution in a more gradual matter. This might be more desirable for real-life systems as the input signal does not change shape massively over the iterations, whilst still converging to approximately the same solution.

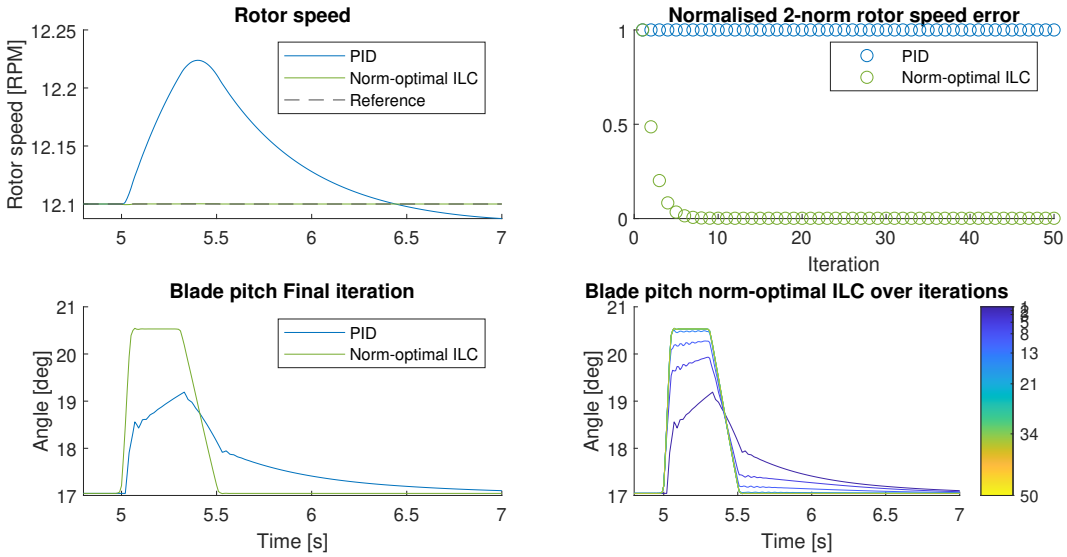


Figure 4-4: Non-causal unconstrained norm-optimal ILC compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1.

4-2-3 Comparison and conclusion of unconstrained ILC on low-fidelity model

Using the results of this subsection, a comparison can be made between the unconstrained PID-type ILC and the unconstrained norm-optimal ILC. In Figure 4-5 one can see a comparison plot of the output error, the 2-norm of the output error and the blade pitch and blade pitch rate. In the figure one can also see that the norm-optimal ILC still has a small rotor speed error as caused by the need for a non-zero zero penalty on the feedforward signal to prevent singular matrix inversion, where the PID-type ILC manages a zero output error. When inspecting the blade pitch signal, one can also see that the transitions of the norm-optimal ILC are not as crisp as those of the PID-type ILC, giving an indication of the worse output error of the norm-optimal ILC.

Though the norm-optimal ILC claims to have superior convergence speed for linear systems, it does, however, perform worse than the PID-type ILC with consistently slower convergence for the here considered nonlinear scenario and tuning. However, the way in which the controller inputs converge to that of the final iteration is different. For the here chosen tuning the PID-type ILC has some overshoot in the input signal, see Figure 4-3, where the norm-optimal ILC has a more gradual convergence towards the final feedforward input applied, see Figure 4-4. This difference is not relevant for this application, but for different systems or tunings one should be alert for this behaviour.

Both controllers in this section have shown that a near-zero output error is possible during a known grid fault when no constraints are applied. This does, however, require blade pitch

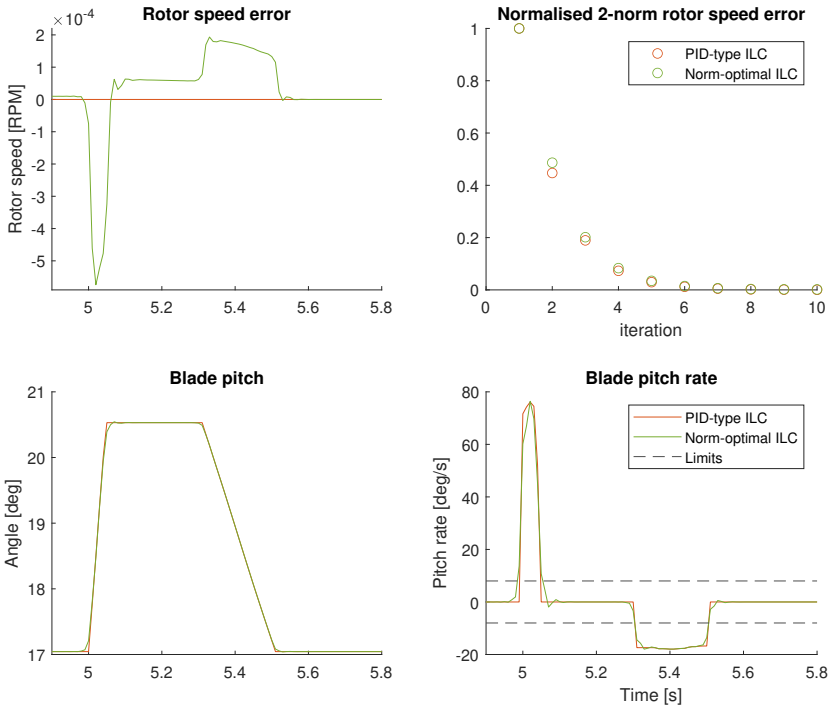


Figure 4-5: Comparison of the non-causal unconstrained PID-type ILC with the non-causal unconstrained norm-optimal ILC. Shown are the rotor speed error, blade pitch and blade pitch rate of the final iterations. Also, the progression of the 2-norm of the output error is shown. In the blade pitch rate plot the hardware limits are shown, but were not used at this point.

rates of almost 80 deg/s, which is tenfold the actual actuator capabilities. In addition, this result is only possible when the controller starts deploying the optimal feedforward signal from the beginning of the actual grid fault, something that would not be possible in real life due to causality violation. The causality problem has been addressed in the next section, where the effects of delayed learning is investigated.

4-3 Unconstrained controllers with causal learning

In the previous section, unconstrained ILC algorithms have been presented. This has been extended to a more realistic scenario with causal learning such that learning occurs only after the fault has started. This implies that one could detect the fault and fault magnitude, and then apply the learnt optimal feedforward signal. Since the learnt feedforward signal does not start before the grid fault has occurred, the wind turbine will have a non-zero rotor speed error, in contrast to the non-causal case where a zero rotor speed error for every time step could be reached. During this test it is investigated how this non-zero rotor speed error will effect the optimal learnt feedforward signal. These results can be seen in Appendix E. Later in this chapter, causal constrained controllers will also be introduced.

4-4 Constrained controllers on low-fidelity model results

Up to this point, only unconstrained controller results have been shown. Although this does give some insight into how the ILC algorithms work in the best-case scenario, it does not provide results that can be expected in the real world. Take, for example, the causal norm-optimal ILC of the previous section that used almost 1000 deg/s blade pitch rate, while the actuator is rate-limited to 8 deg/s. Therefore, there is the need to add input constraints in the controllers. This is first tested on the low-fidelity wind turbine model, and in the next section also on the high-fidelity model.

4-4-1 Constrained PID-type ILC on low-fidelity model

The first constrained controller shown here is the PID-type ILC. Here, constraints are handled by employing saturation on the input and the input rate of change, as described in Section 2-3-1. For the unconstrained case, the PID-type ILC manages to completely reduce the output error to zero. Doing this does, however, require large amounts of input, and more importantly an extremely high blade input rate of change. This is not feasible for real-life systems, and as such, here a constrained PID-type ILC is presented.

This time the controller tuning is started with the controller parameters found for the causal unconstrained controller. It is expected that the 2-norm of the output error will not reduce completely to zero for this case, but instead to a nonzero steady-state value. Therefore, an important tuning parameter to look at is the output error 2-norm in the final iteration. What will also be looked at is that there is no big increase in 2-norm in any intermediate iterations of the ILC algorithms. This can be indicated by a maximum 2-norm equal to one and by looking at the individual rotor speed error 2-norm for every iteration.

K_p	K_i	K_d	mean 2-norm	max 2-norm	final iter 2-norm
0.40	$1 \cdot 10^{-5}$	0.30	0.7932	1.0000	0.7866
0.50	$1 \cdot 10^{-5}$	0.30	0.7674	1.0000	0.7565
0.60	$1 \cdot 10^{-5}$	0.30	0.7971	1.2018	0.7429
0.50	$1 \cdot 10^{-5}$	0.40	0.8057	1.0000	0.7902
0.50	$1 \cdot 10^{-5}$	0.20	0.7397	1.0000	0.7258
0.50	$1 \cdot 10^{-5}$	0.10	0.7828	1.0000	0.7564
0.50	$1 \cdot 10^{-5}$	0.15	0.7379	1.0000	0.7180
0.50	$1 \cdot 10^{-4}$	0.15	0.7505	1.0000	0.7191
0.50	$1 \cdot 10^{-6}$	0.15	0.7374	1.0000	0.7180

Table 4-3: Tuning of the causal constrained PID-type ILC. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

The results of the tuning can be seen in Figure 4-6 and Table 4-3. First, one can see that, in general, it takes a few more iterations to converge to the final solution compared to the unconstrained version in Figure 4-3 where full convergence was already achieved in approximately 8 iterations. That said, there are quite some variations in between the different controller tunings, where many of the used values have large intermediate peaks in the 2-norm, indicating that the performance after certain iterations deteriorates compared to the respective proceeding iteration. However, this is not the case for the controller values $K_\mathrm{p} = 0.5$, $K_\mathrm{i} = 1 \cdot 10^{-5}$, and $K_\mathrm{d} = 0.15$, which also yields the lowest final 2-norm value. This tuning does not have the fastest convergence in the first few iterations, this is reserved for $K_\mathrm{p} = 0.5$, $K_\mathrm{i} = 1 \cdot 10^{-5}$ and $K_\mathrm{d} = 0.2$, but it does have a lower final 2-norm. Since here simulations are used, the preference is to the lowest final 2-norm, but when using field data, one might prefer the second tuning with a faster convergence and slightly higher final 2-norm.

The time domain results of the tuned constrained PID-type ILC can be seen in Figure 4-7. One can see that the maximum rotor overspeeding is reduced compared to the baseline PID

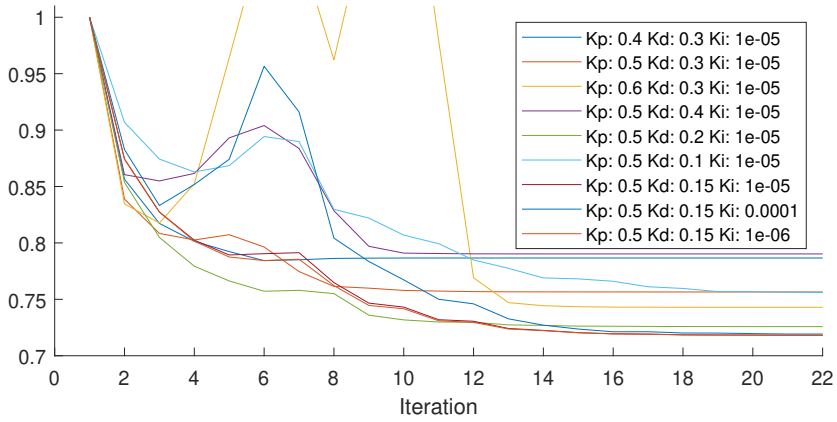


Figure 4-6: The 2-norm of the output error for the causal constrained PID-type ILC shown for different controller settings. Shown is the progression of the 2-norm, for different controller settings, after each ILC iteration.

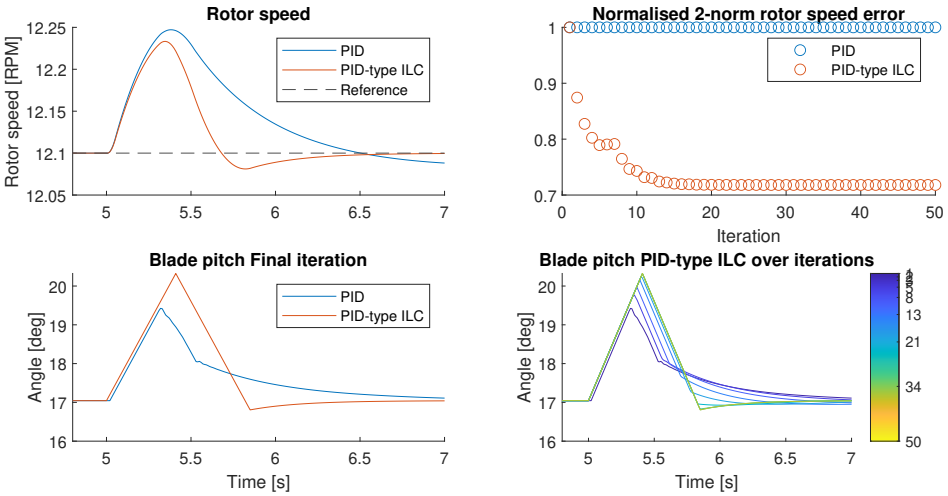


Figure 4-7: Causal constrained PID-type ILC compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

controller. In addition, the recovery speed of the rotor speed back to the reference signal is much higher, which is where the greatest improvement lies in this controller. When inspecting the blade pitch input signal used, one can see that the PID-type ILC continues to use more of the blade input right up to the point where the rotor speed error is significantly reduced, where the baseline controller fails to completely saturate the controller input signal.

4-4-2 Constrained norm-optimal ILC using saturation on low-fidelity model

The constrained norm-optimal ILC will also make use of the saturation block, as was the case for the PID-type ILC. In principle, one could also use an increasingly higher input penalty, and as such decrease the input signal used, but this did not yield enough of a reduction in input rate of change used. The results of this method can be seen in Appendix F, but for the remainder of this section the saturation method is used. An implementation of constrained

W_e	W_f	$W_{\Delta f}$	mean 2-norm	max 2-norm	final iter 2-norm
300	0.01	0	0.7476	1.0000	0.6750
100	0.01	0	0.7030	1.0000	0.6106
10	0.01	0	0.5832	1.0000	0.4888
1.00	0.01	0	0.3898	1.0000	0.2454
1.00	0.10	0	0.5222	1.0000	0.4385
2.00	0.01	0	0.4945	1.0000	0.4206
0.50	0.01	0	0.3449	1.0000	0.2214
0.40	0.01	0	0.3553	1.0000	0.2383

Table 4-4: Tuning of the non-causal constrained norm-optimal ILC using saturation. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

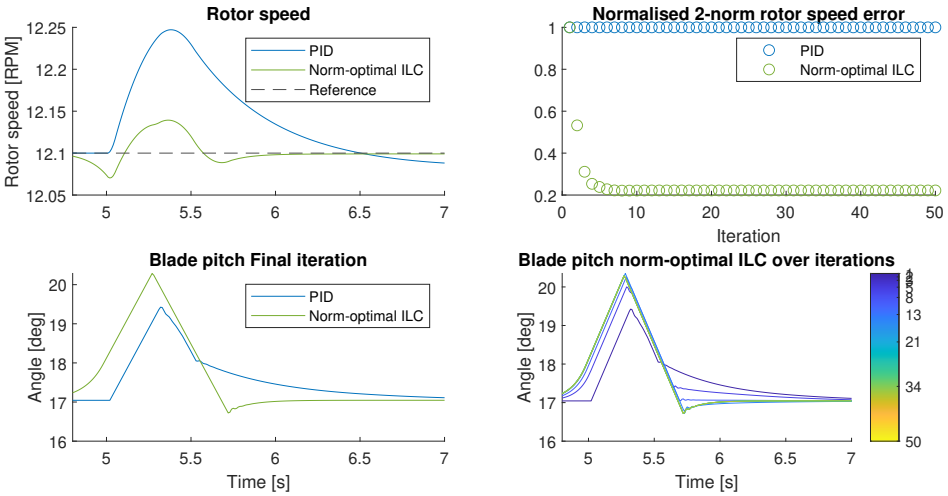


Figure 4-8: Non-causal constrained norm-optimal ILC using saturation compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

norm-optimal ILC using optimisation can be seen in Section 4-4-3

The tuning method for the constrained norm-optimal ILC is similar to that of the constrained PID-type ILC of the previous section. Here, one wants the lowest 2-norm in the final iteration, and also no intermediate loss in performance indicated by a peak in the 2-norm or a maximum 2-norm greater than one. The results of the tuning can be seen in Table 4-4. One can see that the values $W_e = 0.5$, $W_f = 0.01$ and $W_{\Delta f} = 0$ produce the lowest final 2-norm of only 0.22 times that of the baseline PID controller. This controller tuning also has monotonic convergence for this scenario, indicating that after every iteration the performance is equal or better in terms of the 2-norm of the output error.

The time domain results of the constrained norm-optimal ILC can be seen in Figure 4-8. Here, one can see that the norm-optimal ILC starts pitching the blades before the fault has even occurred. This tactic results in a lower peak overshoot compared to the baseline controller, and in term results in a significant improvement of the output 2-norm.

4-4-3 Constrained norm-optimal ILC using optimisation on low-fidelity model

The constrained norm-optimal ILC as presented above yields satisfactory results in that it is able to lower the 2-norm of the output error. It is however not clear if the found solution is optimal since the input constraints are handled outside of the controller, raising the question whether the found solution is actually an optimal solution. From one viewpoint, one could consider the saturation block outside the controller part of the nonlinear system dynamics, and as such accept that the found solution is optimal. However, from a second viewpoint, one could say that the handling of the constraints should be done inside of the controller using the limited system knowledge that is available. To investigate this phenomenon, an optimisation based constrained norm-optimal ILC is presented in which the minimisation and constraint handling are handled all inside of the controller structure.

Way of handling constraints

In Chapter 2 the constrained norm-optimal ILC has been introduced, but with one important omission in the shown constraints. Namely, all the constraints shown are on $\mathbf{f}_{j+1}$, the feedforward input, and not on the combined feedforward and feedback input signals. In practise, this is a substantial difference, as only limiting the feedforward signal can still result in a violation of the input constraints for the combined feedforward and feedback signal. Therefore, a new formulation will have to be made in which the feedforward input combined with the feedback input is penalised. Or in other words, an estimate for the combined feedforward and feedback input signals will have to be made. The following approximation is proposed for the input signal.

$$\hat{\mathbf{u}}_{j+1} = \mathbf{f}_{j+1} + \mathbf{u}_{j,\text{fb}} \tag{4-1}$$

Here $\hat{\mathbf{u}}_{j+1}$ is the estimated total input for the next iteration, $\mathbf{f}_{j+1}$ is the feedforward input for the next iteration, and $\mathbf{u}_{j,\text{fb}}$ the feedback input used in the previous iteration. Following is the intuition of why this input estimate could work. First, notice that the real next input $\mathbf{u}_{j+1}$ can be described as $\mathbf{u}_{j+1} = \mathbf{f}_{j+1} + \mathbf{u}_{j+1,\text{fb}}$ and, thus, the task can be reduced to finding a proper estimate for the feedback input term. Now, since after every iteration the feedforward input should steer the system to a lower output error, it seems reasonable to assume that $\mathbf{u}_{j+1,\text{fb}} \leq \mathbf{u}_{j,\text{fb}}$. Additionally, the norm-optimal ILC usually reaches some steady-state value after some iterations, meaning that in this case $\mathbf{u}_{j+1,\text{fb}} = \mathbf{u}_{j,\text{fb}}$ holds true. For these reasons, this input estimate is used to define the input constraints.

The input constraints are the same as in the previous sections, with a constraint on the value of the input itself and one on the rate of change of the input. The one difference is that here the input $\mathbf{u}_{j+1}$ will not be used, but instead $\hat{\mathbf{u}}_{j+1}$ as defined in Equation 4-1. The complete optimisation formulation can be seen below. Other variants of the formulation of the optimisation problem have also been tried and can be seen in Appendix G, but with less successful results.

$$\begin{aligned}
\mathbf{f}^*_{j+1} &= \arg\min_{\mathbf{f}_{j+1}} \mathcal{J}(\mathbf{f}_{j+1}) \\
&\text{s.t.} \left\| \hat{\mathbf{u}}^{2:N}_{j+1} - \hat{\mathbf{u}}^{1:N-1}_{j+1} \right\| / T_\text{s} \leq u_{\Delta\max}, \\
&\quad \hat{\mathbf{u}}_{j+1} \leq u_{\max}, \\
&\quad \hat{\mathbf{u}}_{j+1} \geq u_{\min}.
\end{aligned} \tag{4-2}$$

With $\mathcal{J}(\mathbf{f}_{j+1})$ as in equation 2-5,T_s the sampling time, $\mathbf{u}_{\Delta\max}$ the maximum rate of change of the input, $u_{\max}$ the maximum input and $u_{\min}$ the minimum input. The superscript $\mathbf{x}^{a:b}_j$ indicates that the samples a to b of iteration j are used in the vector $\mathbf{x}$. For more details on the terms in the above equation, see Section 2-3-2. To solve the above minimisation problem, Yalmip [63] is used from within MATLAB. An important notion for this constraint to work is that the feedback input term should not be to dominant or otherwise the performance of the combined controller will suffer, which is also demonstrated in Appendix G. If this is not done, than the feedback controller will react too aggressively on the added feedforward contribution resulting in oscillations in the blade pitch signal causing a violation of the input constraints.

Tuning the controller

As mentioned before, the optimisation based constrained norm-optimal ILC works best when used with a less aggressive feedback controller. Therefore, the feedback controller values $K_\mathrm{p} = 0.5$, $K_\mathrm{i} = 0$ and $K_\mathrm{d} = 0.023$ have been used, which are different from those of Section 4-1. Also, for the ILC algorithms, the controller variables will have to be tuned. The results of this tuning can be seen in Table 4-5. Unlike the tuning of previous controllers, here the maximum 2-norm is higher than one due to the tuned-down PID feedback controller that is dominant for the first iteration. That said, there is little to no difference in the tuning values of the optimisation based constrained norm-optimal ILC in terms of the mean and final 2-norm. Only when the error weight is set extremely low, the final 2-norm of the output error starts to increase.

Looking at the progression of the output errors 2-norm, one can see that there is quick convergence already in the third step, that is, only two iterations including an active feedforward signal have been used. This is then also one of the claimed strengths of this controller, the fast convergence to the optimal feedforward signal.

The time domain results of this optimisation based constrained norm-optimal ILC can be seen in Figure 4-9. Directly one can see that the ILC algorithm starts to adjust the blade pitch already before the grid fault has begun, and as such builds a sort of buffer in the rotor speed error. This is compared to the baseline PID controller, which only starts taking action after the actual grid fault has occurred. Also visible in the bottom-right of the plot is the composition of the ILC input signal from the feedforward and feedback parts. Here one can see that before at certain times the feedforward signal is even greater than the blade pitch rate of change constraint allows, but that once the feedback signal is added, an exact saturation of the input is achieved. This ability of the norm-optimal ILC to fully saturate the actuator at the right times leads to the dramatic decrease of rotor speed error observed here.

4-4-4 Comparison and conclusion of constrained ILC on low-fidelity model

In Figure 4-10 a comparison can be seen between the constrained PID-type ILC, the saturation based constrained norm-optimal ILC and the optimisation based constrained norm-optimal ILCs. The first observation one can make is that the norm-optimal ILCs perform significantly better than the PID-type ILC with a final 2-norm with approximately 0.2 for the saturation

W_e	W_f	$W_{\Delta\mathrm{f}}$	mean 2-norm	max 2-norm	final iter 2-norm
300	0.01	0	0.5071	1.9579	0.1365
100	0.01	0	0.5071	1.9579	0.1365
10	0.01	0	0.5076	1.9579	0.1372
1	0.01	0	0.5713	1.9579	0.2055
1000	0.01	0	0.5071	1.9579	0.1365

Table 4-5: Tuning of the non-causal optimisation based constrained norm-optimal ILC. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a low max 2-norm.

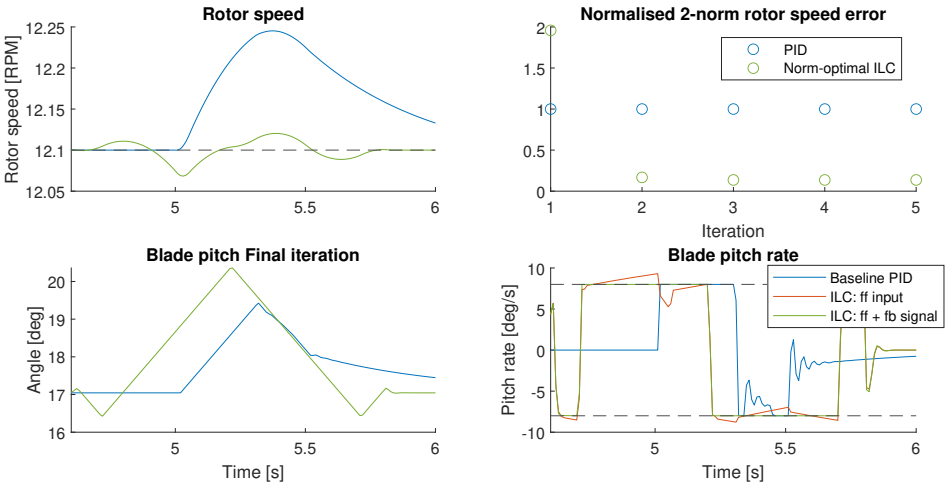


Figure 4-9: Non-causal optimisation based constrained norm-optimal ILC compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1. The bottom right plot shows the blade input rate of change of the baseline PID controller and the ILC, where the ILC is split in the feedforward, ff, and feedback, fb, parts respectively.

based and 0.14 for the optimisation based norm-optimal ILC compared to the 0.8 of the PID-type ILC. In other words, the norm-optimal ILCs have around a four times lower final 2-norm compared to the PID-type ILC, with the optimisation based norm-optimal ILC having the lowest final 2-norm.

A contributor to the lower 2-norm for the norm-optimal ILC is its ability to anticipate the effects of a feedforward input in the future by means of the J-matrix, enabling it to take relevant actions even before the fault occurs. Both versions of the norm-optimal ILC start pitching the blades before the five second mark, where the fault starts, resulting in a negative rotor speed error at the time the grid fault starts. This negative rotor speed error then acts as a sort of buffer on the rotor speed error once the actual grid fault begins. This tactic makes the norm-optimal ILC much more effective in reducing the output error 2-norm.

Comparing both norm-optimal ILCs, one can see that the optimisation based norm-optimal ILC manages to get a lower final 2-norm of the output error, and also faster convergence in three steps. This faster convergence does come at the expense of higher computation cost, but for the scope of this thesis, the computation cost is considered cheap compared to the cost of data, so this is not a big problem. That said, the optimisation based norm-optimal ILC also has a slightly different control approach compared to the saturated norm-optimal ILC. The optimisation based ILC starts pitching to the other direction from 4.5 to around 4.7 seconds, causing a positive rotor speed error. Additionally, the optimisation based ILC starts even earlier with pitching the blades compared to the saturated norm-optimal ILC.

All three controllers manage to achieve a performance gain compared to the baseline PID controller, but the norm-optimal ILCs do a better job of this. These controllers manage to anticipate the grid fault, resulting in a four-time lower 2-norm compared to the PID-type ILC. Therefore, the norm-optimal ILC would also be preferred in this situation. Within the norm-optimal ILCs, the optimisation based one works better than the saturation based with a lower final 2-norm of the output error, and faster convergence. Additionally, the optimisation

based norm-optimal ILCs performance is not that dependent on the specific controller tuning and could thus be easier to implement due to its simpler tuning. Note, however, that for all of these controllers the unrealistic assumption of knowing the exact grid fault characteristics is made, including when the fault is going to occur. For this reason, causal constrained ILC algorithms will also be constructed in section 4-6, where the optimal feedforward signal is only added after the fault has occurred.

4-5 Constrained controllers on high-fidelity model results

In the previous section the non-causal constrained ILC algorithms on the low-fidelity wind turbine model have been presented. These controllers will in this section also be applied on the high-fidelity wind turbine model.

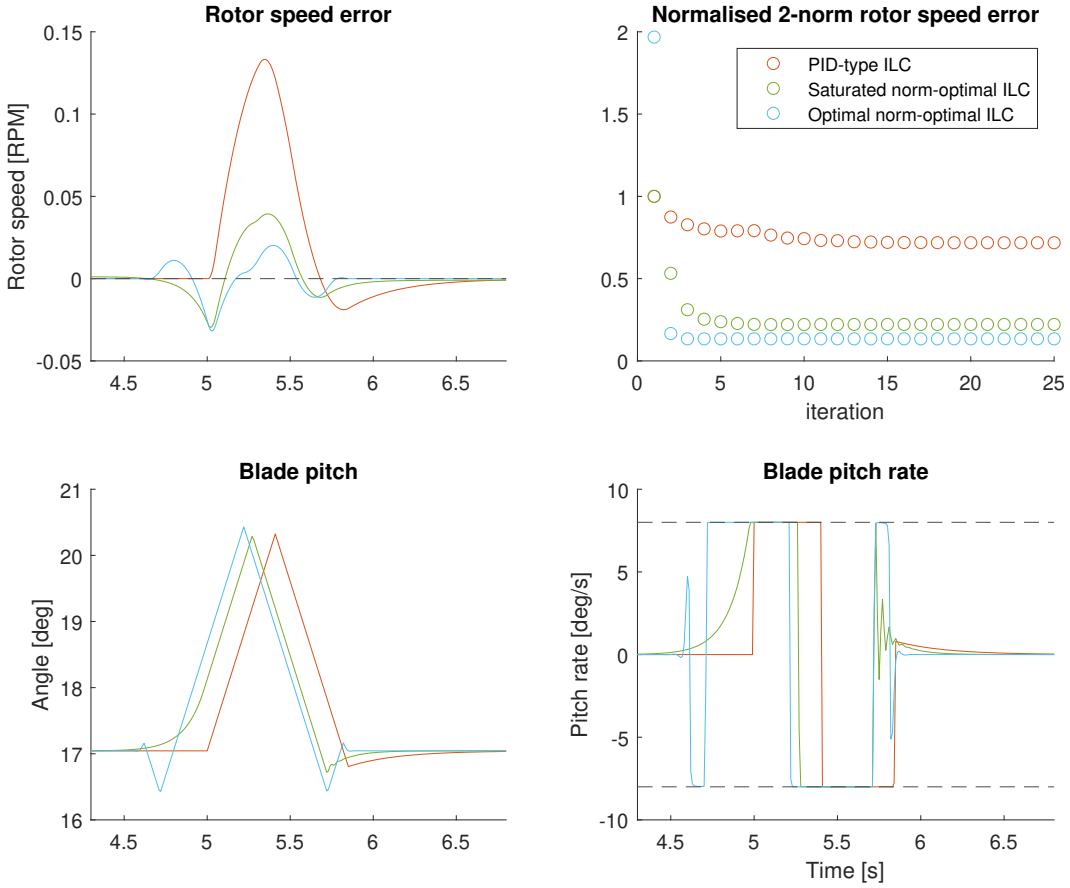


Figure 4-10: Comparison of the constrained non-causal PID-type ILC with the non-causal saturation based- and optimisation based constrained norm-optimal ILC, denoted as 'Saturation norm-optimal ILC' and 'Optimal norm-optimal ILC' respectively. This all on the low-fidelity wind turbine model. Shown are the rotor speed error, blade pitch and blade pitch rate of the final iterations. Also, the progression of the 2-norm of the output error is shown. Visible in the plots is that both controller fully utilise the actuator limits. A major difference is that the norm-optimal ILC already starts decreasing the rotor speed before the grid fault has started, resulting in a lower maximum rotor speed error compared to the PID-type ILC.

4-5-1 Constrained PID-type ILC on high-fidelity model

In the previous subsections, constrained ILC algorithms on the low-fidelity model were presented, where the norm-optimal controller showed far superior performance compared to the PID-type ILC. This was, however, done on the low-fidelity model, and is validated on the high-fidelity model in the following few subsections. Due to a higher computational effort for each simulated iteration, most of the results presented here are for a lower number of iterations. That said, the data should still be rich enough to provide sufficient insight into the performance of each individual ILC.

The first controller tested is the constrained PID-type ILC. The tuning of this controller can be seen in Table 4-6. The best found controller tuning uses $K_\mathrm{p} = 0.15$, $K_\mathrm{i} = 0$, $K_\mathrm{d} = 0.01$ and a cut-off frequency of 2 Hz for the low-pass filter that is applied after the output of the ILC algorithm. This configuration yields a reduction of the 2-norm to 0.906 after ten iterations. The reason that the 2-norm of the output error starts at a value unequal to one is because the ILC starts with constant initial condition and not the input used by the baseline controller.

The time domain results of this controller can be seen in Figure 4-11. From the results one can see that there remain quite some oscillations in the output of the rotor speed. However, the PID-type ILC manages to reduce the 2-norm of the rotor speed error to around 0.85 times that of the baseline controller.

K_p	K_i	K_d	filter cut-off	mean 2-norm	max 2-norm	final iter 2-norm
0.05	0.0001	0.005	10	0.9728	1.0698	0.9073
0.05	0	0.005	10	1.0521	1.0698	1.0160
0.15	0	0.010	10	1.0296	1.0779	0.9219
0.05	0	0.005	30	1.0909	1.1302	1.1302
0.15	0	0.010	5	0.9794	1.0698	0.9054
0.15	0	0.010	2	0.9696	1.0698	0.9063

Table 4-6: Tuning of the non-causal constrained PID-type ILC on high-fidelity wind turbine model. The filter cut-off is the cut off frequency for a low pass filter that is applied directly on the output of the ILC algorithm. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm and a low max 2-norm.

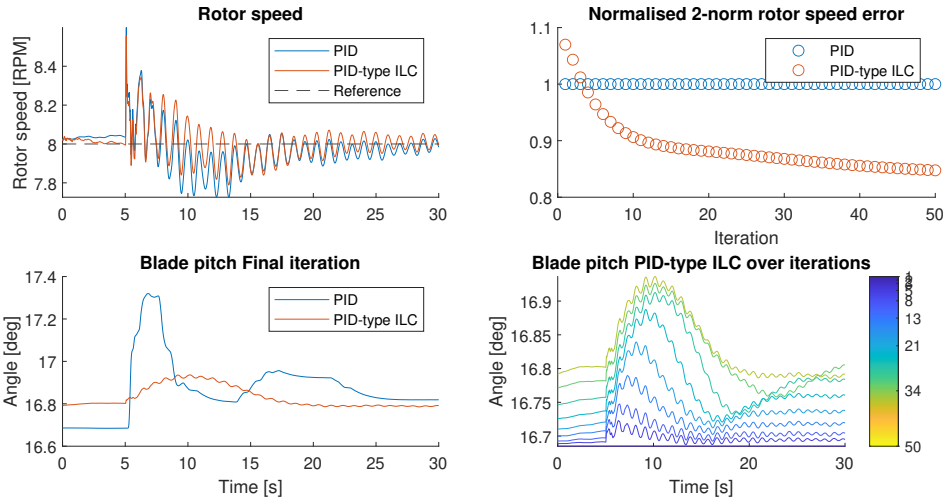


Figure 4-11: Non-causal constrained PID-type ILC compared with the baseline PID on the high-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

4-5-2 Constrained norm-optimal ILC using saturation on high-fidelity model

In addition to the PID-type ILC, also the constrained norm-optimal ILC using saturation is tested on the high-fidelity model. Again, the controller structure is equal to that used on the low-fidelity wind turbine model. The tuning results of this controller can be seen in Table 4-7. From the tuning process, one can see that the best performance was with $W_e = 1$, $W_f = 1$ and $W_{\Delta f} = 0$. Somewhat counter intuitive, for lower penalties on the feedforward input signal, the controller performs worse. A possible explanation is that the lower penalties in combination with the external saturation block drastically changes the form of the feedforward input, which in terms leads to negative effects for the 2-norm of the output error.

In Figure 4-12 one can see the time-domain results of this controller. One can see that the controller successfully decreases the magnitude of the output error oscillations, but does not completely remove them. There is, however, an observable reduction of the 2-norm to 0.82 times that of the baseline controller.

W_e	W_f	$W_{\Delta f}$	mean 2-norm	max 2-norm	final iter 2-norm
10	0.01	0	6.5646	14.900	14.900
1	0.01	0	2.5069	5.9973	5.6273
1	0.1	0	0.9225	1.0000	0.8707
1	1	0	0.8703	1.0000	0.8288
1	10	0	0.9702	1.0000	0.9620

Table 4-7: Tuning of the non-causal constrained norm-optimal ILC on high-fidelity wind turbine model with the objective to reduce the rotor speed error. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one. For low values of W_f the controller becomes unstable, which might be a result of the saturation-based constraint implementation which can significantly alter the shape of the learnt feedforward signal outside the controller.

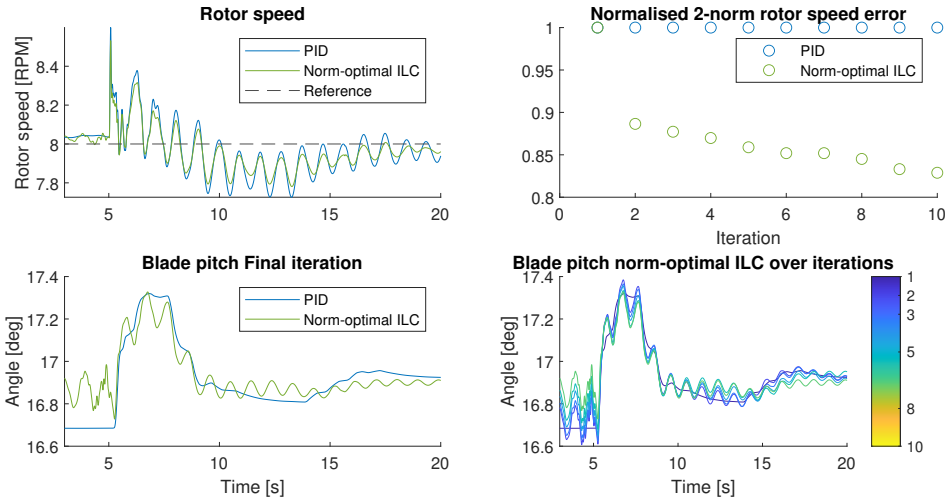


Figure 4-12: Non-causal constrained norm-optimal ILC compared with the baseline PID on the high-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

4-5-3 Constrained norm-optimal ILC using optimisation on high-fidelity model

The optimisation based constrained norm-optimal ILC will not be presented in this section, but is instead presented in Section H-6. This has been done since the optimisation step is otherwise too big to fit into memory, and sometimes times-out using a high-performance virtual machine with multiple Intel Xeon Platinum 8268 CPU's with 48 threads, and 24.0 GB for each client. Some effort has been made to reduce the control frequency, but with moderate success. Therefore, only the more relevant causal version is presented in Section H-6.

4-5-4 Comparison and conclusion of constrained ILC on high-fidelity model

In this section, non-causal constrained PID-type ILC and norm-optimal ILC are compared and a conclusion is made. First of all, due to technical limitations the optimisation based constrained norm-optimal ILC has not been presented in this section, ruling this controller out as a recommendation. This is due to the sheer magnitude of the optimisation problem that needs to be solved to calculate the optimal feedforward signal. It was already concluded that if this solution is truly desirable, more efficient solvers or perhaps other optimisation methods will have to be investigated further. That said, this causal variant of the optimisation based constrained norm-optimal ILC is more attractive a solution for this thesis, and as such the non-causal optimisation based constrained norm-optimal ILC is not recommended to be used for this use case.

The two successfully implemented controllers, the constrained PID-type ILC and saturation

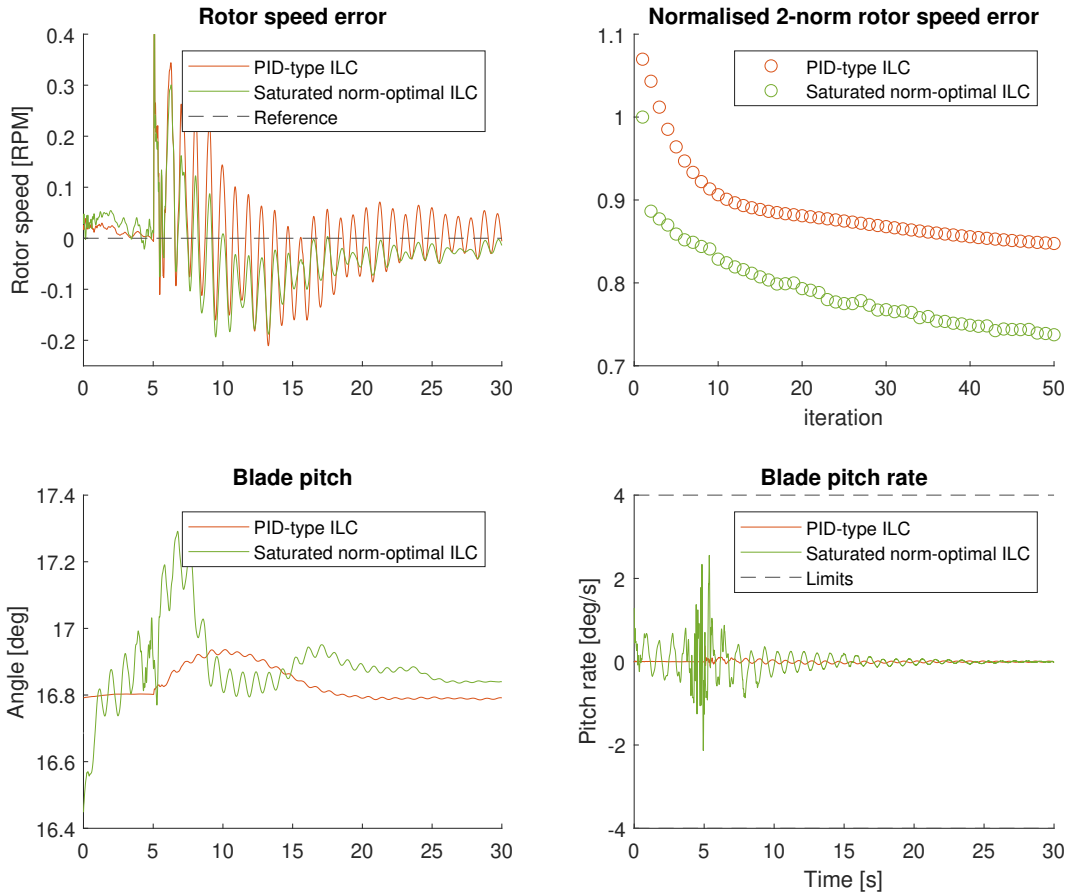


Figure 4-13: Comparison of the non-causal constrained PID-type ILC with the non-causal saturation based constrained norm-optimal ILC on the high-fidelity wind turbine model. Shown are the rotor speed error, blade pitch and blade pitch rate of the final iterations. Also, the progression of the 2-norm of the output error is shown. Visible in the plots is that both controller fully utilise the actuator limits. A major difference is that the norm-optimal ILC already starts decreasing the rotor speed before the grid fault has started, resulting in a lower maximum rotor speed error compared to the PID-type ILC.

based constrained norm-optimal ILC are compared in Figure 4-13. From this time-domain comparison, one can see that both controllers use significantly different blade pitch signals. This also results in a large difference between the resulting rotor speed errors. The rotor speed error of both controllers has large amounts of oscillations present, but the magnitude of these oscillations is smaller for the norm-optimal ILC, and after approximately 25 seconds almost completely gone. To do this, the norm-optimal ILC also employs a blade pitch signal that changes much more compared to the PID-type ILC.

From the results, one can see that the 2-norm of the output error is lower for the norm-optimal ILC compared to the PID-type ILC. That said, both ILC algorithms have large transients in the 2-norm and do not settle to a steady-state value even after 50 iterations. This in comparison to the non-causal constrained ILC algorithms on the low-fidelity model, which had a steady 2-norm of the output error within ten iterations or less. This difference compared to the low-fidelity model can most likely be attributed to the fact that the high-fidelity model has a limited bandwidth of the pitch actuator compared to that of the low-fidelity model and that there are more nonlinearities present that conflict with the linearity assumptions used in the ILC models.

Overall, one can conclude that the saturation based constrained norm-optimal ILC is the better ILC algorithm when used on the high-fidelity wind turbine model. Using this controller, a reduction of the output error oscillations can be observed 20 to 25 seconds after the grid fault. Additionally, this controller has the quickest converging and lowest 2-norm of the output error. Even though the 2-norm of the output error does not settle as quickly as was the case for the low-fidelity model, the norm-optimal ILC still provides a 10% reduction of the 2-norm after the first iteration and a 25% reduction after 50 iterations. However, for this approach, it is best to gather the optimal feedforward signals using simulations, and not just using real-life data due to the relatively slow convergence. To further improve these results, causal constrained ILC algorithms are tested in the next section. These controllers will only be activated after a grid fault has actually started where, assuming that the form of the grid fault is known, causality is regained for the controllers.

4-6 Constrained controllers with causal learning

In this section, the constrained controller is extended to also be causal with respect to the grid fault. This means that the (supplemental) ILC feedforward signal will only be introduced into the system after the fault has occurred. This section should give more information about how much real-life performance could be expected to be gained compared to the baseline controllers. In this section the comparison and conclusions is shown for both the low-fidelity and high-fidelity models, where the tuning and some additional time-domain results are shown in Appendix H.

4-6-1 Comparison and conclusion of causal constrained ILC on low-fidelity model

In Figure 4-14 a comparison between the causal constrained PID-type ILC, the causal saturation based constrained norm-optimal ILC and the causal optimisation based constrained norm-optimal ILC applied on the low-fidelity wind turbine model can be seen. A first obser-

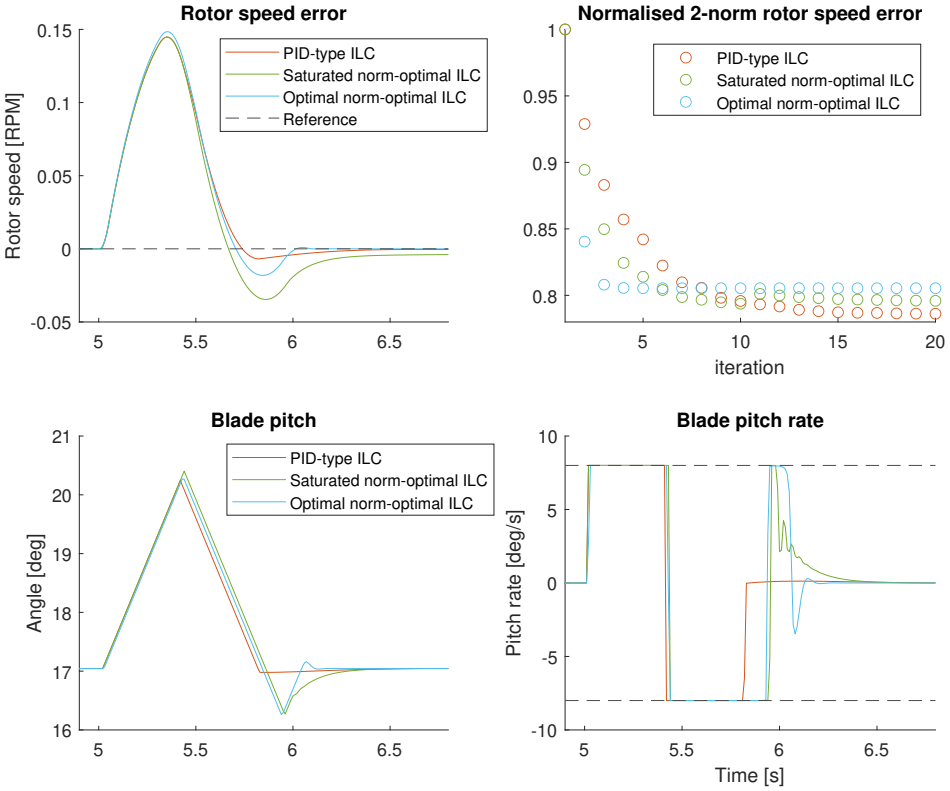


Figure 4-14: Comparison of the causal constrained PID-type ILC with the causal saturation based- and optimisation based constrained norm-optimal ILC, denoted as 'Saturation norm-optimal ILC' and 'Optimal norm-optimal ILC' respectively. This all on the low-fidelity wind turbine model. Shown are the rotor speed error, blade pitch and blade pitch rate of the final iterations. Also, the progression of the 2-norm of the output error is shown. Visible in the plots is that both controllers fully utilise the actuator limits. A major difference is that the norm-optimal ILC already starts decreasing the rotor speed before the grid fault has started, resulting in a lower maximum rotor speed error compared to the PID-type ILC.

vation of these results is that the actual improvement over the baseline is significantly smaller compared to the non-causal controllers. This is also to be expected, since no preemptive compensation of the rotor speed can be applied before the grid fault has occurred. That said, still all three controllers managed to decrease the 2-norm of the output error to around 0.8 times that of the baseline controller.

The biggest improvement for causal ILC algorithms compared to the baseline controller lies in higher utilisation of the input combined with more accurate placement of the switching points of the controller input. This combined increases the speed with which the rotor speed error recovers to zero. When comparing the actual rotor speed error of the three ILC algorithms, one can see that the PID-type ILC has high amounts of damping, resulting in only a small overshoot after the rotor speed error first crosses zero. Both of the norm-optimal ILCs have higher overshoots in the rotor speed error, where the saturation based norm-optimal ILC struggles to fully recover to a zero rotor speed error after the grid fault.

When looking at the difference between the blade pitch signals of the ILC algorithms, the PID-type ILC is the first controller to start reducing the blade pitch after the fault, with the optimisation based norm-optimal controller the last. This is also reflected in the optimisation based norm-optimal ILC having the largest positive rotor speed error after the fault. When looking at the blade pitch signal after around 5.8 seconds, the PID-type ILC quickly returns to the steady-state blade pitch signal, where the two norm-optimal ILCs have some sort of overshoot in the blade pitch signal. Looking from the outside, it is not clear why the norm-optimal ILC has this overshoot in the blade pitch signal. This also happens since the PID-type ILC has a much smaller overshoot in the rotor speed error compared to the norm-optimal ILCs.

Based on the comparison made in this section, one can conclude that the PID-type ILC manages to yield the lowest 2-norm of the output error and is thus the preferred method for situations where the absolute lowest 2-norm is wanted. That said, the optimisation based norm-optimal ILC has a faster convergence of the 2-norm to its final value, meaning that it is probably more suitable when used in combination with hard to acquire real-world data. This does come at the cost of absolute performance after some number of iterations.

4-6-2 Comparison and conclusion of causal constrained ILC on high-fidelity model

In Figure 4-15 the comparison between the three different causal constrained ILC algorithms can be seen. Immediately clear when looking at the 2-norm of the output error is that the PID-type controller underperforms compared to the norm-optimal ILCs. The norm-optimal ILCs have a significantly lower final 2-norm with in addition also a fast decrease of the 2-norm in the first few iterations.

Comparing the 2-norms of the norm-optimal ILCs, one can see that the optimal norm-optimal ILC manages to keep on decreasing the 2-norm of the output error, whilst the saturated norm-optimal ILC does not do this. This leads to favourable performance for the optimisation based constrained norm-optimal ILC.

Looking at the resulting rotor speed errors, it is also clear why the different 2-norms of the output error occur. The PID-type ILC has relatively speaking the higher magnitude of oscillations, with second the saturation based norm-optimal ILC and the best performing

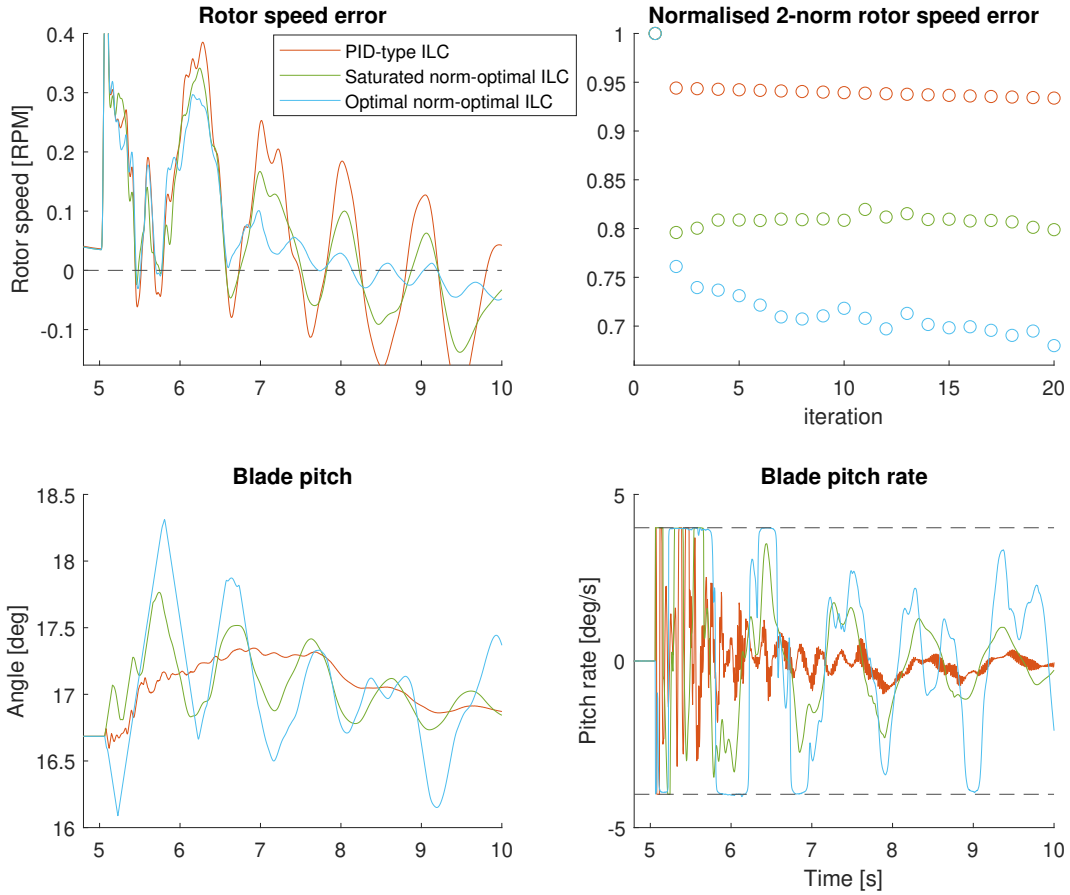


Figure 4-15: Comparison of the causal constrained PID-type ILC with the causal saturation based- and optimisation based constrained norm-optimal ILC, denoted as 'Saturation norm-optimal ILC' and 'Optimal norm-optimal ILC' respectively. This all on the high-fidelity wind turbine model. Shown are the rotor speed error, blade pitch and blade pitch rate of the final iterations. Also, the progression of the 2-norm of the output error is shown.

the optimisation based norm-optimal ILC. Especially when looking from around six to eight seconds, it is clear how the different ILC algorithms manage to better reduce the magnitude of the rotor speed oscillations. Also, when one starts to look at the blade pitch signal, this effect is clear, but then in reverse. The optimisation based norm-optimal ILC has the largest swing in blade pitch, with the PID-type ILC having the least amount of movement in the blade pitch signal. This phenomenon can also be seen well between the 5.5 and 7 second marks in Figure 4-16, where a detailed view of the blade pitch rate after the grid fault can be seen.

Overall, the norm-optimal ILC have better performance with regard to the 2-norm of the output error, with especially the optimisation based ILC performing best. In addition, the PID-type ILC is quite tricky to tune properly as has been shown in section H-4, whilst the norm-optimal ILCs are relatively easy to tune. One disadvantage that pairs with the superior performance of the optimisation based norm-optimal ILC is the additional computation time, making it for these experiments around three times slower to run a single iteration. That said, implementation of the optimisation based norm-optimal ILC within the nominal controller

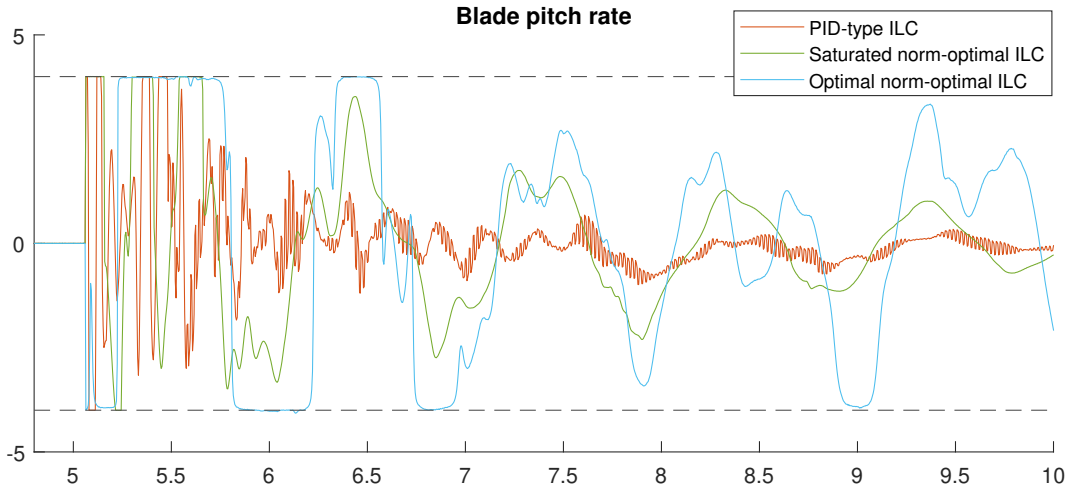


Figure 4-16: Detailed view of the blade pitch rate, comparison of the causal constrained PID-type ILC with the causal saturation based- and optimisation based constrained norm-optimal ILC, denoted as 'Saturation norm-optimal ILC' and 'Optimal norm-optimal ILC' respectively. Visible is that the optimisation based norm-optimal ILC uses less switching compared to the other ILC methods shown and additionally uses more of the available blade pitch rate for a longer period of time.

should be relatively easy, since in its current form it already only supplements an additional feedforward signal for seven seconds after the fault, after which the baseline controller takes over. This is in line with current field deployed grid fault controllers that switch to a separate controller only for a short period of time after a grid fault.

4-7 Grid fault controller further testing

Up to this point of this chapter, grid fault controllers with the objective to reduce the rotor speed error and its associated 2-norm have been shown. However, what has not been discussed is the effect on the rest of the turbine or what the effects are of other objective functions. Therefore, in this section a more detailed overview of the effects of the controllers on mechanical loading is discussed. To limit the scope somewhat, a more detailed analysis on the tower-bottom moment is made. Here, the tower-bottom moment is the moment at the lowest point of the tower structure, just above the water line, with the moment perpendicular to the rotor blades. This channel gets loaded significantly after a grid fault, and has been pointed out by Siemens Gamesa engineers to be an interesting channel to minimise loading where in the best case scenario these load reductions could lead to a lighter tower structure or a longer lifespan of the tower.

4-7-1 Effects of tower-bottom moment using rotor speed objective on high-fidelity model with causal learning

for this first subsection, a deeper dive is made in how the tower-bottom moment is affected for the causal constrained ILC algorithms on the high-fidelity model with the objective of

reducing the rotor speed error. Or, in other words, how the final controllers presented in Section 4-6-2 effect the mechanical loading of the wind turbine structure itself. In Figure 4-17 one can see the 2-norm of the rotor speed error, but also from the tower-bottom moment error and the actual tower-bottom moment. The first observation to be made is that for all ILC algorithms the 2-norm of the tower-bottom moment is below one for every iteration, indicating that at each time point the energy in the tower-bottom moment has been reduced compared to the baseline PID controller. This is good news, as this means that the ILC algorithms found during this thesis at least do not increase the mechanical loading of the wind turbine structure on this important channel.

Although all of the ILC algorithms lower the 2-norm of the tower-bottom moment error, the effectiveness of each controller to do so is different. For the PID-type ILC the 2-norm of the tower-bottom moment stays at an equal level, as is the 2-norm of the linked rotor speed error. However, when looking at the optimisation based norm-optimal ILC, it is clear that the reduction in tower-bottom moment error might not remain for higher number of iterations, as indicated by the increase of its 2-norm after the fifth iteration up to the 20th iteration.

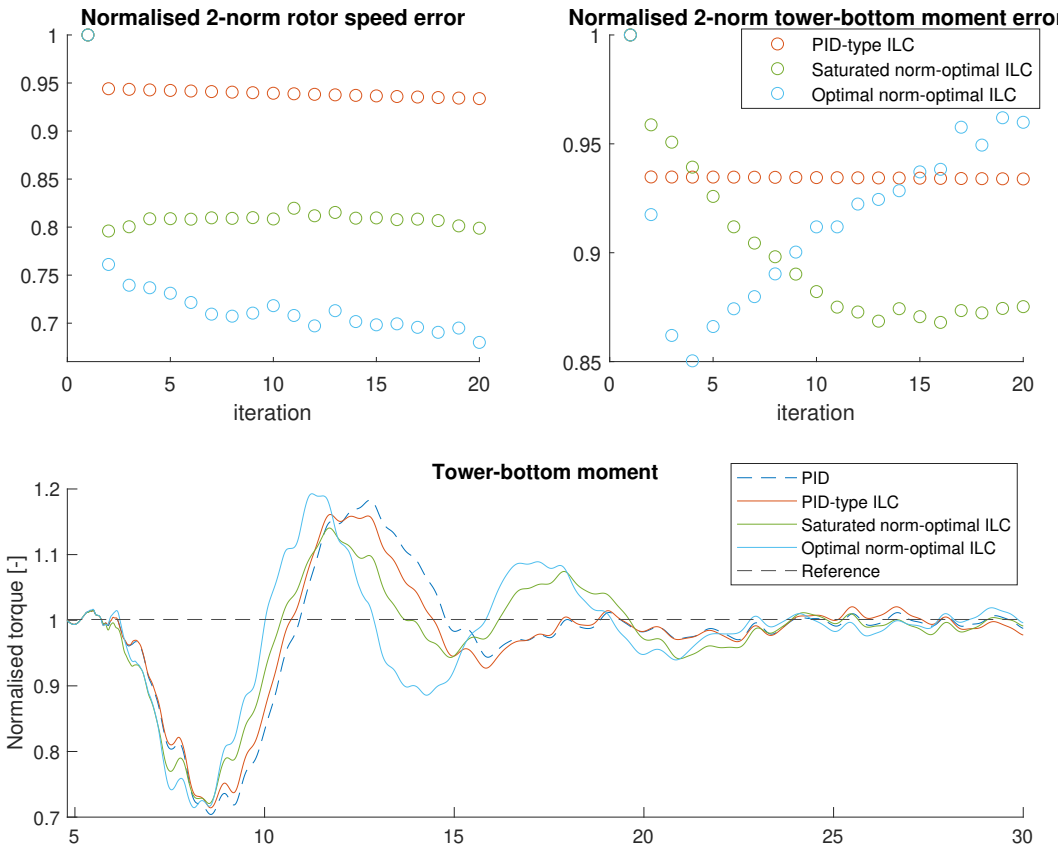


Figure 4-17: Causal constrained ILC on high-fidelity model with objective to reduce the rotor speed error. Shown are how the objective of reducing the rotor speed error effects the tower-bottom moment, an important mechanical loading channel. Shown are the progression of the 2-norm of said channels, and a plot of the tower-bottom moment itself. From the plot one can see that though reducing the tower-bottom moment is not the objective, it is still accomplished up to some degree.

4-7-2 Dedicated tower-bottom moment objective on high-fidelity model with non-causal learning

In the previous subsection, the tower-bottom moment norm was shown for the ILC algorithms with the rotor speed objective function. However, as has been touched upon multiple times in the report, the ILC algorithms can work with different types of objective functions. Therefore, here a dedicated ILC algorithms with the objective of reducing the tower-bottom moment error is presented.

As is also the case for the rotor speed error objective function, the optimisation based norm-optimal ILC does only work for short periods of time due to technical limitations, see Section 4-5, and as such will only be included for the causal variant. A overview of how the controllers are tuned can be seen in Tables 4-8 and 4-9. Based on this tuning, the best performing controller tunings have compared, as can be seen in Figure 4-18. This comparison gives some interesting and perhaps unexpected results based on the knowledge gained on the rotor speed error objective function. For the tower-bottom moment objective function, the PID-type ILC has the lowest 2-norm during every iteration. Also, when investigating the actual tower-bottom moment signal, a clear difference between the two ILC algorithms is visible. The PID-type ILC does not drastically change the tower-bottom moment error before the grid fault has occurred, but the saturated norm-optimal ILC actually decreases the performance with respect to the objective function before the grid fault has occurred. Also, after the grid fault has started, the norm-optimal ILC has larger peak errors at the tower-bottom moment compared to the PID-type ILC.

The finding of this objective function is that the PID-type ILC performs better than the saturation based constrained norm-optimal ILC. This conclusion is opposite to that of the rotor speed error objective function, where the norm-optimal ILC manages to yield a lower final 2-norm. That said, the norm-optimal ILC does yield a lower 2-norm on the rotor speed error compared to the PID-type ILC, but this is not the objective function. Therefore, the data suggest that based on the objective function, either the non-causal constrained PID-type ILC or the non-causal saturation based constrained norm-optimal ILC works best. This

K_p	K_i	K_d	filter cut-off	mean 2-norm	max 2-norm	final iter 2-norm
$1{\cdot}10^{-8}$	0	$5{\cdot}10^{-6}$	1.5	0.80184	1	0.65252
$1{\cdot}10^{-7}$	0	$5{\cdot}10^{-6}$	1.5	0.80019	1	0.64921
$1{\cdot}10^{-6}$	0	$5{\cdot}10^{-6}$	1.5	0.78542	1	0.61971
$1{\cdot}10^{-5}$	0	$5{\cdot}10^{-6}$	1.5	0.85362	1	0.78420
$1{\cdot}10^{-6}$	0	$1{\cdot}10^{-6}$	1.5	0.83605	1	0.69609
$1{\cdot}10^{-6}$	0	$1{\cdot}10^{-7}$	1.5	0.85330	1	0.72635
$1{\cdot}10^{-6}$	$1{\cdot}10^{-10}$	$5{\cdot}10^{-6}$	1.5	0.78531	1	0.61925
$1{\cdot}10^{-6}$	$1{\cdot}10^{-8}$	$5{\cdot}10^{-6}$	1.5	0.81747	1	0.66601

Table 4-8: Tuning of the non-causal constrained PID-type ILC on the high-fidelity wind turbine model with the objective to reduce the tower-bottom moment error. The filter cut-off corresponds to the cut-off frequency off the low-pass filter after the ILC algorithms output, where Inf corresponds to a disabled cut-off filter. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

W_{e}	W_{f}	$W_{\Delta \mathrm{f}}$	mean 2-norm	max 2-norm	final iter 2-norm
$1 \cdot 10^{1}$	1	0	1.2980	1.5607	1.5607
$1 \cdot 10^{-2}$	1	0	1.0824	1.1648	1.1648
$1 \cdot 10^{-5}$	1	0	0.9612	1.0000	0.9251
$1 \cdot 10^{-6}$	1	0	0.9787	1.0000	0.9594
$1 \cdot 10^{-7}$	1	0	1.7974	2.5949	2.5949

Table 4-9: Tuning of the non-causal saturation based constrained norm-optimal ILC on the high-fidelity wind turbine model with the objective to reduce the tower-bottom moment error. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

finding then also makes generalising the conclusion harder, as the results are mixed between the two tested objective functions.

Although the final 2-norm of the PID-type ILC is lower for this test, it does not mean that it could not be improved any further. The author suspects that changing the baseline feedback controller that is operational within the norm-optimal ILC could help its performance. In the here presented implementation, the same baseline feedback controller has been used, but as one might remember, the baseline controller is made to reduce the rotor overspeeding, and not to reduce the tower-bottom moment. Perhaps if the feedback controller used within the norm-optimal ILC were to be redesigned to prioritise the reduction of the tower-bottom moment error, the results would shift in favour of the norm-optimal ILC. The fact that the norm-optimal controller has the integrated feedback controller aimed at reducing the rotor overspeeding possibly also explains why the norm-optimal ILC has a lower 2-norm on the rotor speed error.

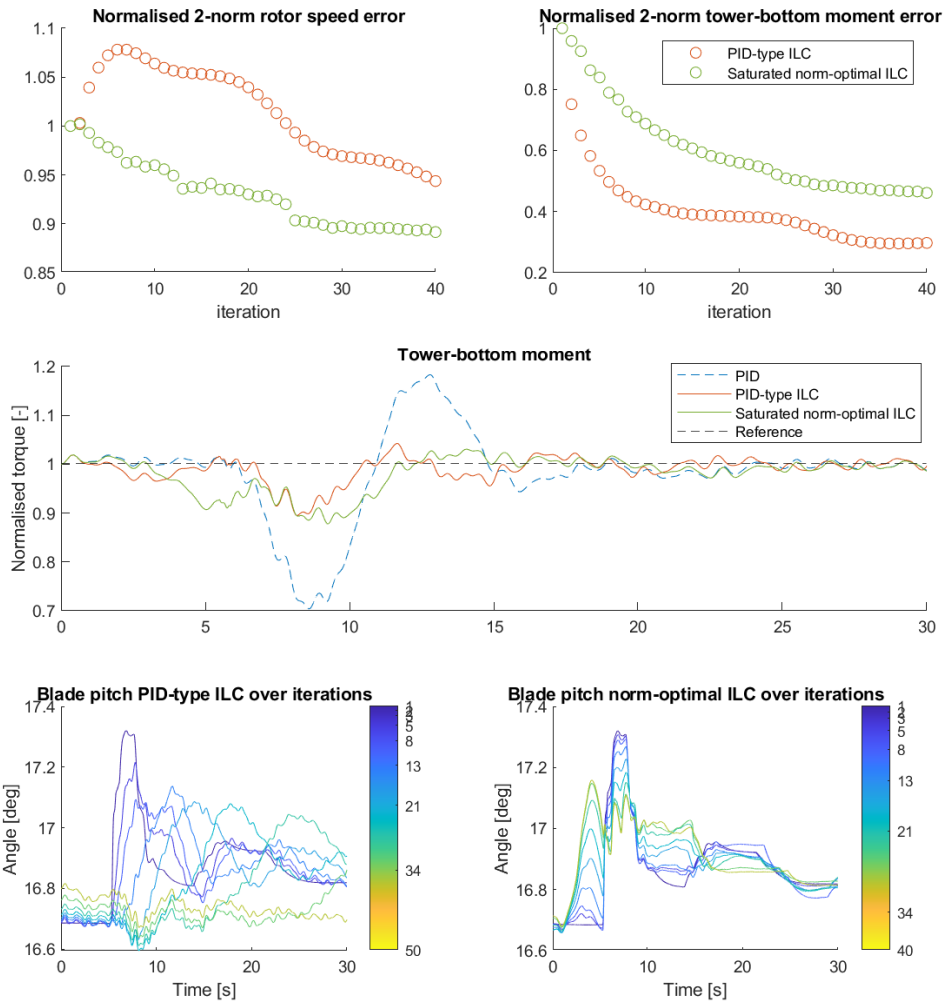


Figure 4-18: Non-causal constrained ILC on high-fidelity wind turbine model with objective to reduce tower-bottom moment error on the PID-type ILC and the saturation based norm-optimal ILC. Shown are the progression of the 2-norm for the rotor speed error, not the main objective, the tower-bottom moment error, main objective, and the actual tower-bottom moment. Visible in the results is that the norm-optimal ILC increases the tower-bottom moment error before the grid fault which partially explains the higher 2-norm w.r.t. the PID-type ILC.

4-7-3 Dedicated tower-bottom moment objective on high-fidelity model with causal learning

To expand the applicability of the controller designed in above subsection, a causal variant has also been designed, which only starts the learning after the grid fault has started. For this variant, the PID-type ILC and the saturation based norm-optimal ILC have been designed. Unfortunately, the optimisation based norm-optimal ILC has not yielded any positive results, as can be seen in Table 4-10. In this table, one can see that for low values of W_e the ILC

W_e	W_f	$W_{\Delta f}$	mean 2-norm	max 2-norm	final iter 2-norm
$1{\cdot}10^{-8}$	1	0	1	1.0000	1.0000
$1{\cdot}10^{-6}$	1	0	1	1.0000	1.0000
$1{\cdot}10^{-5}$	1	0	1.1802	1.3234	1.3234
$1{\cdot}10^{-4}$	1	0	1.1096	1.2193	1.2193
1	1	0	1.2021	1.4041	1.4041

Table 4-10: Tuning of the causal optimisation based constrained norm-optimal ILC on the high-fidelity wind turbine model with the objective to reduce the tower-bottom moment error. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Visible from the results is that low values of W_e lead to no additional feedforward input, and higher values of W_e lead to solutions with a higher 2-norm of the tower-bottom moment error. As such, the controller in this configuration is unable to fulfil its objective of reducing the 2-norm.

algorithm did not apply any feedforward signal due to the small weight on the error. However, if the value of W_e was increased, the ILC algorithm was unable to lower the 2-norm of the tower-bottom moment and instead only increase it. No middle ground has been found in which the controller worked. Why these results are as such is unclear, as the saturation based norm-optimal ILC that shares a great deal of the underlying mechanisms and code did work as intended. Also here the suspicion is that the feedback controller present inside the optimisation based norm-optimal ILC should be redesigned to work on lowering the tower-bottom moment.

Due to the issues with the causal optimisation based constrained norm-optimal ILC, this controller will not be used for the remainder of this section. This then leaves the PID-type ILC and the saturation based norm-optimal ILCs. The tuning of these two controllers can be seen in Tables 4-11 and 4-12. As one can see, for both of these controllers there are controller settings for which the 2-norm of the output error decreases indicating that these controllers are successful in fulfilling their objective functions.

K_p	K_i	K_d	filter cut-off	mean 2-norm	max 2-norm	final iter 2-norm
$1{\cdot}10^{-7}$	0	$5{\cdot}10^{-6}$	1.5	0.79897	1	0.64737
$1{\cdot}10^{-6}$	0	$5{\cdot}10^{-6}$	1.5	0.78464	1	0.61877
$1{\cdot}10^{-5}$	0	$5{\cdot}10^{-6}$	1.5	0.89041	1	0.78081
$1{\cdot}10^{-6}$	0	$1{\cdot}10^{-6}$	1.5	0.90613	1	0.81226
$1{\cdot}10^{-6}$	0	$1{\cdot}10^{-5}$	1.5	0.84694	1	0.69387
$1{\cdot}10^{-6}$	$1{\cdot}10^{-10}$	$5{\cdot}10^{-6}$	1.5	0.86768	1	0.73536

Table 4-11: Tuning of the causal constrained PID-type ILC on the high-fidelity wind turbine model with the objective to reduce the tower-bottom moment error. The filter cut-off corresponds to the cut-off frequency off the low-pass filter after the ILC algorithms output, where Inf corresponds to a disabled cut-off filter. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

W_e	W_f	$W_{\Delta f}$	mean 2-norm	max 2-norm	final iter 2-norm
$1{\cdot}10^{-10}$	1	0	1.0062"	1.0125	1.01250
$1{\cdot}10^{-12}$	1	0	0.98369	1.0000	0.98103
$1{\cdot}10^{-14}$	1	0	0.99992	1.0000	0.99983
$1{\cdot}10^{-8}$	1	0	1.0831"	1.1662	1.16620
$1{\cdot}10^{-6}$	1	0	1.1469"	1.2938	1.29380
$1{\cdot}10^{-11}$	1	0	0.93372	1.0000	0.89900
$1{\cdot}10^{-13}$	1	0	0.99868	1.0000	0.99839

Table 4-12: Tuning of the causal saturation based constrained norm-optimal ILC on the high-fidelity wind turbine model with the objective to reduce the tower-bottom moment error. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

In Figure 4-19 one can see the comparison between the two causal ILC algorithms with the objective of reducing the tower-bottom moment error. A first observation that can be made from the results is that the PID-type ILC is significantly more successful in reducing the tower-bottom moment error, with a final 2-norm of around 0.3. This in contrast to the saturation based norm-optimal ILC that only manages to get a final 2-norm of around 0.9. This then also means that even though the PID-type ILC was the lowest performing ILC algorithm that has been tested on the rotor speed error objective function, the algorithm itself can perform significantly better for the right objective function.

As was the case for the non-causal variant, also here the PID-type ILC significantly alters the shape of the blade pitch input signal. This while the final blade pitch signal from the norm-optimal ILCs resembles more that of the original controller, as can also be seen in Figure 4-19. Here again, the suspicion is that if the feedback controller within the norm-optimal ILC were to be reconfigured to focus more on the tower-bottom moment error, then the norm-optimal ILC would perform better. This theory should, however, still be tested to either confirm or deny its premises. However, this difference does indicate a more flexible structure within the PID-type ILC which is not bounded by a, possibly externally designed, feedback controller.

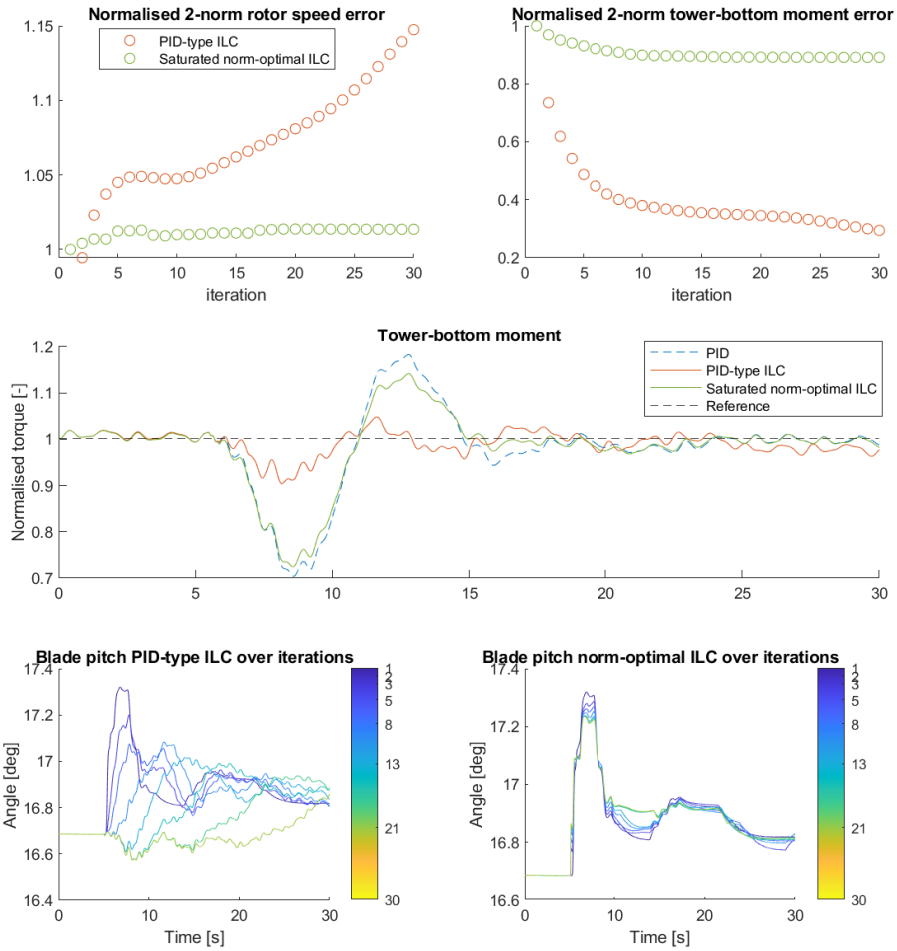


Figure 4-19: Causal constrained ILC on high-fidelity wind turbine model with objective to reduce tower-bottom moment error on the PID-type ILC and the saturation based norm-optimal ILC. Shown are the progression of the 2-norm for the rotor speed error, not the main objective, the tower-bottom moment error, main objective, and the actual tower-bottom moment. Visible in the results is that the norm-optimal ILC increases the tower-bottom moment error before the grid fault which partially explains the higher 2-norm w.r.t. the PID-type ILC.

4-8 Conclusion and reflection

The non-causal unconstrained ILC algorithms have shown that it is possible to get a (near) zero output error with a limited number of iterations for both the PID-type ILC and norm-optimal ILC on a nonlinear system. Expanding on these results, causal unconstrained ILC algorithms have been made where learning only starts after the fault has occurred. The results of these controllers have shown the first noticeable difference between the PID-type and the norm-optimal ILC. The norm-optimal ILC starts with a large spike in the blade pitch that quickly causes the output error to reduce to zero. However, the PID-type ILC does not do

this, leading to a slower transient of the output error to zero. The norm-optimal ILC than also has a significantly lower 2-norm compared to the PID-type ILC. This difference can be contributed to the data-driven system knowledge that is used in the norm-optimal ILC, and not in the PID-type ILC.

The non-causal and causal unconstrained ILC algorithms have also been implemented on the high-fidelity wind turbine model. The results have shown that in general the norm-optimal ILC with especially the optimisation based norm-optimal ILC performs best. The PID-type ILC did manage to compare on the causal constrained test using the low-fidelity model, but once the switch to the high-fidelity model was made it performed much worse. The PID-type ILC has a higher final 2-norm, and additionally has proven to be quite challenging to tune. Another possible disadvantage of the PID-type ILC is the lack of feedback within the iteration. Only after a full iteration measurements are again processed and used to improve the output response, this in comparison to the norm-optimal ILCs which employ a combination of the baseline feedback controller with a supplemental learnt optimal feedforward signal. This setup of the norm-optimal ILC of still using the baseline feedback controller also makes it quite easy to actually implement, as all that needs to be done is to add a supplemental input signal after a grid fault has occurred.

When looking at ILC algorithms with the objective of reducing the mechanical loading in the high-fidelity model, the norm-optimal ILC suddenly does not perform best, but instead the PID-type ILC does. This discrepancy in conclusion compared to that made from the results of the ILC schemes with the objective of reducing the rotor speed is suspected to originate from the unchanged baseline controller that is used as a feedback controller within the norm-optimal ILC. Since the baseline controller of the high-fidelity model has been tuned to reduce rotor speed and not the mechanical loading channel considered here, it is likely that the objectives of this feedback controller react too aggressively on the actions of the feedforward signal that aims to accomplish a different objective. To, however, fully understand this behaviour more research will have to be done, where for now the PID-type ILC can still be used on the load reduction objective function.

The final conclusion and recommendation of this chapter is to use the causal optimisation based constrained norm-optimal ILC on the high-fidelity wind turbine model, or an actual wind turbine. Granted that the optimisation based norm-optimal ILC does take more time to run due to the additional optimisation scheme that needs to be solved, this controller does yield the lower final 2-norm of the output error and has fast convergence. If load reduction is the main objective, then the PID-type ILC can be used.

Chapter 5

Conclusion

The field of (offshore) wind turbine control has many challenges not only through internal interactions, but also from regulatory bodies. A prime example of this is the relatively new low-voltage ride-through (LVRT) requirements imposed by the Transmission System Operator (TSO). These LVRT requirements are established to ensure a stable electrical grid during short intermittent local failures within the electrical grid that can cause a near-instantaneous drop in the grid voltage. The LVRT regulation effects how the physical wind turbine structure shall react during grid faults, necessitating new control schemes to be developed. Currently, control schemes used for such grid faults are quite rudimentary, causing higher wear on the physical turbine structure.

During the LVRT event, a lower grid voltage is present, reducing the wind turbines' ability to 'slow down' the rotor by releasing energy to the grid, or more correctly a loss of counter torque. Due to this loss in torque output, the wind turbines rotor will start to accelerate, potentially triggering the overspeed limit if no sufficient control action is taken. The effects of the LVRT event are not limited to the rotor speed, but will also cause changes in the structural loading of the wind turbine.

Scientific contributions of the thesis. The two most important contributions of this thesis to the state of the art is applying ILC to wind turbines during a grid fault, and expanding the norm-optimal ILC to work with an optimisation framework that also takes input constraints into account using an estimate of the next combined feedforward and feedback input. This has been done whilst answering the following research question:

How can data-driven algorithm(s) be used to learn the optimal (feedforward) control signal to resume regular operation after a predefined low-voltage ride-through scenario?

With the following sub-questions:

1. Which data-driven optimal algorithm(s) can be used for low-voltage ride-through?
2. Which costs or signals should be taken into account in the design of a cost or objective function for the optimal controller?

3. What type of unconstrained algorithm or configuration achieves the best performance?
4. What is the performance impact of including input constraints to the optimisation problem for adhering to actuation limits?

First sub-question

The literature research has concluded that the industry still largely relies on (gain-scheduled) PI(D) controllers. How these controllers are precisely tuned is unclear, but usually with rotor overspeed prevention as the main goal of said controller. These controllers do not include any optimality and might leave much performance measures out of consideration. Academic solutions are more advanced, with examples such as MPC. Most of these controllers are, however, not strictly data-driven and require large amounts of system knowledge. Additionally, no feedforward grid fault controller was found even though the conditions of the grid fault are quite well known through the LVRT regulations. Using a feedforward controller in combination with the knowledge of how the grid fault will look like could bring large improvements over only using a traditional feedback controller. During literature research, Iterative Learning Control (ILC) has been found to be a possible candidate feedforward controller. These ILC algorithms learn an optimal feedforward signal based on iterations or repetition. As no literature has been found on ILC for wind turbine grid fault controllers, the primary focus of this thesis was making the ILC algorithms work for grid fault control. These ILC algorithms have been concluded to improve the response of wind turbines after a fault in the grid. This has been tested for multiple ILC algorithm variants, as is discussed for the third sub-question.

Second sub-question

The literature research has concluded that the state of the art grid fault control in industry and academia is primarily aimed at reducing the imbalance between the rotor torque and electrical torque output. However, this leaves out an important objective, namely, reducing the mechanical load on the wind turbine structure. Although the main focus of this thesis has also been to minimise rotor speed error, there was an interest in investigating the effects on the mechanical structure of the wind turbine. One of the main results of this thesis are the causal constrained ILC algorithms in the high-fidelity wind turbine model with the objective of reducing the rotor speed error. Although these controllers have only been tuned with the aim of lowering the 2-norm of the output error, it has been concluded that these controllers do not increase the tower-bottom moment, an important indicator of the mechanical loading after a grid fault. In fact, these controllers have even lowered the 2-norm of the tower-bottom moment after a grid fault, even though this was not the explicit objective. This then also shows that reducing the rotor speed error can help reduce other important output channels of the wind turbine. That said, tests have also been performed with only the tower-bottom moment as an objective function. Using the reduction of the mechanical loading as the main objective function has then also reduced the 2-norm of this output error even more than was the case when using the rotor speed objective function. Although the objective function of the tower-bottom moment reduced the mechanical load, it did increase the rotor speed error 2-norm. Thus, one should always carefully consider which objective function should be used. The results from the thesis indicate that using the rotor speed objective function does

not harm the mechanical loading and, in fact, improves upon it, but the reverse is not true. Therefore, either the rotor speed objective or some multi-objective cost functions should be among the top objective functions to be considered when using an ILC grid fault controller.

Third sub-question

During this thesis two important classes of ILC have been used, the PID-type ILC and the norm-optimal ILC. Both of these controllers have been tested in an unconstrained case, but only on the low-fidelity model. Results of a non-causal unconstrained ILC on the low-fidelity model have shown what the (theoretical) best-case performance of the ILC algorithms is. It has been shown that both the PID-type ILC and the norm-optimal ILC are capable of bringing the 2-norm of the output error to (near) zero within a limited number of iterations. This has proven that these ILC algorithms are, in fact, capable of controlling a nonlinear system to zero output error.

Expanding on the results of the non-causal unconstrained ILC, a causal version has been made. Here, learning has only started after the fault has occurred, preventing controllers from preemptively starting to compensate for the upcoming grid fault. Instead, controllers can only start to adjust the input signal after the grid fault is up to a certain level. This resulted in, as expected, higher output errors. However, this is the first time that a proper distinction between the PID-type ILC and norm-optimal ILC can be seen with the later performing better. Where the PID-type ILC algorithms' input signal is almost a carbon copy of that of the non-causal controller after the grid fault has occurred, the norm-optimal ILC algorithm is better able to handle this scenario. The causal norm-optimal ILC starts with an immediate spike in the blade pitch once its learning starts, causing the rotor speed error to effectively reduce to zero near instantaneously. This in term also results in a significantly lower final 2-norm of the output error for the norm-optimal ILC of only 0.043 times that of the baseline controller, where the PID-type ILC has a final 2-norm of 0.17 times that of the baseline controller. This means that both controllers have significant reductions of their respective 2-norms, but the causal unconstrained norm-optimal ILC performs significantly better in this regard.

Though results have shown that the causal unconstrained ILC algorithm is highly effective in reducing the 2-norm of the output error, these are not greatly effective controllers for any real-world applications as the actuator constraints have been exceeded multifold and as such can never be implemented on actual wind turbines. These tests have however shown the potential effectiveness of the ILC algorithms and show that a great potential is available within the ILC framework. That said, more effort and importance should be focused on the constrained and or causal variants, as is discussed for the fourth sub-question.

Fourth sub-question

As has quickly been discovered during this thesis, the inclusion of input constraint is fundamental in gaining representable results for the ILC algorithms. This mainly due to the significant input constraint violations of the unconstrained ILC algorithms, making them unsuitable for real-life wind turbines. To add constraints to the PID-type ILC a saturation scheme has been employed, which saturates the input, and the inputs first derivative. For

the norm-optimal ILC two versions have been used, the first uses saturation based constraint similar as in the constrained PID-type ILC. The second alterations is an optimisation based constrained norm-optimal ILC in which the objective function together with input constraints are input to an optimisation problem.

First results on the non-causal constrained ILC algorithms show some interesting results, especially on the low-fidelity wind turbine model. Using the low-fidelity wind turbine model the norm-optimal ILC algorithms start to adjust the blade pitch signal some time before the actual grid fault occurs, leading to a negative rotor speed error before the grid fault. This negative rotor speed error built up before the grid fault leads to a lower peak rotor speed error after the actual grid fault, and in fact to a lower 2-norm of the output error. This is in contrast to the PID-type ILC, which does not start adjusting the pitch signal before the fault has occurred. This difference is also understandable when one looks at how these ILC algorithms work. The PID-type ILC only looks at the grid fault at data from same time instance but from the previous iteration. This stands in contrast to the norm-optimal ILC, which uses the behaviour of an impulse response on the system to calculate what the effect of a specific input signal on the output over the whole time range. Using this information of the system, the norm-optimal ILC manages to preemptively correct for the upcoming grid fault and, as such, has significantly lower final 2-norm of the output error. Although this behaviour is clearly seen on the low-fidelity wind turbine model, it is less so observed for the high-fidelity model. Still, also on the high-fidelity model the norm-optimal ILC manages to yield a lower 2-norm of the output error.

To combine the results of the previous ILC algorithms, lastly a causal constrained ILC has been presented. These controllers adhere to the input constraints of the respective systems, while at the same time having causal learning, only learning after the grid fault has started. This would then also be the variant that could be implemented in real-life wind turbines. As was already expected, the effectiveness of the ILC algorithms to lower the 2-norm of the output error has been diminished for this variant, as the learning and use of input has been constrained the most. That said, the PID-type ILC, the saturation based norm-optimal ILC and the optimisation based norm-optimal ILC all managed to lower the 2-norm of the output error on both the low-fidelity and the high-fidelity models.

On the low-fidelity model, for the first time, the PID-type ILC manages to outperform the norm-optimal ILCs in terms of the final 2-norm. However, this result does not align with those of the high-fidelity model, in which the PID-type ILC performs the worst in terms of the final 2-norm. This difference in the conclusion of which ILC performs best is most likely due to the much higher complexity of the high-fidelity wind turbine model compared to the low-fidelity wind turbine model. Where for the low-fidelity wind turbine model the input and output still relate in a close to linear fashion under certain conditions, this is not the case for the high-fidelity wind turbine model after a grid fault. Comparing the norm-optimal ILCs on the high-fidelity model, it has been observed that the optimisation based controller performs better, with a lower final 2-norm of the output error. This is also reflected in the way the requested input signal of the optimisation based norm-optimal controller has much less switching compared to the saturation based norm-optimal ILC, and also utilises more of the available input with higher blade pitch rates for longer periods of time.

For a changed objective function, reducing the tower-bottom moment error, the conclusion completely changes. For this scenario the PID-type ILC manages to yield a significantly lower

final 2-norm of the output error compared to the saturation based norm-optimal ILC. More so, no suitable tuning for the optimisation based norm-optimal ILC has been found in which the 2-norm is lowered. Even though the controller structure and use of the impulse response is identical to that of the rotor speed error objective function, here the norm-optimal ILCs perform significantly worse. This is then also a curious conclusion of this thesis, where for one of the objective functions the optimisation based norm-optimal ILC performs best, and for the other the PID-type ILC. Although no proper conclusion can be made on why this discrepancy exists, the author does have some suggestions. Note that the norm-optimal ILC makes use of the baseline controller which has been tuned to reduce rotor speed error. It is possible that since the feedback controller inside the norm-optimal ILC tries to counteract the rotor speed error, and the overall norm-optimal ILCs objective is to reduce the tower-bottom moment error, these interests collide in such a significant manner that the performance of the norm-optimal ILC suffers. If this were to be true, then a differently configured feedback controller would have to be used within the norm-optimal ILC on the high-fidelity model. To, however, fully understand this problem, more research should be done. The recommendation for now is to always try both ILC algorithms when significantly changing the controllers objective function, where for the current configuration of the controllers in the high-fidelity model, the causal optimisation based constrained norm-optimal ILC should be used for the rotor speed error objective and the PID-type ILC for the tower-bottom moment objective function.

Future work

Following the completion of the thesis work there are multiple recommendations for future work that can improve or expand on the already acquired results.

- Investigate in more detail the difference between the rotor speed error and tower-bottom moment objective functions. When switching between these two objective functions, either the PID-type ILC or the norm-optimal ILC performs best. There is the suspicion that this difference could be explained by the non-changing feedback controller present within the norm-optimal ILC, and that perhaps for different objective functions the norm-optimal ILC should also get different feedback controller configurations.
- During this thesis, it has always been assumed that the grid fault characteristics are known and deterministic. This could, however, be expanded to consider stochastic grid fault characteristics, so changing shapes of the grid fault, with a fault detection algorithm that can detect when a grid fault is occurring and which type of grid fault it is.
- Instead of using a single objective function, multi-objective cost functions can be implemented. It could be investigated if a multi-objective cost function can successfully provide the operator a trade-off between different objectives, since preliminary results show that the objectives are not decoupled. Additionally, it could be investigated how these types of controller can be tuned; for example, should the feedback controller within the norm-optimal ILC be adjusted, are time-varying objective functions a viable option, or other considerations.

- Based on the best-case load reduction prognoses as simulated during this thesis, a prognosis on the benefits of the ILC algorithms developed here could be made. Here the benefits imply for example a life expectancy improvement and or a steel reduction due to the decrease in overspeeding and or mechanical loading.

Appendix A

A broad overview of Iterative Learning Control algorithms

Though one might associate Iterative Learning Control (ILC) with a singular control algorithm, it is in fact a class of controllers that all work in different ways. Roughly speaking, ILC algorithms can be classified into classical, nonlinear, constrained, and other. The classical ILC is focused on linear systems, but is claimed to also work for some nonlinear systems [45], as the WT is. The nonlinear ILC is directly designed for nonlinear systems and could therefore yield better results for the nonlinear event of LVRT in a WT compared to the classical ILC. Constrained ILC is often an extension of the previously ILC algorithms, where state and/or input constraints are considered. The last category, other, contains ILC algorithms who do not clearly fit in the for-mentioned categories. An example of such an algorithm is a stochastic ILC that allows the modelling of an uncertain signal or event. In this chapter, multiple algorithms are discussed, followed by a brief discussion on which algorithms could be used during this thesis.

A-1 Classical Iterative Learning Control algorithms

Classical ILC is designed for discrete- or continuous-time linear time-invariant systems with state dynamics $\dot{x}_j(n) = Ax_j(n) + Bu_j(n)$ and output dynamic $y_j(n) = Cx_j(n)$. It is assumed that; (1) every iteration has a fixed duration time, (2) the initial conditions of each iteration are equal, and (3) the system dynamics is invariant over the iterations [46].

A simple classical learning rule is one of the Arimoto type described by $u_{j+1} = u_j + \Gamma\dot{e}_j$, with Γ being a diagonal matrix. This learning rule is suitable for systems with a relative degree of one or less [46]. The Arimoto-type learning rule guarantees the following convergence $\lim_{j\to\infty} y_j(n) \to y_\mathrm{d}(n)\,\forall\, n \in [0, N]$ if $||I - CB\Gamma||_i < 1$ for $i \in \mathbb{Z}^+$ [46] with $||\cdot||_\mathrm{i}$ an operator norm. For more general systems, PID-type learning rules are suggested. This PID-type learning rule is expressed by $u_{j+1} = u_j + \Phi e_j + \Gamma\dot{e}_j + \Psi \int e_j\, dt$. Asymptotic convergence of

the PID-type learning rule can also be proven for linear systems, as can be seen in Appendix B.

The above-mentioned classical learning rules use only the information from one previous iteration. There exist also higher-order ILC learning rules that use more than one previous iteration, as is done in the flowing P-type learning rule $u_{j+1} = u_j + \sum_{i=j}^{i=j-l} \Gamma_i e_i$. However, the effectiveness of these higher-order learning rules is still unclear, as stated by [64], although a more recent study suggested that theoretically the convergence property of higher-order learning rules is better [65]. However, the increased complexity of the higher-order learning rule does not seem to outweigh the benefit of a theoretically better convergence.

A-2 Nonlinear Iterative Learning Control algorithms

The classical ILC described above is designed for linear systems. It is suggested that these also work for many nonlinear systems [45], but it is not clear under which conditions this is. Therefore, a special nonlinear ILC has been made. One of such nonlinear ILC is presented in the work of [66] which considers a wide class of nonlinear systems where some bounded uncertainties are allowed. In this paper, a scheme is presented that combines a high-gain state feedback controller to ensure stability and a learnt feedforward controller for tracking. The feedforward controller is learnt using a linear update rule based on previous feedback and feedforward control inputs u. The feedback controller uses knowledge of the eigenvalues and eigenvectors of the system, which might be problematic for unknown or uncertain system models. It is stated that large uncertainties in the system model can prevent the learning rule from working as intended. More recent papers related to nonlinear ILC are published in [46].

A new type of nonlinear ILC algorithm is gradient-based ILC. Gradient-based ILC has become popular due to well-defined convergence and robustness properties [67]. However, it does need a model since it is gradient based. The construction of a sufficiently rich nonlinear model can, however, be time consuming and expensive. Therefore, a model-free gradient-based ILC for nonlinear systems is presented in the work of [68]. In the paper, the output data are used to estimate the gradient of the state function. The authors claim to achieve similar results as in model-based gradient ILC, and better tracking compared to using a model-free gradient ILC for linear systems. More gradient-based model-free ILC algorithms can be found in [67, 69, 70, 71].

A-3 Constrained Iterative Learning Control algorithms

In an industrial application such as WT control, there are often limits and constraints imposed on the system. These constraints can be in terms of the (internal) state(s), but also on the control input(s) and or system output(s). For these applications, there are constrained ILC algorithms. In [72] a constrained ILC for linear systems is presented. In this work the ILC is rewritten as a quadratic optimisation problem, and solved using a barrier function and Newton's method to allow for inequality constraints on the input. The work of [73] uses a similar optimisation based approach in which ILC is used for linear systems with input and state constraints. In the work of [74] ILC for linear systems with input, input rate of change

and soft output constraints is presented, and also solved using a optimisation based method. In the work of [75] a model-free ILC is presented for Single Input Single Output (SISO) linear systems with input constraints. In the paper, they reformulate the problem as an optimisation problem where the new input, or learning rule, is acquired by calculating the coefficients of a causal finite impulse response filter that minimises the next output error. It is claimed that learning is complete in a single step if the output measurements are noise free and if some simple constraints hold.

The work of [76] presents ILC for nonlinear systems modelled by Lagrange's formulation with constraints on the state, first derivative of the state, and input. The work of [77] presents an ILC for more general nonlinear systems with state constraints, and the work of [78] presents an ILC for nonlinear systems with input constraints. The work of [79] presents a model-free ILC for nonlinear systems with output, input, and change rate of input constraints, but only for SISO systems. No ILC has been found for nonlinear MIMO systems with both input and state constraints.

A-4 Other Iterative Learning Control variations

In addition to the ILC algorithms described above, there are many variations that use some form of ILC. These algorithms often aim to solve a specific task that was not yet possible with other ILC algorithms. The paper of [80] presents a fault-tolerant fuzzy ILC method. In the paper, the system is represented using a fuzzy T-S model, which also includes fault dynamics. In this fuzzy model, ILC is used to learn the optimal control, which is also claimed to be valid during faulty conditions. In the work of [54] a norm-optimal ILC is presented. In this work, a feedback and feedforward ILC algorithm are combined, in which the feedforward controller tries to minimise a cost function. In solving this problem, an impulse response of the closed-loop system is used, which enables one-step convergence for linear systems, under some conditions, by effectively employing model inversion. Additionally, the work allows for easy expansion of hard constraints by rewriting the problem as an optimisation problem. Main advantages claimed by the author are fast convergence, robustness, and intuitive tuning abilities. In [81] an ILC using neural networks is presented. The authors claim that this provides a more flexible solution that can adapt to changing working conditions. In [82] a survey is conducted on stochastic ILC. In stochastic ILC, the system model includes a signal described by a random variable, allowing uncertainty to be taken into account. This random variable could, for instance, be used to capture the uncertain dynamics of a grid voltage drop.

A-5 Discussion Iterative Learning Control algorithms for grid faults

In this chapter ILC algorithm classified as classical, nonlinear, constrained and other have have been presented. The presented classical ILC algorithms are simple in implementation and do not require any system knowledge. Still, it is claimed that the classical ILC can be used for some nonlinear systems. The nonlinear ILC algorithms presented are either gradient based, a technique in which system knowledge is exploited, or a combination of feedback and feedforward control. These controllers are claimed to be more robust compared to the classical controllers but at the expense of additional complexity in the controller structure. To

formally add constraints to the problem, the constrained ILC algorithms provide a solution. Such algorithms exist for linear and nonlinear systems and usually employ a technique in which the ILC problem gets rewritten as an optimisation problem, but also model-free versions exist. In the last category, other, multiple ILC algorithms have been presented. Here, especially noteworthy are the norm-optimal ILC that has similarities with the nonlinear and constrained ILC algorithms algorithms presented before, but in a more modular framework, and the stochastic ILC that can include signals modelled by a random variable, e.g. the grid voltage.

Since not all algorithms can be implemented, a selection has to be made. As is common practise, start simple, then add complexity if needed. In that spirit, the PID-type learning rule, under the classical ILC category, is recommended as a first algorithm to test. It is simple in implementation and should work for many systems. Constraints can be added after the controller itself, so that they effectively become part of the system dynamics, see Chapter 2-3-1. As a second algorithm, the norm-optimal ILC is appealing due to its modular structure. The algorithm uses system knowledge by including an impulse response, which should result in higher performance while still remaining model-free. For unconstrained problems, an analytical solution is available, and for constrained problems, it should be relatively easy to convert the problem to an optimisation problem. A more elaborate discussion of the inner workings of these algorithms can be seen in Chapter 2.

Appendix B

Linear ILC convergence proof

Asymptotic convergence of the learning process can be proven for the linear PID-type Iterative Learning Control (ILC) algorithm on a Single Input Single Output (SISO) discrete-time system can be proven. More specifically, it can be proven that the output error goes to zero. This proof is based on the work of [50] where convergence of PD-type ILC is considered. The results of the proof can be used to tune the ILC algorithm gains but do require a linear system with full knowledge of the dynamics. For this proof, the subscript j denotes the iteration index and the time step is indicated by $n \in [0, N]$. The dynamics of the known system can be described as $x_j(n+1) = Ax_j(n) + Bu_j(n)$ and $y_j(n) = Cx_j(n)$. The system output error with respect to the reference signal $y_\mathrm{d}(n)$ can be noted as $e_j(n) = y_j(n) - y_\mathrm{d}(n)$. Converting the system dynamics to supervector notation results in

$$Y_j = H_\mathrm{p} U_j. \tag{B-1}$$

Where H_p are the Markov parameters, Y_j the system output and U_j the system input described by:

$$\begin{aligned} H_\mathrm{p} &= \begin{bmatrix} CAB & 0 & \cdots & 0 \\ CA^2B & CAB & & 0 \\ \vdots & & \ddots & \vdots \\ CA^NB & CA^{N-1}B & \cdots & CAB \end{bmatrix}, \\ Y_j &= \begin{bmatrix} y_j(1), & y_j(2) & \cdots & y_j(N) \end{bmatrix}^T, \\ U_j &= \begin{bmatrix} u_j(0), & y_j(1) & \cdots & u_j(N-1) \end{bmatrix}^T. \end{aligned} \tag{B-2}$$

For PID-based linear ILC the learning rule [83] is described by $u_{j+1}(n) = u_j(n) + k_\mathrm{p} e_j(n+1) + k_\mathrm{i} \sum_{m=1}^{n+1} e_j(m) + k_\mathrm{d}(e_j(n+1) - e_j(n))$, which converted to supervector notation becomes $U_{j+1} = U_j + k_\mathrm{p} I E_j + k_\mathrm{i} T_1 E_j + k_\mathrm{d}(I - T_2) E_j$ with the following definitions.

$$E_j = \begin{bmatrix} e_j(1), & e_j(2) & \cdots & e_j(N) \end{bmatrix}^T,$$

$$T_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdot & 1 & 0 \end{bmatrix}, \; T_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdot & 1 & 0 \end{bmatrix}.$$

Now the difference in the error of the current and next iteration is described by

$$\begin{aligned} E_{j+1} - E_j &= H_\mathrm{p} U_j - H_\mathrm{p} U_{j+1} \\ &= H_\mathrm{p}(U_j - (U_j + (k_\mathrm{p} I + k_\mathrm{i} T_1 + k_\mathrm{d}(I - T_2))E_j) \end{aligned} \tag{B-3}$$

Which can be rewritten to get to the recursion function of the error function

$$\begin{aligned} E_{j+1} &= (I - (k_\mathrm{p} + k_\mathrm{d}) H_\mathrm{p} I - k_\mathrm{i} H_\mathrm{p} T_1 + k_\mathrm{d} H_\mathrm{p} T_2) E_j \\ &= H_\mathrm{e} E_j \end{aligned} \tag{B-4}$$

The definition of the error dynamics by matrix H_e allows for a simple check of convergence. If the inequality $||H_\mathrm{e}||_i < 1$ for $i = 1, 2, \infty$ holds, not only asymptotic but also monotonic convergence of the output error is guaranteed [50, 83].

Appendix C

Baseline controller on grid fault results

For comparison purposes, two baseline controllers have been defined. The controllers are for the low-fidelity model, and the high-fidelity model as described in Section 3-3. The baseline controllers used here are PID-based for the low-fidelity, and the field deployed controller for the high-fidelity model, with more information about the controllers can be seen in Section 4-1. In this Chapter the performance of both controllers for a grid fault is shown, with a static nominal wind speed of 20m/s, and the grid fault as defined in 3-1.

The actual results of the controllers can be seen in Figure C-1 and C-2 for the low- and high-fidelity models respectively. The 5MW low-fidelity model manages to return to oscillation free nominal conditions after only an approximate seven seconds. Comparing this to the 15MW high-fidelity model, one can see that after 30 seconds the system still has oscillations in the rotor speed. This might partially be due to the slower system and blade dynamics that come with the larger-scale wind turbine of 15 MW compared to 5 MW for the low-fidelity model, but could also be due to a more conservative controller setup. One can also see that the high-fidelity controller does not saturate its inputs, meaning that there could still be performance to be gained.

When looking closely at the blade pitch plot of the high-fidelity model, one can see that there is poor tracking of the blade pitch setpoint. This even occurs when actuator constraints are not violated. Since the actual control of the blade pitch is considered a black-box for the intents of this thesis, one can only speculate on why this is. It could be that the actuator blade pitch limits are a low lower, but more likely it seems that the blade pitch controller used does not prioritise following the setpoint and instead tries to the wear of the hardware.

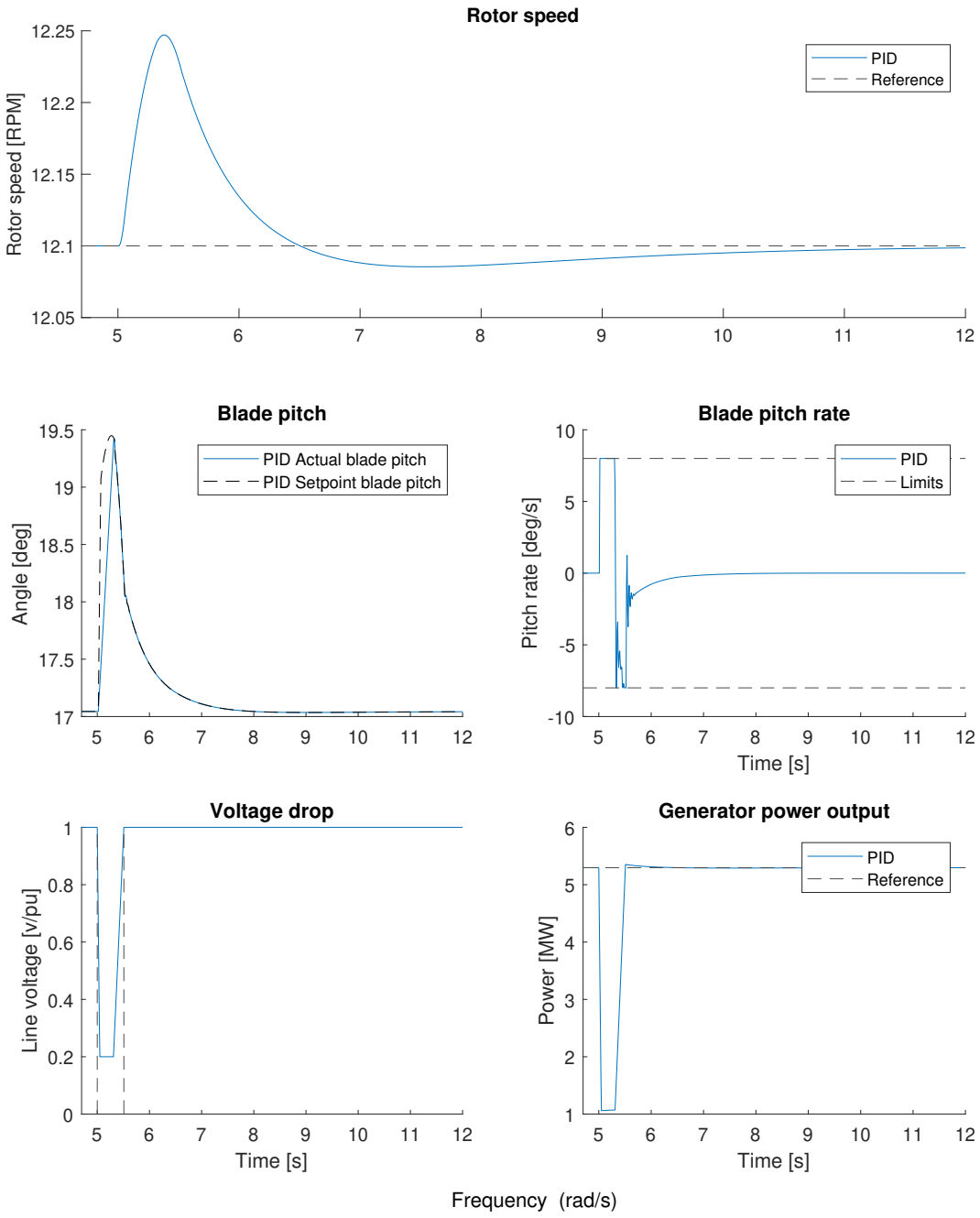


Figure C-1: Results of the baseline PID-controller on the 5 MW low-fidelity wind turbine model with phisical actuator constraints. Control objective is to follow the reference rotor speed signal, with an external disturbance on the line voltage. Blade pitch is limited by actuator constraints with the actual blade pitch and the controllers blade pitch setpoint both shown.

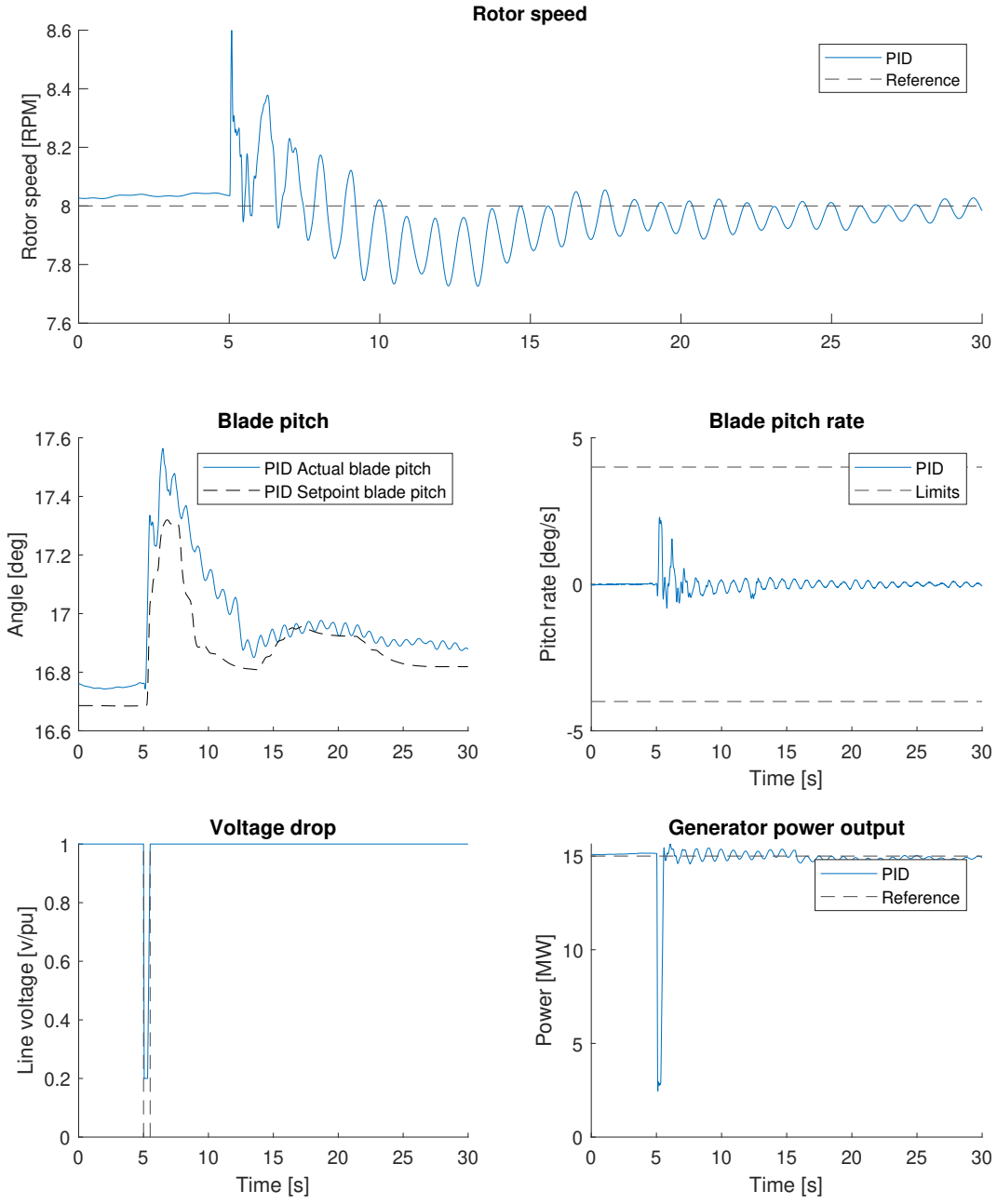


Figure C-2: Results of the baseline field-deployed controller on the 15 MW high-fidelity wind turbine model. Control objective is to follow the reference rotor speed signal, with an external disturbance on the line voltage. Blade pitch is limited by actuator constraints with the actual blade pitch and the controllers blade pitch setpoint both shown.

Appendix D

Calculation of J matrix for low-fidelity wind turbine model

In this chapter the calculation of the J matrix, as defined in Section 2-3-2, is shown in more detail for the low-fidelity wind turbine model with rotor speed objective function. The principles shown here also apply for the high-fidelity wind turbine model and other objective functions.

The J matrix captures data from an impulse response into the systems input, that is the input that the norm-optimal ILC uses to minimise the output error. In this case this input is the blade pitch. As such, an impulse on the blade pitch signal will have to be simulated. However,

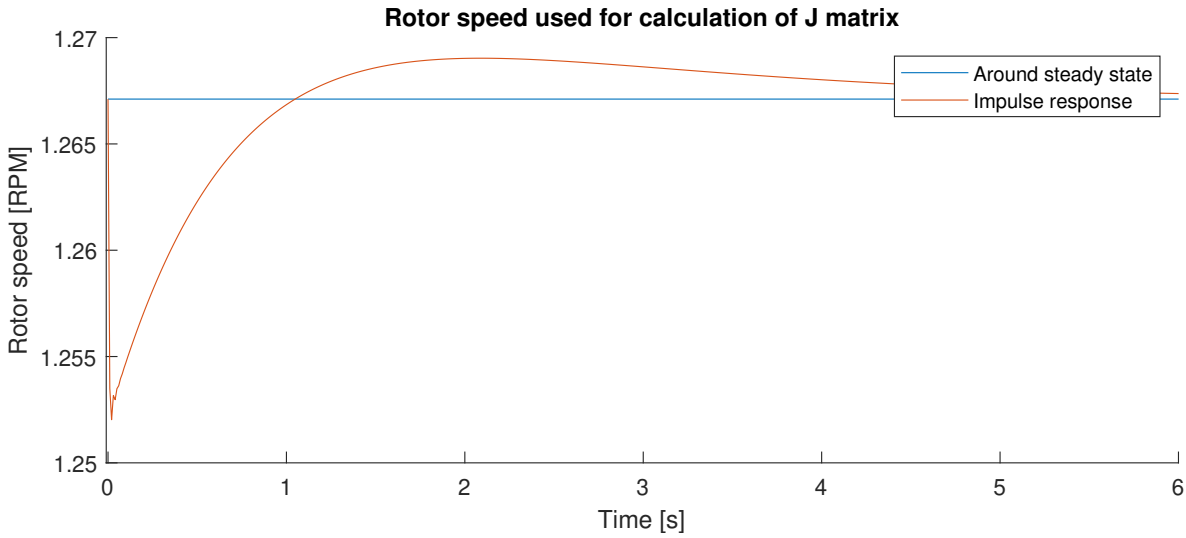


Figure D-1: Rotor speed output as results of an impulse response on the blade pitch input as indicated by 'Impulse response', and the rotor speed output without the impulse response as indicated by 'Around steady state'. Notice that the impulse response right after time is zero decreases the rotor speed and gradually returns to the steady state value

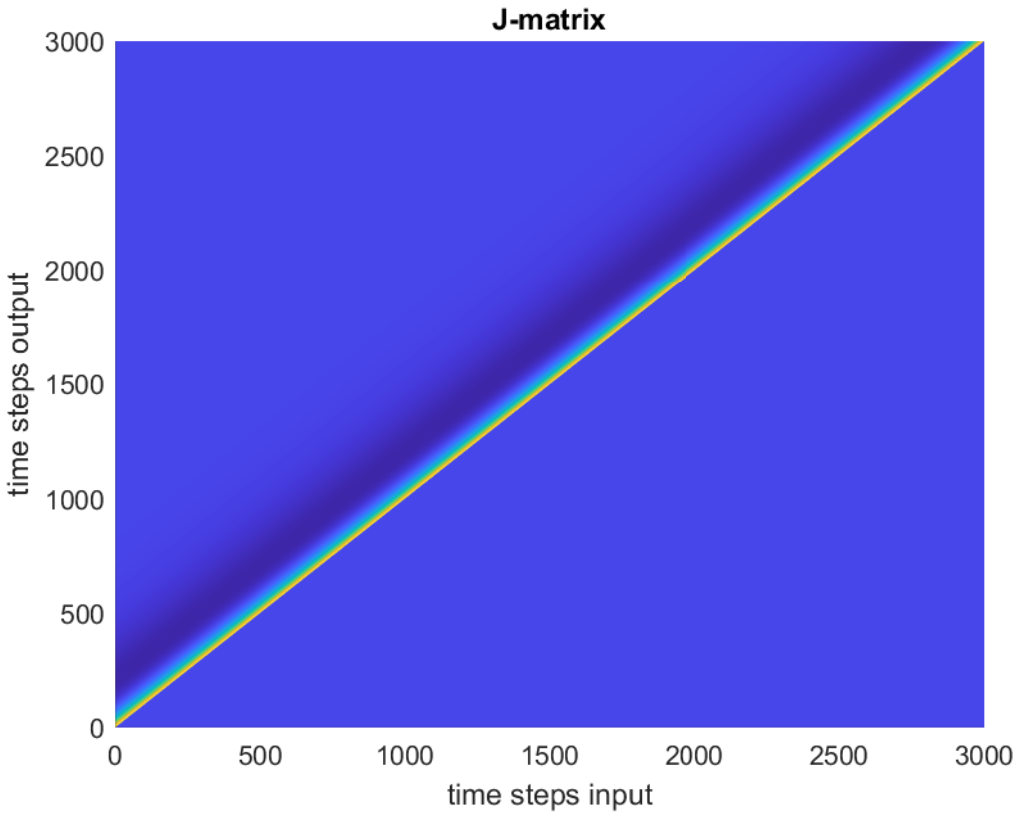


Figure D-2: The resulting J matrix used inside of the norm-optimal ILC. On the x-axis is the input time steps and on the y-axis the output time steps. The J matrix shows how an input signal for any time step effects the output signal for any time step.

since the blade pitch signal used here is not centered around zero, but instead around a sort of linearisation point, two simulations are done. One for the turbine with no impulse response, and a second with an impulse response applied. Out of these simulations, here the rotor speed is saved since the objective is to reduce the rotor speed and as such the J matrix will capture how the blade pitch input reflects on the rotor speed output. The results of these simulations can be seen in Figure D-1.

Using this two simulations as shown the previous figure, now the difference between the two is taken which gives a plot that effectively shows the effect an impulse response has around the linearisation point. Using this vector, the convolution matrix J is calculated. Now matrix J effectively shows the effect of applying an input at any time-step to the output at all time-steps. Note that any and all time-steps refers to time-steps inside the simulation time of the simulation, which in this case is 30 seconds.

Appendix E

Unconstrained ILC with causal learning

In chapter 4-2 the unconstrained ILC is presented. These controllers have had the unrealistic approach of starting the ILC learning before the fault has started. For real-world applications, this would imply that you know exactly when the grid fault starts happening. As this is an unrealistic assumption, the previous controllers are extended to be causal learning controllers, where learning only starts after the full fault magnitude is present. This still leaves the assumption of known fault length and shape intact, but should provide a more implementable scenario. That said, still the unconstrained controller is used, where later in this chapter input constraints are included.

E-1 Causal unconstrained PID-type ILC on low-fidelity model

Implementing a causal PID-type ILC learning is not complicated, as there is no fundamental change to the learning structure. One only needs to determine at which point the learning should start, and before this point change the learning rule to $u_{j+1}[n] = u_j[n]$ so that the learning is effectively turned off. The point at which learning will start is one step after the full fault magnitude has been reached.

The tuning criterion is slightly changed compared to the non-causal controllers, as a non-zero output error is to be expected due to the later start of the learning controller. Therefore, the main tuning criteria is the magnitude of the 2-norm at the end of the learning. In addition, it is still desirable to not get any deteriorating performance in intermediate iterations, meaning that it is desirable to have the maximum of the 2-norm equal to one.

The results of the tuning can be seen in Figure E-1 and Table E-1. Here one can see that if the same tuning value as in the non-causal learning were to be used, the 2-norm only decreases to about 0.6. The best results acquired here are with $K_\mathrm{p} = 0.4$, $K_\mathrm{i} = 1 \cdot 10^{-5}$ and $K_\mathrm{d} = 0.3$ producing a final 2-norm of only 0.17, a significant improvement compared to the 0.6 with the tuning of the previous section.

The time domain results of the tuned controller can be seen in Figure E-2. As expected, one can see that the rotor speed error is not equal to zero due to the causal learning. Only after

K_p	K_i	K_d	mean 2-norm	max 2-norm	final iter 2-norm
0.1	$1 \cdot 10^{-4}$	1.00	0.6027	1.0000	0.5903
0.2	$1 \cdot 10^{-4}$	1.00	0.4594	1.0000	0.4439
0.3	$1 \cdot 10^{-4}$	1.00	0.3859	1.0000	0.3675
0.4	$1 \cdot 10^{-4}$	1.00	0.3801	1.0000	0.3204
0.3	$1 \cdot 10^{-4}$	0.50	0.2825	1.0000	0.2637
0.3	$1 \cdot 10^{-4}$	0.40	0.2600	1.0000	0.2368
0.3	$1 \cdot 10^{-4}$	0.30	0.2372	1.0000	0.2060
0.4	$1 \cdot 10^{-4}$	0.30	0.2158	1.0000	0.1794
0.5	$1 \cdot 10^{-4}$	0.30	0.2594	1.0000	0.1612
0.4	$1 \cdot 10^{-4}$	0.35	0.2239	1.0000	0.1931
0.4	$1 \cdot 10^{-3}$	0.30	3.1282	40.358	0.1799
0.4	$1 \cdot 10^{-5}$	0.30	0.2129	1.0000	0.1797
0.4	0	0.30	0.2130	1.0000	0.1799

Table E-1: Tuning of the causal unconstrained PID-type ILC on the low-fidelity wind turbine model. Shown are the mean, max and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

the learning starts, one can see a decrease in the rotor speed happening. Interesting to see is that the shape of the input matches that of one of the non-causal PID-type ILCs, where here off course the beginning of the input signal is different due to the later starting of learning.

E-2 Causal unconstrained norm-optimal ILC on low-fidelity model

Delaying the learning in the norm-optimal ILC can be done by changing the J matrix of Equation 2-5 to have zero entries at the times when one does not want learning to occur. Alternatively, one could define Equation 2-2 only in the domain where learning must occur.

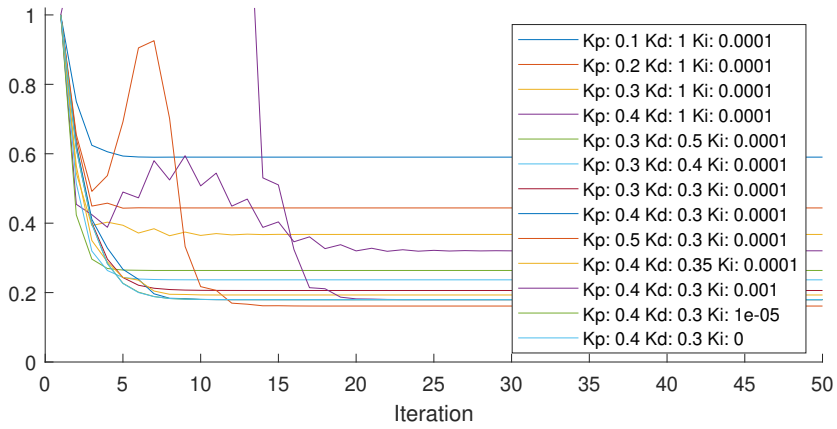


Figure E-1: The 2-norm of the output error for the causal unconstrained PID-type ILC on the low-fidelity wind turbine model shown for different controller settings. Shown is the progression of the 2-norm, for different controller settings, after each ILC iteration.

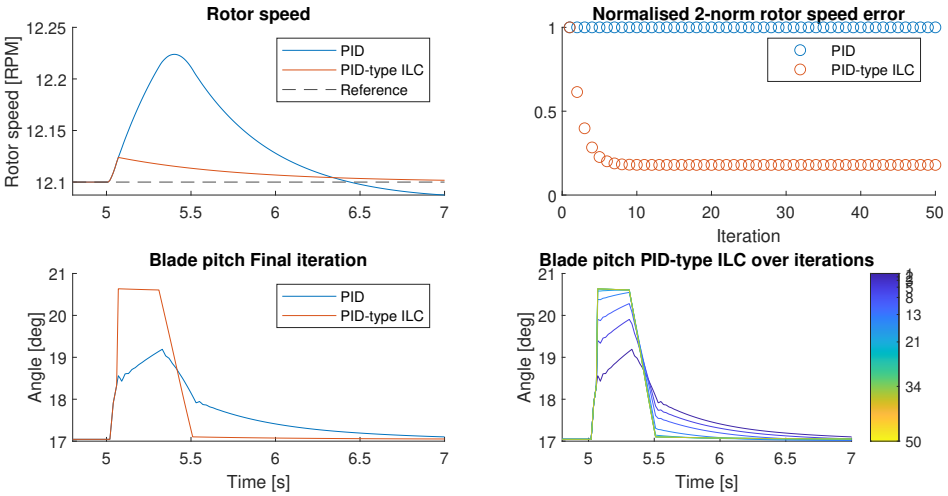


Figure E-2: Causal unconstrained PID-type ILC compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

Just as in the PID-type ILC, this means that the baseline PID controller will control the wind turbine before the feedforward learning starts.

The tuning of the causal norm-optimal ILC can be seen in Figure E-3 and Table E-2. One can see that changing the tuning parameters does not have a huge effect on the optimal solutions, as indicated in all nearly identical shapes in Figure E-3. Still, there was some slight difference in the magnitude of the 2-norm of the final iteration, with the best tuning values found being $W_e = 300$, $W_f = 0.01$, and $W_{\Delta f} = 0$. This yielded a 2-norm for the final iteration of only 4.3% of the baseline controller.

In Figure E-4 one can see the time-domain results of the causal norm-optimal ILC. Here, one can see that the causal norm-optimal ILC manages to almost immediately decrease the output error to near zero after the learning has started. This is indicated by the sharp drop in the rotor speed to the reference in the figure. To do this, the norm-optimal ILC uses large amounts of input directly after the learning has started.

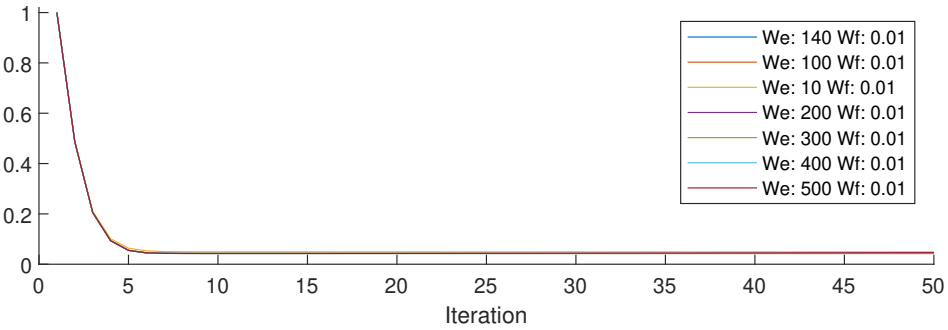


Figure E-3: The 2-norm of the output error for the causal unconstrained norm-optimal ILC on the low-fidelity wind turbine model shown for different controller settings. Shown is the progression of the 2-norm, for different controller settings, after each ILC iteration.

W_e	W_f	$W_{\Delta f}$	mean 2-norm	max 2-norm	final iter 2-norm
140	0.01	0	0.0755	1.0000	0.0429
100	0.01	0	0.0757	1.0000	0.0431
10	0.01	0	0.0817	1.0000	0.0492
200	0.01	0	0.0753	1.0000	0.0427
300	0.01	0	0.0752	1.0000	0.0426
400	0.01	0	0.0754	1.0000	0.0434
500	0.01	0	0.0762	1.0000	0.0460

Table E-2: Tuning of the causal unconstrained norm-optimal ILC on the low-fidelity wind turbine model. Shown are the mean, max and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

E-3 Comparison and conclusion of causal unconstrained ILC on low-fidelity model

Both the PID-type ILC and the norm-optimal ILC provided significant improvements compared to the baseline controller for grid fault control, as can be seen in Figure E-5. Where for the non-causal case the difference between the two controllers was quite small, there one can see that the norm-optimal ILC manages to perform significantly better than the PID-type ILC. The PID-type ILC converges to an input signal quite similar in shape as in the non-causal case, as can be seen in Figure 4-5. This, however, does not yield as fast a convergence to zero output error as the norm-optimal ILC manages to accomplish. Once the learning starts, the norm-optimal ILC gives a high spike in the blade pitch signal, steering the rotor speed error to near zero immediately.

Even though both controllers have quite different approaches and accompanying results, for both controllers, the rotor speed error is suppressed to zero quicker than the baseline PID

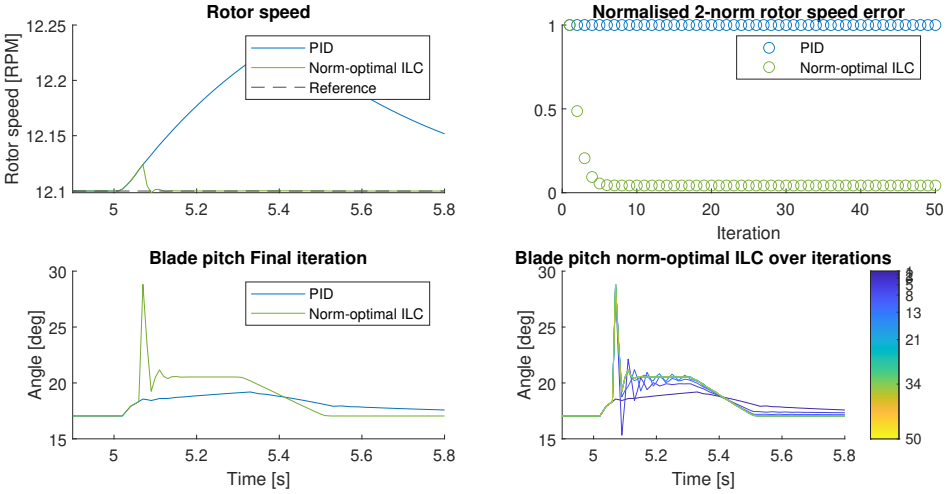


Figure E-4: Causal unconstrained norm-optimal ILC compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

controller manages to do, where also the 2-norm of the output error is significantly decreased. That said, for this scenario with no input constraints, the norm-optimal ILC manages to do a significantly better job in decreasing the rotor speed error and the accompanying 2-norm. To do this, the norm-optimal ILC has to use even more of the input signal than was the case for the non-causal controller, but this is within the controller's scope.

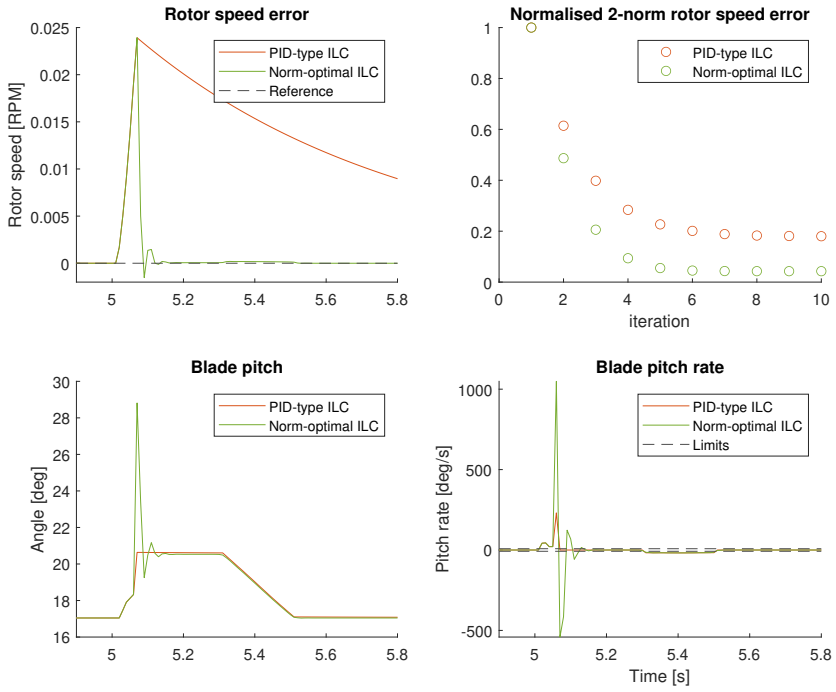


Figure E-5: Comparison of the causal unconstrained PID-type ILC with the causal unconstrained norm-optimal ILC on the low-fidelity wind turbine model. Shown are the rotor speed error, blade pitch and blade pitch rate of the final iterations. Also, the progression of the 2-norm of the output error is shown. In the blade pitch rate plot the hardware limits are shown, but were not used at this point. Visible in the plot is that the causal unconstrained norm-optimal ILC manages to utilise more of the input signal resulting in an improved 2-norm of the output error.

Appendix F

High input penalties in constrained norm-optimal ILC on low-fidelity

In Section 4-4-2 a non-optimisation based constrained norm-optimal ILC has been presented. There an external saturation block of the input and input rate of change has been used. However, in Chapter 2 also a different non-optimisation based approach was presented, namely using a high penalty weight W_{f} with also a second variant where W_{f} penalises the rate of change. The results of these methods were not able to actually get the input within bounds, where specifically the input rate of change was problematic. Yet, in this Chapter some results of these methods are presented.

In Figures F-1, F-2, F-3 and F-4 one can see the results of tuning, and a time-domain simulation. These simulations are done with W_{f} shaped as in equation F-1 and also for W_{f} shaped as in equation F-2. For both of these cases, the maximum input rate of change does not decrease any more upon an increase of the magnitude of W_{f} at some point. The resulting controller still violates the input constraints, meaning that this scheme is not sufficient for as a grid fault controller for wind turbines. Working non-optimisation based alternatives are demonstrated in Chapter 2.

$$W_{\mathrm{f}} = w_{\mathrm{e}} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}, \tag{F-1}$$

$$W_{\mathrm{f,roc}} = w_{\mathrm{e}} \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}, \tag{F-2}$$

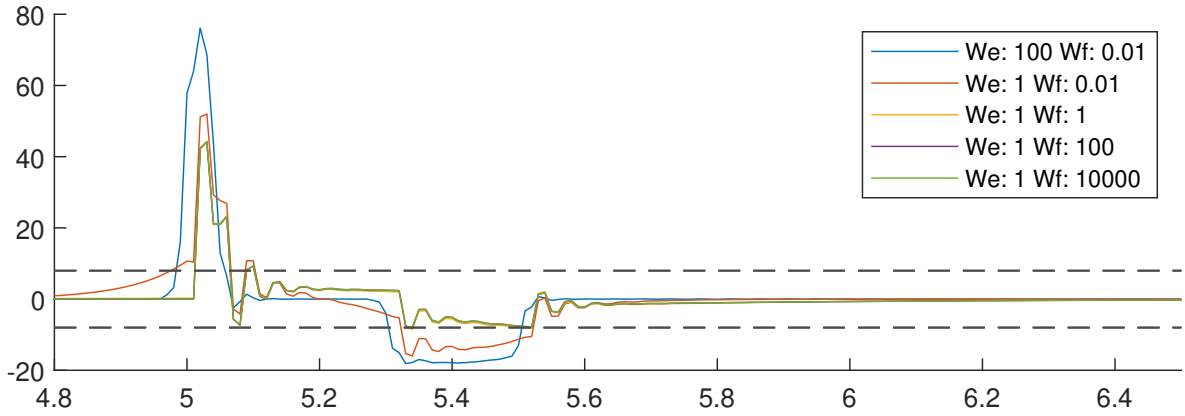


Figure F-1: Soft-constrained norm-optimal ILC. Shown are the ROC, the hardest input constraints, for different controller values. Visible is that for every controller setting the constraints are violated. Here W_{f} is shaped as equation F-1

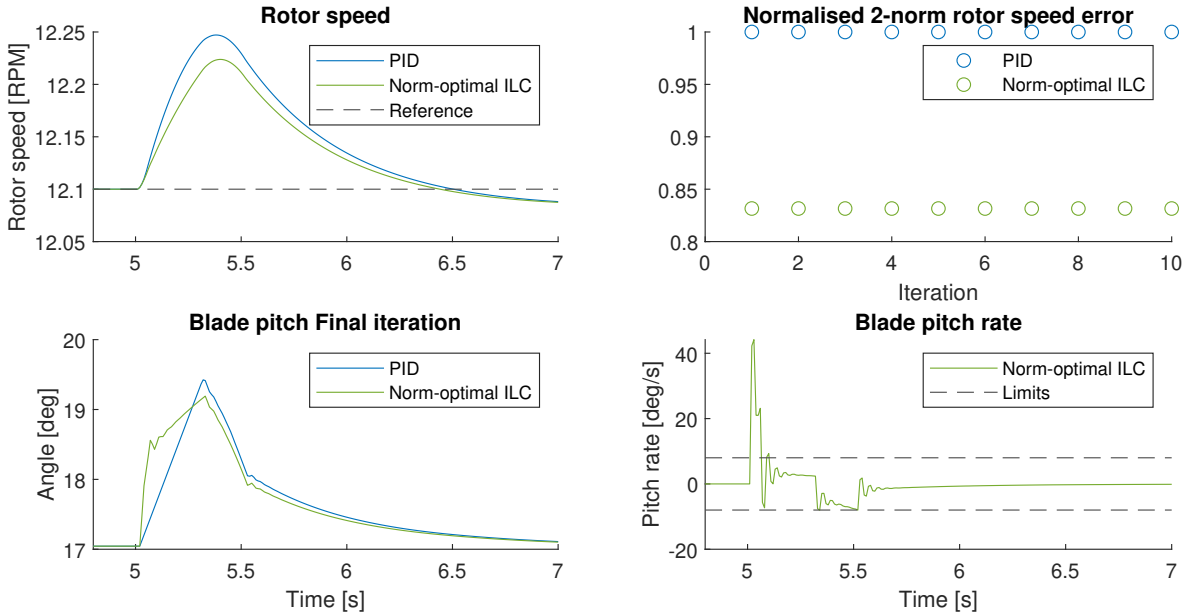


Figure F-2: Soft-constrained norm-optimal ILC results using $W_{\mathrm{e}} = 1$ and $W_{\mathrm{f}} = 1 \cdot 10^4$ with W_{f} shaped as equation F-1. Visible is the the input rate of change constraints are still violated.

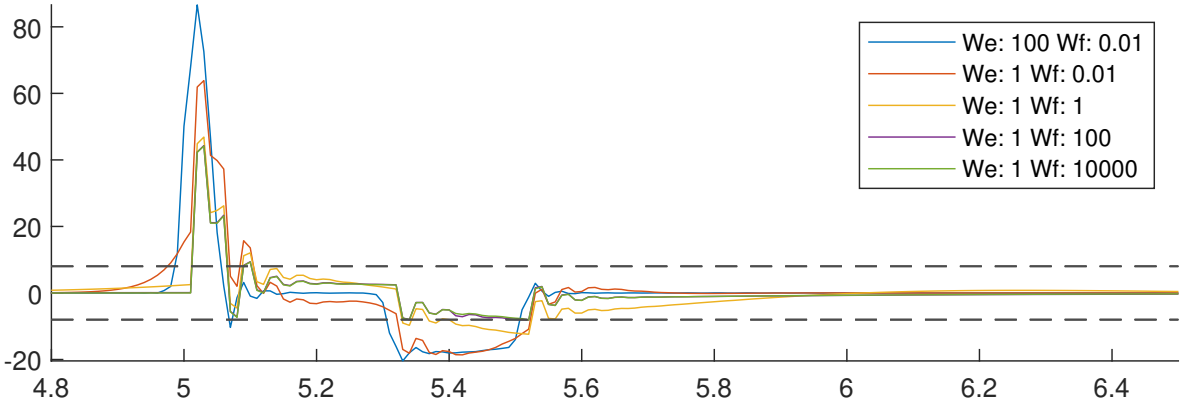


Figure F-3: Soft-constrained norm-optimal ILC. Shown are the ROC, the hardest input constraints, for different controller values. Visible is that for every controller setting the constraints are violated. Here W_{f} is shaped as equation F-2

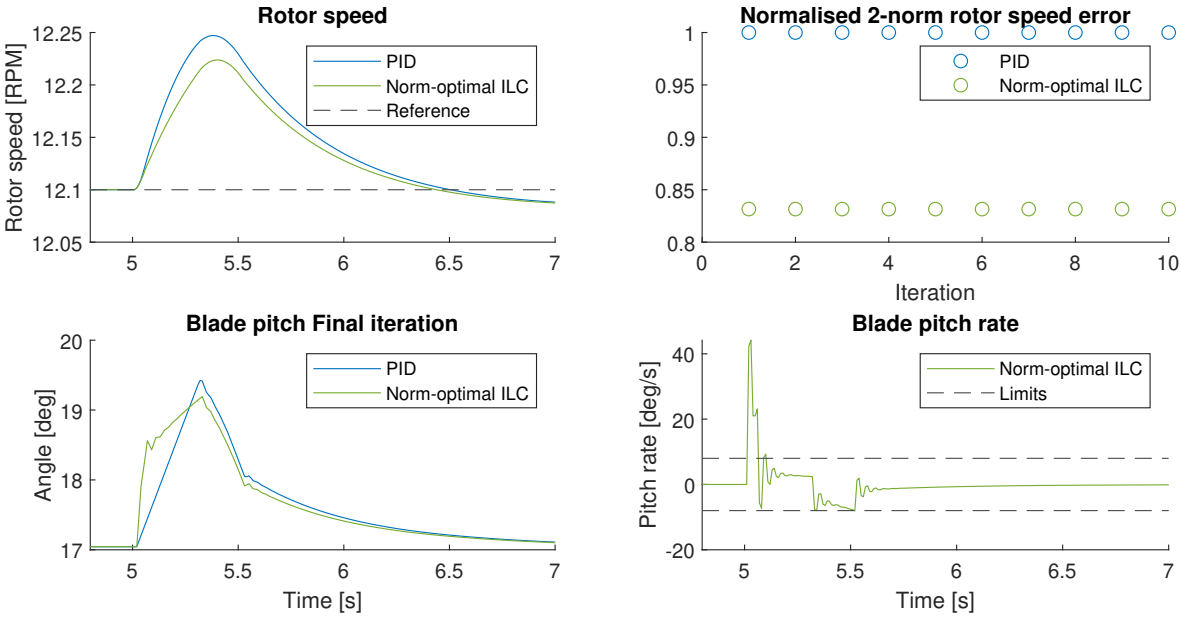


Figure F-4: Soft-constrained norm-optimal ILC results using $W_{\mathrm{e}} = 1$ and $W_{\mathrm{f}} = 1 \cdot 10^4$ with W_{f} shaped as equation F-2. Visible is the the input rate of change constraints are still violated.

Appendix G

Constrained norm-optimal ILC using optimisation on low-fidelity model, alternative problem formulations

In this chapter some variants of the optimisation based constrained norm-optimal ILC, as in Section 4-4-3, are presented. These variants have a different way of handling the input rate of change constraints, but with less satisfactory results, meaning that the input constraints are still violated. This is because the rate of change constraints have to use an approximation of the feedback input signal used for the next iteration, leading to possible violation of said constraint. The standard formulation used for this chapter is as follows, where the next estimated input $\hat{\mathbf{u}}_{j+1}$ is changed.

$$\begin{aligned} \mathbf{f}_{j+1}^{*} &= \arg\min_{\mathbf{f}_{j+1}} \mathcal{J}(\mathbf{f}_{j+1}) \\ \text{s.t.} g(\hat{\mathbf{u}}_{j+1}) &\leq u_{\Delta\max}, \\ \hat{\mathbf{u}}_{j+1} &\leq u_{\max}, \\ \hat{\mathbf{u}}_{j+1} &\geq u_{\min}. \end{aligned} \tag{G-1}$$

With $\mathcal{J}(\mathbf{f}_{j+1})$ as in equation 2-5, T_s the sampling time, $\mathbf{u}_{\Delta\max}$ the maximum rate of change of the input, $u_{\max}$ the maximum input and $u_{\min}$ the minimum input. The superscript $\mathbf{x}_j^{a:b}$ indicates that the samples a up to and including b of iteration j are used of the variable $\mathbf{x} \in \mathbb{R}^N$.

Working configuration, but without a tuned-down feedback controller

The first variant that is presented is the same as in Section 4-4-3 but without a tuned-down feedback controller within the norm-optimal ILC. So in other words, here the same PID-controller tuning is used as is in the baseline PID-controller. For clarification, this uses the following rate of change formulation:

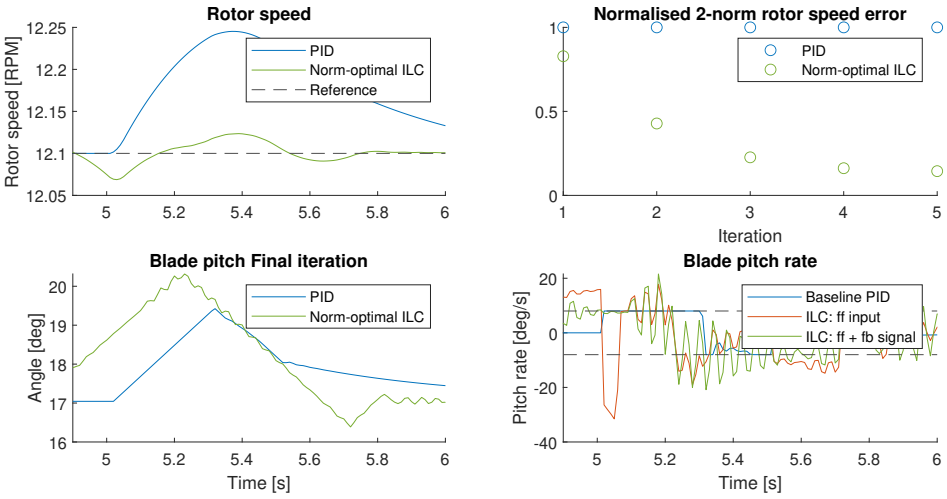


Figure G-1: Results of optimisation based input constrained norm-optimal ILC with rate of change constraint as in Equation G-2 during a grid fault, not shown in figure, as defined in Section 3-1. In the bottom right one can see the feedforward and feedback parts of the ILC algorithm split up, where one can see that the combined feedforward and feedback controller violate the input constraints.

$$g(\hat{\mathbf{u}}_{j+1}) = \left\| \hat{\mathbf{u}}_{j+1}^{2:N} - \hat{\mathbf{u}}_{j+1}^{1:N-1} \right\| / T_\mathrm{s} \leq u_{\Delta\max} \tag{G-2}$$

This is combined with $\hat{\mathbf{u}}_{j+1} = \mathbf{f}_{j+1} + \mathbf{u}_{\mathrm{j,fb}}$. The results of this controller setup can be seen in Figure G-1. One can see that this ILC algorithm manages to exceed the performance of the baseline PID controller, but in doing this, the input constraints are violated. Additionally, the convergence of the output errors 2-norm happens over multiple steps, where in the constrained norm-optimal controller with tuned-down feedback controller has convergence in a single step.

Direct estimation of feedback controller input using estimated next output error

In the previous section, the input signal of the feedback controller has been approximated by using the previous input signal of the feedback controller. An alternative approach is to directly calculate the next expected feedback signal, using the estimate $\hat{\mathbf{e}}_{j+1} = \mathbf{e}_j - J(\mathbf{f}_{j+1} - \mathbf{f}_j)$ used within the ILC algorithm itself, as described in Chapter 2. Using this estimate of the next input combined with a super-vector notation of the feedback PID-controller used within the norm-optimal ILC, one can estimate the next input by the following description:

$$\hat{\mathbf{u}}_{j+1} = \mathbf{f}_{j+1} + \left(k_\mathrm{p} I + k_\mathrm{i} T_1 + k_\mathrm{d}(I - T_2)\right) \hat{\mathbf{e}}_{j+1} \tag{G-3}$$

using some of the matrix notation of Appendix B for matrices T_1 and T_2. Now, equal to the previous section, one can define the actual input rate of change constraint as $g(\mathbf{f}_{j+1}) = \left\| \hat{\mathbf{u}}_{j+1}^{2:N} - \hat{\mathbf{u}}_{j+1}^{1:N-1} \right\| / T_\mathrm{s} \leq \mathbf{u}_{\Delta\max}$. The results of this configuration, the optimisation based constrained norm-optimal ILC can be seen in Figure G-2. One can see that the performance of the controller in terms of the output error 2-norm is about what is to be expected, but

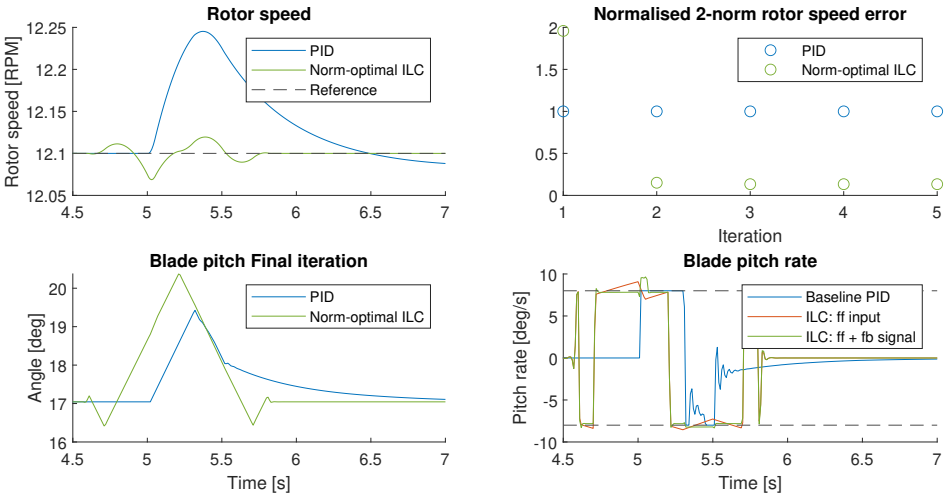


Figure G-2: Results of optimisation based input constrained norm-optimal ILC with input estimation as in Equation G-3 during a grid fault, not shown in figure, as defined in Section 3-1. In the bottom right one can see the feedforward and feedback parts of the ILC algorithm split up, where one can see that the combined feedforward and feedback controller violate the input constraints.

unfortunately the input constraints are violated. Here, mainly the input rate of change constraints are violated right after the fault has started.

To mitigate this problem, one could reduce the input rate of change value $u_{\Delta\max}$ to 0.8 times its original value, resulting in no further violation of the original constraint. See also Figure G-3. In this plot the harsher input constraint does not result in a violation of the input

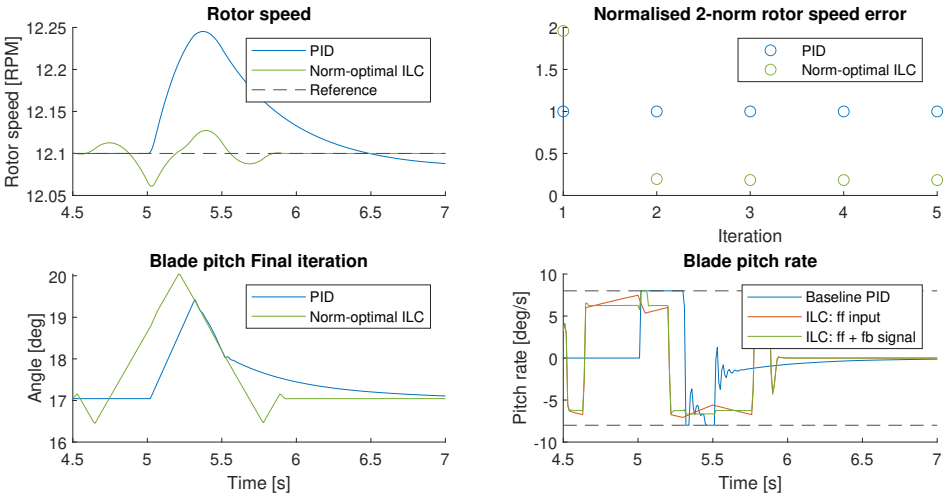


Figure G-3: Results of optimisation based input constrained norm-optimal ILC with input estimation as in Equation G-3 during a grid fault, not shown in figure, as defined in Section 3-1. The rate of change constraint has here been increased 0.8 times the nominal level to make sure that the actual input constraint is not violated. In the bottom right one can see the feedforward and feedback parts of the ILC algorithm split up, where one can see that the combined feedforward and feedback controller violate the input constraints.

constraints for the actual input u_{j+1}. This does, however, come at the expense of some performance, since over the whole time range the input rate of change constraint is stricter, whilst only in a small portion of the time frame an actual violation of said constraint is observed. Though this method does yield the results that are desired, the implementation is not neat or robust in any way, and as such is not recommended.

Direct estimation of feedback controller input using previous output error

In the previous section a direct calculation of the next input signal has been tried using knowledge of the feedback controller. For this calculation, the next estimated output error has been used. The results of this controller setup were unsatisfactory since the input constraints were still violated. Therefore, a second attempt will also be proposed and tested using a slightly modified input estimate. Here, instead of the estimate $\hat{\mathbf{e}}_{j+1}$, the previous measured error $\mathbf{e}_j$ is used, resulting in the following formulation for the input constraint:

$$\hat{\mathbf{u}}_{j+1} = \mathbf{f}_{j+1} + (k_\mathrm{p} I + k_\mathrm{i} T_1 + k_\mathrm{d}(I - T_2))\,\mathbf{e}_j \tag{G-4}$$

The idea behind this slight alteration of the input estimation is that here no estimate input error $\hat{\mathbf{e}}_{j+1}$ will have to be used, but instead the previous input error. And since it is assumed that the output error settles to some steady-state value, the assumption used here of $\mathbf{e}_j \approx \mathbf{e}_{j+1}$ seems reasonable. The results of this setup can be seen in Figure G-4, where unfortunately the input rate of change constraint is still violated.

A solution for this violation of the input constraint is to set a more stringent constraint in the optimisation software, using $g(\hat{\mathbf{u}}_{j+1}) \leq 0.8 u_{\Delta\max}$. This setup works, as can be seen in Figure G-5, but is not a reliable method of using input constraints. Therefore, also this method is not recommenced.

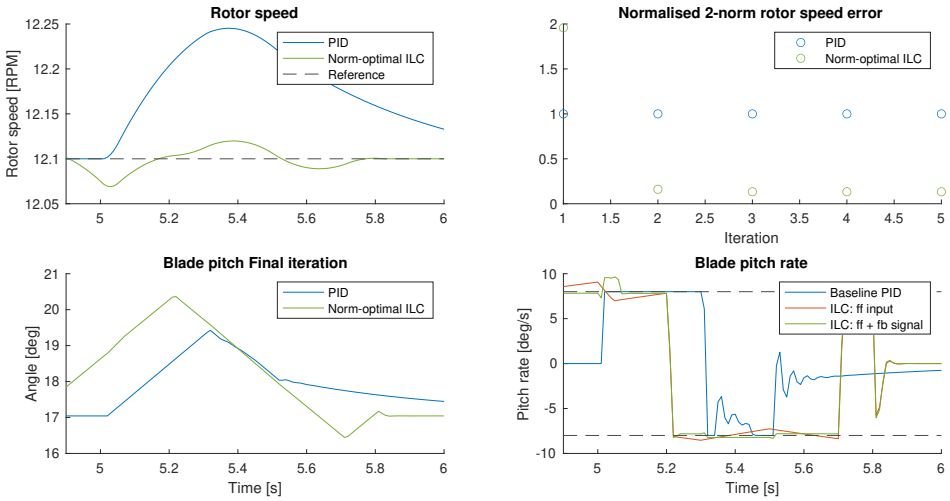


Figure G-4: Results of optimisation based input constrained norm-optimal ILC with input estimation as in Equation G-4 during a grid fault, not shown in figure, as defined in Section 3-1. In the bottom right one can see the feedforward and feedback parts of the ILC algorithm split up, where one can see that the combined feedforward and feedback controller violate the input constraints.

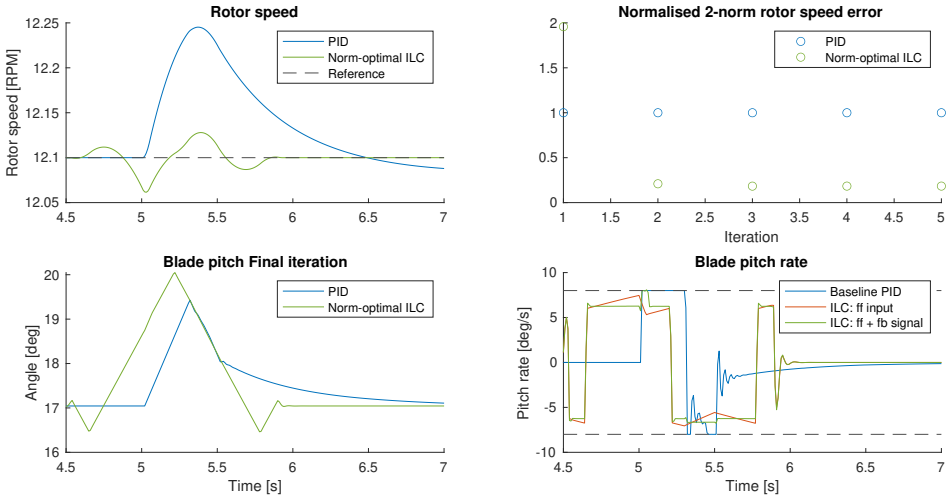


Figure G-5: Results of optimisation based input constrained norm-optimal ILC with input estimation as in Equation G-4 during a grid fault, not shown in figure, as defined in Section 3-1. The rate of change constraint has here been increased 0.8 times the nominal level to make sure that the actual input constraint is not violated. In the bottom right one can see the feedforward and feedback parts of the ILC algorithm split up, where one can see that the combined feedforward and feedback controller violate the input constraints.

Conclusion

In this chapter multiple variants of the optimisation based constrained norm-optimal ILC have been shown that will not be used. More specifically, multiple variants of the input estimates for the next iteration have been demonstrated. The reasons why these variants do not work as intended is due to violation of the input constraints caused by inaccurate input estimates.

Appendix H

Constrained controllers with causal learning

In this chapter, the constrained controller is extended to also be causal with respect to the grid fault. This means that the (supplemental) ILC control signal will only be introduced into the system after the fault has occurred. Before this time, the baseline controllers is active. For this section, the learning will start once the line voltage is only 80% of the nominal value. This section should give more information about how much real-life performance could be expected to be gained compared to the baseline controllers. In this section only the tuning of the controllers are shown, with the comparison and conclusion in Chapter 4.

H-1 Causal constrained PID-type ILC on low-fidelity model

In this section, the causal constrained PID-type ILC is presented. The main difference between it and the non-causal variant of the previous sections is that the learning starts only after the grid fault has begun. The tuning of the controller is as such quite similar, and can be seen in Table H-1. Here, one can see that using the same values as was the case for the non-causal variant works best, with $K_\mathrm{p} = 0.5$, $K_\mathrm{i} = 1 \cdot 10^{-5}$ and $K_\mathrm{d} = 0.15$. This controller tuning does not have the absolute lowest final 2-norm, that is, for the controller with $K_\mathrm{p} = 0.6$, but for that controller tuning, the maximum 2-norm is higher than one, as can also be seen in the plot, and as such has worse performance for some of the intermediate iterations.

The time-domain results of this tuned controller can be seen in Figure H-1. Here, one can see that in the first moments after the grid fault has started, the baseline and PID-type ILC perform the same action, and in doing so fully saturate the controller. The difference between the two controllers starts after the fault has ended, where the baseline controller starts reducing the blade pitch angle much faster than the PID-type ILC. Due to the better control actions of the PID-type ILC, it manages to bring the rotor speed back to zero error much faster, and also with only very little overshoot, indicating a well damped system.

The convergence of the PID-type ILC occurs within around 15 iterations in a monotonic fashion. The final 2-norm of the output error is reduced to 0.78 times that of the baseline controller, indicating a respectable improvement.

K_p	K_i	K_d	mean 2-norm	max 2-norm	final iter 2-norm
0.50	$1 \cdot 10^{-5}$	0.15	0.8001	1.0000	0.7861
0.60	$1 \cdot 10^{-5}$	0.15	0.8383	1.1489	0.7853
0.40	$1 \cdot 10^{-5}$	0.15	0.8076	1.0000	0.7953
0.50	$1 \cdot 10^{-5}$	0.10	0.8266	1.0000	0.8122
0.50	$1 \cdot 10^{-5}$	0.20	0.8162	1.0000	0.8046
0.50	$1 \cdot 10^{-4}$	0.15	0.8021	1.0000	0.7873
0.50	$1 \cdot 10^{-3}$	0.15	5.2031	6.3336	0.8044
0.50	$1 \cdot 10^{-6}$	0.15	0.8001	1.0000	0.7860

Table H-1: Tuning of the causal constrained PID-type ILC on the low-fidelity wind turbine model. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

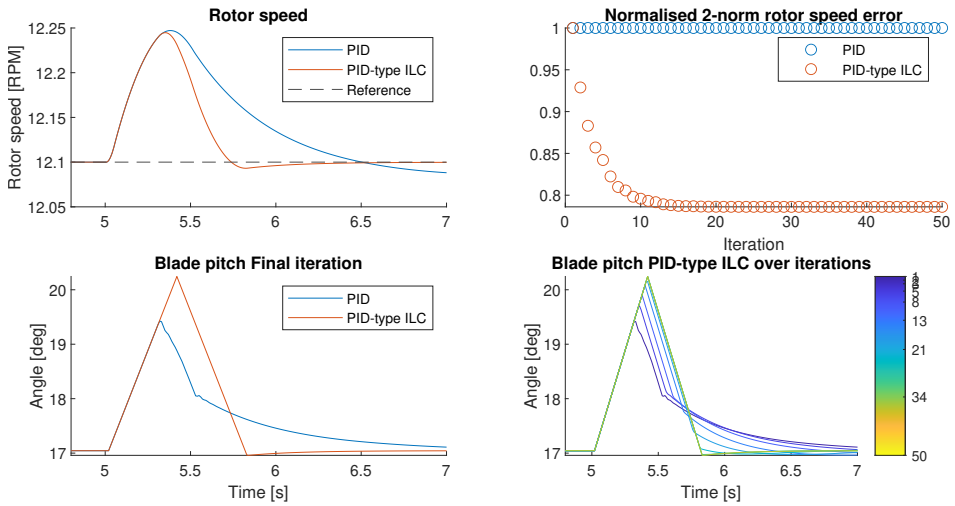


Figure H-1: Causal constrained PID-type ILC compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

H-2 Causal constrained norm-optimal ILC using saturation on low-fidelity model

Here the causal constrained norm-optimal ILC using saturation for the input constraints on the low-fidelity model is shown. The tuning of the controller is started with the tuning values used for the non-causal constrained norm-optimal ILC using saturation. The results of the tuning can be seen in Table H-2. As one can see, using the same values as for the non-causal controller yields the best results. This was also the case for the causal PID-type ILC, indicating that perhaps the controller tuning is invariant for when learning starts. For the final controller tuning values, $W_{\mathrm{e}} = 0.5$, $W_{\mathrm{f}} = 0.01$, and $W_{\Delta \mathrm{f}} = 0$, the controller manages to reduce the 2-norm of the output error to 0.79 times that of the baseline controller.

One can see from the plots that the convergence of the causal constrained norm-optimal ILC using saturation occurs in a monotonic-like fashion for this scenario. Moreover, this monotonic-like behaviour can be seen for all of the controller tunings. This indicates that the controller is much more robust against bad controller tuning compared to the causal constrained PID-type ILC. Also noteworthy is that the performance of this constrained norm-optimal ILC does not change significantly with different tuning values and only starts to lose significant performance when the controller tuning is off by a large degree. This is also a positive aspect of the causal constrained norm-optimal ILC using saturation.

In Figure H-2 one can see the time-domain performance of the casual constrained norm-optimal ILC. One can see that the norm-optimal ILC manages to steer the rotor speed much faster to the reference signal but does suffer from some overshoot in the output. Convergence of the 2-norm happens in around seven iterations, but does show some strange jumping behaviour after the tenth iteration, where the 2-norm increases slightly. The output error 2-norm does, however, again converge to a lower value after this sudden slight jump.

H-3 Causal constrained norm-optimal ILC using optimisation on low-fidelity model

In this section a causal constrained norm-optimal ILC using optimisation on the low-fidelity wind turbine model is presented. The controller tuning can be seen in Table H-3. Tuning has started with the controller values found for the non-causal constrained norm-optimal

W_{e}	W_{f}	$W_{\Delta \mathrm{f}}$	mean 2-norm	max 2-norm	final iter 2-norm
0.5	0.01	0	0.8173	1.0000	0.7959
1.0	0.01	0	0.8226	1.0000	0.7982
10	0.01	0	0.8560	1.0000	0.8308
10	0.10	0	0.8226	1.0000	0.7982
10	1.00	0	0.8292	1.0000	0.8113

Table H-2: Tuning of the causal constrained norm-optimal ILC using saturation on the low-fidelity wind turbine model. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

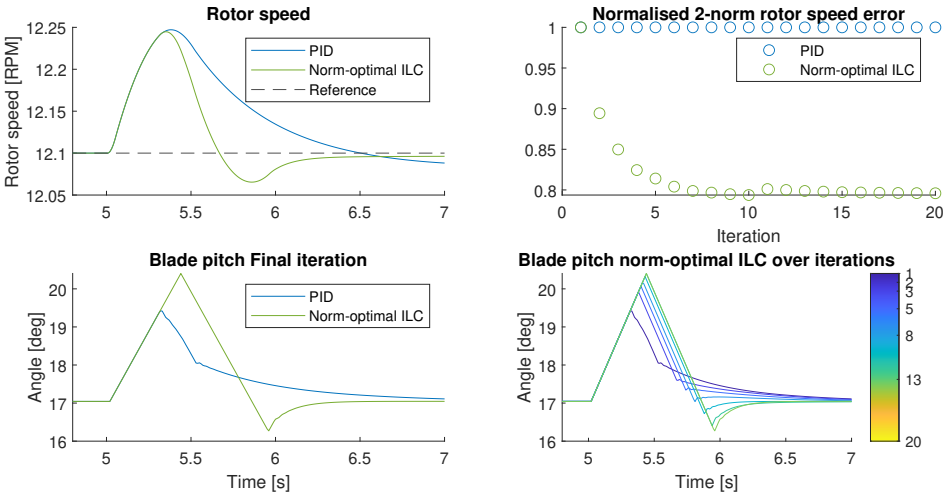


Figure H-2: Causal constrained norm-optimal ILC using saturation compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

using optimisation as a starting point. Similarly as for the previous causal controller, using the controller tuning values from the non-causal controller yields a properly tuned controller, using $W_{\mathrm{e}} = 300$, $W_{\mathrm{f}} = 0.01$, and $W_{\Delta \mathrm{f}} = 0$. The controller is quite robust for bad tuning values. This is also evident by the near-identical performance of the controller for a large spread of the controller tunings as can be seen in the table. Only once one pushes the controller tuning, a significant decrease in performance is observed.

The time domain results of the tuned causal optimisation based constrained norm-optimal ILC can be seen in Figure H-3. From the results one can see that the norm-optimal ILCs start pitching the blades a bit later compared to the baseline PID controller. This also results in a slight increase of the maximum rotor speed error. That said, the norm-optimal ILC still manages to reduce the 2-norm of the output error to about 0.8 times that of the baseline controller. Also, the convergence rate of this controller is good with convergence after the third iteration.

W_{e}	W_{f}	$W_{\Delta \mathrm{f}}$	mean 2-norm	max 2-norm	final iter 2-norm
300	0.01	0	1.0465	1.9697	0.8062
100	0.01	0	1.0465	1.9697	0.8062
10	0.01	0	1.0466	1.9697	0.8063
1	0.01	0	1.0552	1.9697	0.8181
1000	0.01	0	1.0463	1.9697	0.8060

Table H-3: Tuning of the optimisation based causal constrained norm-optimal ILC on the low-fidelity wind turbine model. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

H-4 Causal constrained PID-type ILC on high-fidelity model

After testing the causal constrained ILC on the low-fidelity model, tests on the high-fidelity model will also be made. We start here with the causal constrained PID-type ILC on the high-fidelity model. This controller will, in the same way as for the low-fidelity model, start learning after the fault has started. This should provide a realistic performance indication of this controller during real-world usage.

The first step is controller tuning, as can be seen in Table H-4. As tuning the causal PID-type ILC was quite a cumbersome process with many attempted controller tunings, only a select number of tuning values have been shown here. The final used controller settings are $K_\mathrm{p} = 0$, $K_\mathrm{i} = 0$, $K_\mathrm{d} = 5 \cdot 10^{-3}$ with the low-pass filter disabled. This resulted in a reduction of the 2-norm to 0.92 times that of the baseline controller for the final iteration.

Though the final controller tuning leads to a reduction of the 2-norm of the output error, users of this controller should be mindful, that many controller tunings lead to a diverging solution with a 2-norm greater than one after some iterations. This is also clear when looking at the complete tuning overview in the Appendix. There are also no guarantees that this will not also happen for the here shown final controller setting. This is especially true since no steady-state value of the output error 2-norm has been reached, as was the case for some of the controllers shown earlier. That said, one could employ mitigating schemes that can, for example, prevent further learning after a fixed number of learning steps or once the output error 2-norm starts increasing. Such schemes could drastically reduce the difficulty of tuning this controller whilst also improving the predictability of the controller in terms of the output errors 2-norm reduction.

The results of the final tuning of the causal constrained PID-type ILC can be seen in Figure H-4. Looking at the final rotor speed of the ILC algorithm, it is clear that the main result of this controller is a slight reduction of the magnitude of the rotor speed oscillations. When also looking at the final blade pitch input signal, it is clear that no significant alterations

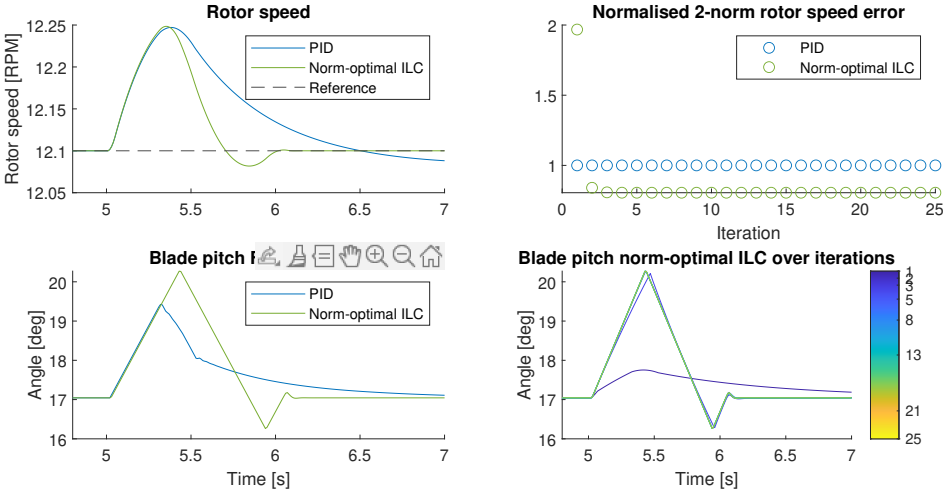


Figure H-3: Optimisation based causal constrained norm-optimal ILC compared with the baseline PID on the low-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

K_p	K_i	K_d	filter cut-off	mean 2-norm	max 2-norm	final iter 2-norm
0	0	0.005	Inf	0.9296	1.0000	0.9248
0	0	0.001	Inf	0.9419	1.0000	0.9338
0	0	0.005	10	1.0355	1.0727	1.0727
0	0	0.005	50	0.9631	1.0000	0.9944

Table H-4: Tuning of the causal constrained PID-type ILC on the high-fidelity wind turbine model. The filter cut-off corresponds to the cut-off frequency off the low-pass filter after the ILC algorithms output, where Inf corresponds to a disabled cut-off filter. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

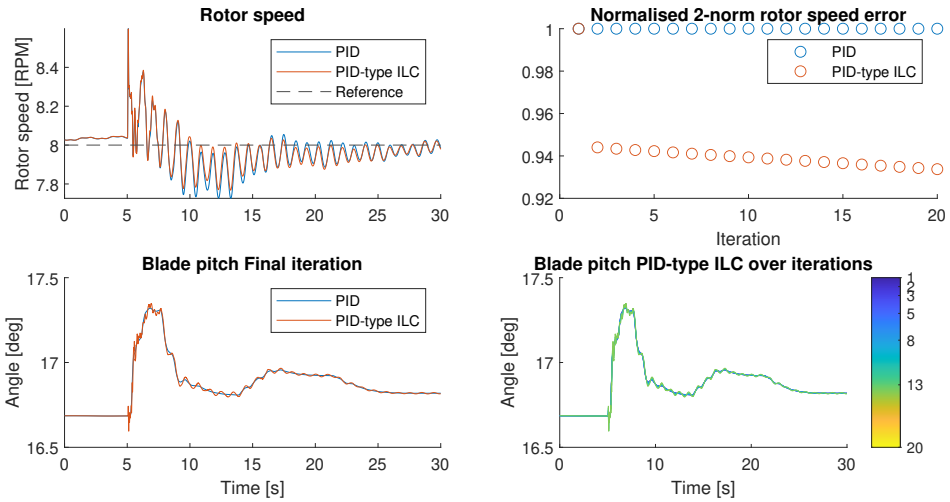


Figure H-4: Causal constrained PID-type ILC compared with the baseline PID on the high-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

have been applied compared to the baseline feedback controller. The general shape of the blade pitch signal is the same, but higher-frequency peaks to the blade pitch signal are added, slightly increasing the performance of the controller compared to the baseline controller. If these results are, however, compared with the non-causal variant on the same high-fidelity model, the results are quite in-line in terms of the final 2-norm of the output error.

H-5 Causal constrained norm-optimal ILC using saturation on high-fidelity model

In this section, a causal constrained norm-optimal ILC using saturation on the high-fidelity wind turbine model is presented. We will start with tuning the controller, as can be seen in Table H-5. The first point to notice with tuning this controller is how easy it is compared to the causal PID-type controller of the previous section. The best results are with the controller tuning $W_\mathrm{e} = 10$, $W_\mathrm{f} = 1$ and $W_{\Delta\mathrm{f}} = 0$. Using these controller settings, a final 2-norm of the output error of 0.80 was achieved. Notice also that the controller quickly converges after

W_e	W_f	$W_{\Delta f}$	mean 2-norm	max 2-norm	final iter 2-norm
0.01	1	0	0.9993	1.0000	0.9992
0.1	1	0	0.9751	1.0000	0.9681
1	1	0	0.8876	1.0000	0.8705
10	1	0	0.8259	1.0000	0.8085
100	1	0	0.8306	1.0000	0.8314
1000	1	0	0.9984	1.0953	1.0953

Table H-5: Tuning of the causal constrained norm-optimal ILC using saturation on the high-fidelity wind turbine model. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

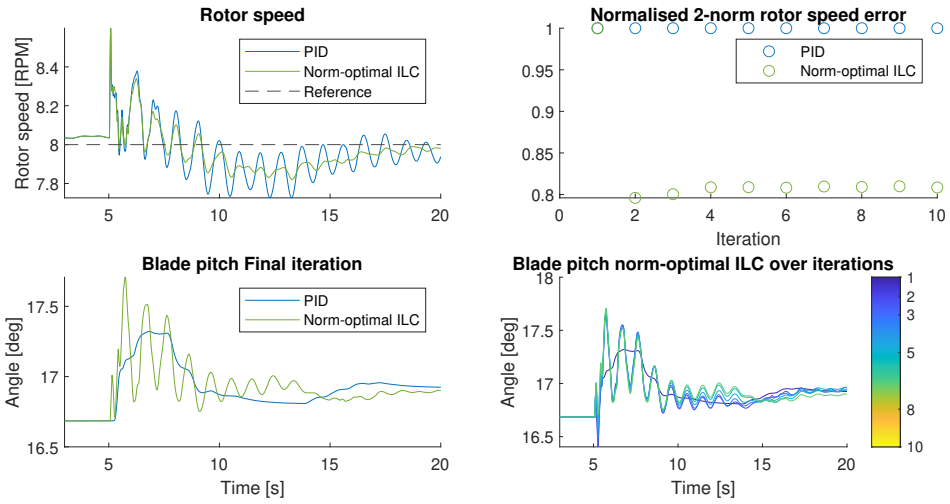


Figure H-5: Causal constrained norm-optimal ILC using saturation compared with the baseline PID on the high-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

the second iteration and stays at this (approximate) steady-state value for the remaining iterations. Only for the last two controller settings, no steady-state value of the output errors 2-norm has been reached. Note also that increasing the penalty on the rotor speed error has an almost direct decrease of the output error 2-norm. Only after exceeding $W_e = 100$, the 2-norm of the output error stops decreasing and even increases after some iterations. This does, however, indicate the ease with which this controller can be tuned.

In Figure H-5 one can see the time-domain results of the causal saturation based constrained norm-optimal ILC. Investigating the rotor speed, one can see that the norm-optimal ILC manages to drastically reduce the magnitude of the rotor speed oscillations present after the ten-second mark. This reduction in oscillations means that after the 20 second mark the wind turbine operates almost in steady-state conditions where the nominal controller could potentially take over without the help of the ILC algorithms additional feedforward signal. The blade pitch signal is also quite different when compared to the baseline feedback controller. The input signal of the baseline controller is relatively smooth, where the norm-optimal input signal has much more movement that is used to reduce the rotor speed oscillations.

H-6 Causal constrained norm-optimal ILC using optimisation on high-fidelity model

The last controller to be presented here is the causal optimisation based constrained norm-optimal ILC. As was already discussed in Section 4-5-3, the optimisation based controller works best when only optimising seven seconds of feedforward input data. This limitation to only seven seconds of feedforward learning is due technical reasons, causing the optimiser to run out of memory for larger time frames. Therefore, all results in this section are performed with seven seconds of feedforward input after the grid fault. This means that this is a causal controller that after said seven seconds relies again on the nominal baseline controller to take over. Advantage of this approach is that the transition back to the nominal controller is also considered here.

As has been the case for the other controller, firstly the controller is tuned as can be seen in Table H-6. Visible are that the tunings of these controllers leads with quite similar results, except for the case of $W_{\mathrm{e}} = 0.1$. Still, a best-performing controller has been identified with the tuning $W_{\mathrm{e}} = 10$, $W_{\mathrm{f}} = 1$, and $W_{\Delta \mathrm{f}} = 0$. This controller tuning does not lead to the absolute lowest final 2-norm, but it does have the lowest mean 2-norm of the output error, indicating that the 2-norm decreases the quickest. That said, all three final tuning values tested work quite well, despite the large range of parameter values of the ILC algorithm. This also indicates how robust this controller is to bad tunings, and that only if the tuning is drastically off, much performance is left untapped.

The controller time domain results can be seen in Figure H-6. Here one can see how the optimisation based norm-optimal controller manages to reduce large amounts of the rotor speed oscillations compared to the baseline controller. This despite the fact that the causal optimisation based constrained norm-optimal ILC only provides a feedforward signal for seven seconds. After these seven seconds of additional feedforward signal the baseline feedback controller again takes over sole control, and manages to keep this reduced rotor speed oscillations for the remaining time. This also makes for easy integration of this controller with the existing control scheme, as it has been shown that if an additional control signal is added to that of the baseline controller for only the first seven seconds after the grid fault, a reduction in the rotor speed error 2-norm and rotor speed oscillations can be observed.

W_{e}	W_{f}	$W_{\Delta \mathrm{f}}$	mean 2-norm	max 2-norm	final iter 2-norm
0.1	1	0	1.0000	1.0000	1.0000
1	1	0	0.7622	1.0000	0.7014
10	1	0	0.7518	1.0000	0.7020
100	1	0	0.7661	1.0000	0.7441

Table H-6: Tuning of the optimisation based causal constrained norm-optimal ILC on the high-fidelity wind turbine model. Shown are the mean, max, and final value of the 2-norm of the controller captured after each ILC iteration. The 2-norm is normalised with respect to the baseline PID controller. Desirable is a low final 2-norm with a max 2-norm equal to one.

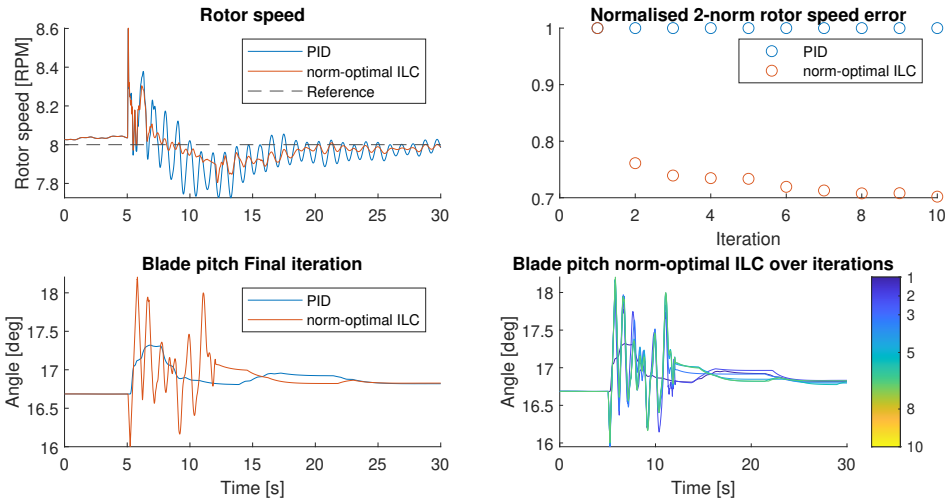


Figure H-6: Optimisation based causal constrained norm-optimal ILC compared with the baseline PID on the high-fidelity wind turbine model. Controller objective, reduce rotor overspeeding caused by grid fault, not shown in figure, as defined in Section 3-1

Bibliography

[1] European Commission, "Liquefied Natural Gas," 2022.

[2] A. Cooban, "Europe still can't live without this Russian energy export," *CNN Business*, Nov 2022.

[3] K. Abnett, "European Commission analysing higher 45% renewable energy target for 2030," *Reuters*, Apr 2022.

[4] P. Messad, "European Parliament backs 45% renewable energy goal for 2030," *Euroactiv*, Sep 2022.

[5] P. Bojek, "Wind energy, technology deep dive," *IEA*, Sep 2022.

[6] K. Whithing, "Explainer: What is offshore wind and what does its future look like?," *weforum*, Nov 2022.

[7] G. Gardiner, "Report forecasts record U.S. wind turbine installations for 2022," *Composit world*, Nov 2021.

[8] L. Y. Pao and K. E. Johnson, "Control of wind turbines," *IEEE Control Systems*, vol. 31, p. 4462, Apr 2011.

[9] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind energy explained: Theory, design and Application*. John Wiley & Sons, 2002.

[10] J. Chen, J. Chen, and C. Gong, "Constant-bandwidth maximum power point tracking strategy for variable-speed wind turbines and its design details," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, p. 50505058, 2013.

[11] R. L. Iglesias, R. L. Arantegui, and M. A. Alonso, "Power Electronics Evolution in wind turbinesa market-based analysis," *Renewable and Sustainable Energy Reviews*, vol. 15, p. 49824993, Sep 2011.

[12] E. J. N. Menezes, A. M. Araújo, and N. S. B. da Silva, "A review on wind turbine control and its associated methods," *Journal of Cleaner Production*, vol. 174, p. 945953, 2018.

[13] F. Blaabjerg, F. Iov, Z. Chen, and K. Ma, "Power Electronics and controls for wind turbine systems," *2010 IEEE International Energy Conference*, 2010.

[14] E. A. Bossanyi, "The design of closed loop controllers for wind turbines," *Wind Energy*, vol. 3, no. 3, p. 149163, 2000.

[15] S. P. Mulders, T. G. Hovgaard, J. D. Grunnet, and J. Wingerden, "Preventing wind turbine tower natural frequency excitation with a quasi-lpv model predictive control scheme," *Wind Energy*, vol. 23, no. 3, p. 627644, 2019.

[16] T. Schneider, "Factsheet Transmission system operators," *Renewables Grid Initiative*, 2015.

[17] Commission regulation (EU) 2016/631, "Establishing a network code on requirements for grid connection of generators," *Official Journal of the European Union*, 2016.

[18] A. D. Hansen, P. Sørensen, F. Iov, and F. Blaabjerg, "Centralised power control of wind farm with doubly fed induction generators," *Renewable energy*, vol. 31, no. 7, pp. 935–951, 2006.

[19] J. M. L. Rubio, F. J. Buendia, and J. C. G. Andujar, "Smart power management during voltage dip in wind turbines," EP 2835529 B1, Aug 31, 2014.

[20] W. Janssen, H. Luetze, A. Buecker, T. Hoffmann, and H. R., "Wind turbine generator with a low voltage ride-through controller and a method for controlling wind turbine components," EP 1590567 B1, Jan 23, 2004.

[21] VDE, "Summary of the draft VDE-AR-N 4110:2017-02," *Verband der Elektrotechnik und Forum Netztechnik Netzbetrieb*, Feb 2017.

[22] A. D. Hansen, N. A. Cutululis, H. Markou, P. Sørensen, and F. Iov, "Grid fault and design-basis for wind turbines - Final report," *National Laboratory for Sustainable Energy*, Jan 2010.

[23] F. Iov, D. A. Hansen, P. Sørensen, and N. A. Cutululis, "Mapping of grid faults and grid codes," *Risø National Laboratory*, Jul 2007.

[24] D. Drossel and U. Harms, "Method for operating a wind turbine to control rotational speed of a wind turbine in the event of a grid error," US 011268492 B2, Apr 2, 2020.

[25] D. Drossel and K. Fischle, "Method for operating a wind turbine in the event of a grid error and a wind turbine for carrying out said method," US 8618685 B2, Dec 31, 2013.

[26] G. Joos, "Wind turbine generator low voltage ride through requirements and solutions," *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 2008.

[27] S. A. Barker, A. Klodowski, J. D. D'Arte, E. Larsen, and G. Drobnjak, "Apparatus for operating electrical machines," EP 1914877 B1, Oct 10, 2007.

[28] S. A. Barker, A. Klodowski, J. D. D'Arte, E. Larsen, and G. Drobnjak, "Wind turbine generator," EP 3460982 B1, Oct 17, 2007.

[29] M. Karimi-Ghartemani, M. Mojiri, A. Safaee, J. A. Walseth, S. A. Khajehoddin, P. Jain, and A. Bakhshai, "A new phase-locked loop system for three-phase applications," *IEEE Transactions on Power Electronics*, vol. 28, no. 3, p. 12081218, 2013.

[30] A. Dixit and S. Suryanarayanan, "Towards pitch-scheduled drive train damping in variable-speed, horizontal-axis large wind turbines," *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec 2005.

[31] A. M. Howlader and T. Senjyu, "A comprehensive review of low voltage ride through capability strategies for the wind energy conversion systems," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 643–658, 2016.

[32] G. Wenming, W. Yun, H. Shuju, and X. Honghua, "A survey on recent low voltage ride-through solutions of large scale wind farm," in *2011 Asia-Pacific Power and Energy Engineering Conference*, 2011.

[33] O. P. Mahela, N. Gupta, M. Khosravy, and N. Patel, "Comprehensive overview of low voltage ride through methods of grid integrated wind generator," *IEEE Access*, vol. 7, pp. 99299–99326, 2019.

[34] E. F. Morgan, O. Abdel-Rahim, T. F. Megahed, J. Suehiro, and S. M. Abdelkader, "Fault Ride-Through Techniques for Permanent Magnet Synchronous Generator Wind Turbines (PMSG-WTGs): A Systematic Literature Review," *Energies*, vol. 15, no. 23, p. 9116, 2022.

[35] A. K. Roy, G. R. Biswal, and P. Basak, "An integrated rule-based power management and dynamic feed-forward low voltage ride through scheme for a grid-connected hybrid energy system," *Journal of Renewable and Sustainable Energy*, vol. 12, no. 5, p. 056303, 2020.

[36] M. Nasiri, S. Mobayen, B. Faridpak, A. Fekih, and A. Chang, "Small-signal modeling of PMSG-based wind turbine for low voltage ride-through and artificial intelligent studies," *Energies*, vol. 13, no. 24, p. 6685, 2020.

[37] Y.-S. Kim, I.-Y. Chung, and S.-I. Moon, "Tuning of the PI controller parameters of a PMSG wind turbine to improve control performance under various wind speeds," *Energies*, vol. 8, no. 2, pp. 1406–1425, 2015.

[38] A. A. Ghany, E. Shehata, A.-H. M. Elsayed, Y. S. Mohamed, H. Haes Alhelou, P. Siano, and A. A. Z. Diab, "Novel switching frequency fcs-mpc of pmsg for grid-connected wind energy conversion system with coordinated low voltage ride through," *Electronics*, vol. 10, no. 4, p. 492, 2021.

[39] M. Zoghlami, A. Kadri, and F. Bacha, "Analysis and application of the sliding mode control approach in the variable-wind speed conversion system for the utility of grid connection," *Energies*, vol. 11, no. 4, p. 720, 2018.

[40] R. DeCarlo and S. Zak, "A quick introduction to sliding mode control and its applications," *elettronica, Università degli Studi di Cagliari (Department of Electrical and Electronic Engineering-DIEE, University of Cagliari, Cagliari CA, Italy)*, 2008.

[41] F. D. Bianchi, B. H. De, and R. J. Mantz, *Wind Turbine Control Systems*, p. 10300. Springer-Verlag London Limited, 2007.

[42] G. C. Pereira, S. V. Muddu, A. D. Roman-Ospino, D. Clancy, B. Igne, C. Airiau, F. J. Muzzio, M. Ierapetritou, R. Ramachandran, and R. Singh, "Combined feedforward/feedback control of an integrated continuous granulation process," *Journal of Pharmaceutical Innovation*, vol. 14, pp. 259–285, 2019.

[43] F. Raza and M. Hayashibe, "Towards robust wheel-legged biped robot system: Combining feedforward and feedback control," in *2021 IEEE/SICE International Symposium on System Integration (SII)*, pp. 606–612, IEEE, 2021.

[44] J. Duan, M. Li, T. C. Lim, M.-R. Lee, M.-T. Cheng, W. Vanhaaften, and T. Abe, "Combined feedforward–feedback active control of road noise inside a vehicle cabin," *Journal of vibration and acoustics*, vol. 136, no. 4, p. 041020, 2014.

[45] D. A. Bristow, M. Tharayil, and A. Alleyne, "A survey of Iterative Learning Control," *IEEE Control Systems*, vol. 26, p. 96114, Jun 2006.

[46] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, p. 10991121, 2007.

[47] M. Sun, Y. Zhan, S. Zou, and X. He, "Adaptive iterative learning control for tracking trajectories with non-equal trail lengths and initial errors," *IEEE Access*, vol. 9, p. 6085360864, Apr 2021.

[48] H. Tao, J. Li, Y. Chen, V. Stojanovic, and H. Yang, "Robust PointtoPoint Iterative Learning Control with trialvarying initial conditions," *IET Control Theory & Applications*, vol. 14, p. 33443350, Sep 2020.

[49] G. R. G. da Silva, A. S. Bazanella, C. Lorenzini, and L. Campestrini, "Data-driven LQR control design," *IEEE control systems letters*, vol. 3, no. 1, pp. 180–185, 2018.

[50] Y. Chen and K. L. Moore, "An optimal design of PD-type iterative learning control with monotonic convergence," in *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, pp. 55–60, IEEE, 2002.

[51] Y. Chen, Z. Gong, and C. Wen, "Analysis of a high-order iterative learning control algorithm for uncertain nonlinear systems with state delays," *Automatica*, vol. 34, no. 3, pp. 345–353, 1998.

[52] F. Memon and C. Shao, "An optimal approach to online tuning method for PID type iterative learning control," *International Journal of Control, Automation and Systems*, vol. 18, pp. 1926–1935, 2020.

[53] J. Shou, D. Pi, and W. Wang, "Sufficient conditions for the convergence of open-closed-loop PID-type iterative learning control for nonlinear time-varying systems," in *SMC03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, vol. 3, pp. 2557–2562, IEEE, 2003.

[54] T. Oomen and C. R. Rojas, "Sparse iterative learning control with application to a wafer stage: Achieving performance, resource efficiency, and task flexibility," *Mechatronics*, vol. 47, pp. 134–147, 2017.

[55] J. van Zundert and T. Oomen, "On inversion-based approaches for feedforward and ILC," *Mechatronics*, vol. 50, pp. 282–291, 2018.

[56] T. Oomen, "Formal conversation at the TU Delft," April 4, 2023.

[57] T. Oomen, "Part II: Lifted ILC - A Design Approach Based on Optimisation," 2023.

[58] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW reference wind turbine for offshore system development," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2009.

[59] S. P. Mulders and J.-W. van Wingerden, "On the importance of the azimuth offset in a combined 1P and 1P SISO IPC implementation for wind turbine fatigue load reductions," in *2019 American Control Conference (ACC)*, pp. 3506–3511, IEEE, 2019.

[60] Q. Lu, R. Bowyer, and B. L. Jones, "Analysis and design of Coleman transform-based individual pitch controllers for wind-turbine load reduction," *Wind Energy*, vol. 18, no. 8, pp. 1451–1468, 2015.

[61] S. Guntur, J. Jonkman, R. Sievers, M. A. Sprague, S. Schreck, and Q. Wang, "A validation and code-to-code verification of FAST for a megawatt-scale wind turbine with aeroelastically tailored blades," *Wind Energy Science*, vol. 2, no. 2, pp. 443–468, 2017.

[62] A. Abdelrahman, A. David, F. Samara, and D. Johnson, "Aeroelastic model validation of an Active Trailing Edge Flap System tested on a 4.3 MW wind turbine," *Journal of Physics*, mar 2014.

[63] YALMIP, "YALMIP. https://yalmip.github.io/ (accessed Jun. 17, 2023)," 2023.

[64] M. Q. Phan and R. W. Longman, "Higher-order iterative learning control by pole placement and noise filtering," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 25–30, 2002.

[65] R. Chi, Z. Hou, S. Jin, and B. Huang, "Computationally efficient data-driven higher order optimal iterative learning control," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 5971–5980, 2018.

[66] T.-Y. Kuc, J. S. Lee, and K. Nam, "An iterative learning control theory for a class of Nonlinear Dynamic Systems," *Automatica*, vol. 28, no. 6, p. 12151221, 1992.

[67] B. Huo, C. T. Freeman, and Y. Liu, "Data-driven gradient-based point-to-point iterative learning control for nonlinear systems," *Nonlinear Dynamics*, vol. 102, p. 269283, Feb 2020.

[68] B. Huo, C. Freeman, and Y. Liu, "Model-free Gradient Iterative Learning Control for Non-linear Systems," *IFAC-PapersOnLine*, vol. 52, no. 29, p. 304309, 2019.

[69] P. Janssens, G. Pipeleers, and J. Swevers, "Model-Free Iterative Learning Control for LTI systems and experimental validation on a linear motor test setup," *Proceedings of the 2011 American Control Conference*, Jul 2011.

[70] X. Bu, S. Wang, Z. Hou, and W. Liu, "Model Free Adaptive Iterative Learning Control for a class of nonlinear systems with randomly varying iteration lengths," *Journal of the Franklin Institute*, vol. 356, p. 24912504, Jan 2019.

[71] Q. Ai, D. Ke, J. Zuo, W. Meng, Q. Liu, Z. Zhang, and S. Q. Xie, "High-order model-free adaptive iterative learning control of pneumatic artificial muscle with enhanced convergence," *IEEE Transactions on Industrial Electronics*, vol. 67, p. 95489559, Nov 2020.

[72] S. Mishra, U. Topcu, and M. Tomizuka, "Optimization-based Constrained Iterative Learning Control," *IEEE Transactions on Control Systems Technology*, vol. 19, p. 16131621, Nov 2011.

[73] D. Liao-McPherson, E. C. Balta, A. Rupenyan, and J. Lygeros, "On robustness in optimization-based constrained iterative learning control," *IEEE Control Systems Letters*, vol. 6, p. 28462851, May 2022.

[74] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, p. 641657, 2000.

[75] P. Janssens, G. Pipeleers, and J. Swevers, "Model-free iterative learning control for LTI systems with actuator constraints," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11556–11561, 2011.

[76] K. Yovchev, K. Delchev, and E. Krastev, "State Space Constrained Iterative Learning Control for robotic manipulators," *Asian Journal of Control*, vol. 20, no. 3, p. 11451150, 2017.

[77] J.-X. Xu and X. Jin, "State-Constrained Iterative Learning Control for a class of MIMO Systems," *IEEE Transactions on Automatic Control*, vol. 58, p. 13221327, Apr 2013.

[78] X. Jin, "Fault tolerant nonrepetitive trajectory tracking for MIMO output constrained nonlinear systems using Iterative Learning Control," *IEEE Transactions on Cybernetics*, vol. 49, p. 31803190, Aug 2019.

[79] R. Chi, X. Liu, R. Zhang, Z. Hou, and B. Huang, "Constrained data-Driven Optimal Iterative Learning Control," *Journal of Process Control*, vol. 55, p. 1029, May 2017.

[80] J. Yu, H. Dang, and L. Wang, "Fuzzy Iterative Learning Control-based design of fault tolerant guaranteed cost controller for Nonlinear Batch Processes," *International Journal of Control, Automation and Systems*, vol. 16, no. 5, p. 25182527, 2018.

[81] K. Patan and M. Patan, "Neural-network-based iterative learning control of Nonlinear Systems," *ISA Transactions*, vol. 98, p. 445453, Sep 2020.

[82] D. Shen and Y. Wang, "Survey on stochastic iterative learning control," *Journal of Process Control*, vol. 24, p. 6477, May 2014.

[83] A. Madady, "PID type iterative learning control with optimal gains," *International Journal of Control, Automation, and Systems*, vol. 6, no. 2, pp. 194–203, 2008.

Glossary

List of Acronyms

AC	Alternating current
BHawC	Bonus Horizontal axis wind turbine simulation Code
DC	Direct current
EU	European Union
ILC	Iterative Learning Control
LVRT	low-voltage ride-through
MIMO	Multiple Input Multiple Output
MPC	Model Predictive Control
PLL	Phase-Locked Loop
PMSG	Permanent Magnet Synchronous Generator
SISO	Single Input Single Output
TSO	Transmission System Operator
UPS	Uninterruptible power supply
WT	wind turbine
ZVRT	zero-voltage ride-through