Iterative data-driven load control for flexible wind turbine rotors

Navalkar, Sachin

DOI
10.4233/uuid:cf1e2110-0ce7-4cc1-956b-f221d5f7b605

Publication date
2016

Document Version
Publisher's PDF, also known as Version of record

Iterative Data-Driven Load Control for Flexible Wind Turbine Rotors

ITERATIVE DATA-DRIVEN LOAD CONTROL FOR FLEXIBLE WIND TURBINE ROTORS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K. C. A. M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 06 december 2016 om 10:00 uur

door

Sachin Tejwant NAVALKAR

werktuigkundig ingenieur
geboren te Mumbai, India.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. G. A. M. van Kuik

Copromotor: Dr. ir. J. W. van Wingerden

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. G. A. M. van Kuik,	Technische Universiteit Delft, promotor
Dr. ir. J. W. van Wingerden,	Technische Universiteit Delft, copromotor
Prof. dr. ir. M. Verhaegen	Technische Universiteit Delft
Prof. dr. G. J. W. van Bussel	Technische Universiteit Delft
Prof. L. Y. Pao	University of Colorado at Boulder
Prof. dr. E. A. Bossanyi	DNV-GL, University of Bristol
Dr. ir. T. A. E. Oomen	Technische Universiteit Eindhoven

Keywords: Load control of wind turbines, data-driven control, recursive identification, repetitive control, iterative feedback tuning, free-floating flaps, individual pitch control, flutter detection

Printed by: Ridderprint

Front & Back: A cover image.

वाऱ्यावर सोडून पहा!

Contents

1

BACKGROUND AND THESIS GOALS

Als je sneller wilt spelen kun je wel harder lopen;
maar in wezen bepaalt de bal de snelheid van het spel.

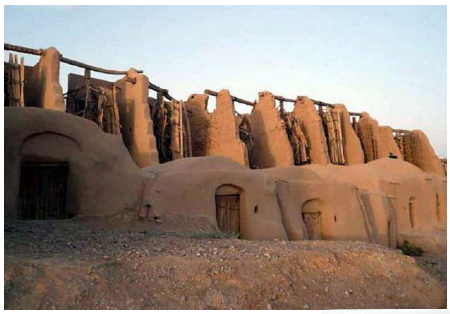
Johan Cruijff, the Netherlands (1947-2016),
describing the effect of exogenous influences on all theoretical efforts.

This introductory chapter sets up the background in which wind energy finds itself, in terms of its origins and geographical distribution today. The chapter discusses the expectations regarding the contribution of wind power towards the energy mix of the future, and enumerates the roadblocks to its implementation, the primary obstacle being the high cost of (offshore) wind energy. The concept of the 'smart' wind turbine, with flexible, active rotors is motivated, along with the need for data-driven control. Finally, the research questions sought to be answered are formulated, and the structure of the thesis is described.

Wind energy as a concept requires little introduction. Large, three-bladed machines, that rotate slowly to convert wind energy into electrical power, are a common sight in many parts of the world. The advantages and limitations of wind energy, and the motivating factors behind wind energy research, are, however, not as readily visible. This chapter presents the current position of wind energy, and the issues that need to be solved in terms of high dynamic lifetime loads, and hence high costs, that impede wind energy deployment. Active control, especially performed in an adaptive manner, can form a part of the solution to this problem, but there remain open research questions, some of which will be addressed in this thesis. First, the chapter studies the climate in which wind turbines are expected to operate, in order to understand the specific nature of challenges faced by wind energy today.

1.1. WIND ENERGY IN THE PAST

Wind energy has been harnessed by humanity since the beginning of recorded history, from the first windmills of ancient Iranians to the iconic Dutch windmills that dried



Iranian windmills,
circa 947 AD

Dutch windmills,
14th - 16th century

The Gedser wind turbine, J. Juul, Denmark, 1957

Figure 1.1: Wind turbine designs over the years, Ragheb (2014).

the land and boosted industrial output to levels then unheard of. With the advent of steam engines, wind energy was pushed to the fringes of economic activity: sailboats, for instance, became a hobbyist's quirk, where once they had been at the forefront of the global voyages of discovery.

Recent years have seen the development of the so-called 'Danish' design of wind turbines – slender, fast-spinning turbines, usually with three blades, producing electricity that is fed directly into the grid. The wind turbine designs can be seen in Fig. 1.1. Modern wind turbines were meant initially for meeting the needs of individual farmers; the design has now been extrapolated to massive utility-scale turbines, of rotor diameters more than twice the wingspan of the largest commercial aircraft.

The global distribution of wind power in 2016 can be seen in Fig. 1.2. In Europe, deployment of these wind power plants onshore has been concentrated in flat, wind-rich regions of low population density with a stable grid connection and a green government policy: the North Sea (and IJsselmeer) coastlines of the Netherlands and Germany, the desert regions of central Spain and, most strikingly, in Denmark, where an approximate 42% of all electricity produced today comes from wind turbines.

One of the other large wind energy producers is the USA, where the central region to the east of the Rocky mountains (the so-called 'mid-west') has significant wind potential

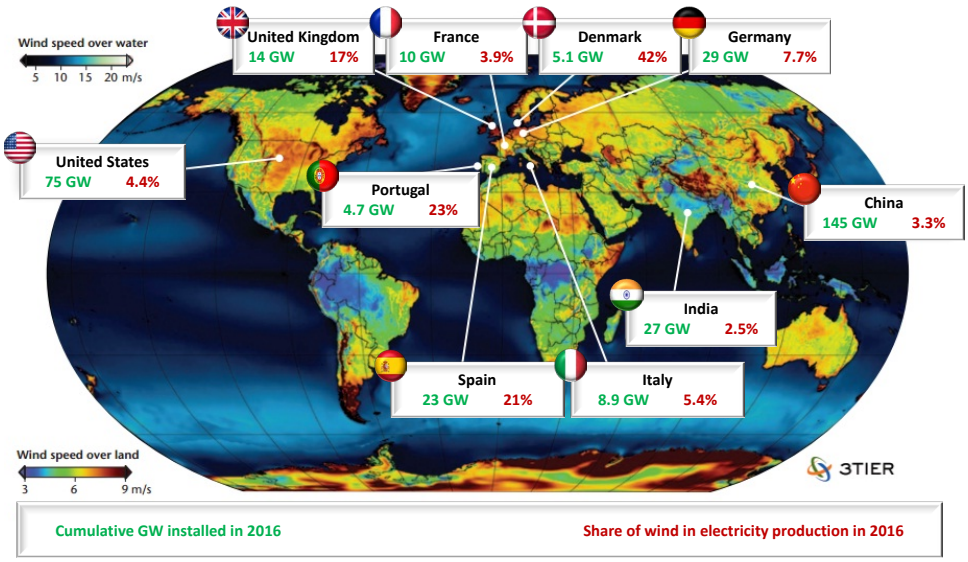


Figure 1.2: Global wind power production in 2016, modified from www.3tier.com.

where wind turbines need not compete with other economic activities for the use of the vast tracts of land (low land-use competition). Wind power plants in the US are hence also typically larger (upto 10x) than onshore farms in Europe. The wind energy market in the US has been significantly more volatile than elsewhere, driven primarily by the vagaries of the national economy and politics, with periods of frenetic wind energy deployment interspersed with periods in which it slowed almost to a standstill.

Two other large wind energy producing nations are China and India, with China being the largest single producer of wind energy today. Good locations for wind turbine siting are Inner Mongolia, the plateaux of Tibet, the west Indian marshlands of the Rann and the plains south of the Nilgiri range. As opposed to wind energy in developed nations, wind power deployment in these countries has been driven mainly by the large growth in energy consumption. However, these nations, like most of the world, face an economic limit in wind energy growth due to grid strength: beyond this limit, wind energy can grow only at a rate proportional to the growth of other energy sources.

Finally, a small but important part of wind energy comes from wind turbines located offshore. These windfarms, starting from the first Danish offshore windfarm of Horns Rev, are located mainly in and around the North Sea, to take advantage of the high wind speeds, and because the wind and wave characteristics of the North Sea are relatively well-understood. Offshore wind farms are also planned along the Atlantic coastline of the US, and in the East and South China Seas.

The map in Fig. 1.2 also shows that the largest and best reserves of wind power occur in relatively inaccessible locations: the mountains of Norway, Scotland and the Andes, the thinly populated regions of weak grid connection in Greenland, Western Sahara and

Somalia; and over Antarctica and the high seas. Barring a massive and unexpected human relocation in the near future, this wind resource is likely to remain untapped.

While the recent decades have shown a revival of wind energy as a viable addendum to energy production capabilities in diverse locations, the continued interest and investment in wind power is still a matter of sociopolitical debate. The next section questions whether wind energy will still make sense in the future.

1.2. A CASE FOR WIND ENERGY IN THE FUTURE

The energy mix is the portfolio of electricity generation plants in an economy. Modern industrial economies rely heavily on fossil fuel-based power plants, and, where favourable resources exist, on large hydroelectric plants. Wind energy is a relative newcomer in this energy mix and brings with it its own unique set of costs and benefits. The energy mix of the world is in flux: the western economy is in the process of decommissioning fossil- and nuclear-based power plants with outdated and potentially hazardous technology, while the developing economies attempt to bring energy consumption upto acceptable levels via the expansion of their energy production capabilities. This implies an impending shortfall in and a large demand for safe and reliable sources of energy. This section establishes the need for and challenges in deploying wind energy in specific, in terms of its primary characteristics: a localised, sustainable source of energy which is hindered by its diffuse nature and relatively high initial investment costs.

1.2.1. ENERGY INDEPENDENCE

It is no coincidence that wind energy development accelerated after the oil price shock in the 1970s. Even today, a majority of investment in wind power aims primarily at insulating electricity production, and hence, economic activity, against fluctuations in the price of oil and other fossil fuels.

1.2.2. SUSTAINABILITY

A longer-term motivation for wind energy is its minimalistic ecological footprint. Recent decades, with the ozone and leaded fuel crises, have shown that human activity can have severe adverse effects on the environment. With large uncertainties involved in the modelling of human-induced climate change, and the potentially catastrophic consequences of unchecked fossil-fuel consumption, global attention is currently focussed on sustainable sources of energy production, in which wind energy figures prominently.

1.2.3. DIFFUSENESS

One of the most significant disadvantages of wind energy is its diffuse nature. To match the nameplate capacity of the nuclear reactors at Doel in Belgium, a modern onshore wind farm would require roughly 1160 modern wind turbines, covering an area of 225 km^2. For comparison, the Doel power station covers an area of 0.8 km^2, excluding the area for mining, transport, storage of nuclear waste. These numbers do not present the full picture, since wind turbine land can also be used simultaneously for other purposes, such as farming. However, they give an idea about the investment required to move from nuclear power to wind power.

Unless it is supplemented by investment in energy storage solutions, wind power is incapable of supporting the electricity grid by itself; its variable nature is today offset by a strong, stable grid balanced by an adequate number of 'base-load' conventional power plants. The diffuse nature of wind energy is however also an advantage; since control over power output can be achieved at a micro-scale, and wind turbines can provide grid support in a much more responsive manner than conventional power plants: wind power plants can help stabilise the grid to reduce grid faults and improve power quality.

1.2.4. COST

What finally makes or breaks the appeal of wind energy is the costs associated, both in terms of capital costs and downtime costs. For a wind-rich onshore site, wind energy can indeed be cheaper than fossil fuel-based energy. However, especially in Western Europe, such sites are increasingly difficult to find due to land-use competition. The move offshore is logical in this sense, since these sites possess many desirable characteristics: high wind speeds, low turbulence, which reduces turbine loads, low land-use competition and a relative proximity to population-dense coastal regions that provide a ready market for energy consumption.

Unfortunately, offshore wind energy is not yet as cost-effective as its onshore counterpart. The primary cost, as can be expected, is the installation of massive rotating structures offshore, with the concomitant exponentially increased tower, foundation and support structure cost and complexity. Equally important is the downtime cost: a turbine that stops working due to a fault is a turbine that stops delivering returns on the initial investment. Maintenance is significantly more difficult offshore; it can become prohibitively expensive to convey maintenance personnel by helicopter or to wait for the right sea conditions to access the turbine.

1.2.5. THE CHALLENGE FOR ENGINEERS

Turbine manufacturers now design wind turbines that are even larger in size to access the economies of scale, upto and exceeding 164 m in diameter. A graph, typically found in most works of wind research, demonstrating the increase in the size of wind turbines, is seen in Fig. 1.3. Already today we have reached the limit of our engineering knowledge: wind turbines form the largest rotating structures ever designed, exposed to significantly stochastic and largely uncertain wind-induced loads. Traditional engineering wisdom is to prevent load-induced failure by compensating for uncertainty by introducing conservatism; in other words, engineers simply make turbines too strong to fail. This conservatism, interestingly, also found in the design of wind turbine controllers, feeds directly into the cost of modern offshore wind energy, making it less attractive as an economic investment.

Hence, the scientific community working in the field of wind energy today places special focus on flexible, lightweight turbine rotors, where the loads are held within acceptable limits using active or passive load control techniques. Such a turbine would potentially be able to compensate for an uncertain dynamically-changing environment by sensing disturbances and acting to counter their detrimental effect on the turbine loads. Such a 'smart' turbine would essentially tailor itself optimally to the conditions it

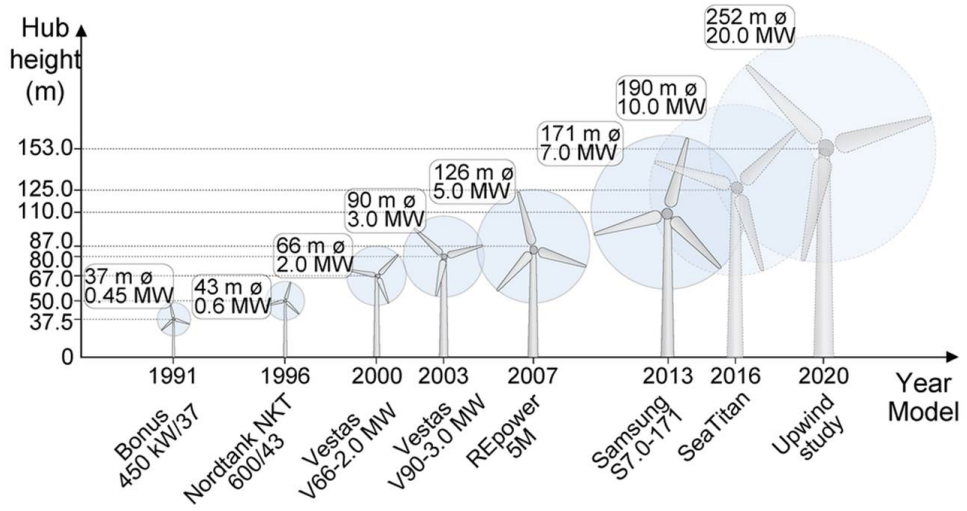


Figure 1.3: Increase in the size of modern wind turbines, Rodrigues et al. (2016).

finds itself in, and can thereby be manufactured to be more cost-effective than its classical counterpart. This forms motivations for the current research, which is part of the EU INNWIND program, a consortium of industrial and academic partners investigating innovations in modern wind energy, INNWIND (2012).

1.3. AERODYNAMIC CONTROL FOR FLEXIBLE ROTORS

Recent literature from the wind research community describes the flexible rotor of the future as a 'smart' rotor; one that can actively measure the incoming wind field, or its effects, and manipulate the flow around its blades so as to control and reduce structural loads. The degree of 'smartness' of a rotor is open to interpretation, covering as it does both conventional pitch control, as well as more exotic instrumentation like plasma actuators. A 'smart' flexible rotor may also react passively or actively. Passive control is the case wherein blades deform structurally in response to an undesirable load, so as to be able to mitigate it. This form of control was proposed in the 1800s for Dutch windmills by Arrenberg (1779), to increase the longevity of the erstwhile wooden blades. Active control involves an active element, such as a hinged flap, which is commanded to deform in response to a measured load.

As compared to active control, passive control can be more robust and more autonomous, with little to no external power consumption. It may also be limited in control authority, and require significantly more hardware modifications as compared to active control. Perhaps a commercial design of a smart wind turbine will eventually combine both passive and active control to exploit the advantages of both.

One of the first uses of active air flow control was the development of the pitch actuator, a device that rotates a blade partially or fully around its longitudinal axis. Commer-

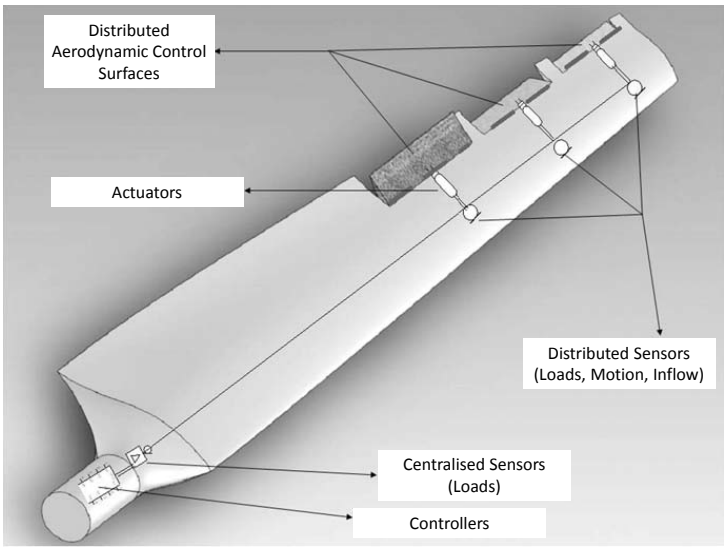


Figure 1.4: An example of an implementation of a smart flexible rotor, adapted from Barlas and Van Kuik (2010).

cial wind turbines today are invariably equipped with full-span pitch actuators, which are massive devices capable of rotating blades each weighing upto and exceeding 33 tonnes. The pitch action serves to limit the energy captured when operating at very high wind speeds, thereby reducing the loads on the turbine. These actuators serve admirably to respond to variations in the mean wind speed, as well as to the first dominant peak in the load frequency spectrum using Individual Pitch Control (IPC), but this appears to be the limit of their capabilities, as suggested by discussions with pitch manufacturers.

Blades equipped with trailing edge flaps, taking inspiration from the helicopter industry, have also been proposed and tested on turbine prototypes, an example layout has been shown in Fig. 1.4. Flaps change the curvature of the blade locally, and induce aerodynamic forces that can counter the wind-induced loads. These actuators serve to extend the applicability of the pitch actuator by addressing higher frequency loads in the load spectrum. While they have less control authority, their higher bandwidth in terms of time and space implies that flaps can potentially address loads and structural modes above and beyond that of pitch actuators. Currently, flaps have shown the potential to reduce loads that can also be targetted by the pitch actuators, the challenge for control engineers is to extend their applicability in combination with pitch actuation.

Other air flow control actuators, such as microtabs, microflaps and synthetic jets, have also been explored in the literature; however they have not yet been demonstrated directly on turbine prototypes. At the other end of the control spectrum, reliable sensors are necessary for the success of smart turbines. Commercial turbines are currently instrumented with strain gauges and accelerometers for measuring turbine response to the incoming wind. Feedforward measurements are also now becoming available via Li-

dar (light detection and ranging) instruments that can measure the incoming wind field and issue a much more suitable turbine actuation command.

The major issue with the new actuators and sensors is that they usually produce signficant changes in the dynamic behaviour of the underlying system. Not only does the altered system dynamics depend upon the unique physical composition, configuration and condition of each turbine, it also varies with wind conditions. With increasing aeroservoelastic complexity as we move towards the high-frequency end of the load spectrum, the task of system modelling, robust control design and controller tuning can become either extremely time-consuming or unnecessarily conservative, with no ultimate guarantee of optimal turbine performance, motivating the need for an iterative autonomous controller design approach, that uses operational data to improve turbine performance.

1.4. INTEGRATED DATA-DRIVEN CONTROL

Commercial wind turbines use control for both regulating power production and for load control. These controllers are (despite the best efforts of the control community) typically PID (proportional-integral-differential), which provides easy tuning knobs for achieving the desired turbine performance. With increasing complexity, such simple controllers are no longer adequate; the unintended cross-influence of actuators on different loads needs to be decoupled using special engineering tricks. With new actuators and sensors, the complexity of decoupling can increase exponentially and it becomes desirable to consider integrated, multivariable control techniques as more viable alternatives to PID controllers. A further drawback of PID controllers is that manual tuning of parameters is unlikely to result in optimal system performance.

To ensure stability and determine the expected performance of a (PID) controller, tuning is usually done in conjunction with a system model. Such a model can be obtained by first principles; however, several turbine parameters (for instance, structural damping) are difficult to estimate, and can vary significantly across turbines. Further, physical models may contain irrelevant dynamics that can lead to very high-order controllers and prolong simulation and validation times needlessly. Finally, since the system model will not be exactly the same as the actual turbine, retuning of the parameters is often done in the field to adjust the performance of each individual turbine, a process that causes substantial delays in the commissioning after construction or recommissioning after maintenance.

One of the alternatives is to use system identification methods, which use experimental input-output data from a wind turbine to determine its dynamic behaviour to synthesise a model. Such a method skips the stage of determining physical parameters and focusses purely on relevant system dynamics, simplifying the controller design process. However, this method still requires manual intervention for conducting experiments on each individual turbine and for tuning an effective controller.

Much interest has been devoted to data-driven control techniques by the control engineering community. In effect, these techniques combine identification and controller synthesis autonomously and require in principle no user interaction. Alternatively, these methods directly use operational data to tune fixed-structure controller gains directly. Such techniques have been demonstrated for simple systems; however, guaranteeing

stability is notoriously hard. Furthermore, unless the structure of the plant is exploited, the computations required for data-driven control can become extremely difficult to implement on a real system.

Data-driven control techniques like Iterative Learning Control and Iterative Feedback Tuning have shown considerable potential for online adjustment of controller performance to adapt to a variety of different systems in industrial environments. As per their name, these techniques are iterative, in that they update the control law over a number of iterations, in such a manner that control performance improves over time with limited computational complexity per iteration. Such techniques, combined with system identification and adjusted to exploit the structure of the modern wind turbine system, could prove to be interesting for developing self-commissioning wind turbines that can autonomously optimise their load alleviation performance given any current and future set of actuators and sensors.

This motivation yields the fundamental question sought to be answered:

Main research question:

How can we use operational data to synthesise a fully multivariable load controller for a flexible wind turbine rotor, that is able to improve its own performance autonomously in real-time?

1.5. STRUCTURE OF THE THESIS

In order to answer the main research question, the thesis, like its title is divided into two parts. The first part explores and extends the concept of iterative data-driven control to make it more applicable for real-time implementation for wind turbine load control, while the second part explores the implementation of these techniques on flexible wind turbine rotors, in the simulation environment and on a scaled wind turbine.

The flow of logic, and the reading order, can be visualised in Fig. 1.5. Each of the chapters in the thesis is geared towards exploring a different aspect of the main research question. As such, the following component questions will be addressed in the sequel:

RESEARCH SUBQUESTIONS

PART I: ITERATIVE DATA-DRIVEN CONTROL

Chapter 2: State of the art

What is the current state of the art in iterative data-driven control, and how can it be made suitable for the current application?

Chapter 3: Closed-loop nuclear norm-based recursive regularisation

How can recent advances in low-variance system identification be modified to fit a closed-loop, online environment?

Chapter 4: Subspace Predictive Repetitive Control

How can online system identification be combined with controller synthesis to minimise periodic loads, with precise control over the shape and smoothness of the actuator commands?

Chapter 5: Iterative Feedback Tuning for LPV systems

How can the gains of low-order fixed-structure LPV controllers be optimally tuned for LPV systems?

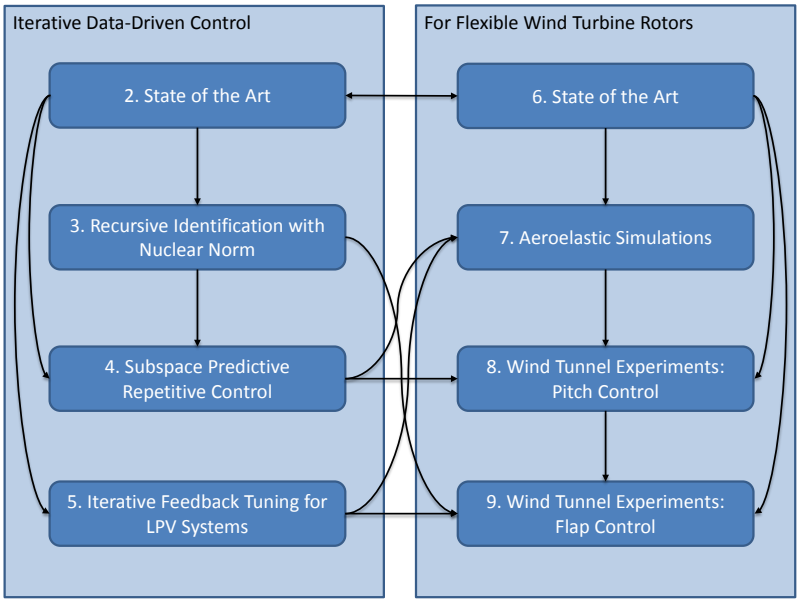


Figure 1.5: Flow of logic, and reading order of this thesis.

PART II: FOR FLEXIBLE WIND TURBINE ROTORS

Chapter 6: State of the art

What is the current state of the art in the load control of flexible rotors, and how can the controller be improved to enhance performance?

Chapter 7: Aeroelastic simulations

How do the control strategies discussed in Part I behave in a fully non-linear simulation environment for commercial wind turbines?

Chapter 8: Wind tunnel experiments: Pitch Control

How do the control strategies discussed in Part I affect the loading behaviour of a scaled prototype of a pitch-controlled wind turbine, under controlled, wind tunnel conditions?

Chapter 9: Wind tunnel experiments: Flap Control

How can a pitch-controlled turbine be extended to include trailing-edge flaps, and how should the control strategies of Chapter 8 be modified to achieve optimal load control for the wind turbine?

Chapter 10: Conclusions and recommendations

I

ITERATIVE DATA-DRIVEN CONTROL

2

STATE OF THE ART

En ekspert er en person, som har begået alle der fejl,
som det er muligt at begå inden for et begrænset område.

Niels Bohr, Denmark (1885-1962),
describing the iterative and data-driven nature of scientific endeavour.

This chapter explores the evolution and current state of the art of data-driven control. Classical control is briefly introduced, as the best-possible theoretical control design approach for ideal systems that can be modelled without uncertainty, subject to disturbances whose stochastic properties are well-known. For practical systems, where such information is not available, robust control design is described as the alternative; however the inherent conservatism of robust control may be considered one of the motivating for seeking other alternatives, such as data-driven control approaches. Two distinct approaches towards data-driven control are then discussed, along with their evolution over the years: in the two-step data-driven approach, input-output data is first used to develop a system dynamics predictor, based on which a (receding horizon-) control law is synthesised and implemented. The second approach discussed is the direct data-driven approach, wherein input-output data is used to construct the gradient of closed-loop performance with respect to controller parameters, and gradient-based optimisation schemes are used to optimally tune these parameters. Both approaches show potential for use in complex realistic applications like wind turbine load control, however further research into these techniques is deemed necessary for their realisation in practice.

2.1. PRELIMINARIES TO DATA-DRIVEN CONTROL

2.1.1. THE CLASSICAL VIEWPOINT: CONTROLLING AN IDEAL SYSTEM

Traditional wisdom suggests that a controller be designed to ensure that the closed loop is stable, and to achieve a certain level of performance in terms of reference tracking or disturbance rejection. Classically, the synthesis of the controller and the analysis of the closed-loop system demands a high-accuracy, low-order model of the system to be

controlled, preferably linear, that still incorporates the relevant dynamics within the frequency band of interest, Ogata (1997). In the case where the states are required to be reconstructed from the measurements, an observer is required to be synthesised: classical control design therefore also demands knowledge of the first- and second-order properties of stochastic disturbances, and/or knowledge about the characteristics of deterministic disturbances. The basic building block of control theory is the Linear Time-Invariant (LTI) system that can be expressed either as a transfer matrix or with a state-space realisation of the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k. \end{aligned} \tag{2.1}$$

Here, the controlled inputs to the system are $u_k \in \mathbb{R}^{n_u}$ and the measured outputs are $y_k \in \mathbb{R}^{n_y}$. The term $x_k \in \mathbb{R}^n$ is the state vector; its dimension is defined as the order of the system. The signals $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^{n_y}$ are the process and measurement noise, uncontrollable quantities that often require stochastic representations. The system behaviour is determined by the matrix tuple (A, B, C, D), of appropriate dimensions. For an LTI system, these matrices are deterministic and do not change over time.

When the system sought to be controlled can be described exactly by a description of the form of (2.1), where stochastic properties of noise are known, it is possible to synthesise an LTI controller (such as an LQG controller, Ogata (1997)) aimed at optimising a certain performance criterion. However, it may be impossible to guarantee the performance of a classical controller, Doyle (1978), since practical systems pose further control challenges on account of the inherent uncertainty in the modelling step.

2.1.2. CONTROL FOR UNCERTAIN SYSTEMS

The modelling of complex systems like wind turbines is rarely exact, Versteijlen et al. (2016): system parameters can often only be estimated to within a certain tolerance. For instance, for a flexible beam, the modal frequencies depend on the mass and structural stiffness, and can typically be predicted with high fidelity. The damping of the different modes, however, is far more difficult to measure, and one is forced to resort to some form of experimental testing to infer the modal damping based on the decay in the vibration measurements. Such parametric uncertainty can strongly influence the analysis of the closed-loop. Further, first-principles modelling often yields unwieldly, non-linear models that have to be simplified for controller design. For instance, the finite-elements model of a wind turbine blade is usually reduced to a second- or fourth-order model for blade load controller design; the (non-linear) effect of the higher-order modes, and of aerodynamic lag are then ignored. Such simplifications produce modelling uncertainty, typically in the high-frequency range of the spectrum.

Such a system, subject to uncertainty, can be modelled using an uncertain system representation using the uncertain matrix tuple (A, B, C, D), where

$$A = \bar{A} + \delta A, \tag{2.2}$$

where $\bar{A}$ is the exactly known 'nominal' or average value of the matrix A, and the term δA is an unknown, but bounded, uncertainty. The other matrices can be described in a similar manner.

A consequence of the uncertainty arising out of modelling is that a classical controller designed for the nominal plant $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ within the set of possible plants may practically cause a degradation of closed-loop performance, and may possibly even lead to closed-loop instability. One solution to this issue is to design the controller for the worst-possible combination of uncertain model parameters so that stability and performance can still be achieved; this is the 'Robust' approach to controller design and usually involves the minimisation of an $\mathcal{H}_\infty$ criterion, Skogestad and Postlethwaite (1996). Robust controllers have enjoyed considerable success in the literature, but their performance is contingent on a reasonably accurately bounded uncertainty description $(\delta A, \delta B, \delta C, \delta D)$. Robust controllers, per definition, are synthesised to be conservative, and may not command the best performance possible from the true system. Further, as the uncertainty set grows, so does the conservatism of the controller. In extreme cases, with large model uncertainties, robust control designs may indicate that a stabilising controller is not feasible.

While time-invariant nominal systems have been considered here, several practical systems show dynamics that vary with time. This time variance can be modelled either as a bounded uncertainty, or, with less conservatism, using a time-varying system description.

2.1.3. CONTROL FOR TIME-VARYING SYSTEMS

Systems that do not admit a Linear, Time-Invariant (LTI) system description, but display dynamics that vary with time, can be modelled with time-varying parameters (A_k, B_k, C_k, D_k). Globally stabilising time-varying system is considerably more complex than the equivalent LTI systems, Kwon and Pearson (1978). Such systems can be linearised around operating points, and LTI controllers can be designed for these operating points. With some form of interpolation, a time-varying controller can then be approximated for the entire range of operation. Such 'gain-scheduled' controllers, Leith and Leithead (2000), may not necessarily be optimal, and a small increase in non-linearity can lead to a combinatorial explosion in the number of operating points required for reasonably adequate performance.

A special case of time-varying systems are Linear Parameter-Varying (LPV) systems, Shamma (2012), where the system dynamics change as a function of scheduling parameters. The system matrices can then be expressed as:

$$A_k = A^{[0]} + \sum_{i=1}^{n_\psi} \psi_i(\mu_k) A^{[i]}, \tag{2.3}$$

and similarly for the other matrices. Here, $\mu_k \in \mathbb{R}^{n_\mu}$ is the vector of scheduling parameters. As seen, the matrices (A_k, B_k, C_k, D_k) are affine combinations of the n_ψ number of basis functions ψ_k scheduled on μ_k. As an example of LPV systems, wind turbine dynamics have been shown to depend strongly on the operating wind conditions, primarily on the ambient wind speed, Van Wingerden et al. (2010a). For LPV systems, it is possible to synthesise a global controller (LTI or LPV) that delivers good performance for all scheduling trajectories, Emedi and Karimi (2013). However, LPV control design is also highly sensitive to uncertainty: modelling errors/approximations or noise in the measurement of the scheduling parameter can drastically degrade controller performance.

Robust approaches can, as before, cause intractability or excessive conservatism.

2.1.4. DATA-DRIVEN CONTROL

An alternative to the robust model-based controller design approaches is the data-driven controller design approach. Here, input-output data collected online is used to synthesise a controller that optimises a certain criterion. Two kinds of data-driven approaches have been studied in the literature: the two-step data-driven control approach, and the direct data-driven control approach. In the two-step approach, the objective is to first identify the system parameters from the input-output data, Ljung (1987), and in the next step synthesise an optimal control law. On the other hand, the direct approach starts with an initial nominally stabilising controller and then, based on the measured performance, iteratively improves the controller parameters until they converge to their optimal values. All data-driven approaches require a persistency of excitation; in other words, the input data has to excite all the relevant modes of the system sought to be controlled.

Data-driven approaches are usually aimed at LTI systems with extremely uncertain properties, or for time-varying systems the changing dynamics of which are difficult to predict.

LTI SYSTEMS

For the case where the underlying system is LTI and the disturbances acting on the system are stationary, both data-driven approaches converge to an LTI control law. Since the controller is then tailored to the specific characteristics of the system, it is typically less conservative than a controller designed using robust control techniques. However, since the data-driven approach involves the estimation of system parameters or performance gradients from noisy data, these estimates are themselves uncertain. Providing robustness proofs for data-driven methods is challenging. Proving the stability of the controlled system, for model and disturbance uncertainty, even in the LTI case, is for many data-driven techniques still an open question, Goodwin et al. (1980), Åström (1987); very few analyses exist in the literature for time-varying systems.

SLOWLY TIME-VARYING SYSTEMS

For the case where the underlying system or systemic constraints change slowly with time, data-driven approaches usually consider the system to be instantaneously LTI, and iteratively update the synthesised control law. Effectively, the control law itself does not remain constant, but changes progressively over time. If the variation in dynamics is slow, the controller may be close to optimal throughout its operation. Here, the data-driven quasi-LTI controller is clearly superior to an LTI controller designed offline, which will be by definition more conservative as it cannot update itself to changes in system characteristics. However, as before, proving stability can be difficult, especially since the variation in system dynamics always causes a lag in the estimation of system parameters or gradients. Also, continuous reëstimation is required to be performed online to ensure that the control law does not become outdated and suboptimal. An alternative is global identification and control law formulation, which often demands a more specific system description, such as the LPV formulation.

LPV SYSTEMS

For an LPV system, it is possible to synthesise an LPV controller in a data-driven manner. Unfortunately, due to the complicating presence of the scheduling dependence, a significantly larger amount of data is demanded by the data-driven approach to optimise the controller parameters, since the persistency of excitation condition applies to both the input data as well as the scheduling parameters. On the other hand, an LPV data-driven approach can in principle converge to a constant global LPV control law that is applicable for all scheduling trajectories. Once convergence is reached, the iterative optimisation process can be terminated, and further controller modification is no longer necessary. In the literature, very little attention has been given to the formal treatment of such controllers.

This chapter studies the literature regarding data-driven control using both the two-step, as well as the direct approaches.

2.2. TWO-STEP APPROACH

The two-step approach to data-driven control, also called the 'self-tuning' approach by Mosca (1995), separates the online, recursive identification of system parameters and control law synthesis into two separate steps.

2.2.1. RECURSIVE ONLINE IDENTIFICATION

The first step, online system identification can typically be performed using either parametric models, Ljung and Söderström (1983), where a specific parameterised structure, such as a Box-Jenkins structure of fixed order is assumed for the underlying dynamic system. Alternatively, subspace identification can be used, as in Houtzager et al. (2012), where the subspaces of data matrices are manipulated to obtain an (LTI) model, often using a rank-revealing intermediate step used for model order reduction. Both methods have their advantages and drawbacks, as described in Verhaegen and Verdult (2003). As long as the identified model lies within the set of parameterised models, parametric identification typically yields accurate models, and affords the possibility to identify non-linear, time-varying models in an online manner. On the other hand, parametric identification often requires the solution of complex, non-convex optimisation criteria that can be difficult to perform online, and are susceptible to the problem of local minima. Subspace identification usually uses convex optimisation criteria that can be efficiently solved online, even for highly multivariable systems. It is, however, non-trivial to prove the accuracy of the identified (typically linear) model, or the effect of modelling error on the adaptively controlled system.

Once system estimates are available, it is possible to proceed to the second step of the two-step approach, and synthesise a control law online.

2.2.2. MODEL PREDICTIVE CONTROL (MPC)

The standard approach to MPC involves the prediction of the system states and outputs over a time horizon, as a function of the control sequence. The deviation of the states (or outputs) from the desired value is penalised, and the optimal control sequence is synthesised, subject to practical system constraints. This process is repeated at every

time instant in order that the optimal control input is continuously updated. The MPC optimisation function, for the case of full state knowledge, is described as:

$$\min_{u_1,\dots u_{N_c}} \sum_{i=1}^{N} (x_i - x_{i,\text{ref}})^T Q (x_i - x_{i,\text{ref}}) + \sum_{i=1}^{N_c} u_i^T R u_i. \tag{2.4}$$

Here, $x_{k,\text{ref}}$ is defined as the ideal state trajectory that the system is required to follow, while Q and R are the weighting matrices that define the trade-off between controller performance and control effort. Further, N and N_c are the prediction and control horizon, respectively. This optimisation function is solved at every time instant, subject to system dynamics and other physical constraints, and control is implemented in a receding horizon fashion.

In Dougherty and Cooper (2003), an interesting half-way step between model-based MPC and data-driven MPC has been proposed. The system is modelled as a set of linear plants, each describing the system dynamics at one specific operating point, and an MPC controller is synthesised for each of these linear models. Based on the current input-output data, the actual control action implemented is a weighted average of the control actions demanded by each LTI MPC controller. While this approach simplifies the system identification component significantly, its drawbacks are similar to those of a gain-scheduled controller, viz. that good linear models are required, the number of models required can increase exponentially, and linear interpolation schemes are rarely amenable to optimality proofs.

MPC can also readily be extended to the two-step approach for data-driven control. At every instant that a new system model estimate is available, the MPC optimisation routine is updated using these estimates, and the controller is able to adapt to the changes in system dynamics.

2.2.3. ADAPTIVE MPC USING PARAMETRIC MODELS

Since the introduction of MPC, several investigations have been made into its synergy with system identification, for instance the books by Bitmead et al. (1990) and Mosca (1995), which combine MPC with parametric identification to obtain a data-driven control law. As described above, it is only when the system admits a model representation linear in parameters that the identification can be carried out efficiently using RLS; otherwise, recourse needs to be taken to iterative methods for identification. Bitmead et al. (1990) highlight the shortcoming of this approach: if the true system does not belong to the model set in which the identification process seeks a solution, there is no guarantee that the devised controller will stabilise the system. For the sake of robustness, therefore, one needs as general a parameterisation as possible. This requirement directly contradicts the constraint that the number of parameters to be optimised should be held low for low-variance estimation from a limited amount of data.

Neural networks form a specific form of parametric modelling, well-suited for the identification of highly non-linear data-generating systems; the extension of neural networks to data-driven control has been described by Narendra and Parthasarthy (1990). However, as with many other parametric identification methods, a non-convex optimisation process is involved in the estimation of the weights and biases of the neurons

in the network, and the real-time implementation of neural network-based data-driven control can be challenging.

Adaptive MPC with parametric models targets SISO (single input single-output) systems specifically, extensions to MIMO systems can greatly increase the number of coefficients to be identified, and hence the amount of data required to achieve good model estimation.

Even for an underlying LTI system, proving convergence and stability of the data-driven approach can be difficult, Åström and Wittenmark (1973); Mosca (1995) provides a convergence proof only for the very specific case where the prediction horizon is one step ahead. For time-varying systems, the stability of data-driven controllers can only hold for a slow rate of adaptation of the controller. If this rate is slower than the rate at which plant dynamics change, then there may occur a controller-plant mismatch which could possibly cause closed-loop instability. Bitmead et al. (1990) also describes synthetic approaches to restoring stability from an unstable operating point: *projection*, or the reinitialisation of plant estimates, and *leakage*, the forcing of plant estimates to physically acceptable values. Both methods are ad hoc in the sense that they require prior knowledge, and yield highly non-linear behaviour that is difficult to analyse.

Several of the adaptive MPC methods described in the references, e.g. Mosca (1995), utilise parametric identification methods that depend, not only on the plant dynamics, but also on the controller dynamics, which, in an adaptive application, change over time. This complicating factor reduces the fidelity of the identification part of the cycle. Subspace-based methods typically do not suffer from this drawback.

2.2.4. SUBSPACE PREDICTIVE CONTROL

Traditional approaches to controller design based on experimental data perform the identification and controller design steps separately, with an intermediate step being the synthesis of a full system model. A subspace-based alternative is the use of canonical variate analysis that uses the covariance structure of the past data to obtain the state trajectory predictions required by the MPC cost function, as in Larimore (1990). With subspace identification, it is also possible to derive the controller directly from the subspace predictor, rendering a state-space realisation (A, B, C, D) unnecessary. Such a combination of (partial) subspace identification and predictive control, called Subspace Predictive Control (SPC), was introduced by Favoreel and De Moor (1999), and extended to cover the functionality of a standard MPC controller with an $\mathcal{H}_2$ cost criterion by Kadali et al. (2003) and an $\mathcal{H}_\infty$ cost criterion by Woodley et al. (2001). The equivalence of the controller derived from subspace identification to a standard MPC controller, and to LQG in the infinite horizon case, has been given in Favoreel et al. (1999), however, the effect of parameter estimation error is not investigated.

To be noted in this discussion is the fact that identification for data-driven control demands closed-loop identification, since the plant is usually operated in closed loop with the controller that is being recursively updated by new parameter estimates from the identification process. Both parametric and subspace methods, Van der Veen et al. (2013), can be augmented to account for the closed-loop nature of the data, while some identification algorithms such as Predictor-Based Subspace Identification (PBSID), Chiuso (2007), are inherently able to perform system identification from

closed-loop data. PBSID has been used for SPC in Dong et al. (2008). In Dong and Verhaegen (2009), the authors evaluate the uncertainties associated with the SPC predictor estimation, and lay down a cautious framework for control design robust to these uncertainties. SPC lends itself to different adaptive applications: Hallouzi and Verhaegen (2008) investigate the extension to fault tolerance: the simultaneous identification and control law synthesis allows the controller to detect the occurrence of faults and to adapt the control action for specific known or anticipated faults.

While subspace identification simplifies the first step of the two-step approach by enforcing convexity, this method of data-driven control is contingent on the system admitting a (temporally local) LTI model that needs continuous reïdentification. Extensions to LPV or time-varying systems, Lovera et al. (2013), require large amounts of experimental data or yield high-variance estimates that can adversely affect the stability of the data-driven control approach. The order of the controller synthesised is also usually significantly larger than strictly necessary for control. As an alternative, direct data-driven approaches have been explored in the literature that avoid the system identification step. Low-order controllers, such as PID controllers, can herewith be optimised directly using gradient-based methods.

2.3. DIRECT DATA-DRIVEN CONTROL

2.3.1. ITERATIVE FEEDBACK TUNING

The concept of Iterative Feedback Tuning (IFT) is explained in Hjalmarsson (2002), and it can be described using Fig. 2.1. The system to be controlled, G is connected in closed loop with a controller $C(\rho)$ that is parameterised using the parameters ρ, initialised to an arbitrary value that stabilises the system. In addition to the control input commanded by the controller, u, the auxiliary input q is fed to the system in order to analyse its behaviour. The terms r and v correspond to exogenous known or unknown disturbances. Based on the system response y, the gradient of system performance J with respect to ρ, viz. $\frac{\partial J}{\partial \rho}$ can be determined. Typically, at least two experiments are needed to find this gradient; they are termed the 'reference' and the 'gradient' experiments. For each of these methods, the respective auxiliary input is given by:

$$q_{\text{reference}} = 0, \tag{2.5}$$

$$q_{\text{gradient}} = \frac{\partial C(\rho)}{\partial \rho}(r - y_{\text{reference}}), \tag{2.6}$$

where the $_{\text{reference}}$ and $_{\text{gradient}}$ subscripts indicate quantities measured in the reference and the gradient experiments, respectively. With these quantities, the (ergodically unbiased) gradient $\frac{\partial J}{\partial \rho}$ can be estimated. Gradient-based methods can then be used to optimise the controller parameters such that the closed-loop system performance criterion is maximised.

Gevers (2002) demonstrates in a review paper that IFT can be applied readily to multiple industrial applications. To be noted is the fact that, although IFT was originally designed for SISO systems, it can be directly extended to apply to multivariable systems. However, multivariable systems require more than one gradient experiment; generally, one gradient experiment per controller parameter is required for controller tuning.

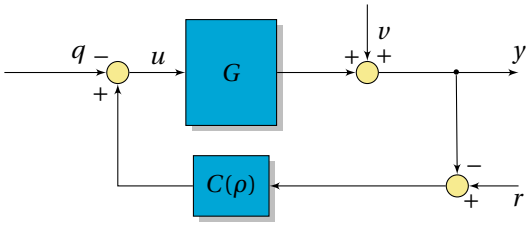


Figure 2.1: Iterative Feedback Tuning experiment layout for an LTI controller in closed loop with an LTI plant.

As for general adaptive systems, proving robust stability for IFT can be challenging. An incorrectly designed performance criterion can lead to closed-loop instability even in the ideal noise-free case. As a result of this, Lyapunov theory and passivity theory have been suggested by Mosca (1995) for implementation in direct data-driven control.

Further, the IFT performance criterion is typically non-convex, implying that optimality may be achieved, depending on the initial controller configuration, only in a local sense. Hjalmarsson (2002) describes a method to track the stability of the closed loop through iterations, wherewith the step size can be manipulated to ensure that closed-loop instability does not occur during the IFT process.

2.3.2. VIRTUAL REFERENCE FEEDBACK TRACKING

In a manner similar to IFT, Virtual Reference Feedback Tracking (VRFT), Campi and Savaresi (2006), utilises a parameterised controller $C(\rho)$. VRFT then postulates an ideal or desired closed-loop transfer function M, which delivers the performance required from the controlled system: this approach is similar to the Model Reference Adaptive Control (MRAC) discussed in Mosca (1995). Based on input-output data, the controller parameters are then optimised to minimise the distance between the actual output and the output delivered by the transfer function M. In other words, the minimisation criterion in VRFT is given by:

$$\min_{\rho} \| C(\rho) e - u \|, \tag{2.7}$$

where $e = (M^{-1} - z^{-1})y$, the deviation of the actual performance from the desired one. It can be directly appreciated that this approach is equivalent to parametric identification of the optimal controller, based on input-output data, and the theory of parametric identification is also applicable here.

The VRFT optimisation cost function forms a convex envelope of the IFT cost function; the VRFT optimisation can be performed in one single step. However, while the IFT controller tuning process eventually leads to a (local) minimum, the convex relaxation implies that the VRFT controller may be arbitrarily far from the optimal controller.

The data-driven techniques developed in the literature have also been extended to apply to LPV systems, Dong et al. (2009), Formentin and Savaresi (2011). The LPV extension generally requires a larger amount of experimental data for convergence; however the techniques are therewith applicable to a larger class of controllable systems.

2.4. CONCLUSIONS

Both the two-step and the direct approaches to data-driven control show potential for the control of complex realistic systems with unknown or time-varying dynamics. However, the data-driven approach also presents considerable challenges in that guarantees for convergence and optimality are difficult to postulate. The two approaches also differ from each other in implementation, and these differences dictate the control setting wherein they would be most suitable.

The two-step approach, specifically when it uses subspace identification, typically poses a convex optimisation problem that can be solved with relative ease online, and is not susceptible to local minima. While the intermediate step of system model realisation is not strictly necessary for control, it provides useful information in practice. For instance, for systems that transition between stable and unstable operating points, the system identification step reveals information about the current regime of operation. Further, the control law can be synthesised to be globally optimal, in an LQG sense. For the case where the system identification provides high-fidelity estimates of the true LTI system parameters, the two-step method approaches classical offline control design in terms of stability and performance.

With increasing system complexity and non-linearity, the direct data-driven approach becomes more attractive. For such complex practical systems that demand control, fixed-structure low-order controllers, such as PID controllers, are typically already in place. However, a manually-tuned fixed-structure controller is rarely optimal, the optimal controller parameters may also drift with time. For such a case, it would be desirable to use a direct approach that starts from the initial stabilising controller parameters and converges to their optimal values. Since they use gradient-based methods, direct methods can readily find the local minimum of the cost function, and can be used to fine-tune other, more global control design techniques. However, since no knowledge of the true system is used in controller design, proving steady-state stability or convergence can be challenging.

The objective of the first part of this thesis is to set up data-driven control approaches in a manner suitable for wind turbine load control; in the sequel the developed algorithms will be validated numerically and experimentally to assess their potential in real-time under realistic conditions of operation.

3

CLOSED-LOOP NUCLEAR NORM-BASED RECURSIVE IDENTIFICATION

For an example of a recursive citation, please refer to Navalkar (2016).

Paolo Pozzi, the Netherlands (2016).

Recursive identification oftens forms the first step in online data-driven control. This chapter explores the extension of standard recursive online identification methods designed to reduce the variance in the estimate of the system parameters, while retaining the sensitivity to changes in system dynamics. First, the closed-loop Predictor Based Subspace Identification (PBSID) cost function is augmented with a nuclear norm-based cost function to this effect. Next, since the new cost function uses the system Markov parameters as optimisation variables, which are assumed to vary slowly over time, a recursive version of this extended PBSID method is readily set up. The recursive solution to this optimisation problem is then synthesised using the Alternating Direction Method of Multipliers (ADMM). This method demands a time-consuming recursive singular value thresholding step, which is replaced by a fast randomised method that also recursively updates the range of the system based on previously used data. The benefits of using the nuclear norm for recursive system identification are then evaluated using a case study.

3.1. INTRODUCTION

The first step of the two-step data-driven control approach is the identification of system dynamics. Since a control law, adaptive to changing dynamics and environments, is to be synthesised on the basis of the identified system parameters, the identification process is designed to be able to operate online, in a recursive manner, and preferably in closed loop.

Parts of this chapter have been published in the Proceedings of the American Control Conference, Boston, USA, Navalkar and Van Wingerden (2016) and submitted to the IEEE Transactions on Control Systems Technology, Navalkar and Van Wingerden (2016).

The online identification of system dynamics also yields interesting insights into the current operational condition of the plant. Several references in the literature investigate the use of online identification for fault diagnosis, Rodrigues et al. (2006), Uppal and Patton (2005). A 'fault' in this case could be defined, for instance, by a change in the estimated poles of the system, from one region to another (either pre-defined or unknown). In the case of applications like flexible aircraft or wind turbine blades, the coupling of aerodynamics and structural dynamics implies that there are certain conditions under which the stable poles of the system migrate into the unstable region, Bernhammer et al. (2013). In such a case, online system identification can be used in order to identify the point where the system becomes unstable, and to adjust the control strategy accordingly.

As discussed in the previous chapter, in order to preserve convexity and for the ease of handling multivariable systems, subspace identification may be considered suitable for online, recursive implementation. Further, as an inherently closed-loop strategy, Predictor-Based Subspace Identification (PBSID), Chiuso (2007), is explored for extensions to online recursive identification. A recursive implementation of this method have been considered using projection approximation for subspace tracking, Lovera (2003). In this method, however, it is not guaranteed that the identified state-space parameters converge to a coherent state basis. The authors in Houtzager et al. (2012), on the other hand, use a propagator-based technique for estimating the state sequence in a coherent basis. In all, three Recursive Least Squares (RLS) problems are solved online in order to estimate the state-space realisation of the system in real time. In the case where the true system is LTI and the persistency of excitation condition is satisfied, the state-space matrices estimated by this method ergodically approaches the true underlying system parameters.

One of the issues faced by Houtzager et al. (2012) in the implementation of recursive PBSID is that the variance of the estimates can be held to within small bounds only by employing a large forgetting factor. In a batchwise sense, this high value is equivalent to using a large past window of data to estimate the current dynamics of the system, which causes a severe lag in the responsiveness of the algorithm to changes in system dynamics. It is therefore desirable to extend the algorithm in such a manner that the dominant system dynamics can be rapidly identified from short batches of data, with low estimate variance.

Recent studies in system identification suggest that such an extension can be achieved by exploiting the low-rank structure of the predictor state matrix, Fazel et al. (2001). In this work, the rank penalty is approximated by the nuclear norm, defined as the sum of singular values of the argument. Since the addition of the nuclear norm retains the convexity of the optimisation function, the regularised identification problem yields a unique solution and can be solved efficiently. Verhaegen and Hansson (2016) use the nuclear norm to extend the open-loop batchwise N2SID subspace identification technique, and demonstrate that the nuclear norm enables low-variance system identification from short batches of data.

As such, in the recursive sense, one might expect the addition of the nuclear norm to reduce the estimate variance with a low value of forgetting factor (which is equivalent to a short data batch). Since the optimisation function now consists of a mixture of norms, RLS cannot be used by itself. However, since the cost function is still convex,

a method such as the Alternating Direction Method of Multipliers (ADMM), Annergren et al. (2012), could be used for online minimisation of the cost function.

The ADMM solution demands the thresholding of singular values; a Singular Value Decomposition (SVD) required for this step can be computationally expensive, and may render the algorithm intractable in real time. Different alternatives have been explored in the literature for speeding up the SVD computations in practice. For instance, Cai (2010) avoids the SVD step entirely. Instead, singular value thresholding is achieved via a polar decomposition step followed by norm shrinkage through projection. This method reduces the computation time of the thresholding step by upto 50%. Another alternative to the full SVD is the fast randomised SVD, which randomly samples the columns of the argument matrix and performs a singular value decomposition of the smaller sampled matrix, Oh et al. (2015). Further savings in computational complexity can be achieved using the concept of 'range propagation' for the case where the range of state predictor matrix remains almost the same across time instants, as is the case for time-varying systems where the dynamics changes slowly over time.

The key contribution of this chapter is thus threefold: firstly, a closed-loop subspace identification method will be extended to include the nuclear norm penalty, to facilitate system identification from short batches of data. Unlike the reference Verhaegen and Hansson (2016), which formulates a nuclear norm extension to the (open-loop) subspace identification method N4SID, the current chapter explicitly formulates an extension of the predictor-based method PBSID. In the method developed, the output variables, which change with time, do not form an optimisation variable in the method developed. Instead, the variables to be optimised are the Markov parameters of the system, which are invariant for an LTI system. For a slowly time-varying system, these parameters can also be expected to vary slowly. As such, this method of identification is especially amenable to implementation in a recursive form, and can be formulated using ADMM for the identification of real-time systems with time-varying dynamics. Finally, the developed algorithm will be made numerically more efficient by replacing the SVD step with a fast randomised version.

The first section sets up the notation and formulates the problem statement. Next, the ADMM approach to an online solution is described. The following section describes the replacement of the SVD with a faster alternative. Finally, a case study is described to evaluate the proposed strategy for a simple system with time-varying dynamics, and conclusions are drawn from the results.

3.2. PROBLEM FORMULATION

Conventional predictor-based subspace identification algorithms, Chiuso (2007), typically take a two-step approach in order to arrive at a state-space model estimate on the basis of input-output data. In the first step, an extended Vector Auto-Regressive with eXogenous inputs (VARX) model is constructed from the experimental data, using a least-squares approach with a 2-norm cost function. In the second step, the state sequence is reconstructed and the state-space matrices are estimated based on the data. Since the goal is the rapid identification of the dominant system dynamics, it would be desirable to synthesise a low-order VARX model in the first step. To achieve this goal, the 2-norm cost function in the first step is augmented with a nuclear norm penalty on the state ma-

trix, which exploits the low-rank property of this matrix. With this penalty, the effect of noise on the VARX model and the eventual state-space model is reduced, thereby reducing estimate variance and increasing the responsiveness of the algorithm to changes in the dominant system dynamics.

Let us consider that the underlying system whose parameters are to be identified can be modelled, at a given instant of time, as an LTI system in the predictor form:

$$\begin{aligned} x_{k+1} &= \tilde{A}x_k + Bu_k + Ky_k, \\ y_k &= Cx_k + Fd_k + e_k. \end{aligned} \tag{3.1}$$

Here, $y_k \in \mathbb{R}^{n_y}$ are the outputs of the system, while $u_k \in \mathbb{R}^{n_u}$ are the inputs. The state vector $x_k \in \mathbb{R}^n$ has an unknown length. The signal $d_k \in \mathbb{R}^{n_d}$ is the deterministic part of the disturbance, and is modelled as a combination of basis functions. On the other hand, the signal $e_k \in \mathbb{R}^{n_y}$ is the innovation sequence, considered to be zero-mean white noise. The state-space matrices $\tilde{A}$, B, C, K and F are unknown; they may also vary slowly over time. The objective of recursive identification is to estimate these matrices at every instant of time.

As discussed earlier, for PBSID, the output of the system can be approximated using a truncated VARX representation in the following manner:

$$y_k \approx C\tilde{\mathcal{K}}^{(s)} z_k + Fd_k + e_k, \tag{3.2}$$

where z_k is the input-output data stacked over a past horizon s. Thus,

$$z_k = \begin{bmatrix} u_{k-s} \\ y_{k-s} \\ u_{k-s+1} \\ y_{k-s+1} \\ \vdots \\ u_{k-1} \\ y_{k-1} \end{bmatrix}. \tag{3.3}$$

The term $\tilde{\mathcal{K}}^{(s)}$ is the extended controllability matrix of the system:

$$\tilde{\mathcal{K}}^{(s)} = \begin{bmatrix} \tilde{A}^{s-1}B & \tilde{A}^{s-1}K & \cdots & B & K \end{bmatrix}. \tag{3.4}$$

The block terms in $C\mathcal{K}^{(s)}$ are defined as the Markov parameters $\Xi \in \mathbb{R}^{n_y \times (n_u+n_y)s}$ of the system, and are required to be identified to complete the first step of PBSID. The Markov parameters are partitioned in the following way:

$$\begin{aligned} \Xi &= \begin{bmatrix} \Xi_1 & \Xi_2 & \cdots & \Xi_{(n_u+n_y)(s-1)} & \Xi_{(n_u+n_y)s} \end{bmatrix} \\ &= C\tilde{\mathcal{K}}^{(s)} = \begin{bmatrix} C\tilde{A}^{s-1}B & C\tilde{A}^{s-1}K & \cdots & CB & CK \end{bmatrix}. \end{aligned} \tag{3.5}$$

From Equation (3.2), it can be seen that the Markov parameters can be obtained by directly minimising the quantity $\|y_k - \Xi z_k\|$ over the past window. This approach is followed in conventional PBSID, as well as the recursive version, Houtzager et al. (2012).

However, as noted, this approach is sensitive to noise in the data, and requires large datasets (or high forgetting factors) to give reasonably accurate results in practical situations.

The objective is to estimate Ξ such that the state predictions are adequately low rank, so that this sensitivity to noise is reduced. The state prediction matrix over the past batch of N data points can be synthesised by extending the system description, Equation (3.1) to form the following data equation:

$$Y_{k,s,N} = \tilde{\mathcal{O}}_s X_{k,1,N} + \tilde{T}_{u,s} U_{k,s,N} + \tilde{T}_{y,s} Y_{k,s,N} + \bar{F} D_{k,s,N} + E_{k,s,N}. \tag{3.6}$$

Here, the output data is arranged in Hankel matrices of the form:

$$Y_{k,s,N} = \begin{bmatrix} y_k & y_{k+1} & \cdots & y_{k+N-s-1} \\ y_{k+1} & y_{k+2} & \cdots & y_{k+N-s} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k+s-1} & y_{k+s} & \cdots & y_{k+N-2} \end{bmatrix}, \tag{3.7}$$

with similar expressions for the known quantities u_k and d_k, and the unknown quantities x_k and e_k. The predictor above involves the unknown system Toeplitz matrices that can be defined as:

$$\tilde{T}_{u,s} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ CB & 0 & 0 & \cdots & 0 \\ C\tilde{A}B & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{s-2}B & C\tilde{A}^{s-3}B & C\tilde{A}^{s-4}B & \cdots & 0 \end{bmatrix}, \tag{3.8}$$

with a similar expression for $\tilde{T}_{y,s}$ obtained by replacing B by K. The term $\tilde{\mathcal{O}}_s$ represents the extended observability matrix of the system, given by:

$$\tilde{\mathcal{O}}_s^T = \begin{bmatrix} C^T & (C\tilde{A})^T & \cdots & (C\tilde{A}^{s-1})^T \end{bmatrix}. \tag{3.9}$$

Finally, the term $\bar{F} \in \mathbb{R}^{n_y s \times n_d s}$ is a block diagonal matrix with the matrices F along the diagonal.

In order to estimate the state sequence $X_{k,1,N}$ in terms of the past input-output data, one can refer back to the expression (3.2), whence the following can be stated:

$$\begin{aligned} X_{k,1,N} &= \tilde{\mathcal{K}}^{(s)} \begin{bmatrix} z_k & z_{k+1} & \cdots & z_{k+N-1} \end{bmatrix} \\ X_{k,1,N} &= \tilde{\mathcal{K}}^{(s)} Z_{k,1,N}. \end{aligned} \tag{3.10}$$

Replacing this estimate in the data equation (3.6),

$$Y_{k,s,N} = \tilde{\mathcal{O}}_s \tilde{\mathcal{K}}^{(s)} Z_{k,1,N} + \tilde{T}_{u,s} U_{k,s,N} + \tilde{T}_{y,s} Y_{k,s,N} + \bar{F} D_{k,s,N} + E_{k,s,N}. \tag{3.11}$$

As noted by Verhaegen and Hansson (2016), the term $\tilde{\mathcal{O}}_s \tilde{\mathcal{K}}^{(s)} Z_{k,1,N}$ is low rank, since the window size s is typically larger than the order n of the system. If the Markov parameter identification problem $\|y_k - \Xi z_k\|$ is conditioned on solutions that simultaneously (in a Pareto optimal sense) minimise the rank of the matrix $\tilde{\mathcal{O}}_s \tilde{\mathcal{K}}^{(s)} Z_{k,1,N}$, then the Markov

parameters can be identified in a manner that is less sensitive to measurement noise. As such, the cost function to be minimised can be stated as:

$$\min_{\Xi} \|\tilde{\mathcal{O}}_s \tilde{\mathcal{K}}^{(s)} Z_{k,1,N}\|_* + \frac{\lambda}{N} \sum_{i=1}^{N} \|y_i - \Xi z_i\|_2^2, \tag{3.12}$$

where λ is a user-defined weighting parameter that can be used to tune the trade-off between accuracy and model complexity.

At this point it is to be noted that the argument of the nuclear norm, $\tilde{\mathcal{O}}_s \tilde{\mathcal{K}}^{(s)} Z_{k,1,N}$, is not an independent variable, but can be expressed as a function of the system Markov parameters in the following manner:

$$\begin{aligned}
\tilde{\mathcal{O}}_s \tilde{\mathcal{K}}^{(s)} Z_{k,1,N} &= \begin{bmatrix} C\tilde{A}^{s-1}B & C\tilde{A}^{s-1}K & \cdots & CB & CK \\ 0 & 0 & \cdots & C\tilde{A}B & C\tilde{A}K \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & C\tilde{A}^{s-1}B & C\tilde{A}^{s-1}K \end{bmatrix} Z_{k,1,N} \\
&= \begin{bmatrix} \Xi_1 & \Xi_2 & \cdots & \Xi_{2s-1} & \Xi_{2s} \\ 0 & 0 & \cdots & \Xi_{2s-3} & \Xi_{2s-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \Xi_1 & \Xi_2 \end{bmatrix} Z_{k,1,N} = \mathcal{A}(\Xi).
\end{aligned}$$

Herewith, the cost function can be expressed purely in terms of the unknown Markov parameters Ξ in the following manner:

$$\min_{\Xi} \|\mathcal{A}(\Xi)\|_* + \frac{\lambda}{N} \sum_{i=1}^{N} \|y_i - \Xi z_i\|_2^2. \tag{3.13}$$

The minimisation of this cost function at every time instant would provide the low-rank VARX model that can be further utilised in the next steps of the recursive PBSID algorithm. The next section details a method for minimising this cost function online, in a recursive manner.

3.3. THE ONLINE ADMM SOLUTION

The online, recursive implementation of PBSID has been discussed in Houtzager et al. (2012). In this reference, since the cost function to find the system Markov parameters only contains a 2-norm term, the minimisation can be performed efficiently in a recursive manner using Recursive Least Squares (RLS). However, the cost function augmented with the nuclear norm can no longer be solved using this direct technique, although this function remains convex and admits a unique solution. This cost function, containing a mixture of norms, can be solved online and in a recursive manner using the Alternating Direction Method of Multipliers (ADMM), as described in Annergren et al. (2012).

In order to make the cost function separable in norms, an auxiliary variable $\mathcal{X} \in \mathbb{R}^{n_y s \times N}$ is introduced in Equation (3.13), such that a constrained optimisation problem

is formulated:

$$\min_{\mathcal{X},\Xi} \|\mathcal{X}\|_* + \text{trace}((\Xi - \hat{\Xi})\mathcal{C}(\Xi - \hat{\Xi})^T), \tag{3.14}$$

$$\text{s. t. } \mathcal{X} = \mathcal{A}(\Xi). \tag{3.15}$$

Here, the term $\hat{\Xi}$ is the RLS solution to the 2-norm minimisation problem, $\|y_k - \Xi z_k\|$, which is updated at every instant of time that new data is available. In the batchwise sense, if the forgetting factor in this RLS solution corresponds to a batch of $\bar{N}$ data points, the value of this quantity becomes:

$$\hat{\Xi} = Y_{k,1,\bar{N}} Z^{\dagger}_{k,1,\bar{N}}, \tag{3.16}$$

where the symbol † represents the Moore-Penrose pseudo-inverse of a matrix. Comparing the equations, it can be concluded that the weighting matrix is given by $\mathcal{C} = \lambda Z_{k,1,N} Z^T_{k,1,N}$.

The constrained optimisation problem can be solved in a straightforward manner by considering its Lagrangean and determining the saddle point of the extended cost function:

$$\max_{\mathcal{Z}} \min_{\mathcal{X},\Xi} \|\mathcal{X}\|_* + \text{trace}((\Xi - \hat{\Xi})\mathcal{C}(\Xi - \hat{\Xi})^T) + \mathcal{Z}^T(\mathcal{X} - \mathcal{A}(\Xi)) + \frac{t}{2}\|\mathcal{X} - \mathcal{A}(\Xi)\|_F^2. \tag{3.17}$$

In this expression, the new dual variable $\mathcal{Z}$ is introduced, over which the Lagrangean is to be maximised, while simultaneously minimising the function over the primal variables $\mathcal{X}$ and Ξ. The penalty term t is held constant and represents the relative trade-off between the primal and dual residuals. The objective of this exercise is to specifically find the minimiser Ξ^*, an estimate of the true system Markov parameters, which can then be used in the next steps of the PBSID algorithm to arrive at a system realisation.

The saddle point of the Lagrangean can be found by alternatingly optimising the function over each variable independently, in the following sequential manner:

1. At the initial instant, initialise the values of the primal and dual variables to 0. With the superscripts corresponding to the iteration number,

$$\Xi^{(0)} = 0, \qquad \mathcal{X}^{(0)} = 0, \qquad \mathcal{Z}^{(0)} = 0. \tag{3.18}$$

   Alternatively, since it is assumed that the system dynamics will not change significantly from one instant of time to the next, the variables can be initialised to the converged values from the previous instant of time.

2. At iteration number i, since the term Ξ only appears in quadratic functions in Equation (3.17), the minimiser can be found by directly setting the gradient of the function with respect to Ξ to zero, giving the following result:

$$\Xi^{(i)} = (\mathcal{C} + tM)^{-1}\mathcal{A}_{\text{adj}}(t\mathcal{X}^{(i-1)} - \mathcal{Z}^{(i-1)} + \mathcal{C}\hat{\Xi}^{(i)}). \tag{3.19}$$

   Here, the terms $\mathcal{A}_{\text{adj}}(.)$ refers to the adjoint of the function $\mathcal{A}(.)$ and the matrix M is defined such that $M(\Xi) = \mathcal{A}_{\text{adj}}(\mathcal{A}(\Xi))$.

3. At this iteration, in order to minimise the Lagrangean with respect to $\mathcal{X}$, it is required to perform a singular value thresholding operation, in the following manner:

$$U\text{diag}(\sigma)V^T = \mathcal{A}(\Xi^{(i)}) + \mathcal{Z}^{(i-1)}/t, \tag{3.20}$$

$$\mathcal{X}^{(i)} = U\text{diag}(\max(0, \sigma - 1/t))V^T. \tag{3.21}$$

This step can be defined using the singular value thresholding operator $\mathbb{S}_t$ as:

$$\mathcal{X}^{(i)} = \mathbb{S}_t(\mathcal{A}(\Xi^{(i)}) + \mathcal{Z}^{(i-1)}/t). \tag{3.22}$$

4. The dual variable $\mathcal{Z}$ can be updated at iteration i in a linear manner:

$$\mathcal{Z}^{(i)} = \mathcal{Z}^{(i-1)} + t(\mathcal{A}(\Xi)^{(i)} - \mathcal{X}^{(i)}). \tag{3.23}$$

5. The steps 2. to 4. are then repeated until convergence is reached, as indicated by an acceptably small value of the primal and dual residuals.

The ADMM algorithm, for an LTI system, always reaches convergence, as shown by Nishihara et al. (2015). If the system dynamics are assumed to vary slowly over time, the optimal values of the variables Ξ, $\mathcal{X}$ and $\mathcal{Z}$ can be considered to also vary slowly over time. As such, it is not necessary to proceed to convergence at every instant of time, as approximate convergence can be considered to be reached after an adequately large number of time samples. Hence, the iterative ADMM process for the purpose of finding the optimal Markov parameters Ξ^* can be terminated after a fixed number of iterations at every instant of time, without significantly impairing the accuracy of the solution. In practical terms, this early termination before convergence can ease the computational burden, which is especially important for real-time applications.

It is possible to arrive at analytical expressions for the terms $\mathcal{A}_{\text{adj}}(.)$ and M, for the given formulation of the function $\mathcal{A}(.)$. These quantities can be obtained in the following manner:

$$\mathcal{A}_{\text{adj}}(\mathcal{X}) = \begin{bmatrix} \sum_{j=1}^{s} \bar{x}_j \bar{z}_{k,j}^T \\ \sum_{j=1}^{s-1} \bar{x}_j \bar{z}_{k,j+1}^T \\ \vdots \\ \bar{x}_1 \bar{z}_{k,s}^T \end{bmatrix}, \tag{3.24}$$

$$M = [m_{ij}], \qquad m_{ij} = \begin{cases} \sum_{p=i}^{2s-i+j} \xi_{p(p+i-j)}, & \text{if } i \geq j \\ m_{ji}, & \text{otherwise.} \end{cases} \tag{3.25}$$

The terms $\bar{x}_j \in \mathbb{R}^{n_y \times N}$ and $\bar{z}_{k,j} \in \mathbb{R}^{(n_u+n_y)\times N}$ are obtained by block partitioning the matrices $\mathcal{X}$ and $Z_{k,1,N}$ respectively:

$$\mathcal{X}^T = \begin{bmatrix} \bar{x}_1^T & \cdots & \bar{x}_s^T \end{bmatrix}, \tag{3.26}$$

$$Z_{k,1,N}^T = \begin{bmatrix} \bar{z}_{k,1}^T & \cdots & \bar{z}_{k,s}^T \end{bmatrix}. \tag{3.27}$$

On the other hand, the terms $\xi_{ij} \in \mathbb{R}^{(n_u+n_y)\times(n_u+n_y)}$ are the block elements of the matrix $Z_{k,1,N} Z_{k,1,N}^T$.

Following the above method, it is possible, at each instant of time k, to have available an optimal regularised estimate of the system Markov parameters, Ξ^*. On the basis of this estimate, it is possible to identify the system parameters, following the approach laid out by Houtzager et al. (2012). However, it must be noted that this process involves the computationally expensive SVD step, which can, for many practical applications, render the algorithm intractable in real-time. In order to circumvent this issue, the fast randomised SVD process used by Oh et al. (2015) is described as a means to reduce computational complexity in the next section.

3.4. FAST SINGULAR VALUE THRESHOLDING

3.4.1. RANDOMISED SINGULAR VALUE THRESHOLDING

In Oh et al. (2015), the problem of computational complexity in image processing with nuclear norm regularisation is dealt with by speeding up the SVD step with the randomised approximation introduced by Halko et al. (2011); this method can also be adapted for the ADMM solution for recursive identification. The bottleneck in the ADMM process described in the previous section lies in step 3., which demands the SVD of a large matrix $\mathcal{A}(\Xi) + \mathcal{Z}/t \in \mathbb{R}^{n_y s \times N}$. However, taking into consideration that the rank n of this matrix is typically far smaller than the size N of the collected data points, a considerable speed-up can be achieved. For this, the columns of the matrix $\mathcal{A}(\Xi) + \mathcal{Z}/t$ are sampled randomly by post-multiplication with a small random Gaussian matrix $\Omega \in \mathbb{R}^{N\times\ell}$:

$$\mathcal{Y} = (\mathcal{A}(\Xi) + \mathcal{Z}/t)\Omega, \qquad \mathcal{Y} \in \mathbb{R}^{n_y s \times \ell}. \tag{3.28}$$

As long as the column size ℓ of the random matrix Ω is larger than the rank n of the original matrix $\mathcal{A}(\Xi) + \mathcal{Z}/t$, the new sampled matrix $\mathcal{Y}$ approximately spans the range of the original matrix. Thus, the singular value thresholding (SVT) operator $\mathbb{S}_t$ can be applied to this smaller matrix $\mathcal{Y}$ instead of the original matrix:

$$\mathbb{S}_t(\mathcal{Y}) \approx \mathbb{S}_t(\mathcal{A}(\Xi) + \mathcal{Z}/t). \tag{3.29}$$

With this approximation, the column size of the matrix, the SVD of which is required, reduces drastically. Since computational costs of the SVD step scale linearly with the column size, there is a proportionate decrease in computational complexity by taking the SVD of the smaller matrix. It has been shown by Halko et al. (2011) that the approximation error reduces exponentially with the value of the quantity $\ell - n$, with the error becoming negligible when this value exceeds 5. Also, for the case where the underlying system is LTI, the repeated application of this randomising step at each time instant implies that the approximation error vanishes ergodically.

Thus, with the randomised sampling of the argument of the SVT operator, a significant reduction in computational complexity is achieved, at the cost of an approximation error that becomes negligible over time.

3.4.2. RANGE PROPAGATION IN SVT

It has been shown in Oh et al. (2015), that a further reduction in the computational burden can be achieved through the use of range propagation. In other words, use can be made of the fact that the range of the matrix $\mathcal{Y}$ changes slowly with the changing dynamics of the system, and remains virtually constant across consecutive instants of time. For an LTI system, this assumption is exactly true. The term $\mathcal{Y}$ is hence decomposed as follows:

$$\mathcal{Y} = Q\mathcal{B}, \tag{3.30}$$

where the orthonormal matrix Q spans the range of $\mathcal{Y}$. The matrix $\mathcal{B}$ can be directly synthesised, given $\mathcal{Y}$ and its range as: $\mathcal{B} = Q^{\mathcal{Y}}$. Since the term Q is orthonormal, the SVT step from Equation (3.29) can be updated as follows:

$$\mathbb{S}_t(\mathcal{A}(\Xi) + \mathcal{Z}/t) = \mathbb{S}_t(\mathcal{Y}) = Q\mathbb{S}_t(\mathcal{B}). \tag{3.31}$$

Since the size of $\mathcal{B}$ is smaller than that of $\mathcal{Y}$, the computational complexity is reduced further. Also, the range at the i^{th} iteration, $Q^{(i)}$ can be updated recursively as new data becomes available, based on its value from the previous iteration, $Q^{(i-1)}$. For instance, the new sample matrix Y_q can be given such that:

$$Y_q = (\mathcal{A}(\Xi^{(i)}) + \mathcal{Z}^{(i-1)})\Omega_q, \qquad \Omega_q \in \mathbb{R}^{n_y s \times q}, \tag{3.32}$$

where q is a small number that indicates the extent to which the range is expected to evolve. Based on this new data, the range can be updated by a partial orthogonalisation and renormalisation step as follows:

$$\tilde{Q} = \text{PartialOrthogonalisation}(\begin{bmatrix} Q^{(i-1)} & Y_q \end{bmatrix}) \tag{3.33}$$

$$Q^{(i)} = \text{Renormalisation}(\tilde{Q}). \tag{3.34}$$

The term $\mathcal{B}$ can then be synthesised as before, and the SVT step can be carried out. Thus, the computational complexity reduces from the full SVD of a large matrix to the SVD of a considerably smaller matrix, in the ideal case of size $n \times \ell$, if n is a known quantity. Further, an additional QR-update step is necessary to track the range of the matrix Q. With these modifications, it is expected that the recursive PBSID algorithm regularised by the nuclear norm becomes tractable in real time for a larger class of systems.

At this point, with the use of fast ADMM, it is possible to arrive at an estimate of the Markov parameters that optimise the nuclear norm-augmented cost criterion (3.13). These Markov parameters can be used further to estimate the state-space parameters of the data-generating system, as described in the next section.

3.5. RECURSIVE PBSID WITH THE NUCLEAR NORM: ALGORITHM

The regularised Markov parameters estimated from the previous sections can be used directly in a data-driven strategy like Subspace Predictive Control (SPC), introduced in the previous chapter; or in Subspace Predictive Repetitive Control, a new strategy for the control of periodic disturbances, that will be developed in detail in the next chapter.

However, if the objective is to analyse the properties of the system, for instance, in terms of the poles of the system, then it may become necessary to be able to get an estimate of the state-space matrices of the system. To arrive at these state-space matrices at every instant of time, starting from an estimate of the system Markov parameters, in an online manner, two more RLS problems can be solved. Each of the RLS problems that is required to be solved in the Recursive PBSID algorithm is associated with a forgetting factor $0 < f \leq 1$, which determines the adaptivity of the algorithm to changes in the underlying data-generating system. Also, the fidelity of the estimate obtained at every step in the RLS method can be measured in terms of the covariance matrix P, of appropriate dimensions. The algorithm for recursive PBSID, extended with the nuclear norm, is given in Algorithm 1.

Herewith, it is possible to obtain an estimate of the A, B, C, K and F matrices of the system, starting from input-output data. It can be seen from the algorithm that the term $f^{(1)}$ refers to the forgetting factor associated with the determination of the Markov parameter estimate based on the data. Reducing the value of the estimate can increase the adaptivity of the algorithm, but it can also cause a strong increase in the estimate variance. However, with the addition of the nuclear norm regularisation, it is possible to reduce the sensitivity of the Markov parameter estimation process to external noise, thereby keeping the estimate variance under control. Thus, with the addition of the nuclear norm, it is expected that the forgetting factor can be reduced, thus increasing the responsiveness of the algorithm to changes in system dynamics, without affecting the variance of the estimate.

The potential of the algorithm will be demonstrated using a case study in the next section. The application of the algorithm to experimental data is deferred to the second part of this book.

3.6. CASE STUDY

The behaviour of the nuclear norm-enhanced closed-loop identification algorithm developed in the previous sections is studied. First, the ability of the algorithm to identify system parameters for an LTI system, both stable and unstable, is investigated. Then, the responsiveness of the algorithm to changes in system dynamics is explored and compared with that of the conventional recursive PBSID algorithm from Houtzager et al. (2012).

3.6.1. LINEAR TIME-INVARIANT DYNAMICS

This section describes the results obtained using two different synthetic LTI systems, one system G_s that is open-loop stable and one open-loop unstable G_u. Both systems are single-input single-output (SISO). The transfer function realisations of these systems are given as follows:

$$G_s = \frac{s/\omega + 1}{(s/\omega)^2 + 2\beta_s(s/\omega) + 1}, \qquad G_u = \frac{s/\omega + 1}{(s/\omega)^2 + 2\beta_u(s/\omega) + 1}. \tag{3.35}$$

As can be seen, the LTI systems have one pair of poles each; for the purposes of this

Algorithm 1 Recursive identification with the nuclear norm

input: s, n, $\begin{bmatrix}\tilde{A}_{-1} & B_{-1} & K_{-1}\end{bmatrix}$, $f^{(1),(2),(3)}$, $P_{-1}^{(1),(2),(3)}$
for $k = 0, 1, 2 \ldots$

input: u_k, y_k
Step 1: Update $\Xi_k^{(0)}$

$$P_k^{(1)} = \tfrac{1}{f^{(1)}} P_{k-1}^{(1)} - \tfrac{1}{f^{(1)}} P_{k-1}^{(1)} z_k (f^{(1)} + z_k^T P_{k-1}^{(1)} z_k)^{-1} z_k^T P_{k-1}^{(1)},$$
$$\Xi_k^{(0)} = \Xi_{k-1}^{(0)} + (y_k - \Xi_{k-1}^{(0)} z_k) z_k^T P_k^{(1)}.$$

Step 2: Update Ξ_k
From $\Xi_k^{(0)}$, obtain Ξ_k using ADMM as described in Section 3.3. The ADMM variables Ξ, $\mathcal{X}$ and $\mathcal{Z}$ are initialised from the values at the previous iteration. The SVT step in ADMM is carried out using the randomised variant, with range propagation, as described in Section 3.4.

Step 3: Estimate x_k
From Ξ_k, construct $\tilde{\mathcal{O}}_s \tilde{\mathcal{K}}^{(s)}$ and $\tilde{T}_{y,s}$.
With selection matrix S described in Houtzager et al. (2012) Section IV-B,
$x_k = S(I_{\ell s} - \tilde{T}_{y,s})^{-1} \tilde{\mathcal{O}}_s \tilde{\mathcal{K}}^{(s)} z_k$.

Step 4: Estimate the C matrix at time k,

$$P_k^{(2)} = \tfrac{1}{f^{(2)}} P_{k-1}^{(2)} x_k (f^{(2)} + x_k^T P_{k-1}^{(2)} x_k)^{-1} x_k^T P_{k-1}^{(2)},$$
$$C_k = C_{k-1} + (y_{k-1} - C_{k-1} x_k) x_k^T P_k^{(2)}.$$

Step 5: Estimate e_k
$e_k = y_k - C_k x_k$.

Step 6: Estimate the $\tilde{A}$, B and K matrices at time k,

$$\Phi_k = \begin{bmatrix} x_{k-1} \\ u_k \\ e_k \end{bmatrix}, \Theta_k = \begin{bmatrix}\tilde{A}_k & B_k & K_k\end{bmatrix},$$
$$P_k^{(3)} = \tfrac{1}{f^{(3)}} P_{k-1}^{(3)} \Phi_k (f^{(3)} + \Phi_k^T P_{k-1}^{(3)} \Phi_k)^{-1} \Phi_k^T P_{k-1}^{(3)},$$
$$\Theta_k = \Theta_{k-1} + (x_k - \Theta_{k-1}\Phi_k)\Phi_k^T P_k^{(3)}.$$

end for

case study, the value of the frequency and damping of the poles is taken as:

$$\omega = 2\pi \text{ rad/s}, \qquad \beta_s = 0.05, \qquad \beta_u = -0.05. \tag{3.36}$$

The computational software Matlab^{TM} is used for simulating the system numerically. A sampling time $T_s = 12.5$ Hz is used for the simulations. Since one of the systems is unstable, the systems are both simulated in closed loop with a nominally stabilising controller (designed using classical loop-shaping), described by:

$$K_c = 0.997z^{-1} + 0.00475\frac{T_s z^{-2}}{1 - z^{-1}}. \tag{3.37}$$

In order to ensure that adequate persistency of excitation is available at the system input, a zero-mean white noise signal, with covariance 1, is added to the output of the controller. Further, the output measurements are considered to be perturbed by an unknown external disturbance, simulated by superposing a zero-mean white noise of covariance 0.01 on to the output data collected from the system. The objective of the identification algorithms is to identify the frequency and damping of the pole pair, in a recursive manner. The recursive PBSID algorithm of Houtzager et al. (2012) (RPBSID) is used as a baseline in this case study in order to compare its results to the results obtained by using the nuclear norm-enhanced recursive PBSID developed in this chapter (RPBSID-NN), as given in Algorithm I.

It should be noted that Algorithm I requires the solution of three RLS problems and delineates the direct process of obtaining an RLS solution. However, the recursively updated inverse-QR method, Sayed (2003), is far more numerically stable, and is utilised in this section to obtain solutions to the three least squares problems in the simulation results described below.

As described in Section 3.3, the ADMM method is used to obtain an estimate of the Pareto optimal Markov parameters at each time instant. However, in pursuit of reducing the computational burden, ADMM is not allowed to proceed to convergence; instead, a single step of the ADMM algorithm is executed at every instant of time. The ADMM primal and dual variables are not reinitialised at each time instant, they are assigned the (non-converged) values obtained from the previous instant of time. In a sense, therefore, ADMM is warm-started at all time steps using data from the previous time steps. This approach is considered valid for the case where the underlying system dynamics change slowly over time; this assumption may break down for rapid changes in system dynamics.

The fast randomised SVD described in Section 3.4 is not utilised in the results in this chapter; however it will be employed in Part II, with the experimental results.

The results obtained using the recursive identification algorithms can be seen in Fig. 3.1. Both methods perform well in the identification of the damping, for the stable system as well as the unstable system. It can be seen that the addition of the nuclear norm significantly reduces the variance in the parameter estimate. However, it can also be observed that, after convergence, there remains a small but finite bias between the true value of the parameter and its online, recursive estimate.

The steady-state estimated Markov parameters can be visualised in Fig. 3.2. As per the figure, the addition of the nuclear norm tends to weight the more recent data to a

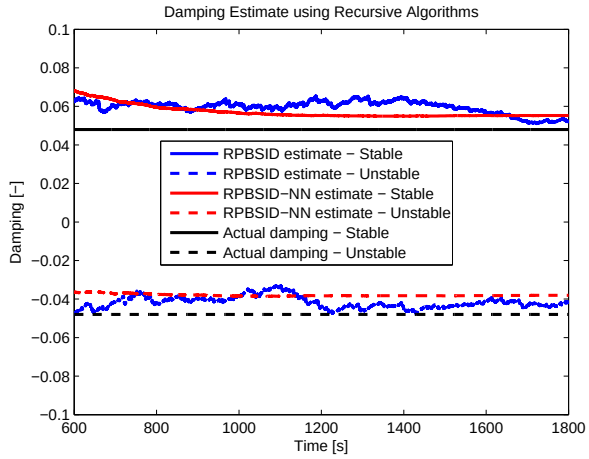


Figure 3.1: Damping estimate using RPBSID and RPBSID-NN

larger extent at the expense of the older data. For a low-order system, the dependence on older data is likely to be corrupted much more by artefacts in the noise, and hence the addition of the nuclear norm reduces the sensitivity of RPBSID to external noise. However, the suppression of the higher-order Markov parameters can be a contributing factor to the bias in the system parameter estimates, which causes, in this case, an overestimation of the damping of the estimated system poles.

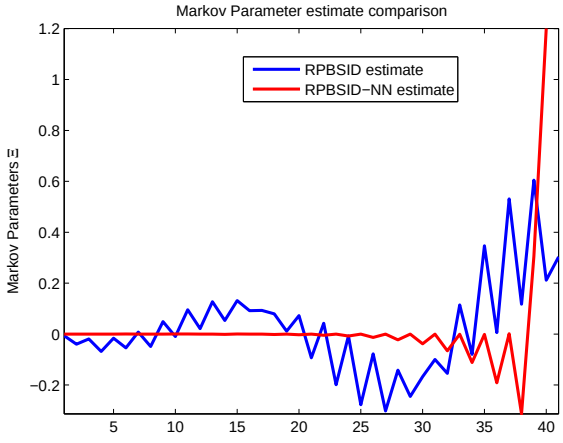


Figure 3.2: Markov parameter estimate using RPBSID and RPBSID-NN

3.6.2. THE BIAS/VARIANCE TRADE-OFF

In the noiseless case, optimal unbiased parameter estimates for the LTI case (with an infinitely large past window) can be obtained by setting the weight on the nuclear norm penalty to 0. In other words, if in Equation (3.13), the weight $\lambda \to \infty$, unbiased system estimates can be obtained. However, from the previous results, a non-zero weight on the nuclear norm is necessary to reduce the variance in the estimates of the system parameters. As such, by tuning the weighting parameter λ, it is possible to achieve the desired bias/variance trade-off. This trade-off is depicted in Fig. 3.3. It can be seen that increasing the weight on the nuclear norm term (reducing λ) causes an increase in estimate bias, while reducing the estimate variance, and vice-versa.

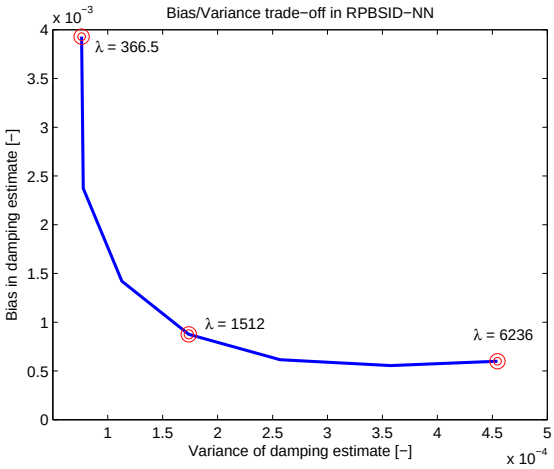


Figure 3.3: Tuning the bias/variance trade-off using λ

3.6.3. TIME-VARYING DYNAMICS

The value of the weight is kept fixed at $\lambda = 160$, and the recursive algorithms are applied to a time-varying system. The transfer function of this system has the same form as G_s and G_u, however the damping of the poles, $\beta(\tau)$, is not constant, but varies with time τ. The damping follows the trajectory:

$$\beta(\tau) = \begin{cases} 0.05, & \text{if } \tau \leq 900 \\ 0.05 - 0.1\frac{\tau-900}{900}, & \text{if } 900 < \tau \leq 1800 \\ -0.05, & \text{if } 1800 < \tau \leq 2700 \\ -0.05 + 0.1\frac{\tau-2700}{900}, & \text{if } 2700 < \tau \leq 3600 \\ 0.05, & \text{if } \tau > 3600. \end{cases} \tag{3.38}$$

Both recursive algorithms are used with the data generated by the system as the pole damping follows the above trajectory; the results can be seen in Fig. 3.4. According to the figure, RPBSID can track the unstable-to-stable transition with low variance and almost

no delay. However, the stable-to-unstable transition causes a sudden large transient in the damping estimate, and a significant delay of nearly 750 s occurs before the instability is detected. Overall, the estimates also show more variance than the estimates with RPBSID-NN. This nuclear norm-enhanced method does not show any unexpected transients, and it is able to smoothly track the changes in the system dynamics. RPBSID-NN is able to track the onset of instability in less than half the time taken by the conventional RPBSID method. Further, the forgetting factor in the RPBSID-NN approach can be reduced from 0.9999 to 0.999, thereby increasing the rate of convergence. This change does not alter the estimate variance significantly. It can be noted that RPBSID-NN shows a bias in the damping estimates, but this bias does not affect the detection of the occurrence of the instability.

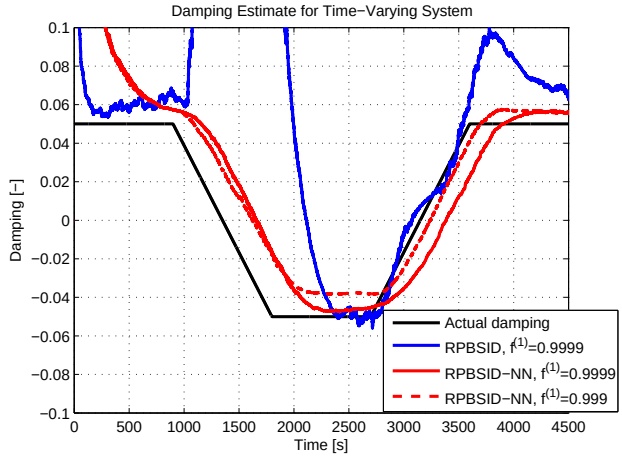


Figure 3.4: Time-varying damping estimate using RPBSID and RPBSID-NN

3.7. CONCLUSIONS

In this chapter, the closed-loop subspace identification algorithm PBSID for slowly time-varying plants was augmented with a nuclear norm rank penalty. The structural knowledge of the low-rank nature of the state predictor matrix was exploited by levying a cost on its nuclear norm. The objective of this extension is to penalise the high-order artefacts produced by noise and therewith reduce the sensitivity of the algorithm to noisy data. The state predictor matrix was reformulated as a function of the system Markov parameters, which then form the only optimisation variable in the identification cost function. Since these parameters vary slowly with time for a slowly time-varying plant, the recursive extension of this method is straightforward. The convex optimisation criterion, a mixture of nuclear and 2-norms, was approximately minimised at every time instant using a few steps of ADMM. While the ADMM process can be sped up by reducing the dimensionality of the SVD step, this step was not implemented in the simulation case study, but will become important when using experimental data.

In the simulation environment, the RPBSID-NN algorithm presented in this chapter was shown able to identify the damping parameters of both stable as well as unstable LTI systems operating in closed-loop. The weighting of the nuclear norm can be tuned to adjust the bias/variance characteristics of the system parameter estimates. Depending on the signal-to-noise ratio (SNR) of the input-output data, increasing the weight on the nuclear norm can increase the bias but leads to a reduction in the estimate variance, and vice-versa. The Pareto weighting parameter can hence be adjusted to achieve the optimal bias/variance balance for the expected SNR.

For plants with time-varying dynamics, the damping can be tracked with more fidelity by RPBSID-NN as compared to conventional RPBSID. Due to its reduced sensitivity to noise, the speed of detecting changes in system dynamics is greater, and can be further improved by reducing the forgetting factor without causing a large increase in estimate variance. In the example, RPBSID-NN is shown to rapidly detect the onset of an instability, in less than half the time taken by the conventional RPBSID algorithm.

Thus, for plants with time-varying dynamics, RPBSID-NN shows potential for tracking changes in pole locations over time. It can also furnish, with low time lag, a reasonable estimate of the system Markov parameters that are relatively uncorrupted by external disturbances. These system parameters can, going forward, be utilised to synthesise a time-varying data-driven control law that is able to adapt to changes in the dynamic behaviour of the underlying data-generating system.

4

SUBSPACE PREDICTIVE REPETITIVE CONTROL

To the best of my knowledge, he has never encountered a charging rhinoceros, but should this contingency occur, I have no doubt that the animal, meeting his eye, would check itself in mid-stride, roll over and lie purring with its legs in the air.

Bertram Wooster in 'Right Ho Jeeves', authored by P. G. Wodehoouse, the UK (1934), describing the extraordinary adaptivity of his controller, Jeeves.

Several industrial systems require the rejection of periodic disturbances, or periodic reference trajectory following. This chapter describes the data-driven implementation of repetitive control, termed as Subspace Predictive Repetitive Control (SPRC). In this method, first, the system Markov parameters are determined recursively from the input-output data: both in the time domain, as described in the previous chapter, as well as in the iteration domain, where an iteration is taken to be a multiple of the period of the periodic disturbance. The dimensionality of the identification in the iteration domain can be reduced by projecting the data into a smaller subspace defined by basis vectors. Next, an output predictor is synthesised, and an LQ-like cost criterion is minimised over a finite and receding horizon. This control law can also be synthesised in a basis function space. The chapter then describes the stability characteristics of the two-step data-driven control approach developed. Finally, with the help of a case study, the implementation of the new control methodology is investigated, specifically for a case with constant and time-varying system dynamics. The case study also explores the benefits and drawbacks of using basis functions in the identification and/or control steps of the data-driven SPRC controller.

Parts of this chapter have been published in Mechatronics **24**, Navalkar et al. (2014), in the Proceedings of the American Control Conference, Portland, USA, Navalkar et al. (2014c), in the Proceedings of the IFAC World Congress, Cape Town, South Africa, Navalkar et al. (2014a) and the IEEE Transactions on Control Systems Technology **23**, Navalkar et al. (2015).

4.1. INTRODUCTION

Data-driven control based on recursively identified system dynamics shows signficant potential for the control of systems with time-varying dynamics, as discussed in the previous chapters. Typically, the system parameters identified online are used to synthesise an MPC law, which is then implemented in a receding horizon sense. For the special case where the objective of the controller is to reduce periodic disturbances, or where the system is required to follow a periodic trajectory, the periodicity can be exploited to enhance the controller performance even further. It should be noted that, from the point of view of control design, the rejection of periodic disturbances and periodic reference trajectory following are exactly identical.

High-performance control design methods for addressing the periodic disturbance rejection issue have been explored in the literature: “learning control", as introduced in Arimoto et al. (1984), is one such methodology. In this approach, a feedforward sequence is successively optimised over a number of iterations to minimise the effect of the periodic disturbance. A distinction is made between “Iterative Learning Control" (ILC), which is applicable to systems that undergo a reset in initial conditions at the beginning of each periodic iteration, and “Repetitive Control" (RC), where the initial conditions are not reset. The current chapter focusses on RC, as described by Longman (2000), and is aimed towards systems that evolve continuously over time and do not undergo jumps in the state sequence. However, it should be noted that the treatment of ILC and RC is very close in practice. The developments in ILC have been reviewed in Bristow et al. (2006), specifically dealing with issues of stability, performance and robustness. Tousain and Van Casteren (2007) show that ILC outperforms the traditional combined feedforward-feedback control design approach in terms of periodic reference trajectory tracking. Since the formulation of ILC, as used by Van de Wijdeven and Bosgra (2007) for the control of flexible structures, is analogous to that of an MPC controller, it can be considered amenable to a data-driven extension. Further, as with Wang et al. (2010), MPC can be combined readily with RC, with a view towards synthesising a control law with constraint handling capabilities.

The objective of RC is to arrive at an optimal periodic “feedforward" control input sequence in order to attenuate the periodic disturbance. However, if the period is large as compared to the sampling time, as is often the case for many practical applications, the optimisation problem has an argument of a considerably large size. Such an ill-conditioned optimisation problem may cause numerical issues or slow the rate of convergence of RC to the ideal feedforward sequence. This problem can be overcome by defining the control input as a linear combination of basis functions, as is done by Van de Wijdeven and Bosgra (2010) and Bolder et al. (2013) in ILC and Shi et al. (2014) for RC. This approach has a few advantages and disadvantages:

- The computational burden is reduced since the length of the optimisation variable reduces to the number of basis functions chosen.

- The control input is constrained to remain within the basis function subspace, thereby allowing precise control over the frequency content or the smoothness of the control input signal.

- If identification is to be done in the lower-dimensional basis function subspace, then persistency of excitation is required only in this subspace.

- For a linear system, the disturbances that lie in the null space of the basis functions cannot be attenuated. On the other hand, these disturbances cannot be magnified either.

Such an implementation has been demonstrated experimentally by Van Wingerden et al. (2010b) on a wind turbine operating inside a wind tunnel, for the purpose of rejecting blade load disturbances that are sinusoidal in nature. The exogenous forcing frequencies 1P (rotor speed) and 2P (2x rotor speed) were significantly attentuated with an RC method designed offline, with control signals synthesised as (optimally) weighted combinations of sinusoidal basis functions of frequency 1P and 2P.

It should be noted that, in all the above references, the starting point is an approximate (LTI or linear periodic, Dijkstra and Bosgra (2002)) model of the system to be controlled, which may not be readily available in all cases and all operating conditions; as such, a data-driven extension may become necessary. Some data-driven extensions to ILC exist: for instance, in Frueh and Phan (2000), the basis functions are optimally synthesised based on input-output data. Offline identification can also be done as a precursor to synthesising an RC control law, as in Ye and Wang (2005). However, an online combination of system identification and repetitive control would be able to address the problem of slowly time-varying dynamics or periodic disturbances; such a combination is not available in the literature.

As discussed in the Chapters 2 and 3, subspace identification can be performed recursively by solving a convex cost function to obtain the system Markov parameters for systems with slowly time-varying dynamics. These parameters can be used, in an RC control design approach, to iteratively synthesise the ideal feedforward sequence that would minimise external periodic disturbances.

The key contribution of this chapter is thus the full integration of an online subspace identification strategy with a repetitive control law formulation for periodic disturbance rejection. Further, both the identification and control steps are cast into a lower-dimensional basis functions subspace, to minimise computational complexity and to ensure control over the smoothness of the control input. Since identification is done by lifting the system over the period of the disturbance, and then casting the data into a basis-function subspace, the method is applicable to LTI systems as well as to systems that admit an (instantaneously) linear periodic state-space realisation. Finally, as the developed method is analogous to MPC, constraint handling is possible. Both equality and inequality constraints are then required to be cast into the basis function subspace in which the algorithm is implemented.

With this data-driven extension of RC, irrespective of changes in dynamics and disturbance properties, the control law is formulated to adapt itself in such a manner that maximal periodic disturbance rejection is still achieved.

In this chapter, first, the problem is introduced mathematically and the input-output data equation is set up. Then, the identification step is explained. The next step involves the synthesis of a repetitive control law based on the parameter estimates obtained from the identification step. A stability metric is set up that can be tracked to ensure that the

uncertainty in the estimates does not affect the closed-loop stability of the data-driven controller. Finally, a numerical case study demonstrates the potential of this method. The experimental application of this method is deferred to the second part of the book.

4.2. SPRC STEP 1: ONLINE SYSTEM IDENTIFICATION

In common with all two-step data-driven approaches to control, the first step is to devise a system description that can be used to identify key system dynamics based on input-output data. As in the previous chapter, the system dynamics can be estimated in terms of the (regularised) Markov parameters identified from input-output data. This section formulates a predictor in terms of such Markov parameters, and defines the system matrices, in terms of the identified parameters, that can be used for synthesising a control law. The identification can be carried out either in the time domain, or in the iteration domain, where an iteration is taken to be an integral multiple of the period of the periodic disturbance to be rejected.

4.2.1. TIME-DOMAIN IDENTIFICATION

The time-domain identification laid out in this section is similar to the first step of the RPBSID method discussed in the previous chapter: the objective is to estimate the system Markov parameters from the input-output data obtained in the time domain. Hence, as in the previous chapter, let us consider that the underlying system to be controlled, at a given instant of time, as an LTI system in the innovation form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ed_k + Ke_k \\ y_k &= Cx_k + Du_k + Fd_k + e_k. \end{aligned} \tag{4.1}$$

Here, as previously defined, the state $x_k \in \mathbb{R}^n$ is the state vector of unknown length n, while the inputs are denoted by $u_k \in \mathbb{R}^{n_u}$ and the outputs by $y_k \in \mathbb{R}^{n_y}$. The periodic disturbances are $d_k \in \mathbb{R}^{n_d}$, while $e_k \in \mathbb{R}^{n_y}$ is the zero-mean white innovation sequence. The system parameters A, B, C, D, E, F and K have the appropriate dimensions and are unknown. These parameters are considered to either remain constant or evolve slowly over time. For brevity, they have not been subscripted with k. However, especially for a linear periodic system which has the same period N as that of the disturbance, the extension is straightforward.

To be able to formulate a state predictor, the system is recast into its predictor form:

$$\begin{aligned} x_{k+1} &= \tilde{A}x_k + \tilde{B}u_k + \tilde{E}d_k + Ky_k, \\ y_k &= Cx_k + Du_k + Fd_k + e_k, \end{aligned} \tag{4.2}$$

where,

$$\tilde{A} = A - KC, \qquad \tilde{B} = B - KD, \qquad \tilde{E} = E - KF. \tag{4.3}$$

To simplify the identification process, the effect of the periodic disturbance on the state and the output is eliminated by using the periodic difference operator δ defined as:

$$\delta y_k = y_k - y_{k-N}, \qquad \delta u_k = u_k - u_{k-N}, \qquad \delta d_k = d_k - d_{k-N} = 0. \tag{4.4}$$

The term N is in this case the period of the periodic disturbance; it is in this case taken to be equal to the size of the data batch that will be used for identification. This is not a strict requirement: the size of the data batch can also be an integral multiple of the period of the periodic disturbance. Applying this operator to the system description (4.2), we get

$$\begin{aligned} \delta x_{k+1} &= \tilde{A}\delta x_k + \tilde{B}\delta u_k + K\delta y_k \\ \delta y_k &= C\delta x_k + D\delta u_k + \delta e_k. \end{aligned} \tag{4.5}$$

The key step in predictive control is the prediction of state evolution as a function of the control input sequence. From the equations above, in a manner analogous to the previous chapter, the differenced state can be predicted in terms of the past input-output data as:

$$\delta x_k = \tilde{A}^s \delta x_{k-s} + \tilde{\mathcal{K}}^{(s)} \delta z_k, \tag{4.6}$$

with the differenced, stacked input-output data $\delta z_k \in \mathbb{R}^{(n_u+n_y)s}$ defined as:

$$\delta z_k = \begin{bmatrix} u_{k-s+1} - u_{k-s-N+1} \\ y_{k-s+1} - y_{k-s-N+1} \\ \vdots \\ u_k - u_{k-N} \\ y_k - y_{k-N} \end{bmatrix}. \tag{4.7}$$

The term $\tilde{\mathcal{K}}^{(s)}$ refers once more to the extended controllability matrix,

$$\tilde{\mathcal{K}}^{(s)} = \begin{bmatrix} \tilde{A}^{s-1}\tilde{B} & \tilde{A}^{s-1}K & \cdots & \tilde{A}\tilde{B} & \tilde{A}K & \tilde{B} & K \end{bmatrix}. \tag{4.8}$$

As before, the approximation is made that $\tilde{A}^j \approx 0$ for $j \geq s$, which is true, for an arbitrarily large value of s for a stable system or a system where K stabilises $A - KC$. With this, it is possible to describe the state purely in terms of the input-output data as:

$$\delta x_k \approx \tilde{\mathcal{K}}^{(s)} \delta z_k. \tag{4.9}$$

Usually, in two-step data-driven control, two data windows are taken into consideration: the past window, from which the system parameters can be estimated, and the future window, over which the output is to be optimised by manipulating the control sequence. The above equation describes the state in terms of the past window of data, of block size s. For control, the output can be predicted over the window size N in the following manner:

$$\delta Y_{k,N,1} = \tilde{\mathcal{O}}_N \delta x_k + \begin{bmatrix} \tilde{T}_{u,N} & \tilde{T}_{y,N} \end{bmatrix} \begin{bmatrix} \delta U_{k,N,1} \\ \delta Y_{k,N,1} \end{bmatrix}. \tag{4.10}$$

Since the innovation sequence is zero-mean, and its actual realisation over a future window is unknown, it is omitted in the equation above. The state is replaced by its estimate from the past input-output data per (4.9),

$$\delta Y_{k,N,1} = \tilde{\mathcal{O}}_N \tilde{\mathcal{K}}^{(s)} \delta z_k + \begin{bmatrix} \tilde{T}_{u,N} & \tilde{T}_{y,N} \end{bmatrix} \begin{bmatrix} \delta U_{k,N,1} \\ \delta Y_{k,N,1} \end{bmatrix}. \tag{4.11}$$

The system matrices in this equation can be described in terms of the system Markov parameters. The Toeplitz matrix from the previous chapter is extended to $\tilde{T}_{u,N} \in \mathbb{R}^{n_y N \times n_u N}$ such that:

$$\tilde{T}_{u,N} = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ C\tilde{B} & D & 0 & \cdots & 0 \\ C\tilde{A}\tilde{B} & C\tilde{B} & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{s-1}\tilde{B} & C\tilde{A}^{s-2}\tilde{B} & C\tilde{A}^{s-3}\tilde{B} & \cdots & 0 \\ 0 & C\tilde{A}^{s-1}\tilde{B} & C\tilde{A}^{s-2}\tilde{B} & \cdots & 0 \\ 0 & 0 & C\tilde{A}^{s-1}\tilde{B} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & D \end{bmatrix}. \tag{4.12}$$

In a similar manner, $\tilde{T}_{y,N} \in \mathbb{R}^{n_y N \times n_y N}$ can be defined, replacing $\tilde{B}$ by K and D by 0. Equivalent open-loop Toeplitz matrices $T_{u,N}$ and $T_{y,N}$ can also be defined in terms of the "open-loop system matrices", replacing $\tilde{A}$ by A and $\tilde{B}$ by B. These matrices will be required in the predictor formulation in the next section. Finally, the extended controllability matrix $\tilde{\mathcal{O}}_N \in \mathbb{R}^{n_y N \times n}$ is defined in the same manner as in the last chapter:

$$\tilde{\mathcal{O}}_N^T = \begin{bmatrix} C^T & \cdots & (C\tilde{A}^{s-2})^T & (C\tilde{A}^{s-1})^T & 0 & \cdots & 0 \end{bmatrix}. \tag{4.13}$$

As for the Toeplitz matrices, an "open-loop" version of the extended controllability matrix, $\mathcal{O}_N$ can also be defined by replacing $\tilde{A}$ in its definition by A. It can be directly observed that the block elements of the extended observability times controllability matrix, $\tilde{\mathcal{O}}_N\tilde{\mathcal{K}}^{(s)}$ are the system Markov parameters that can readily identified from the input-output data:

$$\tilde{\mathcal{O}}_N\tilde{\mathcal{K}}^{(s)} = \begin{bmatrix} C\tilde{A}^{s-1}\tilde{B} & C\tilde{A}^{s-1}K & C\tilde{A}^{s-2}\tilde{B} & C\tilde{A}^{s-2}K & \cdots & C\tilde{B} & CK \\ 0 & 0 & C\tilde{A}^{s-1}\tilde{B} & C\tilde{A}^{s-1}K & \cdots & C\tilde{A}\tilde{B} & C\tilde{A}K \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & C\tilde{A}^{s-1}\tilde{B} & C\tilde{A}^{s-1}K \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{4.14}$$

Thus, were the Markov parameters of the system known, it would be possible to predict system output over the period N using Equation (4.11). As done in the previous chapter, the Markov parameters of the system can be identified by considering the following regression problem obtained from the first block row of (4.11):

$$\delta y_k = C\tilde{\mathcal{K}}^{(s)}\delta z_k + D\delta u_k. \tag{4.15}$$

As defined in the previous chapter, $C\tilde{\mathcal{K}}^{(s)} = \Xi$, the system Markov parameters. Once identified, the Markov parameters can be used to reconstruct $\tilde{\mathcal{O}}_N\tilde{\mathcal{K}}^{(s)}$, as well as the Toeplitz matrices defined above. On the basis of these matrices, the system output can

be predicted as a function of the future control sequence and a predictive control law can be set up. Hence, the objective of the first step of the two-step data-driven control design approach is the identification of the system Markov parameters from the regression problem set up in (4.15). Taking into account the fact that the actual measurements are perturbed by the innovation sequence, we have that:

$$\delta y_k = \Xi \delta z_k + D\delta u_k + \delta e_k. \tag{4.16}$$

Given measurements of input-output data, the objective of the first step is to identify Ξ and D such that they minimise the norm:

$$\min_{\Xi,D} \sum_{i=1}^{\infty} \|\delta y_i - \Xi \delta z_i - D\delta u_i\|_2^2. \tag{4.17}$$

For low estimate variance and rapid convergence, this minimisation function can be regularised with the nuclear norm, as described in the previous chapter. The method devised in that chapter needs to be modified minimally to include differenced input-output signals and a direct feed-through term, but it is otherwise directly applicable to obtain optimal estimates of the system Markov parameters Ξ. As discussed, the cost function above (extended with a nuclear norm if necessary) can be solved recursively and online to yield a unique minimising solution $\hat{\Xi}_k$ at every instant of time, as long as the persistency of excitation condition holds.

Once the Markov parameters estimate $\hat{\Xi}_k$ is available, it can be partitioned and rearranged to obtain the matrix $\tilde{\mathcal{O}}_N \tilde{\mathcal{K}}^{(s)}$ and the Toeplitz matrices $\tilde{T}_{u,N}$ and $\tilde{T}_{y,N}$. From these matrices, it is possible to formulate a predictive control law.

One of the drawbacks of time-domain identification is that persistency of excitation in all possible directions is required for uniqueness of the estimate. This condition could interfere with the normal operation of the plant, especially since identification and control is expected to occur simultaneously in this data-driven approach. Alternatively, it is possible, using iteration-domain identification, to excite the system only along specific directions important for control, and perform identification only in this limited subspace. This approach is described in the next section.

4.2.2. ITERATION-DOMAIN IDENTIFICATION

In this section, identification will be performed at the level of the iteration, defined as N, an integral multiple of the period of the system. As opposed to the previous section, the system description will not be differenced, instead it is directly lifted over the period N as in Bamieh et al. (1991):

$$\begin{aligned} x_{k+N} &= \tilde{A}^N x_k + \tilde{\mathcal{K}}^{(N)} Z_k^{(N)} + \tilde{\mathcal{K}}_d^{(N)} \bar{D} \\ Y_{k,N,1} &= \tilde{\mathcal{O}}_N x_k + \tilde{T}_{u,N} U_{k,N,1} + \tilde{T}_{d,N} \bar{D} + E_{k,N,1}. \end{aligned} \tag{4.18}$$

It should be noted that $\tilde{\mathcal{K}}^{(N)}$ and $Z_k^{(N)}$ are defined in the same way as $\tilde{\mathcal{K}}^{(s)}$ and z_k, replacing the window size s by N. The term $\bar{D}$ denotes the stacked disturbance, since it is periodic with period N, $\bar{D}$ is constant across iterations. The term $\tilde{\mathcal{K}}_d^{(N)}$ is defined as:

$$\tilde{\mathcal{K}}_d^{(N)} = \begin{bmatrix} 0 & 0 & \cdots & \tilde{A}^{s-1}\tilde{E} & \tilde{A}^{s-2}\tilde{E} & \cdots & \tilde{E} \end{bmatrix}. \tag{4.19}$$

Further, the Toeplitz matrix $\tilde{T}_{d,N}$ is defined in the same manner as $\tilde{T}_{u,N}$, replacing $\tilde{B}$ by $\tilde{E}$ and D by F.

Since identification will be done in the iteration domain, the time index $(k, k+N, ...)$ is replaced by the iteration index $(\bar{k}, \bar{k}+1, ...)$. In the previous section, the assumption has been made that $\tilde{A}^j \approx 0$ for $j \geq s$. Typically, the length of the iteration is much larger than the window size, $N >> s$, and as such, it can be assumed that $\tilde{A}^N \approx 0$. The system description (4.18) reads:

$$\begin{aligned} x_{\bar{k}+1} &= \tilde{A}^N x_{\bar{k}} + \tilde{\mathcal{K}}^{(N)} Z_{\bar{k}}^{(N)} + \tilde{\mathcal{K}}_d^{(N)} \bar{D} \\ Y_{\bar{k},N,1} &= \tilde{\mathcal{O}}_N x_{\bar{k}} + \tilde{T}_{u,N} U_{\bar{k},N,1} + \tilde{T}_{d,N} \bar{D} + E_{\bar{k},N,1}. \end{aligned} \tag{4.20}$$

The state and output equations in the iteration domain become:

$$\begin{aligned} x_{\bar{k}+1} &= \begin{bmatrix} \tilde{\mathcal{K}}^{(N)} & \tilde{\mathcal{K}}_d^{(N)} \bar{D} \end{bmatrix} \begin{bmatrix} Z_{\bar{k}}^{(N)} \\ 1 \end{bmatrix} \\ Y_{\bar{k},N,1} &= \begin{bmatrix} \tilde{\mathcal{O}}_N \tilde{\mathcal{K}}^{(N)} & \tilde{T}_{u,N} & (\tilde{\mathcal{O}}_N \tilde{\mathcal{K}}_d^{(N)} + \tilde{T}_{d,N}) \bar{D} \end{bmatrix} \begin{bmatrix} Z_{\bar{k}-1}^{(N)} \\ U_{\bar{k},N,1} \\ 1 \end{bmatrix} + E_{\bar{k},N,1}. \end{aligned} \tag{4.21}$$

In the above data equation, it should be noted that the innovation sequence $E_{\bar{k},N,1}$ is uncorrelated with the input-output data of the previous iteration, $Z_{\bar{k}-1}^{(N)}$. Also, if an RC law is the only form of control used, it is updated at the end of every iteration. So, while $U_{\bar{k},N,1}$ is correlated with $E_{\bar{k}-1,N,1}$, it is not correlated with $E_{\bar{k},N,1}$. As such, $E_{\bar{k}-1,N,1}$ in Equation (4.21) is an uncorrelated white noise and the equation can be used to estimate the values of the coefficients $\begin{bmatrix} \tilde{\mathcal{O}}_N \tilde{\mathcal{K}}^{(N)} & \tilde{T}_{u,N} & (\tilde{\mathcal{O}}_N \tilde{\mathcal{K}}_d^{(N)} + \tilde{T}_{d,N}) \bar{D} \end{bmatrix}$ in a least-squares sense. This regression can be performed online, recursively, as and when data becomes available.

However, at this point, using Equation (4.21) instead of Equation (4.15) is not attractive, since the dimensionality of the optimisation variables in (4.21) is significantly large as the input-output data has been stacked over a length of N. If the input-output data can be projected into a subspace of a lower dimensionality, the regression (4.21) could become more viable.

BASIS FUNCTION PROJECTION

In order to reduce the dimensionality of the regression problem (4.21), the stacked input-output data can be projected into a lower-dimensional subspace using basis functions. The limitation of this approach is that the identified parameters are then able to describe the behaviour of the system only in this limited-dimensional space. This aspect is not necessarily a drawback for SPRC, if the target of the controller is to achieve performance in this specific subspace. The input basis functions ϕ_u are used to project the stacked input data into a lower-dimensional space. This projection is also used when synthesising a control law, to ensure that the shape and smoothness of the commanded input signals can be exactly controlled. Output basis functions ϕ_y are used to shape the performance requirements and/or to signify that a restricted subspace of the stacked output space is controllable by the restricted control inputs.

Accordingly, the stacked input projected into the input basis function subspace is denoted by $\bar{U}_{\bar{k}} = \phi_u U_{\bar{k},N,1}$ and similarly for the stacked output and stacked input-output data. The projection matrices are spanned by the basis vectors as:

$$\phi_u = \begin{bmatrix} \phi_{u,1}^T & \phi_{u,2}^T & \cdots & \phi_{u,b_u}^T \end{bmatrix}^T, \tag{4.22}$$

where the basis vectors $\phi_{u,i} \in \mathbb{R}^{n_u N}, i = 1,2,...,b_u$. In practice, a limited number of basis vectors is chosen, such that $n_u N \gg b_u$. In this case, the basis function projection strongly reduces the dimensionality of the regression problem and therewith the computational complexity. If the basis function matrix ϕ_u is chosen to be the identity matrix, then the full input space is recovered. As such, the original regression problem can be considered to be a special case of the problem projected into the basis function subspace, where the projection matrix is identity. A similar treatment can be done for the output and the input-output projection matrices.

Since the control input is required in the full-dimensional subspace, it can be reconstructed using the Moore-Penrose pseudo-inverse of the projection matrix, denoted by the symbol †. Thus, the original control signal is:

$$U_{\bar{k},N,1} = \phi_u^\dagger \bar{U}_{\bar{k}}. \tag{4.23}$$

By construction, $U_{\bar{k},N,1}$ will contain energy only along the basis function directions. On the other hand, the output, especially when perturbed by an exogenous source, will contain energy both along the basis vector directions as well as orthogonal to it:

$$Y_{\bar{k},N,1} = \phi_y^\dagger \bar{Y}_{\bar{k}} + \phi_y^\perp Y_{\bar{k}}^\perp. \tag{4.24}$$

The residual signal $Y_{\bar{k}}^\perp$ lies outside the output basis function subspace, mapped by $\phi_y^\perp$. Since $\phi_y \phi_y^\perp$ is identically zero, the effect of the residual ouptut is projected away by using the output basis functions. Since, by design, the residual output can either not be reached by the input, or this part of the output is not relevant for the controller performance. In both cases, it is then unnecessary to identify the transfer between $\bar{U}_{\bar{k}}$ and $Y_{\bar{k}}^\perp$. A similar consideration can be done for the noise $E_{\bar{k},N,1}$ and the effect of the periodic disturbance $(\tilde{\mathcal{O}}_N \tilde{\mathcal{K}}_d^{(N)} + \tilde{T}_{d,N})\bar{D}$. Thus, projecting (4.21) into the basis function subspace,

$$\bar{Y}_{\bar{k}} = \begin{bmatrix} \phi_y \tilde{\mathcal{O}}_N \tilde{\mathcal{K}}^{(N)} \phi_z^\dagger & \phi_y \tilde{T}_{u,N} \phi_u^\dagger & \phi_y (\tilde{\mathcal{O}}_N \tilde{\mathcal{K}}_d^{(N)} + \tilde{T}_{d,N}) \bar{D} \end{bmatrix} \begin{bmatrix} \bar{Z}_{\bar{k}-1}^{(N)} \\ \bar{U}_{\bar{k}} \\ 1 \end{bmatrix} + \phi_y E_{\bar{k},N,1}. \tag{4.25}$$

The objective of the iteration domain identification step is then to identify the dependence of the past projected input-output data on the future projected output, so that this dependence can be used to synthesise a predictive control law. The above equation can be used to formulate a regression problem by defining the projected system Markov parameters Ξ_ϕ as:

$$\Xi_\phi = \begin{bmatrix} \phi_y \tilde{\mathcal{O}}_N \tilde{\mathcal{K}}^{(N)} \phi_z^\dagger & \phi_y \tilde{T}_{u,N} \phi_u^\dagger & \phi_y (\tilde{\mathcal{O}}_N \tilde{\mathcal{K}}_d^{(N)} + \tilde{T}_{d,N}) \bar{D} \end{bmatrix}. \tag{4.26}$$

Given input-output data in iteration $\bar{k}-1$, it is possible to generate an estimate of $\hat{\Xi}_{\phi,\bar{k}}$ at iteration $\bar{k}$, in a recursive and online manner in closed loop RC control operation. Thus, from the relation:

$$\bar{Y}_{\bar{k}} = \Xi_{\phi} \begin{bmatrix} \bar{Z}^{(N)}_{\bar{k}-1} \\ \bar{U}_{\bar{k}} \\ 1 \end{bmatrix} + \phi_y E_{\bar{k},N,1}, \tag{4.27}$$

an RLS approach can be taken to obtain the estimate $\hat{\Xi}_{\phi,\bar{k}}$ that solves the minimisation problem:

$$\hat{\Xi}_{\phi,\bar{k}} = \arg\min_{\Xi_{\phi}} \sum_{i=1}^{\infty} \left\| \bar{Y}_{\bar{k}} - \Xi_{\phi} \begin{bmatrix} \bar{Z}^{(N)}_{\bar{k}-1} \\ \bar{U}_{\bar{k}} \\ 1 \end{bmatrix} \right\|_2^2. \tag{4.28}$$

4

It should be noted that, for the case of an RC control law, $E_{\bar{k},N,1}$ is uncorrelated with the input-output data in the iteration $\bar{k}$. Further, the product of $E_{\bar{k},N,1}$ with the non-zero constant projection matrix ϕ_y will remain uncorrelated with the projected input-output data. As such, RLS will yield unbiased unique estimates of Ξ_{ϕ} as long as persistency of excitation holds in the projected space. This condition is less strict than the requirement of persistency of excitation in the time domain, required by time-domain identification. Also, since the basis function projections can take the controller bandwidth into account, excitation energy is also typically required only in the low-frequency region, thereby reducing the actuator duty that would otherwise be required for full-dimensional persistency of excitation.

Herewith, the iteration-domain identification yields, at every time step, an estimate of the projected Markov parameters $\hat{\Xi}_{\phi,\bar{k}}$ which can then be partitioned and manipulated to obtain a prediction of the future behaviour of the plant, and hence formulate an RC control law.

4.3. STEP 2: INFINITE HORIZON REPETITIVE CONTROL

From the previous section, it is possible to use input-output data to obtain the system Markov parameters, either in the full-dimensional space, $\hat{\Xi}$ or in the reduced dimension basis function space, $\hat{\Xi}_{\phi}$. If the commanded control input is to be constrained to remain in the space defined by the basis functions, then the time-domain full-dimensional Markov parameters from Section 4.2.1 can be directly projected into the basis function space to obtain $\phi_y \hat{\Xi}_{\bar{k}} \phi_z^{\dagger}$. On the other hand, if control is expected to be performed in the full-dimensional input-output space, then the input and output basis function projection matrices can be chosen to be identity (of the appropriate dimensions).

The RC law can be formulated based on the identified Markov parameters to obtain a law of the generic form from Van de Wijdeven and Bosgra (2010):

$$\bar{U}_{\bar{k}+1} = \alpha \bar{U}_{\bar{k}} + \beta \begin{bmatrix} x_{\bar{k}} \\ \epsilon_{\bar{k}-1} \end{bmatrix}. \tag{4.29}$$

Thus, the control input sequence over the next iteration is obtained by updating the control input from the last iteration with a weighted combination of the new initial state and the error from the previous iteration, $\epsilon_{\bar{k}-1}$. The term $\beta \in \mathbb{R}^{b_u \times (n+b_y)}$ is defined as

the learning gain matrix, which will be synthesised in this section. The term α is used to denote the "Q-filter" that is typically used in ILC/RC literature to alleviate robustness concerns.

The objective of the controller developed in this section is to minimise the output over an infinite horizon, assuming that the output channel coincides with the performance channel. For the case where the performance is distinct from the output, the extension is straightforward. Further, in this section, a "Q-filter" will not be used, specifically to avoid compromising on performance. Instead, an infinite-horizon cost function is minimised, which yields, for the case of exact parameter identification, a stable closed-loop system. Now, reformulating the data equation (4.11) as a predictor equation in terms of the estimated system properties,

$$\delta Y_{\bar{k},N,1} = \widehat{(\tilde{\mathcal{O}}_N \tilde{\mathcal{K}}^{(s)})}_{\bar{k}} \delta z_{\bar{k}} + \hat{\tilde{T}}_{u,N,\bar{k}} \delta U_{\bar{k},N,1} + \hat{\tilde{T}}_{y,N,\bar{k}} \delta Y_{\bar{k},N,1}. \tag{4.30}$$

If the Markov parameters are obtained from iteration-domain identification, then the window size $s = N$. Eliminating the predicted output from the right-hand side of the equation,

$$\delta Y_{\bar{k},N,1} = \widehat{(\mathcal{O}_N \tilde{\mathcal{K}}^{(s)})}_{\bar{k}} \delta z_{\bar{k}} + \hat{T}_{u,N,\bar{k}} \delta U_{\bar{k},N,1}. \tag{4.31}$$

This step involves making use of the following equalities:

$$(I_{n_y N} - \tilde{T}_{y,N})^{-1} \tilde{\mathcal{O}}_N \tilde{\mathcal{K}}^{(s)} = \mathcal{O}_N \tilde{\mathcal{K}}^{(s)}, \qquad (I_{n_y N} - \tilde{T}_{y,N})^{-1} \tilde{T}_{u,N} = T_{u,N}. \tag{4.32}$$

In order to predict the absolute value over an infinite horizon, the left hand side of the equation (4.31) is expanded:

$$Y_{\bar{k},N,1} = \begin{bmatrix} \widehat{(\mathcal{O}_N \tilde{\mathcal{K}}^{(s)})}_{\bar{k}} & I_{n_y N} \end{bmatrix} \begin{bmatrix} \delta z_{\bar{k}} \\ Y_{\bar{k}-1,N,1} \end{bmatrix} + \hat{T}_{u,N,\bar{k}} \delta U_{\bar{k},N,1}. \tag{4.33}$$

The objective is to write this statement in a state-transition form. For this, the term $\tilde{\mathcal{K}}^{(s)} \delta z_{\bar{k}}$ is rearranged and partitioned in such a manner that:

$$\tilde{\mathcal{K}}^{(s)} \delta z_{\bar{k}} = \begin{bmatrix} \tilde{\mathcal{K}}_u^{(N)} & \tilde{\mathcal{K}}_y^{(N)} \end{bmatrix} \begin{bmatrix} \delta U_{\bar{k},N,1} \\ \delta Y_{\bar{k},N,1} \end{bmatrix}. \tag{4.34}$$

Here, the data window has been extended from s to N by introducing zero padding at the appropriate locations. Substituting this separation into (4.33),

$$Y_{\bar{k},N,1} = \begin{bmatrix} \widehat{(\mathcal{O}_N \tilde{\mathcal{K}}_u^{(N)})}_{\bar{k}} & \widehat{(\mathcal{O}_N \tilde{\mathcal{K}}_y^{(N)})}_{\bar{k}} & I_{n_y N} \end{bmatrix} \begin{bmatrix} \delta U_{\bar{k},N,1} \\ \delta Y_{\bar{k},N,1} \\ Y_{\bar{k}-1,N,1} \end{bmatrix} + \hat{T}_{u,N,\bar{k}} \delta U_{\bar{k},N,1}. \tag{4.35}$$

This statement is reformulated into a state-transition equation that operates in the iteration domain instead of the time domain, such that an LQR-like control law can be devised. Such a method has been followed by Dijkstra and Bosgra (2002); however the current approach formulates the state-transition matrices directly from data and can hence be considered to be an adaptive extension of Dijkstra and Bosgra (2002). Since an

LQR-type controller is set up, for the case where the true system parameters are identified by the system, stability of the closed-loop system is guaranteed. The state-transition form of (4.35) is:

$$\underbrace{\begin{bmatrix} Y_{\bar{k},N,1} \\ \delta U_{\bar{k},N,1} \\ \delta Y_{\bar{k},N,1} \end{bmatrix}}_{\tilde{\mathcal{X}}_{\bar{k}+1}} = \underbrace{\begin{bmatrix} I_{n_yN} & \widehat{(\mathcal{O}_N\tilde{\mathcal{K}}_u^{(N)})}_{\bar{k}} & \widehat{(\mathcal{O}_N\tilde{\mathcal{K}}_y^{(N)})}_{\bar{k}} \\ 0_{n_uN\times n_yN} & 0_{n_uN\times n_uN} & 0_{n_uN\times n_yN} \\ 0_{n_yN\times n_yN} & \widehat{(\mathcal{O}_N\tilde{\mathcal{K}}_u^{(N)})}_{\bar{k}} & \widehat{(\mathcal{O}_N\tilde{\mathcal{K}}_y^{(N)})}_{\bar{k}} \end{bmatrix}}_{\tilde{A}_{\bar{k}}} \underbrace{\begin{bmatrix} Y_{\bar{k}-1,N,1} \\ \delta U_{\bar{k}-1,N,1} \\ \delta Y_{\bar{k}-1,N,1} \end{bmatrix}}_{\tilde{\mathcal{X}}_{\bar{k}}} + \underbrace{\begin{bmatrix} \hat{T}_{u,N,\bar{k}} \\ I_{n_uN} \\ \hat{T}_{u,N,\bar{k}} \end{bmatrix}}_{\tilde{B}_{\bar{k}}} \delta U_{\bar{k},N,1}. \tag{4.36}$$

It should be noted that, in this case, the system matrices $\tilde{\mathcal{A}}_{\bar{k}}$ and $\tilde{\mathcal{B}}_{\bar{k}}$ are updated once every iteration, based on the system identification from input-output data. Since it is assumed that the change in system dynamics is slow, this system description is considered to remain approximately correct for generating the system control law. If the identification was performed in the basis function space, or if the objective of RC is to achieve control only in the restricted input-output basis function space, then the state-transition equation (4.37) can be projected into this smaller subspace as follows:

$$\underbrace{\begin{bmatrix} \bar{Y}_{\bar{k},N,1} \\ \delta\bar{U}_{\bar{k},N,1} \\ \delta\bar{Y}_{\bar{k},N,1} \end{bmatrix}}_{\bar{\mathcal{X}}_{\bar{k}+1}} = \underbrace{\begin{bmatrix} I_{n_yN} & \phi_y\widehat{(\mathcal{O}_N\tilde{\mathcal{K}}_u^{(N)})}\phi_u^\dagger & \phi_y\widehat{(\mathcal{O}_N\tilde{\mathcal{K}}_y^{(N)})}\phi_y^\dagger \\ 0_{n_uN\times n_yN} & 0_{n_uN\times n_uN} & 0_{n_uN\times n_yN} \\ 0_{n_yN\times n_yN} & \phi_y\widehat{(\mathcal{O}_N\tilde{\mathcal{K}}_u^{(N)})}\phi_u^\dagger & \phi_y\widehat{(\mathcal{O}_N\tilde{\mathcal{K}}_y^{(N)})}\phi_y^\dagger \end{bmatrix}}_{\bar{A}_{\bar{k}}} \underbrace{\begin{bmatrix} \bar{Y}_{\bar{k}-1,N,1} \\ \delta\bar{U}_{\bar{k}-1,N,1} \\ \delta\bar{Y}_{\bar{k}-1,N,1} \end{bmatrix}}_{\bar{\mathcal{X}}_{\bar{k}}}$$
$$+ \underbrace{\begin{bmatrix} \phi_y\hat{T}_{u,N,\bar{k}}\phi_u^\dagger \\ I_{n_uN} \\ \phi_y\hat{T}_{u,N,\bar{k}}\phi_u^\dagger \end{bmatrix}}_{\bar{B}_{\bar{k}}} \delta\bar{U}_{\bar{k},N,1}. \tag{4.37}$$

The projected system matrices can either be obtained directly from the iteration-domain identification, as in Section 4.2.2, or by projecting the identified time-domain system matrices into the basis function subspace. Thus, at this point, an iteration-domain state-transition description of the system is available. The objective of the SPRC controller is then to minimise the (performance) output over an infinite horizon. In other words, the following minimisation problem is to be solved:

$$\min \sum_{\bar{k}=1}^{\infty} (\bar{\mathcal{X}}_{\bar{k}+1})^T Q_f \bar{\mathcal{X}}_{\bar{k}+1} + (\delta U_{\bar{k}})^T R_f \delta U_{\bar{k}}, \tag{4.38}$$

where Q_f and R_f are user-defined positive (semi-)definite weighting matrices. These terms can be used to tune the weighting on the control effort. A high value of Q_f and a low value of R_f indicate cheap control effort, and achieve larger disturbance rejection, while the opposite is true for a low value of Q_f an a high value of R_f. Increasing R_f typically also increases the robustness of the controller. Finally, the weighting matrices can be taken as diagonal for the sake of ease of tuning.

The similarities between the above minimisation problem and the conventional LQR minimisation problem are evident, however it should be noted that this problem is formulated in the iteration domain (projected into a basis function subspace), and not in the time domain, as with a conventional LQR problem. Making use of its similarity to the LQR problem it is possible to synthesise an iteration-domain state-feedback matrix K_f. Specifically, the Discrete Algebraic Riccati Equation (DARE) can be solved efficiently online, at each iteration, to obtain the (updated) value of K_f. Solution methods for the DARE are given in Arnold III and Laub (1984). For instance, starting with an initial estimate of $P_{R,\bar{k}}$, a positive definite solution of the DARE, the solution is iteratively optimised using the same equation repeatedly until convergence is achieved:

$$P_{R,\bar{k}+1} = Q_f + \bar{\mathcal{A}}_{\bar{k}}^T (P_{R,\bar{k}} - P_{R,\bar{k}} \bar{\mathcal{B}}_{\bar{k}}^T (R_f + \bar{\mathcal{B}}_{\bar{k}}^T P_{R,\bar{k}} \bar{\mathcal{B}}_{\bar{k}}^{-1} \bar{\mathcal{B}}_{\bar{k}}^T P_{R,\bar{k}} \bar{\mathcal{A}}_{\bar{k}}). \tag{4.39}$$

The state feedback matrix in this case is given by:

$$K_{f,\bar{k}} = (R_f + \bar{\mathcal{B}}_{\bar{k}}^T P_{R,\bar{k}} \bar{\mathcal{B}}_{\bar{k}})^{-1} \bar{\mathcal{B}}_{\bar{k}}^T P_{R,\bar{k}} \bar{\mathcal{A}}_{\bar{k}}. \tag{4.40}$$

Thus, it is possible to synthesise a repetitive control law directly on the basis of the system Markov parameters, identified as per the previous section or the previous chapter. The repetitive control law gives the optimal input control sequence for the next iteration:

$$U_{\bar{k},N,1} = U_{\bar{k}-1,N,1} + \phi_u^{\dagger} K_{f,\bar{k}} \begin{bmatrix} \phi_y Y_{\bar{k}-1,N,1} \\ \phi_u \delta U_{\bar{k}-1,N,1} \\ \phi_y \delta Y_{\bar{k}-1,N,1} \end{bmatrix}. \tag{4.41}$$

Since the control input only has energy along the basis vector directions, precise control over the shape and smoothness of the actuator signal is possible. Hereby, it is possible to formulate an RC control law starting from input-output data, where very limited knowledge about system dynamics is available.

4.4. STABILITY OF SPRC

For two-step data-driven algorithms, guarantees of stability are difficult to postulate. As shown in Dong and Verhaegen (2008), the two-step data-driven approach of Subspace Predictive Control has been shown to be equivalent to LQG control, and does not admit performance guarantees.

The discussion on stability in this section will proceed in the following manner: first, the nominal stability of the algorithm is discussed. Next, the stability of the system identification process and the resultant parameter estimate error is analysed. The effect of this error on the synthesis of the control law and the robust stability of the closed-loop system is described. Finally, the practical implications for the implementation of SPRC will be discussed. It should be noted that this discussion is valid for a slow evolution of plant dynamics; a breakdown in this assumption will imply that the stability of the algorithm cannot be guaranteed.

4.4.1. NOMINAL STABILITY

For the case where the underlying system is LTI and the identification step converges instantaneously to the true system parameters, the LQR formulation of the control law ensures that nominal stability of the closed-loop system is achieved.

4.4.2. STABILITY AND ERROR IN THE IDENTIFICATION STEP

First, it is assumed that the true Markov parameters of the system evolve slowly in such a manner that their evolution can be modelled as a random walk:

$$\Xi_{\bar{k}} = \Xi_{\bar{k}} + \gamma w_k, \tag{4.42}$$

where w_k is a zero-mean white noise sequence. As per Equation (4.16), the output can be predicted in terms of the past input-output data as:

$$\delta y_k = \Xi_k \delta z_k + \delta e_k. \tag{4.43}$$

Here, the feedthrough term has been taken to be zero for simplicity, but it can also be directly absorbed into the array of Markov parameters by extending the definition of z_k to include the current input u_k. The aim of this section is to determine the estimation error $\tilde{X}_k$, defined as the deviation of the Markov parameter estimate from its true value, $\tilde{\Xi}_k = \hat{\Xi}_k - \Xi_k$. If RLS is used for parameter estimation, with forgetting factor f and covariance matrix P_k, then the estimation error is given by Ljung and Gunnarsson (1990) to be:

$$\tilde{\Xi}_k = f P_k P_{k-1} \tilde{\Xi}_{k-1} + P_k (\delta z_k^T \delta e_k - P_k^{-1} \gamma w_k). \tag{4.44}$$

Also, according to Ljung and Gunnarsson (1990), the RLS algorithm is convergent for the case where:

- The condition of persistency of excitation holds.
- The inverse covariance P_k^{-1} increases at a rate that is smaller than exponential.

With an arbitrarily large value of persistency of excitation, and an adequately large value of f, these conditions can be satisfied, given that the closed-loop system is stable.

In order to approximate the parameter estimation error, the random perturbing disturbances are considered to admit a bounded description with arbitrarily high confidence. Hence, redefining the disturbance signals in terms of the real bounded uncertainty $\Delta_K \in [-1, 1]$,

$$w_k \approx \bar{W} \Delta_k, \qquad \delta e_k \approx \bar{E} \Delta_k, \tag{4.45}$$

are obtained with scaling factors $\bar{W}$ and $\bar{E}$. Substituting this description in Equation (4.44),

$$\tilde{\Xi}_k = f P_k P_{k-1} \tilde{\Xi}_{k-1} + P_k (\delta z_k^T \bar{E} - P_k^{-1} \gamma \bar{W}) \Delta_k. \tag{4.46}$$

Thus, at every instant of time, it is possible to obtain a bound on the system Markov parameters estimation error,

$$\Xi_k = \hat{\Xi}_k + \Delta_{\Xi,k}. \tag{4.47}$$

Herewith, it is possible to get an estimate of the system parameters and an uncertainty description. The effect of the uncertainty on the stability of the closed loop is discussed in the next section.

4.4.3. ROBUST STABILITY OF THE CLOSED LOOP

At this point, the system description is available in the form of a nominal system description perturbed by an unknown, but bounded, uncertainty. Based on the estimated Markov parameter $\hat{\Xi}_{\bar{k}}$ and the uncertainty bounds $\Delta_{\Xi,\bar{k}}$, it is possible to synthesise the nominal state-transition and input matrices $\hat{\bar{\mathcal{A}}}_{\bar{k}}$ and $\hat{\bar{\mathcal{B}}}_{\bar{k}}$, along with the corresponding uncertain matrices $\Delta_{\mathcal{A},\bar{k}}$ and $\Delta_{\mathcal{B},\bar{k}}$ [1]:

$$\bar{\mathcal{A}}_{\bar{k}} = \hat{\bar{\mathcal{A}}}_{\bar{k}} + \Delta_{\mathcal{A},\bar{k}}, \qquad \bar{\mathcal{B}}_{\bar{k}} = \hat{\bar{\mathcal{B}}}_{\bar{k}} + \Delta_{\mathcal{B},\bar{k}}. \tag{4.48}$$

Here the uncertain matrices are constructed by considering $\Delta_{\Xi,\bar{k}}$ elementwise as follows:

$$\Delta_{\mathcal{A},\bar{k}} = \sum_{i=1}^{n_a} D_i f_{i,j} C_i, \qquad \Delta_{\mathcal{B},\bar{k}} = \sum_{i=n_a+1}^{n_b} D_i f_{i,j} C_i, \tag{4.49}$$

with C_i and D_i the appropriate selection matrices, and $f_{i,j} \in [-1,1]$ the bounded uncertainty. The stability parameter can be defined, according to Neto et al. (1992) as:

$$\zeta_{\bar{k}} = \bar{\mathcal{X}}_{\bar{k}}^T (Q_f + K_{f,\bar{k}}^T R_f K_{f,\bar{k}}) \bar{\mathcal{X}}_{\bar{k}} - \sum_{i=1}^{n_b} \bar{\lambda} f_{i,j}^2 \bar{\mathcal{X}}_{\bar{k}}^T C_i^T C_i \bar{\mathcal{X}}_{\bar{k}} \tag{4.50}$$

$$- 2 \sum_{i=1}^{n_b} f_{i,j} \left(\bar{\mathcal{X}}_{\bar{k}}^T (\hat{\bar{\mathcal{A}}}_{\bar{k}} - \hat{\bar{\mathcal{B}}}_{\bar{k}})^T P_{R,j} D_i D_i^T P_{R,\bar{k}} (\hat{\bar{\mathcal{A}}}_{\bar{k}} - \hat{\bar{\mathcal{B}}}_{\bar{k}}) \bar{\mathcal{X}}_{\bar{k}} \bar{\mathcal{X}}_{\bar{k}}^T C_i^T C_i \bar{\mathcal{X}}_{\bar{k}} \right)^{\frac{1}{2}}. \tag{4.51}$$

In this equation, $\bar{\lambda}$ is the maximum eigenvalue of $\mathcal{D}^T P_{R,\bar{k}} \mathcal{D}$, with $\mathcal{D} = \begin{bmatrix} D_1 & D_2 & \cdots & D_{n_b} \end{bmatrix}$. For closed-loop stability, the following condition has to hold:

$$\zeta_{\bar{k}} > 0. \tag{4.52}$$

Herewith, the adaptive controller developed in the previous section yields a closed-loop stable system if the stability parameter is strictly positive.

4.4.4. PRACTICAL IMPLICATIONS

Since the bounds on the parameter estimation error and the stability parameter can be tracked online, it is possible to predict whether the synthesised control law will stabilise the system. If the stability parameter $\zeta_{\bar{k}} \leq 0$, then the control law may result in an unstable control loop. In such a case, if the nominal open-loop system is stable, the control input can be set trivially to zero.

In practice, when cold-starting the algorithm, several computational variables, such as the estimate covariance matrices in the RLS algorithm, may not have converged to their quasi-steady values, and as such the estimate of the stability parameter may be signficantly inaccurate. In such a situation, it would be advisable to use the stability parameter as an indication of closed-loop stability only after an adequate amount of data has been collected, and the Markov parameter estimates and their covariance matrices have converged to relatively stable values. It should also be noted that the computations required for tracking the stability parameter are numerically expensive, and computing this parameter in real-time may prove intractable. In the next section, the use of SPRC for rejecting periodic disturbances is described using a case study.

[1] The time index k has been replaced by iteration index $\bar{k}$ since the control law is synthesised in the iteration domain

4.5. RESULTS

The methodology of SPRC is demonstrated using a case study. First, SPRC applied to an LTI system is described, in terms of the performance of identification and control. Next, the effect of a sudden change in plant dynamics on SPRC disturbance rejection is evaluated. Finally, the advantages and disadvantages of the use of basis functions in the identification and the control step are investigated. In all simulations, the controller starts with zero knowledge about the system dynamics, and iteratively synthesises a control law.

4.5.1. LTI SYSTEM

An arbitrary LTI system G is generated, with three inputs and three outputs; the norm of the output is intended to be minimised. The block diagram representation of the SPRC controller can be seen in Fig. 4.1. In order to simulate the effect on a realistic plant, the plant order is taken to be relatively high, with the number of states $n = 49$. At each of the output, a periodic disturbance v is present; this disturbed output is to be rejected such that the net output y of the controlled system is 0.

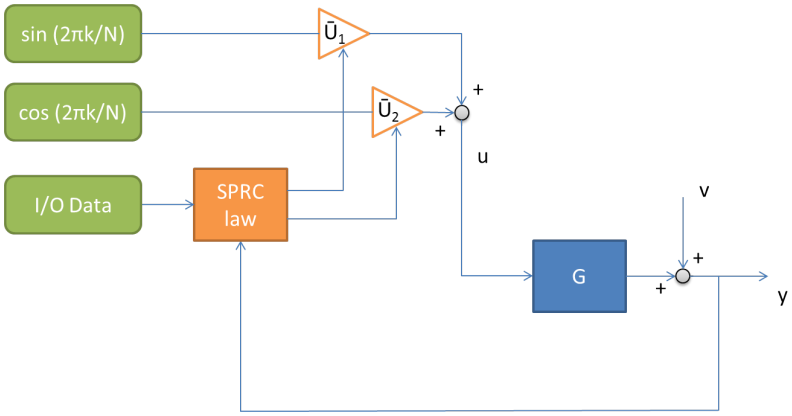


Figure 4.1: Scheme for the implementation of SPRC

The periodic disturbance is a superposition of two sinusoids such that:

$$v_k = \begin{bmatrix} \sin(2\pi k/N + \Psi_1) + \sin(4\pi k/N + \Psi_2) \\ \sin(2\pi k/N + \Psi_3) + \sin(4\pi k/N + \Psi_4) \\ \sin(2\pi k/N + \Psi_5) + \sin(4\pi k/N + \Psi_6) \end{bmatrix}, \tag{4.53}$$

where the term Ψ_* refers to an arbitrary value of the phase of the sinusoid.

As such, since the system is linear, it is expected that an input combination of sinusoids of the same frequency will be able to exactly cancel the periodic disturbance. Hence, the input is taken to be a combination of sinusoidal basis functions of the same

frequency:

$$u_k = \begin{bmatrix} \bar{U}_1 & \bar{U}_2 & \bar{U}_3 & \bar{U}_4 \\ \bar{U}_5 & \bar{U}_6 & \bar{U}_7 & \bar{U}_8 \\ \bar{U}_9 & \bar{U}_{10} & \bar{U}_{11} & \bar{U}_{12} \end{bmatrix} \begin{bmatrix} \sin(2\pi k/N) \\ \cos(2\pi k/N) \\ \sin(4\pi k/N) \\ \cos(4\pi k/N) \end{bmatrix}. \tag{4.54}$$

The objective of the SPRC algorithm is then to iteratively optimise the values of $\bar{U}_i, i = 1, ..., 12$, in order to minimise the effect of the disturbance on the output. For the sake of brevity, Fig. 4.1 only shows control inputs $\bar{U}_1$ and $\bar{U}_2$.

In order to formulate the stacked vectors for the SPRC law, the lifting window is taken the same as the period of the disturbance, N, where for these simulations, $N = 334$ samples. The lifting window is treated as the size of the iteration, thus the objective of SPRC is to update the control sequence over each iteration to achieve disturbance rejection. Identification of system parameters is done in the time domain, however the control sequence is optimised in the iteration domain. Since the argument of the optimisation would be large (of size $U_{\bar{k},N,1} \in \mathbb{R}^{n_u N}$, or 1002 elements long), the control input is projected into the basis function space defined by the projection matrices:

$$\phi_u = \begin{bmatrix} \sin(2\pi/N) & \sin(4\pi/N) & \cdots & \sin(2\pi) \\ \cos(2\pi/N) & \cos(4\pi/N) & \cdots & \cos(2\pi) \\ \sin(4\pi/N) & \sin(8\pi/N) & \cdots & \sin(4\pi) \\ \cos(4\pi/N) & \cos(8\pi/N) & \cdots & \cos(4\pi) \end{bmatrix}, \qquad \bar{U}_{\bar{k}} = \phi_u U_{\bar{k},N,1}. \tag{4.55}$$

It can be seen that the new optimisation variable $\bar{U}_{\bar{k}}$ is now only $n_u b = 12$ elements long. Similarly, since the input can only influence sinusoids in the output of the same frequency, the stacked output is also projected into the same basis function subspace using projection matrix $\phi_y = \phi_u$. Herewith, the dimensions of the problem can be strongly reduced without compromising on disturbance attenuation properties of the control law.

The iterative behaviour of SPRC in terms of rejecting the periodic disturbances acting on the output of the linear system can be observed in Fig. 4.2. In this figure, the control error has been normalised such that its value at the initial instant of time is unity. The performance of the identification part of the algorithm is measured in terms of the Variance Accounted For (VAF), Verhaegen and Verdult (2003). A value close to a 100% VAF implies that the system dynamics are described well by the identified model. It can be seen from the figure that, within 6 iterations, the identification part of SPRC reaches a VAF value greater than 90%. In the first two iterations, since the system dynamics have not been identified well (as shown by the low value of VAF), the SPRC control law amplifies the periodic disturbance instead of attenuating it. However, as a good model becomes available, the performance improves rapidly, with nearly perfect disturbance rejection after 30 iterations. It must be noted that the control input is always overlaid with a small random component (with power 1% that of the periodic disturbance), which is required by the identification part of SPRC to provide persistency of excitation. However, it can be seen that this component does not significantly degrade the performance of the SPRC control law.

The behaviour of the control input and system output over successive iterations can be seen in Fig. 4.3. In the initial iteration, the control input is zero, and the output contains a large periodic component. Over the iterations, the control input is updated until

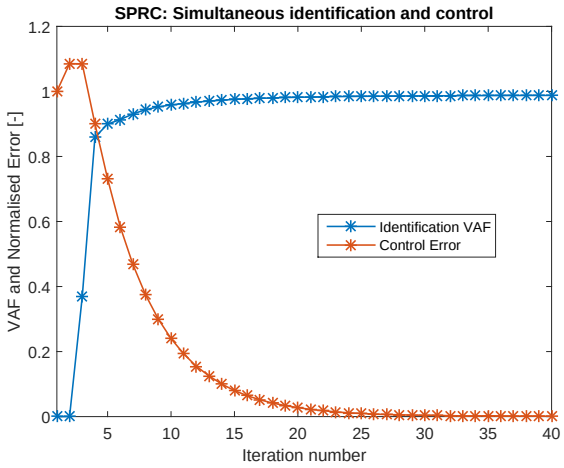


Figure 4.2: Iterative improvement in Identification VAF and disturbance rejection over iterations

it reaches its optimal value, at which point the effect of the periodic disturbance on the output is almost entirely eliminated. It can be seen that the control input is not exactly sinusoidal, but it also contains a small random variation which provides the required persistency of excitation.

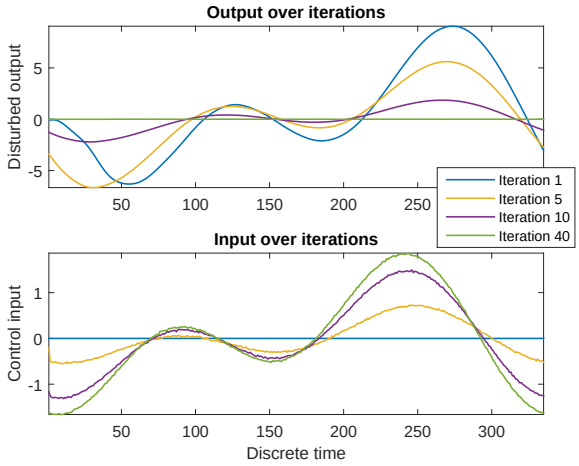


Figure 4.3: Iterative improvement of system output and control input over iterations

4.5.2. TIME-VARYING CHANGES IN DYNAMICS

The purpose of SPRC is to track changes in plant dynamics over time and accordingly change the control law so as to account for these changes. The same LTI system used

in the previous section is simulated for a duration of 50 iterations. Then, the sign of the transfer functions abruptly inverts itself at iteration number 51, and it is evaluated whether SPRC is able to estimate the changed system Markov parameters and adjust the control input sequence accordingly. The results can be seen in Fig. 4.4. As before, the cold-started SPRC algorithm converges to a good system estimate and a near-optimal control law after 20 iterations. However, after iteration 51, due to the large change in system Markov parameters, the identification VAF drops drastically to 18%. As a result of the model mismatch, the output error increases dramatically, upto a factor twice that of the uncontrolled system. However, the system identification step is able to recalibrate itself and reëstimate the system Markov parameters, once again achieving a VAF exceeding 90% within 10 iterations. As the system parameter estimate becomes better, the control law is also able to attenuate the periodic disturbances, and reach near-optimal disturbance rejection once more after 30 iterations.

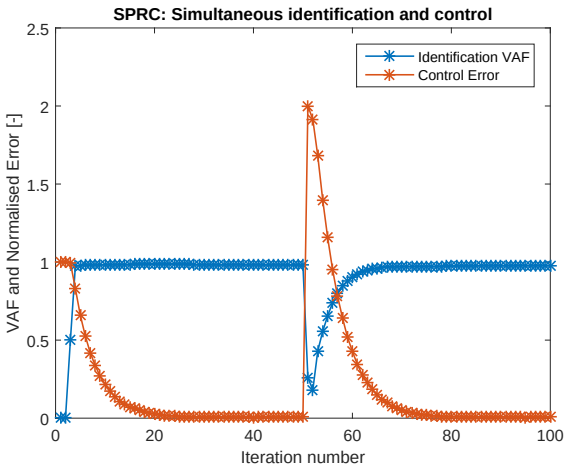


Figure 4.4: SPRC Identification VAF and disturbance rejection over iterations with changed plant dynamics at iteration 51

The change in the control input and performance output can be seen in Fig. 4.5. Initially, the control input is zero, and the periodic disturbance is directly fed through to the output. As the SPRC iterations proceed, the control input is successively updated, and at iteration 40, it is near ideal, such that the system output is virtually zero over time. However, when the system changes sign, the control input is exactly 180° out of phase, and it amplifies the disturbance instead of reducing it; this can be observed in the system response of iteration 55. However, with recursive reïdentification, SPRC is able to detect the changes in the system dynamics, and converge to the near-optimal control sequence at iteration 90, which is almost identically mirrored from the optimal sequence of iteration 40. Thus, SPRC is shown able to handle sudden large changes in system dynamics.

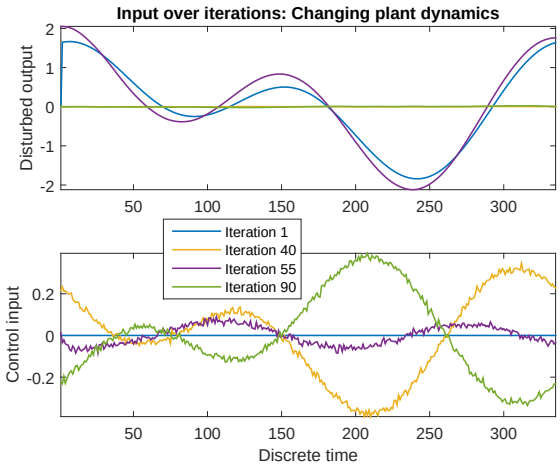


Figure 4.5: System output and control input with changing plant dynamics

4.5.3. THE RÔLE OF BASIS FUNCTIONS IN CONTROL

Thus far, it has been assumed that the frequency content of the periodic disturbance is exactly known, so that the basis functions used in the controller form sinusoids of these frequencies. However, in case the periodic disturbance contains frequency content outside of these frequencies, this component cannot be attenuated by SPRC control inputs that are constrained to remain within the basis functions space. On the other hand, if full-dimensional SPRC is used, then the entire periodic content of the disturbance signal can be targetted. This section compares basis-function SPRC with full-dimensional SPRC.

Since full-dimensional SPRC is a computationally heavy problem, a simpler LTI system is simulated in this section. An arbitrary second-order SISO system is generated for this purpose; the size N is limited to 100. Two different realisations of the periodic disturbance are used:

$$v_{k,1} = \sin(2\pi/N + \Psi_1) + \sin(4\pi/N + \Psi_2), \tag{4.56}$$

$$v_{k,2} = \sin(2\pi/N + \Psi_3) + \sin(4\pi/N + \Psi_4) + \sin(10\pi/N + \Psi_5). \tag{4.57}$$

However, the basis functions used are the same as those used previously:

$$u_k = \begin{bmatrix} \bar{U}_1 & \bar{U}_2 & \bar{U}_3 & \bar{U}_4 \end{bmatrix} \begin{bmatrix} \sin(2\pi k/N) \\ \cos(2\pi k/N) \\ \sin(4\pi k/N) \\ \cos(4\pi k/N) \end{bmatrix}. \tag{4.58}$$

Thus, the control input can provide energy only along the two lower frequencies. As such, when the SPRC law is implemented, the output error converges to zero for $v_{k,1}$, but remains non-zero for $v_{k,2}$, as seen in Fig. 4.6. The reason can be immediately deduced from Fig. 4.7. The optimised control input for both disturbance realisations is almost

identical, and contains energy only at the two low frequencies. While this is sufficient to drastically reduce the effect of $v_{k,1}$ on the output, the effect of $v_{k,2}$ is only partially reduced, and it still contains a high-frequency periodic disturbance.

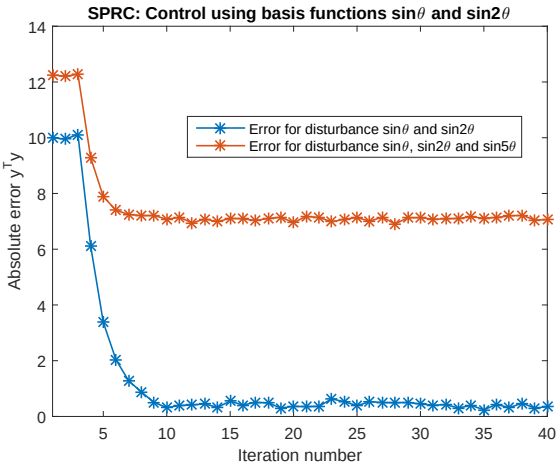


Figure 4.6: Control using basis functions $\sin\theta$ and $\sin 2\theta$

On the other hand, if full-dimensional SPRC is used, then, for both realisations of the periodic disturbance, SPRC is able to reduce the output error to nearly zero, as seen in Fig. 4.8. From Fig. 4.9, it can be seen that, for both $v_{k,1}$ and $v_{k,2}$, the output is reduced nearly to zero. It can be seen that in the former case, the optimised control input contains energy only at the two lower frequencies, and is identical to the control input synthesised by using the appropriate basis functions. On the other hand, for the case of rejection of periodic disturbance $v_{k,2}$, the optimal control input is different from the previous case, and it contains energy at three discrete frequencies in the spectrum, corresponding to the three frequencies in the periodic disturbance.

Thus, full-dimensional SPRC can give more accurate results for the case where the frequency content of the periodic disturbance is not exactly known. However, the computational complexity increases considerably if full-dimensional SPRC is used; this topic will be discussed in detail in the next section.

4.5.4. THE RÔLE OF BASIS FUNCTIONS IN IDENTIFICATION

As discussed in Section 1.2.2, identification can also be performed in the iteration domain, with the stacked input-output data projected into the basis functions subspace. This projection has the potential to drastically reduce the size of the identification problem. It also yields system Markov parameters directly suitable for use in the SPRC control law, formulated in the same basis. However, this form of identification is only able to predict the transfer along the basis vector directions.

The LTI system used in the previous section is used here once again to perform identification and control in the basis function space, using the same basis functions as described in the previous section. The performance of SPRC using basis functions for iden-

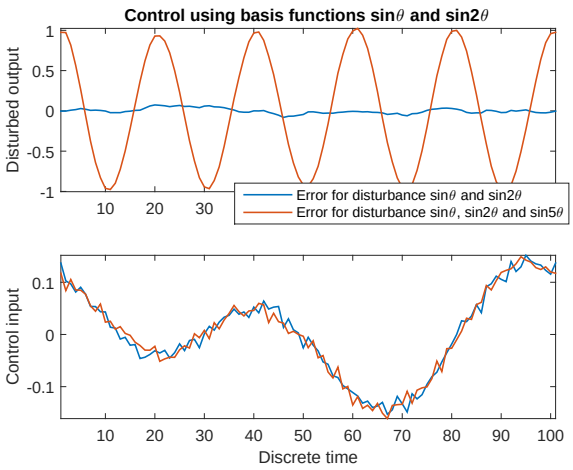


Figure 4.7: Performance output and control input using basis functions $\sin\theta$ and $\sin 2\theta$

tification and control is compared in Fig. 4.10 to that of SPRC using basis functions only for control. It can be seen that basis function space identification increases the time required by the algorithm to reach convergence.

One of the advantages of performing identification in the basis function space is that persistency of excitation is also required only in this subspace. Thus, instead of superposing a random white noise on the control signal, the persistency of excitation is provided in this case by superposing the signal $u_{\text{pers},k}$ such that:

$$u_{\text{pers},k} = 0.01(\sin(2\pi k/N + \Psi_1) + \sin(4\pi k/N + \Psi_2)), \tag{4.59}$$

where, as before, the terms Ψ_* refer to randomly generated sine phases. As a result of this, the control input of SPRC using basis functions for identification is smoother than the control input of standard SPRC, as depicted in Fig. 4.11.

Finally, the three versions of the SPRC algorithm, with and without basis functions for identification and/or control, can be compared against each other in Table 9.3. The computational times given are representative averages, for a problem of the size described in this section, run in a Matlab environment on an i3 core laptop.

From the table, it can be seen that the choice of using basis functions depends upon the intended application of SPRC. All algorithms tested in this section show good convergence properties, and are able to synthesise a control law starting from very limited system knowledge.

4.6. CONCLUSIONS

The new control strategy Subspace Predictive Repetitive Control (SPRC) was developed in this chapter specifically as a data-driven method for rejecting periodic disturbances or for periodic trajectory tracking. This two-step data-driven control approach uses black-box identification techniques, discussed in the previous chapter, to obtain the current

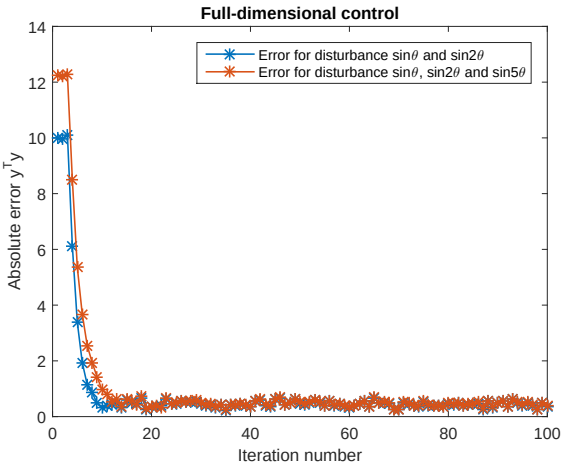


Figure 4.8: Full-dimensional SPRC Control

Table 4.1: SPRC: The Use of basis functions

	SPRC: Full-dimensional	SPRC: Basis functions only control	SPRC: Basis functions identification and control
Time per iteration (s)	760.4	10.01	1.821
Convergence	Fast	Fast	Slow
Accuracy	High	Restricted	Restricted
Actuator Duty	Unconstrained	Moderate	Extremely smooth

system Markov parameters. Based on these parameters, an LQR-like control law is synthesised online to reject periodic disturbances in a lifted 'iteration' domain. Since SPRC uses a state-space representation of the plant for the identification and control synthesis steps, this method is inherently MIMO by nature.

Simulations using a case study show that SPRC is able to correctly identify the system parameters and synthesise a control law to achieve near-optimal rejection of periodic disturbances, for the case where appropriate basis functions are used. Further, if the plant undergoes sudden changes in system dynamics, SPRC is able to detect these changes and reïdentify the system. Herewith, the control law can adapt to the changes and achieve near-optimal disturbance rejection again, once convergence has been reached.

The rôle of basis functions in the identification and control steps of SPRC has been studied. For full-dimensional SPRC, the stacked control sequence is optimised over a time horizon, taken to be an integral multiple of the period of the periodic disturbance. High accuracy is observed, however it involves a large dimensional optimisation problem, and is hence computationally expensive. When the stacked input-output data is

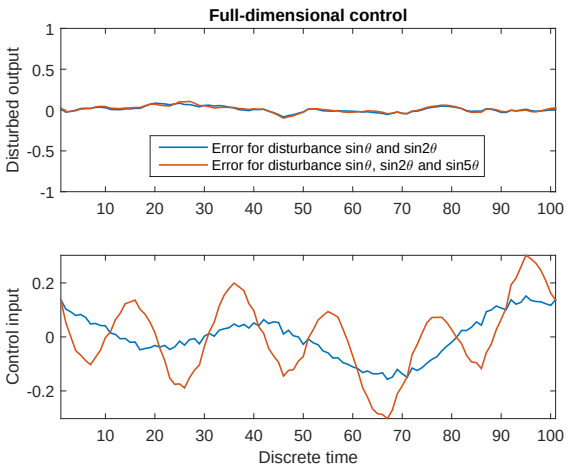


Figure 4.9: Performance output and control input using full-dimensional SPRC

projected into a basis function space for synthesising the control law, computational complexity is reduced, however, all components of the periodic disturbance cannot be rejected. While smooth control input sequences are synthesised, the identification step requires persistency of excitation across the frequency spectrum. When identification is performed in the basis function space, persistency of excitation is only required in this space; hence the control inputs demanded by SPRC are extremely smooth. This approach strongly reduces the computational time per iteration, however, it requires multiple iterations for convergence. The choice of using basis functions for identification and/or control thus depends mainly upon the application at hand.

This chapter explores the use of a predictive control cost function that is optimised over an infinite horizon. This implementation ensures that when the true system parameters are available to SPRC, the control law synthesised will always be stabilising. However, in case hard constraints are to be imposed on the inputs or outputs of the system, optimisation can be performed over a finite horizon subject to these constraints.

Finally, the overall stability of the algorithm depends upon the uncertainty involved in the estimation of the system Markov parameters in the identification step. The effect of this uncertainty can be tracked using a stability parameter developed in this chapter. However, the recursive calculation of this parameter is computationally expensive; it is also sensitive to initial conditions.

Thus, it may be concluded that a cautious implementation of SPRC may be useful for the control of periodic disturbances for the case where the plant dynamics are unknown or highly uncertain. Hence, this algorithm will be validated for the application of wind turbine load control in the second part of this book. One of the drawbacks of this approach is that continuous reïdentification is necessary to track changes in system dynamics, which appears superfluous for plants known to admit LPV realisations. The repeated identification step can be avoided by formulating an LPV extension of SPRC. Alternatively an LPV extension of a direct data-driven approach can be developed, such

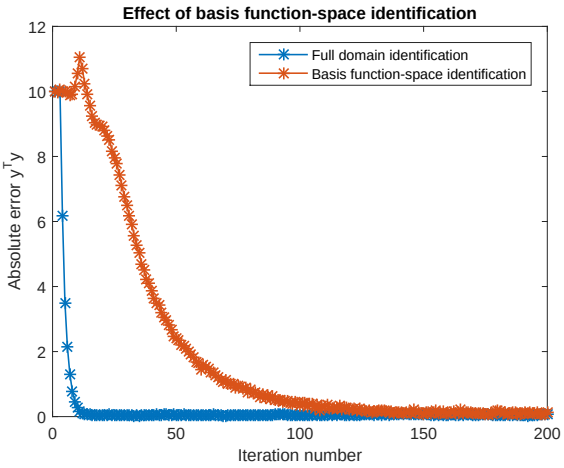


Figure 4.10: SPRC using identification in the basis function space, comparison with standard SPRC

an approach is investigated in the next chapter.

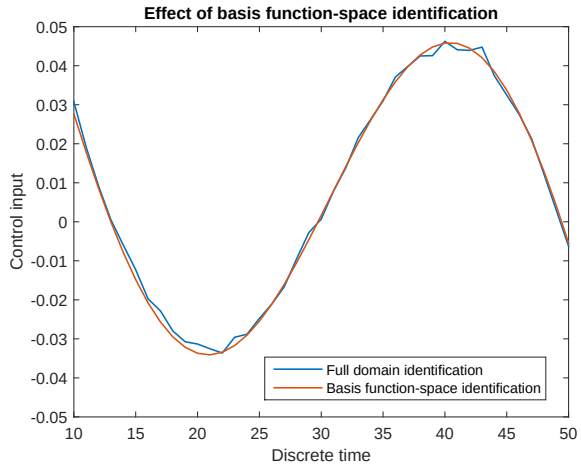


Figure 4.11: Control input synthesised by SPRC using identification in the basis function space, comparison with standard SPRC

5

ITERATIVE FEEDBACK TUNING FOR LPV SYSTEMS

Dans ses écrits, un sage Italien;
Dit que le mieux est l'ennemi du bien.

Voltaire quoting Proverbi Italiani, France (1772),
extolling the virtues of local optimisation.

While the previous chapter dealt with the two-step data-driven approach to control, this chapter will focus on the Iterative Feedback Tuning (IFT) of controllers, which is a direct data-driven approach to control. This method directly optimises controller parameters from data, and is hence able to autonomously tune low-order fixed structure controllers, such as PID controllers, online. The chapter explores an extension of IFT to Linear Parameter-Varying (LPV) systems, such that a global controller can be synthesised that is valid for any scheduling sequence that the plant is subject to. First, the methodology of IFT is extended to the tuning of feedforward controllers for LPV systems. Next, this methodology is extended to IFT for the tuning of feedback controllers in closed loop, where the curse of dimensionality is encountered, which implies that exact gradient estimation demands a combinatorial explosion in the number of IFT experiments. A practical method is then described, designed to perform IFT by approximating the performance gradient estimate. Finally, a case study is used to describe the performance of IFT for LPV systems.

5.1. INTRODUCTION

The previous chapters describe data-driven control using a two-step approach, with recursive online identification and control law synthesis forming two distinct steps. While

Parts of this chapter have been published in the Proceedings of the IFAC Workshop on Linear Parameter Varying Systems, Grenoble, France, Navalkar and Van Wingerden (2015), the Proceedings of the IFAC Symposium on System Identification, Beijing, China, Navalkar et al. (2015a) and under review in Wind Energy Science, Navalkar et al. (2016).

the identification step may provide additional data, for instance regarding the open-loop stability of the plant, it is not strictly necessary to synthesise the control law, and it may introduce an avoidable source of error. Secondly, while both the identification and control involve convex optimisations, the dimensionality may become arbitrarily large and intractable in real time; they may also yield unnecessarily high-order controllers which are typically sought to be avoided in practical applications. Finally, while two-step approaches have been extended to handle non-linear, especially Linear Parameter-Varying (LPV) systems, the identification step typically demands a large amount of data that may affect the feasibility of their implementation in practice.

These drawbacks motivate the use of a direct data-driven approach for autonomously optimising low-order controllers. Iterative Feedback Tuning (IFT), described in Chapter 2, is considered suitable to locally optimise fixed-structure controllers in closed loop. To briefly recapitulate, IFT assumes that there exists a nominally stabilising, parameterised controller in closed loop with the plant. Based on a set of the so-called 'reference' and 'gradient' experiments, IFT determines the performance gradient with respect to the controller parameters. Once such a gradient estimate becomes available, a standard gradient descent method can be used to optimise the parameters. IFT offers no guarantees of stability; indeed, if the cost function is poorly defined, IFT can yield an unstable closed loop after convergence. However, for a correctly defined optimisation criterion, if sufficiently small steps are taken in the gradient descent algorithm, IFT will find the local minimum of the defined (non-convex) cost function.

As referenced in Chapter 2, IFT has been applied successfully to applications where the underlying plant is considered Linear Time-Invariant (LTI). However, this method cannot be directly applied to LPV plants, since the gradient estimates from the IFT experiments are then a function of the scheduling trajectory during the experiments. If IFT-LTI experiments are done for an LPV plant, the controller gains would not stabilise to their (locally) optimal values, but would instead vary with time. One method to cope with the LPV nature of the plant would be to find the optimal values of the controller gains for (approximately) constant operating points, and then interpolate a gain schedule between these points. However, such an approach revisits the drawbacks of gain-scheduled controllers: the exponential increase in the number of operating points required with an increase in non-linearity or dimensionality, and the non-optimality of the gain schedule at the interpolated operating points. Some other alternatives found in the literature include the treatment of scheduling variation as an uncertainty, Van der Velden et al. (2014), which is then dealt with by synthesising a robust controller. Another possible alternative is the use of a switching controller, Koumboulis et al. (2007), tuned using IFT, that detects changes in the plant dynamics and switches to the appropriate LTI controller for the corresponding operating point. However, the tuning of a fully LPV controller that globally optimises a fully LPV plant using IFT has not yet been addressed.

The first contribution of this chapter hereby involves the recasting of the IFT framework into a state-space form, such that multivariable systems and fully LPV systems can be handled with ease. Initially, the use of IFT, for optimally tuning a gain schedule for an LPV plant, is discussed. The system matrices are then decomposed into two factors: a known, scheduling-dependent factor that varies with time, and an unknown factor that stays constant over time. The objective of IFT here is to perform an adequate number

of gradient experiments to compensate for this unknown factor and estimate the performance gradient in an unbiased manner. As done in standard IFT, the gradient can then be used to optimise the parameters of the full LPV controller.

To achieve LPV control, the next section describes the methodology of IFT used for tuning a gain-scheduled feedforward controller. In the following section, IFT will be used for synthesising an LPV feedforward controller. Finally, the method of IFT will be employed for the tuning of a fully LPV feedback controller. A case study will be used to demonstrate the working of such an IFT-LPV strategy, and this chapter will end with conclusions.

5.2. IFT OF A GAIN-SCHEDULED FEEDFORWARD CONTROLLER FOR LPV SYSTEMS

As a first step, standard IFT for LTI systems will be extended such that it is able to tune a gain schedule for an LPV system, using input-output data. First, the notation will be set up. Then, the IFT experiments required to obtain the performance gradients will be defined. Finally, the synthesis of a gain schedule based on these experiments will be discussed.

5.2.1. PRELIMINARIES AND NOTATION

Let us consider that the system to be controlled admits an LPV state-space realisation given by:

$$x_{k+1} = A_k x_k + B_k u_k \tag{5.1}$$

$$y_k = C_k x_k + D_k u_k + v_k. \tag{5.2}$$

Here, it is assumed that the system is stable. If it is not open-loop stable, then it is assumed that the system is operating in closed loop with a nominally stabilising controller such that its closed-loop representation admits a stable LPV state-space representation like the one given above. Here, $x_k \in \mathbb{R}^n$ is the state vector, of unknown length. The input signal is represented by $u_k \in \mathbb{R}^{n_u}$, while the output signal is represented by $y_k \in \mathbb{R}^{n_y}$. The signal $v_k \in \mathbb{R}^{n_y}$ is the measurement noise; although no process noise has been included in this model, the extension thereto is direct. The system matrices A_k, B_k, C_k and D_k are of the appropriate dimensions, and vary with time.

Since the system is taken to be LPV, the same matrices are at any point of time considered to be affinely dependent on a weighted combination of basis functions that take the instantaneous value of the scheduling variable, $\mu_k \in \mathbb{R}^{n_\mu}$ as argument:

$$A_k = A^{[0]} + \sum_{i=1}^{n_\psi} \psi_i(\mu_k) A^{[i]}, \tag{5.3}$$

and similarly for B_k, C_k and D_k. Here, $\psi_i : \mathbb{R}^{n_\mu} \to \mathbb{R}, i = 1,..,n_\mu$ is a set of basis functions chosen such that they best describe the dependence of the system matrices on the scheduling variable. It is assumed here and throughout this chapter that μ_k is exogenous, perfectly measurable, and cannot be controlled.

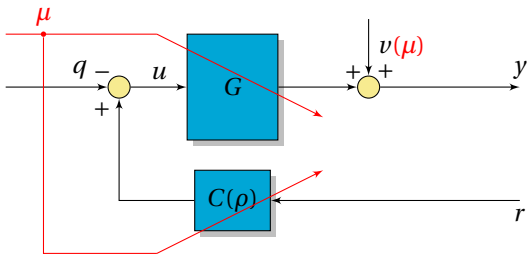


Figure 5.1: IFT experiment for tuning an LPV feedforward controller for an LPV plant.

The objective of IFT is to synthesise a parameterised controller that adjusts to the instantaneous value of μ_k and generates the appropriate control inputs, described in its state-space form as:

$$\begin{aligned} x_{c,k+1} &= A_{c,k}(\rho)x_{c,k} + B_{c,k}(\rho)r_k \\ u_k &= C_{c,k}(\rho)x_{c,k} + D_{c,k}(\rho)r_k - q_k. \end{aligned} \tag{5.4}$$

Here, the state-space matrix $A_{c,k}(\rho)$ is also scheduled on the scheduling variable μ in the following manner:

$$A_{c,k} = A_c^{[0]} + \sum_{i=1}^{n_\psi} \psi_i(\mu_k) A_c^{[i]}, \tag{5.5}$$

and similarly for $B_{c,k}$, $C_{c,k}$ and $D_{c,k}$.

The controller is taken to be a fixed-structure controller, such as a PID controller. Hence, the size of the state vector $x_{c,k} \in \mathbb{R}^{n_c}$ is small and fixed. The controller is parameterised using the controller parameters $\rho \in \mathbb{R}^{n_\rho}$; these parameters are sought to be optimised by using IFT. The term $q_k \in \mathbb{R}^{n_u}$ refers to an auxiliary input required for the IFT gradient experiments. The signals can be visualised in Fig. 5.1.

The objective of IFT is to tune the controller parameters, such that a performance criterion J is minimised. Preferably, the performance should be convex and differentiable, for instance, this chapter considers a quadratic cost criterion given as:

$$J = \frac{1}{2N}((R_{k,N,1} - Y_{k,N,1})^T(R_{k,N,1} - Y_{k,N,1}) + \lambda_J U_{k,N,1}^T U_{k,N,1}). \tag{5.6}$$

It should be noted here that, since IFT is used to tune a feedforward controller, the system will always be stable. The effect of the control effort vis-à-vis the output error is weighted by the user-defined tuning parameter λ_J. As in the previous chapters, the vectors $Y_{k,N,1}^T = \begin{bmatrix} y_k^T & y_{k+1}^T & \cdots & y_{k+N-1}^T \end{bmatrix}$ are stacked vectors, with similar expressions for the stacked reference signal $R_{k,N,1}$ and the input signal, $U_{k,N,1}$. The objective of the IFT experiments is to estimate, from input-output data, the gradient of this performance criterion J, with respect to the controller parameters ρ, given by:

$$\frac{\partial J}{\partial \rho} = \frac{1}{N}(-\frac{\partial Y_{k,N,1}^T}{\partial \rho}(R_{k,N,1} - Y_{k,N,1}) + \lambda_J \frac{\partial U_{k,N,1}^T}{\partial \rho} U_{k,N,1}). \tag{5.7}$$

As in the previous chapters, the system behaviour over a time horizon is described in terms of the Toeplitz matrix $T_{u,N,k} \in \mathbb{R}^{n_y N \times n_u N}$, defined in this LPV case as follows:

$$T_{u,N,k} = \begin{bmatrix} D_k & 0 & \cdots & 0 \\ C_{k+1}B_k & D_{k+1} & \cdots & 0 \\ C_{k+2}A_{k+1}B_k & C_{k+2}B_{k+1} & \cdots & 0 \\ C_{k+3}A_{k+2}A_{k+1}B_k & C_{k+3}A_{k+2}B_{k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{k+N-1}A_{k+N-2}\cdots B_k & C_{k+N-1}A_{k+N-2}\cdots B_{k+1} & \cdots & D_{k+N-1} \end{bmatrix} \tag{5.8}$$

A similar Toeplitz matrix can be defined using the controller state-space matrices, $T_{c,N,k} \in \mathbb{R}^{n_u N \times n_y N}$. As distinct from the previous chapters, these Toeplitz matrices explicitly vary with time, specifically, with the scheduling variable μ_k. For the case where the scheduling variable is approximately constant at μ_*, these matrices take the constant values $T_{u,N}(\mu_*)$ and $T_{c,N}(\mu_*)$. A series of IFT-LTI experiments can then be conducted to obtain the ideal controller parameters at this operating point.

5

5.2.2. IFT EXPERIMENTS

As per Hjalmarsson (2002), given an LTI plant, three IFT experiments are necessary for an unbiased estimate of the performance gradient with respect to controller parameters. Since the scheduling variable is here held constant, the plant can be considered to be an LTI system, and the performance gradient can be directly obtained using these standard IFT experiments. This approach is recast in a state-space framework in this section, so that it is directly applicable to multivariable systems.

For the first experiment, also called the 'reference' experiment, the auxiliary input is set to zero, $Q^I_{k,N,1} = 0$. The data equation then becomes:

$$Y^I_{k,N,1} = T_{u,N}(\mu_*)T_{c,N}(\mu_*)R_{k,N,1} + V^I_{k,N,1}(\mu_*). \tag{5.9}$$

In order to find the performance gradient, from Equation (5.7), it is necessary to evaluate the gradient of the system output with respect to the controller parameters, $\frac{\partial Y^I_{k,N,1}}{\partial \rho}$. So, the equation above is differentiated with respect to each controller parameter ρ_{j_ρ}, $j_\rho = 1, 2, ..., n_\rho$, giving:

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = T_{u,N}(\mu_*)\frac{\partial T_{c,N}(\mu_*)}{\partial \rho_{j_\rho}}R_{k,N,1}. \tag{5.10}$$

The system matrix $T_{u,N}(\mu_*)$ is the only unknown in this equation. To be able to estimate the required gradient, hence, one more experiment is required, the so-called 'gradient experiment'. For this experiment, the auxiliary input is taken to be:

$$Q^{II}_{k,N,1} = \frac{\partial T_{c,N}(\mu_*)}{\partial \rho_{j_\rho}}R_{k,N,1}. \tag{5.11}$$

In this case, the output then becomes:

$$Y^{II}_{k,N,1} = (T_{u,N}(\mu_*)T_{c,N}(\mu_*) - T_{u,N}(\mu_*)\frac{\partial T_{c,N}(\mu_*)}{\partial \rho_{j_\rho}})R_{k,N,1} + V^{II}_{k,N,1}. \tag{5.12}$$

From direct observation, the required output gradient $\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}}$ can be constructed as:

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = Y^I_{k,N,1} - Y^{II}_{k,N,1} + V^{II}_{k,N,1}(\mu_*) - V^I_{k,N,I}(\mu_*). \tag{5.13}$$

In the case where the signal v_k represents pure white noise, or white noise overlaid with a periodic component of period N, the term $V^{II}_{k,N,1}(\mu_*) - V^I_{k,N,I}(\mu_*)$ is also a stacked zero mean white noise sequence in the iteration domain. Hence, if the output gradient is constructed in the manner given below, it will be ergodically unbiased:

$$\frac{\partial \hat{Y}^I_{k,N,1}}{\partial \rho_{j_\rho}} = Y^I_{k,N,1} - Y^{II}_{k,N,1}. \tag{5.14}$$

However, the performance gradient term contains a product of the gradient with the output error, which cannot be directly estimated as:

$$-(\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}})^T (R_{k,N,1} - Y^I_{k,N,1}) = (Y^{II}_{k,N,1} - Y^I_{k,N,1})^T (R_{k,N,1} - Y^I_{k,N,1}). \tag{5.15}$$

This is because the noise in the output gradient estimate is correlated with the noise in the output error, and hence the estimate of the performance gradient would be biased. In order to remove the biasing influence of the noise, another reference experiment is conducted, once again taking the auxiliary input as 0. In this case, the output $Y^{III}_{k,N,1}$ will be deterministically equivalent to the output of reference experiment I, but its value will be perturbed by an uncorrelated realisation of noise. As such, the first term of cost equation (5.7) is estimated as:

$$-(\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}})^T (R_{k,N,1} - Y^I_{k,N,1}) = (Y^{II}_{k,N,1} - Y^I_{k,N,1})^T (R_{k,N,1} - Y^{III}_{k,N,1}). \tag{5.16}$$

Finally, the input gradient with respect to the controller parameters can be directly synthesised as:

$$\frac{\partial U^I_{k,N,1}}{\partial \rho_{j_\rho}} = \frac{\partial T_{c,N}(\mu_*)}{\partial \rho_{j_\rho}} R_{k,N,1}. \tag{5.17}$$

Based on the above gradients, it is possible to estimate the performance gradient, $\frac{\partial J}{\partial \rho_{j_\rho}}\Big|_{\mu=\mu_*}$, for a specific operating point denoted by μ_*, using equation (5.7). At this juncture, it is possible to use a gradient-based optimisation method to optimise the controller parameters. It should be noted that these parameters will be (locally) optimal only for the specific value of the scheduling variable, $\mu = \mu_*$. A typical gradient descent step could be as follows:

$$\rho^{\bar{k}+1}_{j_\rho}(\mu_*) = \rho^{\bar{k}}_{j_\rho}(\mu_*) - \gamma^{\bar{k}} R_H^{-1} \frac{\partial J}{\partial \rho_{j_\rho}}\Big|_{\mu=\mu_*}. \tag{5.18}$$

In this equation, the term $\gamma^{\bar{k}}$ is the (iteration-dependent) step size in the optimisation algorithm. As long as this step size is adequately low, the optimisation procedure will

converge to a local optimum. The term R_H signifies a positive definite matrix, which is identity for the steepest descent method. The rate of convergence can be increased by using an estimate of the Hessian of the optimisation cost function. Such an estimate can be constructed from the data collected in the gradient experiment and the two reference experiments, but it will be biased. Unbiased Hessian estimates require more IFT experiments.

Thus, for a constant value of the scheduling variable, the optimal controller gains can be determined. This method can also be directly extended to tune a gain schedule, in a data-driven manner, that is globally optimal for all values of the scheduling variable.

5.2.3. DATA-DRIVEN GAIN SCHEDULE SYNTHESIS

From the previous section, it is possible to derive the ideal controller parameters ρ^* for every scheduling variable operating point μ_*. At iteration $\bar{k}$, it is assumed that the dependence of these ideal parameters can be described as a weighted affine sum of basis functions scheduled on the current value of the scheduling variable, $\mu_{\bar{k}}$:

$$\rho^*(\mu_{\bar{k}}) = \rho^{[0],*} + \sum_{i=1}^{n_\psi} \psi_i(\mu_{\bar{k}})\rho^{[i],*}. \tag{5.19}$$

It has here been assumed that the gain schedule uses the same basis functions as the ones used for defining the LPV structure of the plant and/or controller. However, such a definition is unnecessary, indeed, determining the best set of basis functions for defining the gain schedule is a non-trivial task. The greater the non-linearity of the underlying system, the larger is the number of basis functions required for describing a good gain schedule. On the other hand, if the input-output data is noisy, estimating a smooth gain schedule may become difficult, and it may be deemed necessary to reduce the number of basis functions used in describing the gain schedule.

IFT can be used to iteratively determine the values of $\rho^{[i],*}, i = 0, ..., n_\psi$ directly from the input-output data, using a least squares approach. Thus, after the following the approach of the previous section, when the pair of values $\rho^{\bar{k}}$ and $\mu_{\bar{k}}$ become available, Recursive Least Squares (RLS) can be used to update the values of $\rho^{[i],*}$ following equation (5.19).

The experimental implementation of this approach will be discussed in the second part of the thesis. It should be noted at this point, that the IFT approach in this section demands that the scheduling remain approximately constant over the three IFT experiments. This strict requirement is relaxed in the next sections and IFT will be discussed for arbitarily varying exogenous scheduling sequences.

5.3. IFT FOR FEEDFORWARD CONTROL OF SYSTEMS LPV IN THE OUTPUT MATRICES

The restriction on the exogenous scheduling is relaxed in this section, and the scheduling sequence can vary arbitrarily over time, and is required to be persistently exciting. This section will focus on the data-driven tuning of a feedforward controller, for the case where the system is LPV in the output matrices. For this special class of LPV systems,

the curse of dimensionality is not encountered, and IFT gradient estimation is still possible using a limited number of IFT experiments. The block diagram representation of the IFT controller is identical to the one from the previous section, depicted in Fig. 5.1. The objective of the IFT method is once more to use input-output data from the IFT experiments to tune the parameters ρ of the parameterised controller $C(\rho)$ such that the output error is minimised.

First, a factorisation of the system Toeplitz matrices is defined. Next, the updated IFT experiments for gradient estimation in the presence of a varying scheduling sequence are described. Finally, the use of the gradients for updating the controller parameters is delineated.

5.3.1. LPV FACTORISATION

Let us consider that the system G in this case is LPV only in its output matrices, described by the state space representation:

$$x_{k+1} = Ax_k + Bu_k \tag{5.20}$$

$$y_k = C_k x_k + D_k u_k + v_k. \tag{5.21}$$

The signals x_k, u_k, y_k and v_k are the state, input, output and disturbance signals as defined in the previous section. The system matrices A and B are in this section taken to be constant, while the matrices C_k and D_k are LPV and depend upon the current value of the scheduling variable, μ_k as follows:

$$C_k = C^{[0]} + \sum_{i=1}^{n_\psi} \psi_i(\mu_k) C^{[i]}, \tag{5.22}$$

and similarly for the LPV matrix D_k. As before, $\psi_i(\mu_k), i = 1, ..., n_\psi$ are a series of basis functions scheduled on the variable μ_k, which is considered exogenous but perfectly measurable. The feedforward controller is described by (5.4), and it is parameterised using the parameters $\rho \in \mathbb{R}^{n_\rho}$. The objective of IFT is to use input-output data to optimise these parameters, with respect to the cost function (5.6). As this cost function is differentiable, the objective is once more to find the performance gradient with respect to the controller parameters, given by (5.7).

Data equations similar to the ones found in the previous section can be set up here. Since the system is here considered to be LPV only in the output matrices, the definition of the lifted system matrix $T_{u,N,k} \in \mathbb{R}^{n_y N \times n_u N}$ is simplified as:

$$T_{u,N,k} = \begin{bmatrix} D_k & 0 & \cdots & 0 \\ C_{k+1}B & D_{k+1} & \cdots & 0 \\ C_{k+2}AB & C_{k+2}B & \cdots & 0 \\ C_{k+3}A^2B & C_{k+3}AB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{k+N-1}A^{N-2}B & C_{k+N-1}A^{N-3}B & \cdots & D_{k+N-1} \end{bmatrix}. \tag{5.23}$$

In order to separate the known time-varying variables μ_k from the unknown time-invariant state-space matrices, the expansion (5.22) is made use of:

$$T_{u,N,k} = \underbrace{\begin{bmatrix} D^{[0]} & 0 & \cdots & 0 \\ C^{[0]}B & D^{[0]} & \cdots & 0 \\ C^{[0]}AB & C^{[0]}B & \cdots & 0 \\ C^{[0]}A^2B & C^{[0]}AB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C^{[0]}A^{N-2}B & C^{[0]}A^{N-3}B & \cdots & D^{[0]} \end{bmatrix}}_{T^{[0]}_{u,N}}$$
$$+ \sum_{i=1}^{n_\psi} \underbrace{\begin{bmatrix} \psi_i(\mu_k)I & 0 & \cdots & 0 \\ 0 & \psi_i(\mu_{k+1})I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_i(\mu_{k+N-1})I \end{bmatrix}}_{M_N(\psi_i(\mu_k))} \underbrace{\begin{bmatrix} D^{[i]} & 0 & \cdots & 0 \\ C^{[i]}B & D^{[i]} & \cdots & 0 \\ C^{[i]}AB & C^{[i]}B & \cdots & 0 \\ C^{[i]}A^2B & C^{[i]}AB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C^{[i]}A^{N-2}B & C^{[i]}A^{N-3}B & \cdots & D^{[i]} \end{bmatrix}}_{T^{[i]}_{u,N}} \tag{5.24}$$

Here, $I \in \mathbb{R}^{n_y \times n_y}$ is the identity matrix. A similar decomposition can be performed for the controller system matrix, $T_{c,N,k}$. Further, it is assumed that the disturbance properties also depend upon the scheduling trajectory in a similar manner:

$$T_{c,N,k} = T^{[0]}_{c,N} + \sum_{i=1}^{n_\psi} M_N(\psi_i(\mu_k))\, T^{[i]}_{c,N} \tag{5.25}$$
$$V_{k,N,1} = V^{[0]}_{k,N,1} + \sum_{i=1}^{n_\psi} M_N(\psi_i(\mu_k))\, V^{[i]}_{k,N,1}.$$

The objective of the IFT experiments is then to estimate the performance gradient in the presence of the perturbing effect of the scheduling variation.

5.3.2. EXPERIMENT I

As in the previous section, the first IFT experiment is the reference experiment, conducted with the auxiliary input q_k set to zero. Herewith, as before, the data equation becomes:

$$Y^I_{k,N,1} = T_{u,N}(\mu^I_k)\, T_{c,N}(\mu^I_k) R_{k,N,1} + V^I_{k,N,1}. \tag{5.26}$$

From equation (5.7), in order to determine the performance gradient, it is necessary to use input-output data to determine the output gradient:

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = T_{u,N}(\mu^I_k) \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}} R_{k,N,1}. \tag{5.27}$$

Factorising the effect of the scheduling variable out of the system matrices, using the equations (5.24)-(5.25):

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = M(\mu^I_k)\bar{T}_{u,N}\frac{\partial T^I_{c,N}}{\partial \rho_{j_\rho}}R_{k,N,1}, \tag{5.28}$$

$$\bar{T}^T_{u,N} = \begin{bmatrix}(T^{[0]}_{u,N})^T & (T^{[1]}_{u,N})^T & \cdots & (T^{[n_\psi]}_{u,N})^T & 0 & \cdots & 0\end{bmatrix}. \tag{5.29}$$

Here, $M(\mu^I_k) \in \mathbb{R}^{n_y N \times \frac{(n_\psi+1)(n_\psi+2)n_y}{2}}$ is partitioned in the following manner:

$$M(\mu^I_k) = \begin{bmatrix} I_{n_y N} & \mathcal{M}(\mu^I_k) & \mathcal{P}(\mu^I_k)\end{bmatrix}, \tag{5.30}$$

such that the block elements of the matrix $\mathcal{M}(\mu^I_k) \in \mathbb{R}^{n_y N \times n_\psi n_y N}$ are given by:

$$\mathcal{M}_i(\mu^I_k) = M_N(\psi_i(\mu^I_k)), \qquad i = 1,...,n_\psi. \tag{5.31}$$

5

and the block elements of the matrix $\mathcal{P}(\mu_k) \in \mathbb{R}^{n_y N \times \frac{n_\psi(n_\psi+1)n_y}{2} N}$ is given as:

$$\mathcal{P}_{i n_\psi - \frac{(i-1)i}{2}+j}(\mu^I_k) = M_N(\psi_i(\mu^I_k))M_N(\psi_{n_\psi-j+1}(\mu^I_k)), \qquad i = 1,...,n_\psi, \qquad j = 1,...,i, \tag{5.32}$$

The data equation (5.26) can be factorised in a similar manner:

$$Y_{k,N,1} = M(\mu^I_k)(\bar{T}_{pc}R_{k,N,1} + \bar{V}_{k,N,1}), \tag{5.33}$$

Here, $\bar{T}_{pc} \in \mathbb{R}^{\frac{n_\psi(n_\psi+1)n_y}{2} N \times n_y N}$, the constant part of the plant time controller matrix, is partitioned in the following manner:

$$\bar{T}^T_{pc} = \begin{bmatrix}(T^{[0]}_{u,N}T^{[0]}_{c,N})^T & \bar{\mathcal{M}}^T & \bar{\mathcal{P}}^T\end{bmatrix}. \tag{5.34}$$

Here, the appropriately dimensioned matrix $\bar{\mathcal{M}}$ is composed of the block rows:

$$\bar{\mathcal{M}}_i = T^{[0]}_{u,N}T^{[i]}_{c,N} + T^{[0]}_{c,N}T^{[i]}_{u,N}, \qquad i = 1,...,n_\psi, \tag{5.35}$$

while the matrix $\bar{\mathcal{P}}$ is composed of the block rows:

$$\bar{\mathcal{P}}_{i n_\psi - \frac{(i-1)i}{2}+j} = \begin{cases} T^{[i]}_{u,N}T^{[n_\psi-j+1]}_{c,N} + T^{[i]}_{u,N}T^{[n_\psi-j+1]}_{c,N}, & \text{for } i = 1,...,n_\psi, j = 1,...,i, i \neq j \\ T^{[i]}_{u,N}T^{[n_\psi-j+1]}_{c,N}, & \text{for } i = 1,...,n_\psi, j = 1,...,i, i = j \end{cases} \tag{5.36}$$

Finally, the scheduling-dependent disturbance $\bar{V}^I_{k,N,1} \in \mathbb{R}^{\frac{n_\psi(n_\psi+1)n_y}{2} N}$ is decomposed such that:

$$\bar{V}^I_{k,N,1} = \begin{bmatrix}(V^{[0],I}_{k,N,1})^T & \cdots & (V^{[n_\psi],I}_{k,N,1})^T & 0 & \cdots & 0\end{bmatrix}^T. \tag{5.37}$$

At this point, in order to be able to estimate the output gradient $\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}}$, a set of gradient experiments is necessary.

5.3.3. EXPERIMENT SET II

As in IFT-LTI, the output gradient is determined by using a specific auxiliary input signal, obtained by filtering the reference through the controller gradient:

$$Q^{II}_{k,N,1} = \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}} R_{k,N,1}. \tag{5.38}$$

Herewith, the data equation becomes:

$$Y^{II}_{k,N,1} = T_{u,N}(\mu^{II}_k) T_{c,N}(\mu^{II}_k) R_{k,N,1} - T_{u,N}(\mu^{II}_k) \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}} R_{k,N,1} + V^{II}_{k,N,1}. \tag{5.39}$$

The same factorisation as the one used in the previous section is used here to separate the known, time-varying effect of the scheduling variable μ_k from the unknown system matrices:

$$Y^{II}_{k,N,1} = M(\mu^{II}_k)(\bar{T}_{pc} R_{k,N,1} - \bar{T}_{u,N} \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}} R_{k,N,1} + \bar{V}^{II}_{k,N,1}). \tag{5.40}$$

In order to eliminate the perturbing effect of the scheduling variable on the gradient estimate, this equation could in principle be pre-multiplied by the left-inverse of $M(\mu^{II}_k)$. However, with a single experiment, the system is underdetermined, and the required left-inverse does not exist. Hence, a sufficient number of experiments with the same auxiliary input are required to be conducted in such a manner that the matrix $M(\mu^{II}_k)$ reaches full row rank (and is well-conditioned). For each such experiment $\hat{j}$,

$$Y^{II,\hat{j}}_{k,N,1} = M(\mu^{II,\hat{j}}_k)(\bar{T}_{pc} R_{k,N,1} - \bar{T}_{u,N} \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}} R_{k,N,1} + \bar{V}^{II,\hat{j}}_{k,N,1}). \tag{5.41}$$

In order to eliminate the influence of the disturbance quantity, the estimate of a differenced output is taken in the following manner such that the biasing influence of the disturbance vanishes ergodically:

$$\Delta Y^{II,\hat{j}}_{k,N,1} = \text{est}\{2Y^{II,\hat{j}}_{k,N,1} - Y^{II,1}_{k,N,1}\}, \qquad \hat{j} > 1, \tag{5.42}$$

$$\Delta Y^{II,\hat{j}}_{k,N,1} = (2M(\mu^{II,\hat{j}}_k) - M(\mu^{II,1}_k))(\bar{T}_{pc} R_{k,N,1} - \bar{T}_{u,N} \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}} R_{k,N,1}). \tag{5.43}$$

When an adequate number of experiments n_{II} has been conducted, the output data from each experiment is stacked into $\Delta Y_{k,n_{II}N,1}$ such that the gradient can be obtained using least squares:

$$\Delta Y_{k,n_{II}N,1} = \begin{bmatrix} \Delta Y^{II,1}_{k,N,1} \\ \Delta Y^{II,2}_{k,N,1} \\ \vdots \\ \Delta Y^{II,n_{II}}_{k,N,1} \end{bmatrix}. \tag{5.44}$$

Lifting the equation (5.41) over the horizon n_{II} to obtain an overdetermined least squares problem,

$$\Delta Y_{k,n_{II}N,1} = \bar{M}(\mu_k^{II})(\bar{T}_{pc}R_{k,N,1} - \bar{T}_{u,N}\frac{\partial T_{c,N}(\mu_k^I)}{\partial \rho_{j_\rho}}R_{k,N,1}), \tag{5.45}$$

where $\bar{M}(\mu_k^{II}) \in \mathbb{R}^{n_{II}n_yN \times \frac{n_\psi(n_\psi+1)n_y}{2}N}$ is a block diagonal matrix with the terms $(2M(\mu_k^{II,\hat{j}}) - M(\mu_k^{II,1}))$ along its block diagonal. The least squares solution to the problem stated above is given as:

$$\bar{M}(\mu_k^{II})^\dagger \Delta Y_{k,n_{II}N,1} = \bar{T}_{pc}R_{k,N,1} - \bar{T}_{u,N}\frac{\partial T_{c,N}(\mu_k^I)}{\partial \rho_{j_\rho}}R_{k,N,1}. \tag{5.46}$$

It can be directly observed that the required output gradient $\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}}$ can be obtained from the second term in this expression. In order to eliminate the effect of the first term, a set of reference experiments is required.

5.3.4. EXPERIMENT SET III

In a similar manner, in order to remove the influence of the direct reference signal on the output gradient estimate, a set of reference experiments is required. For these experiments, the auxiliary input signal is set to zero, $Q^{III}_{k,N,1} = 0$. With this, the first term in equation (5.46) can be expressed as:

$$\bar{M}(\mu_k^{III})^\dagger \Delta Y_{k,n_{III}N,1} = \bar{T}_{pc}R_{k,N,1}. \tag{5.47}$$

Here, the term $\Delta Y_{k,n_{III}N,1}$ is the stacked output over an n_{III} number of experiments, conducted such that $\bar{M}(\mu_k^{III})$ has full row rank and is well-conditioned. Substituting this estimate of the direct influence of the reference into equation (5.46), we get:

$$\bar{M}(\mu_k^{III})^\dagger \Delta Y_{k,n_{III}N,1} - \bar{M}(\mu_k^{II})^\dagger \Delta Y_{k,n_{II}N,1} = \bar{T}_{pc}\frac{\partial T_{c,N}(\mu_k^I)}{\partial \rho_{j_\rho}}R_{k,N,1}. \tag{5.48}$$

At this point, it is possible to obtain an estimate of the output gradient with respect to the controller parameters, from equation (5.28) as:

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = M(\mu_k^I)(\bar{M}(\mu_k^{III})^\dagger \Delta Y_{k,n_{III}N,1} - \bar{M}(\mu_k^{II})^\dagger \Delta Y_{k,n_{II}N,1}). \tag{5.49}$$

At this point, the input gradient can be estimated, and the performance gradient with respect to the controller parameters can be synthesised, as in equation (5.7), following the procedure in Section 5.2.2. With the gradient estimate available, as before, a gradient-based optimisation method can be used to iterate to the optimal parameters that satisfy the user-defined cost function.

Thus, in accordance with the steps laid out in this section, it is possible to use IFT for tuning an LPV feedforward controller for a plant that is LPV in its output matrices. Such

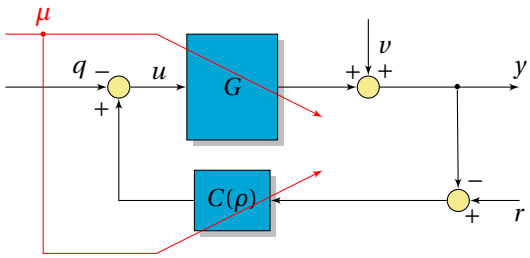


Figure 5.2: Iterative Feedback Tuning experiment layout for an LPV controller in closed loop with an LPV plant.

a method may be considered applicable for load control of wind turbines; this application will be discussed in the second part of the book. Here, it should be noted that the key element is the LPV factorisation of the system matrices: the inner dimension scales quadratically with n_ψ and linearly with N. The number of IFT experiments required for the inverse $\bar{M}(\mu_k)^\dagger$ to exist, and hence, for this method to be applicable, is directly related to this inner dimension.

For the case where the underlying system is fully LPV, and a feedback controller is to be tuned in closed loop, the size of the inner dimension of this factorisation increases exponentially with window size N, this is the well-known curse of dimensionality encountered in LPV identification and control theory. The extension of IFT-LPV for such a case will be dealt with in the next section.

5.4. IFT FOR FEEDBACK CONTROL OF LPV SYSTEMS

In this section, the parameters of a fixed-structure LPV controller is tuned using IFT for the case of a plant with LPV state-transition properties. Once more, the exogenous scheduling sequence μ is assumed perfectly measurable but uncontrollable, and persistently exciting. The experimental layout is depicted in Fig. 5.2, and is similar to the layout in the previous sections. It should, however, be noted that the parameterised controller acts on the output error, e_k, the difference between the reference r_k and the output y_k, thus $e_k = r_k - y_k$. As before, the auxiliary input q_k is added to the control input in order to obtain performance gradients using IFT experiments. First, the notation for this closed-loop setting is defined. Then, the closed-loop LPV factorisation is described. Based on this new factorisation, the three sets of IFT experiments are then updated, such that the performance gradient with respect to the controller parameters can be obtained, which is used for tuning the controller as in the previous sections.

5.4.1. NOTATION

As distinct from the previous section, it is assumed here that the system is LPV in its state-transition properties and admits a realisation of the form:

$$x_{k+1} = A_k x_k + B_k u_k \tag{5.50}$$

$$y_k = C x_k + v_k. \tag{5.51}$$

The output matrix C has been taken to be constant, the direct feedthrough matrix D is taken to be zero. The case for fully LPV non-zero output matrices follows as a direct extension of the concept described in this section. The system input u_k, state x_k, output y_k and the perturbing disturbance v_k have already been defined in the previous sections. In this section, the matrices A_k and B_k are considered to depend on the scheduling sequence μ_k in the following manner:

$$A_k = A^{[0]} + \sum_{i=1}^{n_\psi} \psi_i(\mu_k) A^{[i]}, \qquad B_k = B^{[0]} + \sum_{i=1}^{n_\psi} \psi_i(\mu_k) B^{[i]}. \tag{5.52}$$

The system is considered to be in closed loop with a nominally stabilising feedback controller, acting on the output error e_k, which is parameterised by $\rho \in \mathbb{R}^{n_\rho}$:

$$x_{c,k+1} = A_{c,k}(\rho) x_{c,k} + B_{c,k}(\rho) e_k \tag{5.53}$$

$$u_k = C_{c,k}(\rho) x_{c,k} + D_{c,k}(\rho) e_k - q_k. \tag{5.54}$$

5

As before, $x_{c,k} \in {}^{n_c}$ is the state of the fixed-order controller, and the controller matrices $A_{c,k}(\rho)$, $B_{c,k}(\rho)$, $C_{c,k}(\rho)$ and $D_{c,k}(\rho)$ are parameterised LPV matrices of the appropriate dimensions, defined in detail in Section 5.2.1. As before, q_k is the auxiliary input that is used in the IFT experiments for the purpose of controller tuning. The objective of IFT, as before, is to tune the controller such that a user-defined performance criterion is minimised. The closed-loop controller criterion can be defined in terms of the stacked output error $E_{k,N,1}$ and control effort, $U_{k,N,1}$, where the signals are stacked in a manner described in the Section 5.2.1. The closed-loop optimisation criterion taken here is quadratic:

$$J = \frac{1}{2N}(E_{k,N,1}^T E_{k,N,1} + \lambda_J U_{k,N,1}^T U_{k,N,1}). \tag{5.55}$$

As before, the prediction horizon is N. For the case of feedback control, it has to be ensured that this value is large enough to ensure stability of the tuned closed-loop system. Since IFT-LPV is designed to use a gradient-based optimisation algorithm to minimise the performance cost by optimising controller parameters, the objective of the IFT experiments is to estimate the gradient of the performance cost with respect to ρ. Thus, this gradient has to be estimated:

$$\frac{\partial J}{\partial \rho} = \frac{1}{N}(\frac{\partial E_{k,N,1}^T}{\partial \rho} E_{k,N,1} + \lambda_J \frac{\partial U_{k,N,1}^T}{\partial \rho} U_{k,N,1}). \tag{5.56}$$

As before, the IFT experiments are designed to be able to obtain estimates of $\frac{\partial E_{k,N,1}^T}{\partial \rho}$ and $\frac{\partial U_{k,N,1}^T}{\partial \rho}$. The data equations can be expressed in terms of closed-loop lifted system matrices. Since the LPV realisation is different in the current section from that in the

previous section, the system matrix $T_{u,N,k}$ is redefined as:

$$T_{u,N,k} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CB_k & 0 & \cdots & 0 \\ CA_{k+1}B_k & CB_{k+1} & \cdots & 0 \\ CA_{k+2}A_{k+1}B_k & CA_{k+2}B_{k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA_{k+N-1}\cdots A_{k+1}B_k & CA_{k+N-1}\cdots A_{k+2}B_{k+1} & \cdots & 0 \end{bmatrix}. \tag{5.57}$$

The lifted system matrix of the feedback controller, $T_{c,N,k}$ can be defined in the same manner as done for the feedforward controller in the previous section. For this closed-loop system, it is also possible to construct the closed-loop lifted system matrices. As the lifted system matrices are lower-triangular, from Oomen et al. (2009), it is possible to define the sensitivity matrix $T_{s,k}$, the complementary sensitivity matrix $T_{cs,k}$ and the process sensitivity matrix $T_{ps,k}$ as:

$$T_{s,k} = (I_{n_y N} + T_{u,N,k}T_{c,N,k})^{-1}, \tag{5.58}$$

$$T_{cs,k} = (I_{n_y N} + T_{u,N,k}T_{c,N,k})^{-1}T_{u,N,k}T_{c,N,k}, \tag{5.59}$$

$$T_{ps,k} = (I_{n_y N} + T_{u,N,k}T_{c,N,k})^{-1}T_{u,N,k}. \tag{5.60}$$

Now, if the underlying plant and controller state-space matrices are LPV, then the closed-loop system matrices are also LPV. Let the tuple $(A_{cc,k}, B_{cc,k}, C_{cc,k}, D_{cc,k})$ represent all three closed-loop systems described in the above equations. Then, according to Chen and Francis (1995), the LPV expansion of each of the closed-loop system matrices is given by:

$$A_{cc,k} = A_{cc}^{[0]} + \sum_{i=1}^{n_{\bar{\psi}}} \bar{\psi}_i(\mu_k) A_{cc}^{[i]}. \tag{5.61}$$

Here, $\bar{\psi}_i(\mu_k), i = 1, ..., n_{\bar{\psi}}$ are inner products of the original set of basis functions $\psi_i(\mu_k)$. The total number of new basis functions is related to the original number of basis functions as $n_{\bar{\psi}} = 2^{n_\psi}$. Similar LPV expansions can be defined for $B_{cc,k}$, $C_{cc,k}$ and $D_{cc,k}$, for all three closed-loop systems.

Thus, the closed-loop system matrices can be described as affine functions of the new basis functions scheduled on μ_k. As in the previous section, these closed-loop matrices can also be expressed as a product of a known time-varying factor that is purely a function of the scheduling variable, and an unknown time-invariant factor.

5.4.2. LPV FACTORISATION

The generic closed-loop lifted system matrix $\mathcal{T}$ can be factorised as follows:

$$\mathcal{T} = \underbrace{\begin{bmatrix} M_{cc,1}(\mu_k) & M_{cc,2}(\mu_k) & \cdots & M_{cc,p}(\mu_k) \end{bmatrix}}_{M_{cc}(\mu_k)} \underbrace{\begin{bmatrix} \mathcal{H} & 0 & \cdots & 0 \\ 0 & \mathcal{H} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{H} \end{bmatrix}}_{\bar{\mathcal{T}}}. \tag{5.62}$$

It should be noted that the term $\bar{\mathcal{T}}$ is time-invariant, but still a function of ρ. For the three closed-loop system matrices, the following factorisations then hold:

$$T_{s,k} = M_{cc}(\mu_k)\bar{T}_s, \qquad T_{cs,k} = M_{cc}(\mu_k)\bar{T}_{cs}, \qquad T_{ps,k} = M_{cc}(\mu_k)\bar{T}_{ps}. \tag{5.63}$$

The components of the factorisation are defined as follows, the term $\mathcal{H} \in \mathbb{R}^{n_z \times n_u}$ is partitioned into block rows such that:

$$\mathcal{H}^T = \begin{bmatrix} \mathcal{F}_1^T & \cdots & \mathcal{F}_p^T \end{bmatrix}. \tag{5.64}$$

The block rows $\mathcal{F}_{j_f} \in \mathbb{R}^{n_y n_{\bar{\psi}}^{j_f} \times n_u}$ are defined as below:

$$\mathcal{F}_1^T = \begin{bmatrix} (D_{cc}^{[0]})^T & \cdots & (D_{cc}^{[n_{\bar{\psi}}]})^T \end{bmatrix}. \tag{5.65}$$

The z^{th} row of $\mathcal{F}_{j_f}$ is given by:

$$\mathcal{F}_{j_f}^{(z)} = C_{cc}^{[i_1^z]} A_{cc}^{[i_2^z]} \cdots A_{cc}^{[i_{j_f-1}^z]} B_{cc}^{[i_{j_f}^z]}, \qquad i_1^z, \cdots, i_{j_f}^z \in \{1, \cdots, n_{\bar{\psi}}\}. \tag{5.66}$$

5

The ordering of the indices i_*^z in the above expression are such that $\sigma_{z+1} > \sigma_z$ where:

$$\sigma_z = \begin{bmatrix} i_1^z & \cdots & i_{j_f}^z \end{bmatrix} \begin{bmatrix} n_{\bar{\psi}}^{j_f-1} \\ \vdots \\ n_{\bar{\psi}} \\ 1 \end{bmatrix}. \tag{5.67}$$

With this definition of $\mathcal{H}$, the unknown time-invariant factors $\bar{T}_s$, $\bar{T}_{cs}$ and $\bar{T}_{ps}$ are fully defined. The measurable time-varying factor $M_{cc}(\mu_k)$ is defined in terms of its block columns in the following manner:

$$M_{cc,1}(\mu_k) = \begin{bmatrix} \bar{\psi}(\mu_k)^T & 0 & \cdots & 0 \\ 0 & \bar{\psi}(\mu_k)^T \otimes \bar{\psi}(\mu_{k+1})^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{\psi}(\mu_k)^T \otimes \cdots \otimes \bar{\psi}(\mu_{k+N-1})^T \end{bmatrix} \otimes I_{n_y} \tag{5.68}$$

$$M_{cc,2}(\mu_k) = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \bar{\psi}(\mu_k)^T & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{\psi}(\mu_k)^T \otimes \cdots \otimes \bar{\psi}(\mu_{k+N-2})^T & 0 \end{bmatrix} \otimes I_{n_y} \tag{5.69}$$

and so on. Herewith, it is possible to fully define the scheduling-dependent factor $M_{cc}(\mu_k)$ and the unknown invariant factors $\bar{\mathcal{T}}$ of the closed-loop system matrices. It needs to be observed that the inner dimension of this factorisation is $n_{cc} = N\sum_{i=1}^{N} n_y n_{\bar{\psi}}^i$. Thus, the size of the factorisation grows exponentially with the size of the lifting window N. This is the curse of dimensionality often also seen in global LPV identification methods; it will have repercussions on the number of IFT experiments required for gradient estimation, described in the next section.

5.4.3. IFT EXPERIMENTS

EXPERIMENT I

As before, the reference experiment is conducted with auxiliary input $Q^I_{k,N,1} = 0$. The data equation is constructed from the input-output relations:

$$Y^I_{k,N,1} = T_{u,N}(\mu^I_k)U^I_{k,N,1} + V^I_{k,N,1}, \tag{5.70}$$

$$U^I_{k,N,1} = T_{c,N}(\mu^I_k)(R_{k,N,1} - Y^I_{k,N,1}). \tag{5.71}$$

Combining the equations,

$$Y_{k,N,1} = T_{u,N}(\mu^I_k)T_{c,N}(\mu^I_k)(R_{k,N,1} - Y^I_{k,N,1}) + V^I_{k,N,1}. \tag{5.72}$$

Since the performance gradient $\frac{\partial J}{\partial \rho}$ is a function of the output gradient $\frac{\partial Y_{k,N,1}}{\partial \rho}$, the above equation is differentiated with respect to each controller parameter as below:

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = T_{u,N}(\mu^I_k)\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}) - T_{u,N}(\mu^I_k)T_{c,N}(\mu^I_k)\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}}, \tag{5.73}$$

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = (I_{n_y N} + T_{u,N}(\mu^I_k)T_{c,N}(\mu^I_k))^{-1}T_{u,N}(\mu^I_k)\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}). \tag{5.74}$$

In terms of the closed-loop matrices, the output and its gradient can be written as:

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = T_{ps}(\mu^I_k)\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}), \tag{5.75}$$

$$Y^I_{k,N,1} = T_{cs}(\mu^I_k)R_{k,N,1} + T_s(\mu^I_k)V^I_{k,N,1}. \tag{5.76}$$

As in the previous section, the LPV factorisation is used to separate the known, time-varying effect of the scheduling sequence from the unknown time-invariant system dynamics:

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = M_{cc}(\mu^I_k)\bar{T}_{ps}\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}), \tag{5.77}$$

$$Y^I_{k,N,1} = M_{cc}(\mu^I_k)(\bar{T}_{cs}R_{k,N,1} + \bar{T}_s V^I_{k,N,1}). \tag{5.78}$$

Next, the gradient experiments are performed to estimate the output gradient with respect to the controller parameters.

EXPERIMENT SET II

As in the previous sections, the auxiliary input, $Q^{II}_{k,N,1}$ is set to be the same as the input to the controller from the reference experiment, filtered through a gradient of the controller with respect to the controller parameter ρ_{j_ρ}:

$$Q^{II}_{k,N,1} = \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}). \tag{5.79}$$

In this case, the data equation becomes:

$$Y^{II}_{k,N,1} = T_{cs}(\mu^{II}_k) - T_{ps}(\mu^{II}_k)\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}) + T_s(\mu^{II}_k)V^{II}_{k,N,1}. \tag{5.80}$$

Using the factorisation once more to separate out the effect of the scheduling sequence,

$$Y^{II}_{k,N,1} = M_{cc}(\mu^{II}_k)(\bar{T}_{cs} - \bar{T}_{ps}\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}) + \bar{T}_s V^{II}_{k,N,1}). \tag{5.81}$$

The middle term on the right hand side can be used directly to construct an estimate of the output gradient; to compensate for the effect of $M_{cc}(\mu^{II}_k)$, it would be necessary to left-multiply the equation by the left-inverse of this matrix. Once again, since the matrix $M_{cc}(\mu^{II}_k)$ is deficient in terms of row rank, such a left-inverse does not exist, and multiple gradient experiments are required to obtain an overdetermined system. For each experiment $\hat{j}$, the above equation holds:

$$Y^{II,\hat{j}}_{k,N,1} = M_{cc}(\mu^{II,\hat{j}}_k)\left(\bar{T}_{cs} - \bar{T}_{ps}\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}) + \bar{T}_s V^{II,\hat{j}}_{k,N,1}\right). \tag{5.82}$$

For the case where the scheduling sequence varies randomly, the effect of the disturbance can be recast as:

$$M_{cc}(\mu^{II,\hat{j}}_k)\bar{T}_s V^{II,\hat{j}}_{k,N,1} = \bar{\mathcal{T}}_s W^{II,\hat{j}}_{k,N,1}. \tag{5.83}$$

The new stacked disturbance vector $W^{\hat{j}}_{k,N,1} \in \mathbb{R}^{n_{cc}n_y}$ is given by:

$$W^{II,\hat{j}}_{k,N,1} = \sum_{i=1}^{N} \nu^{II,\hat{j}}_{k+i-1} \otimes M_{cc,i}(\mu^{II,\hat{j}}_k). \tag{5.84}$$

This new signal is considered to be zero-mean, and its biasing effect on the output gradient estimate should decay over time. Further, $\bar{\mathcal{T}}_s$ is time-invariant. Stacking equation (5.82) over an adequate number of experiments n_{II},

$$Y_{k,n_{II}N,1} = \bar{M}_{cc}(\mu^{II}_k)\left(\bar{T}_{cs}R_{k,N,1} - \bar{T}_{ps}\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1})\right) + \bar{\bar{\mathcal{T}}}_s W^{II}_{k,n_{II}N,1}. \tag{5.85}$$

Here, $\bar{M}_{cc}(\mu^{II}_k)$ and $\bar{\bar{\mathcal{T}}}_s$ are square block diagonal matrices of block size n_{II}, with the terms $M_{cc}(\mu^{II}_k)$ and $\bar{\mathcal{T}}_s$ on their block diagonals, respectively. In order to get an estimate of the output gradient, the equation is pre-multiplied by the left-inverse of $\bar{M}_{cc}(\mu^{II}_k)$, and the estimate of the disturbance is set to zero, such that:

$$\bar{M}_{cc}(\mu^{II}_k)^\dagger Y_{k,n_{II}N,1} = \bar{T}_{cs}R_{k,N,1} - \bar{T}_{ps}\frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}}(R_{k,N,1} - Y^I_{k,N,1}). \tag{5.86}$$

The second term in this expression can be used to estimate the output gradient. In order to remove the influence of the first term, which describes the direct influence of the reference on the output, a third set of experiments is required to be conducted.

EXPERIMENT SET III

In this experiment set, the auxiliary input $Q^{III}_{k,N,1}$ is once again set to zero, such that the first term in equation (5.86) can be estimated. Following the same set of steps as in the previous section, this term can be estimated as:

$$\bar{M}_{cc}(\mu^{III}_k)^\dagger Y_{k,n_{III}N,1} = \bar{T}_{cs} R_{k,N,1}. \tag{5.87}$$

Subtracting (5.86) from the above equation,

$$\bar{M}_{cc}(\mu^{III}_k)^\dagger Y_{k,n_{III}N,1} - \bar{M}_{cc}(\mu^{II}_k)^\dagger Y_{k,n_{II}N,1} = \bar{T}_{ps} \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}} (R_{k,N,1} - Y^I_{k,N,1}). \tag{5.88}$$

Referring to equation (5.77), the output gradient can be directly obtained as:

$$\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}} = M_{cc}(\mu^I_k)(\bar{M}_{cc}(\mu^{III}_k)^\dagger Y_{k,n_{III}N,1} - \bar{M}_{cc}(\mu^{II}_k)^\dagger Y_{k,n_{II}N,1}). \tag{5.89}$$

Since the performance criterion is expressed in terms of the output error gradient $\frac{\partial E^I_{k,N,1}}{\partial \rho_{j_\rho}}$ and the input gradient $\frac{\partial U^I_{k,N,1}}{\partial \rho_{j_\rho}}$, these quantities can be derived in a straightforward manner from the value of the output gradient $\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}}$ as:

$$\frac{\partial E^I_{k,N,1}}{\partial \rho_{j_\rho}} = -\frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}}, \tag{5.90}$$

$$\frac{\partial U^I_{k,N,1}}{\partial \rho_{j_\rho}} = \frac{\partial T_{c,N}(\mu^I_k)}{\partial \rho_{j_\rho}} (R_{k,N,1} - Y^I_{k,N,1}) - T_{c,N}(\mu^I_k) \frac{\partial Y^I_{k,N,1}}{\partial \rho_{j_\rho}}. \tag{5.91}$$

Thus, finally, the required performance gradient with respect to the controller parameters can be obtained as:

$$\frac{\partial J}{\partial \rho_{j_\rho}} = \frac{1}{N}\left(\left(\frac{\partial E^I_{k,N,1}}{\partial \rho_{j_\rho}}\right)^T E_{k,N,1} + \lambda_J \left(\frac{\partial U^I_{k,N,1}}{\partial \rho_{j_\rho}}\right)^T \frac{\partial U^I_{k,N,1}}{\partial \rho_{j_\rho}}\right). \tag{5.92}$$

With this gradient, as in standard IFT, a gradient-descent method can be used to optimise each controller parameter ρ_{j_ρ} of the fully LPV controller, in closed loop with a fully LPV plant, such that the performance criterion is (locally) optimised. A case study is taken up next to describe the implementation of such a strategy.

5.4.4. CASE STUDY: IFT-LPV FOR A SWITCHED SYSTEM

The main requirement for performing IFT-LPV is the ability to factorise the lifted system matrices into a known, time-varying component, and an unknown, time-invariant component. The previous section showed that for IFT for full LPV feedback control of LPV plants, the inner dimension of the factorisation increases exponentially with the size of the lifting window. In order to uniquely estimate the effect of these unknown system

dynamics on the performance gradient in a least-squares sense, it becomes necessary to perform an exponentially increasing number of IFT experiments, which may prove intractable for real-time implementation of the algorithm. As such, it is advisable to exploit the structure of the lifted system matrices as much as possible to reduce this curse of dimensionality.

In this section, an example of a switched system is considered, which simplifies the factorisation. The use of full IFT-LPV for tuning a feedback controller for this special case is then explored.

LPV FACTORISATION

Let us consider that the system switches between two realisations, described respectively using the tuples $(A^{[0]}, B^{[0]}, C, 0)$ and $(A^{[0]} + A^{[1]}, B^{[0]} + B^{[1]}, C, 0)$. The system switches between these two models at an arbitrary instant of time, denoted by τ. That is to say, the scheduling variable $\mu_k \in \mathbb{R}$ can be described as:

$$\mu_k = \begin{cases} 0, & k \leq \tau \\ 1, & k > \tau. \end{cases} \tag{5.93}$$

The term τ is thus an integer and the switching occurs within the prediction horizon N, thus $1 \leq \tau \leq N$. Now, with an LPV controller in closed-loop with the system, the closed-loop system matrices before the time of switching become $(A_{cc}^{[0]}, B_{cc}^{[0]}, C_{cc}, D_{cc})$. After time instant τ, the state transition matrix changes to $A'_{cc} = A_{cc}^{[0]} + A_{cc}^{[1]}$ with a similar expression for B'_{cc}. The closed-loop system matrix lifted over the horizon N can then be described as:

$$\mathcal{T}(\tau) = \begin{bmatrix} D_{cc} & 0 & \cdots & 0 \\ C_{cc}B_{cc}^{[0]} & D_{cc} & \cdots & 0 \\ C_{cc}A_{cc}^{[0]}B_{cc}^{[0]} & C_{cc}B_{cc}^{[0]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{cc}(A_{cc}^{[0]})^{\tau-1}B_{cc}^{[0]} & C_{cc}(A_{cc}^{[0]})^{\tau-2}B_{cc}^{[0]} & \cdots & 0 \\ C_{cc}(A'_{cc})(A_{cc}^{[0]})^{\tau-1}B_{cc}^{[0]} & C_{cc}(A'_{cc})(A_{cc}^{[0]})^{\tau-2}B_{cc}^{[0]} & \cdots & 0 \\ C_{cc}(A'_{cc})^2(A_{cc}^{[0]})^{\tau-1}B_{cc}^{[0]} & C_{cc}(A'_{cc})^2(A_{cc}^{[0]})^{\tau-2}B_{cc}^{[0]} & \cdots & 0 \\ C_{cc}(A'_{cc})^{N-\tau}(A_{cc}^{[0]})^{\tau-1}B_{cc}^{[0]} & C_{cc}(A'_{cc})^{N-\tau}(A_{cc}^{[0]})^{\tau-2}B_{cc}^{[0]} & \cdots & D_{cc} \end{bmatrix}. \tag{5.94}$$

This matrix can be approximated as the sum of the lifted system matrix of the original system before the switching instant, and a weighted lifted system matrix of the updated system parameters, as follows:

$$\mathcal{T}(\tau) \approx \underbrace{\begin{bmatrix} I_N & Q_N(\tau) \end{bmatrix}}_{M_{ss}(\tau)} \underbrace{\begin{bmatrix} \mathcal{T}^{[0]} \\ \mathcal{T}^{[1]} \end{bmatrix}}_{\bar{\mathcal{T}}}, \tag{5.95}$$

$$Q_N(\tau) = \mathrm{diag}(\underbrace{0, 0, \cdots, 0}_{\tau \text{ terms}} \underbrace{1, 1, \cdots, 1}_{N-\tau \text{ terms}}). \tag{5.96}$$

the time-invariant, unknown system matrices are given as:

$$\mathcal{T}^{[0]} = \begin{bmatrix} D_{cc} & 0 & \cdots & 0 \\ C_{cc}B_{cc}^{[0]} & D_{cc} & \cdots & 0 \\ C_{cc}A_{cc}^{[0]}B_{cc}^{[0]} & C_{cc}B_{cc}^{[0]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{cc}(A_{cc}^{[0]})^{N-2}B_{cc}^{[0]} & C_{cc}(A_{cc}^{[0]})^{N-3}B_{cc}^{[0]} & \cdots & 0 \end{bmatrix}, \tag{5.97}$$

$$\mathcal{T}^{[1]} = \begin{bmatrix} D_{cc} & 0 & \cdots & 0 \\ C_{cc}B_{cc}^{[1]} & D_{cc} & \cdots & 0 \\ C_{cc}A_{cc}^{[1]}B_{cc}^{[0]} & C_{cc}B_{cc}^{[1]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{cc}(A_{cc}^{[1]})^{N-\bar{\tau}}(A_{cc}^{[0]})^{\bar{\tau}-1}B_{cc}^{[0]} & C_{cc}(A_{cc}^{[1]})^{N-\bar{\tau}-1}(A_{cc}^{[0]})^{\bar{\tau}-1}B_{cc}^{[0]} & \cdots & 0 \end{bmatrix}. \tag{5.98}$$

Here, $\bar{\tau}$ is the average value of τ expected over iterations. Thus, this special low-dimensional factorisation yields a component $M_{ss}(\tau)$ that varies per iteration, and an unknown constant component $\bar{\mathcal{T}}$. This factorisation can be used for the closed-loop system matrices in the IFT experiments as described in the previous section, in order to tune the values of a fixed-structure LPV controller for an LPV plant.

CASE STUDY RESULTS

This section uses the LPV model of a morphing aerofoil as a test case. The model is described in Lee and Singh (2007), and it is a fourth-order model where the dynamics change based on the ambient wind speed V_k (which here forms the scheduling μ_k). The controller is to be designed to track aerofoil motion, by commanding the deflection of a small trailing edge flap.

The system is recast as a switched system by considering that the operating wind speed of the aerofoil is 6 m/s, but there occurs a sudden change in wind speed to a value of 8 m/s, at an arbitrary instant of time within the time horizon, taken to be $N = 100$ samples. For this special case, the LPV factorisation (5.95) is then applicable. The reference trajectory to be followed consists of a positive step at the initial instant of time, and a negative step half-way through the time series.

A fixed-structure LPV PI controller was taken to be in closed-loop with the system, and its time-varying gain values $K_{p,k}$ and $K_{i,k}$ were assigned such that the closed-loop tracking performance was arbitrarily poor. The gain values were taken to be affine functions of the scheduling variable (wind speed) such that:

$$K_p = K_p^{[0]} + V_k K_p^{[1]}, \qquad K_p = K_i^{[0]} + V_k K_i^{[1]}. \tag{5.99}$$

Thus, overall, an $n_\rho = 4$ number of parameters were tuned for this fully LPV fixed-structure controller to optimise the closed-loop performance of the LPV plant. It can be seen in Fig. 5.3 that, the algorithm is able to tune the four parameters, even in the presence of noise, which is simulated as an additive zero-mean white measurement noise signal with SNR 20.

The improvement in closed-loop performance can be observed from Fig. 5.4. The initial controller, with arbitrary values of controller parameters, performs poorly in

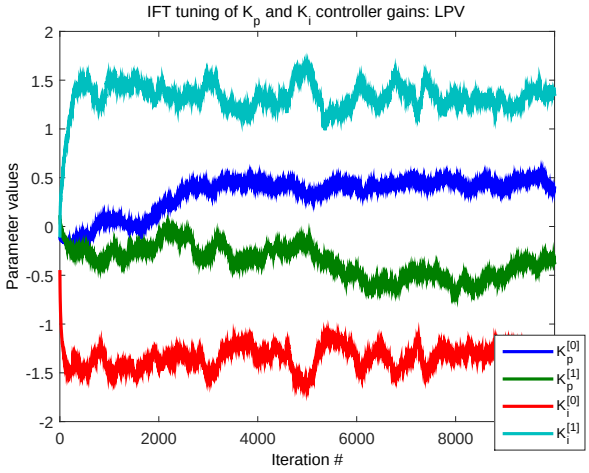


Figure 5.3: IFT tuning of K_p and K_i parameters for LPV system and LPV controller.

5

terms of reference trajectory tracking. On the other hand, IFT-LPV is able to modify the controller gains, purely based on input-output data, to achieve good controller performance.

The performance of an LPV controller is compared, in Fig. 5.5, to that of an LTI controller, both tuned using IFT. It can be seen that the LPV controller is able to adjust to the changes in wind speed dynamics and accordingly change the controller gains such that the performance deviates minimally irrespective of the actual scheduling sequence. On the other hand, the LTI controller is unable to compensate for the changes in wind speed, and shows poorer performance under off-design conditions.

While a switched-system setup has been explored here, the next section explores a fully LPV factorisation modified for practical tractability.

5.4.5. CASE STUDY: EARLY TERMINATION OF IFT EXPERIMENTS

One of the drawbacks of the IFT-LPV approach developed in this section is that, for the unique estimation of the performance gradient, the number of IFT experiments required increases exponentially with the size of the lifting window. At this point, a heuristic is defined that forms a measure of the approximation error in the estimate of the gradient, and set a limit on the number of IFT experiments required. Herefor, two quantities related to each set of IFT experiments (I, II and III) are defined below:

Definition 1. *The characteristic vector $\eta(\mu_k^{*,\hat{j}})$ is defined as the vectorised form of all unique non-zero terms in the corresponding scheduling sequence factor $M_{cc}(\mu_k^{*,\hat{j}})$, where $\hat{j} \in \{1, \cdots, n_*\}$ is the experiment number, and $*$ corresponds to (I, II or III).*

Example. For instance, for the case where the lifting window has size $N = 5$, and a single basis function $\psi(\mu_k)$ is scheduled on the variable μ_k, then the characteristic vector $\eta(\mu_k^I)$

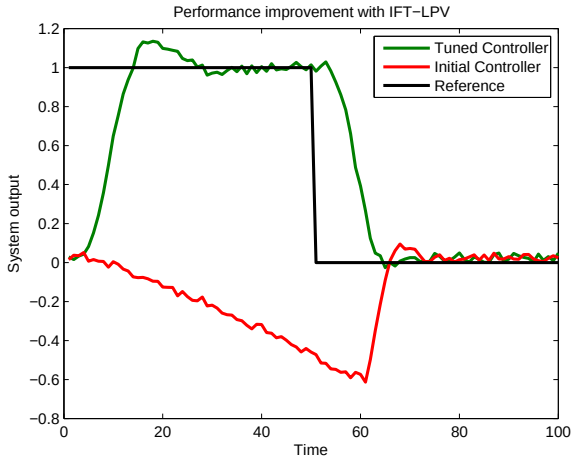


Figure 5.4: Comparison of controller performance before and after data-based controller tuning with IFT.

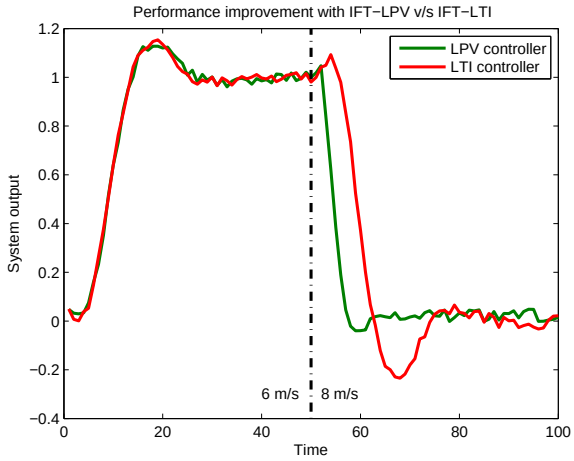


Figure 5.5: Comparison of LTI and LPV controllers.

for the experiment I is given by:

$$\eta(\mu_k^I) = \begin{bmatrix} 1 \\ \psi(\mu_k^I) \\ \psi(\mu_{k+1}^I) \\ \psi(\mu_{k+2}^I) \\ \psi(\mu_{k+3}^I) \\ \psi(\mu_k^I)\psi(\mu_{k+1}^I) \\ \psi(\mu_k^I)\psi(\mu_{k+2}^I) \\ \psi(\mu_k^I)\psi(\mu_{k+3}^I) \\ \psi(\mu_{k+1}^I)\psi(\mu_{k+2}^I) \\ \psi(\mu_{k+1}^I)\psi(\mu_{k+3}^I) \\ \psi(\mu_{k+2}^I)\psi(\mu_{k+3}^I) \\ \psi(\mu_k^I)\psi(\mu_{k+1}^I)\psi(\mu_{k+2}^I) \\ \psi(\mu_k^I)\psi(\mu_{k+1}^I)\psi(\mu_{k+3}^I) \\ \psi(\mu_{k+1}^I)\psi(\mu_{k+2}^I)\psi(\mu_{k+3}^I) \\ \psi(\mu_k^I)\psi(\mu_{k+1}^I)\psi(\mu_{k+2}^I)\psi(\mu_{k+3}^I) \end{bmatrix} \tag{5.100}$$

.

Definition 2. *The residual metric ϵ_m^* for experiment sets II and III is defined as the norm of the projection of the characteristic vector $\eta(\mu_k^I)$ of experiment I on the null space of the matrix $\bar{\eta}_* = \begin{bmatrix} \eta(\mu_k^{*,1}) & \cdots & \eta(\mu_k^{*,n_*}) \end{bmatrix}$, where $*$ stands for II and III.*

If the number of experiments conducted is not sufficient such that the equations (5.86)-(5.87) represent an undetermined system of equations, then it is still possible to generate a unique minimum-norm solution to the least-squares problem. However, such a solution may be arbitrarily far from the true least-squares minimiser required for an unbiased estimate of the performance gradient. The residual metric defined above provides a measure of the distance between the true solution and the minimum-norm solution obtained from a limited number of experiments. By thresholding this metric such that the approximation error in the gradient estimate is within user-defined tolerance limits, the IFT experiments can be terminated early, thereby increasing the feasibility of the online implementation of IFT-LPV.

For example, consider the case where the scheduling trajectory is identical for experiments I and the first experiments of set II and III, which is the case for LTI systems, or systems with periodic scheduling where N is a multiple of the period. The characterstic vector $\eta(\mu_k^I)$ is here identical to the vectors $\eta(\mu_k^{II,1})$ and $\eta(\mu_k^{III,1})$. In this case, the residual metric $\epsilon_m^{II} = \epsilon_m^{III} = 0$ and the IFT experiments can be stopped after the first experiments of set II and set III. This special degenerate case of IFT-LPV is thus equivalent to IFT-LTI.

On the other hand, if $\eta(\mu_k^I)$ is not collinear with $\eta(\mu_k^{II,1})$, but coplanar with $\eta(\mu_k^{II,1})$ and $\eta(\mu_k^{II,2})$, then the number of experiments required in set II is equal to two, at which point the residual metric $\epsilon_m^{II} = 0$. Thus, as the complexity in the variation of the scheduling trajectory grows, the number of IFT experiments required for the accurate estimation of the performance gradient also increases.

The algorithm is laid out in full in Algorithm 2, and the effect of early termination on the optimum-seeking nature of IFT-LPV is described using two case studies.

IFT-LPV: STATE OF THE ART COMPARISON

The case study used by Formentin et al. (2013) is considered in this section. The authors in the reference use a direct data-driven approach for optimising the controller parameters of a simple LPV system, given as below:

$$x_{k+1} = \mu_k x_k + u_k \tag{5.101}$$

$$y_k = x_k. \tag{5.102}$$

The system is connected in closed loop with a fully LPV controller, with a state-space representation as in (5.4). The parameterised controller matrices are given as below:

$$A_{c,k}(\rho) = \begin{bmatrix} \mu_k - \rho_0(\mu_k) & 1 \\ -\rho_0(\mu_k) - \rho_1(\mu_k) & 1 \end{bmatrix}, \qquad B_{c,k} = \begin{bmatrix} \rho_0(\mu_k) \\ \rho_0(\mu_k) + \rho_1(\mu_k) \end{bmatrix}, \tag{5.103}$$

$$C_{c,k} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D_{c,k} = 0. \tag{5.104}$$

Here, the controller parameters are themselves LPV:

$$\rho_0(\mu_k) = \rho_0^{[0]} + \mu_k \rho_0^{[1]}, \qquad \rho_1(\mu_k) = \rho_1^{[0]} + \mu_{k-1} \rho_1^{[1]}. \tag{5.105}$$

The objective of the controller, as per Formentin et al. (2013), is to achieve closed-loop behaviour that can be represented by the plant:

$$x_{cc,k+1} = \begin{bmatrix} -1 & 1 \\ -1 - \mu_k + \mu_{k-1} & 1 \end{bmatrix} x_{cc,k} + \begin{bmatrix} 1 + \mu_k \\ 1 + \mu_k - \mu_{k-1} \end{bmatrix} r_k \tag{5.106}$$

$$y_{M,k} = \begin{bmatrix} 1 & 0 \end{bmatrix} x_{cc,k}. \tag{5.107}$$

Here, $y_{M,k}$ is then the output expected from the closed-loop system, tuned using IFT. It should be noted that the IFT cost function is slightly different in the sense that the output error is redefined as $e_k = y_{M,k} - y_k$, and the control effort weight $\lambda_J = 0$. However, this change does not alter the procedure of IFT-LPV. Formentin et al. (2013) also provide the analytically optimal controller parameters, which take the values:

$$\rho_0^{[0]} = 1, \qquad \rho_0^{[1]} = 1, \qquad \rho_1^{[0]} = 0, \qquad \rho_1^{[1]} = -1. \tag{5.108}$$

It is shown in this section that the IFT-LPV method is able to iterate to these analytically optimal values using the algorithm described in the previous section. In this case, a lifting window size of N is used, and the scheduling sequence is taken to be such that:

$$\mu_k = 0.5 \sin\left(\frac{2\pi k}{N} + \Psi_1\right), \tag{5.109}$$

where Ψ_1 is a random phase that changes every iteration. Further, the threshold for the metric $\epsilon_m^* = 1$. The convergence of IFT-LPV to the analytically calculated optimal values can be seen in Fig. 5.6. It can be seen that the developed IFT-LPV methodology is

Algorithm 2 IFT-LPV with a limited number of experiments

for $j = 0, 1, 2 \ldots$
for $j_\rho = 0, 1, 2 \ldots n_\rho$

Step 1: Perform reference experiment I with $Q^I_{k,N,1} = 0$.
Obtain $Y^I_{k,N,1}$ and μ^I_k.
Calculate $M_{cc}(\mu^I_k)$ and $\eta(\mu^I_k)$.
Calculate $Q^{II}_{k,N,1}$.

Set $\hat{j} = 1$. Set $\bar{\eta}_{II} = \emptyset$, $\bar{M}_{cc}(\mu^{II}_k) = \emptyset$, $Y^{II}_{k,n_{II}N,1} = \emptyset$.
Step 2: Perform gradient experiment $II^{\hat{j}}$ with $Q^{II}_{k,N,1}$.
Obtain $Y^{II,\hat{j}}_{k,N,1}$ and $\mu^{II,\hat{j}}_k$.
Calculate $M_{cc}(\mu^{II,\hat{j}}_k)$ and $\eta(\mu^{II,\hat{j}}_k)$.
Extend the block diagonal matrix $\bar{M}_{cc}(\mu^{II}_k)$ with new diagonal element $M_{cc}(\mu^{II,\hat{j}}_k)$.
Extend the matrix $Y^{II}_{k,n_{II}N,1} = [Y^{II}_{k,n_{II}N,1}; Y^{II,\hat{j}}_{k,N,1}]$.
Extend the matrix $\bar{\eta}_{II} = [\bar{\eta}_{II}, \eta(\mu^{II,\hat{j}}_k)]$.
Compute ϵ^{II}_m the norm of the projection of $\eta(\mu^I_k)$ on the null space of $\bar{\eta}_{II}$.
if $\epsilon^{II}_m > \varepsilon$, increment $\hat{j}$ and repeat Step 2.
else, compute
$z^I_{t,II} = M_{cc}(\mu^I_k)\bar{M}_{cc}(\mu^{II}_k)^T(\bar{M}_{cc}(\mu^{II}_k)\bar{M}_{cc}(\mu^{II}_k)^T)^{-1}Y^{II}_{k,n_{II}N,1}$.
end if

Set $\hat{j} = 1$. Set $\bar{\eta}_{III} = \emptyset$, $\bar{M}_{cc}(\mu^{III}_k) = \emptyset$, $Y^{III}_{k,n_{III}N,1} = \emptyset$.
Step 3: Perform reference experiment $III^{\hat{j}}$ with $Q^{III}_{k,N,1} = 0$.
Obtain $Y^{III,\hat{j}}_{k,N,1}$ and $\mu^{III,\hat{j}}_k$.
Calculate $M_{cc}(\mu^{III,\hat{j}}_k)$ and $\eta(\mu^{III,\hat{j}}_k)$.
Extend the block diagonal matrix $\bar{M}_{cc}(\mu^{III}_k)$ with diagonal element $M_{cc}(\mu^{III,\hat{j}}_k)$.
Extend the matrix $Y^{III}_{k,n_{III}N,1} = [Y^{III}_{k,n_{III}N,1}; Y^{III,\hat{j}}_{k,N,1}]$.
Extend the matrix $\bar{\eta}_{III} = [\bar{\eta}_{III}, \eta(\mu^{III,\hat{j}}_k)]$.
Compute ϵ^{III}_m the norm of the projection of $\eta(\mu^I_k)$ on the null space of $\bar{\eta}_{III}$.
if $\epsilon^{III}_m > \varepsilon$, increment $\hat{j}$ and repeat Step 2.
else, compute
$z^I_{t,III} = M_{cc}(\mu^I_k)\bar{M}_{cc}(\mu^{III}_k)^T(\bar{M}_{cc}(\mu^{III}_k)\bar{M}_{cc}(\mu^{III}_k)^T)^{-1}Y^{III}_{k,n_{III}N,1}$.
end if

Calculate the gradient $\frac{\partial y_I}{\partial \rho_{j_\rho}} = z^I_{t,III} - z^I_{t,II}$.
Calculate the performance gradient as $\frac{\partial J}{\partial \rho_{j_\rho}} = \frac{\partial y_I}{\partial \rho_{j_\rho}}^T y_I$.
Using the performance gradient, apply a gradient descent method and update ρ_{j_ρ}.

end for
end for

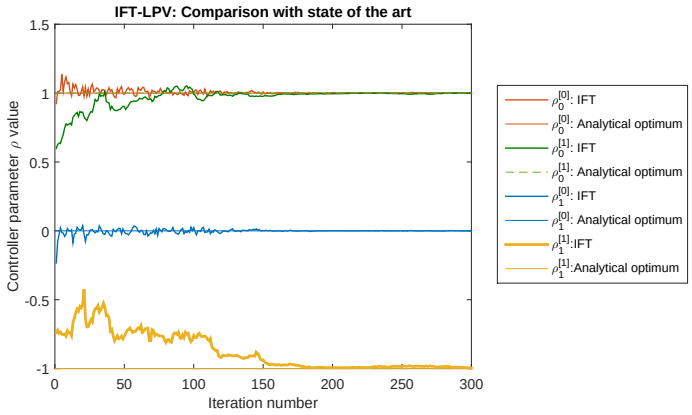


Figure 5.6: Iterative optimisation of controller parameters with IFT-LPV.

able to converge to the theoretically optimal values of the controller parameters under arbitrarily varying scheduling trajectories.

The total number of IFT experiments required for each iteration are given in Fig. 5.7. It can be seen that, since the threshold on the residual metric is relatively high, the number of experiments required per iteration are very low, with most iterations requiring a single gradient and a single reference experiment. There are no iterations in which the number of IFTexperiments demanded is greater than 5. It should be noted that, for ideal least-squares gradient estimation, the total number of experiments per iteration required would be 16.

While it appears that the convergence of the parameters takes around 200 iterations to complete, it should be noted that performance cost functions often have a broad minimum, and exact optimisation using gradient descent methods can become inefficient. However, if the descent along the cost function is visualised as in Fig. 5.8, it becomes apparent that excellent performance is already achieved after 35 iterations, where the tuned parameters are slightly different from their analytically optimal values. As such, for practical applications, it may be possible and even advisable to terminate the IFT tuning process once the desired level of performance has been achieved.

Comparing the method developed in this chapter with the direct data-driven method developed by Formentin et al. (2013) (VRFT-LPV), it can be seen that the current approach is applicable to a large range of LPV systems with full LPV controllers, and a reference model describing the desired behaviour of the plant is not required in this method. Further, while Formentin et al. (2013) describe controller tuning only for the case of periodic scheduling, this is not the case for IFT-LPV, where the scheduling trajectory can be arbitrarily complex. While VRFT-LPV only requires one single iteration for controller tuning, the tuned parameters are only approximately optimal, while it can be seen that IFT-LPV will, given an adequate number of iterations, arrive at the true (locally) optimal solution to the controller tuning problem.

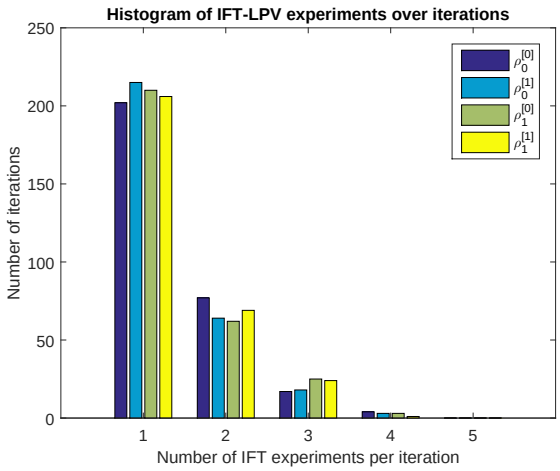


Figure 5.7: Histogram of the number of IFT experiments per iteration.

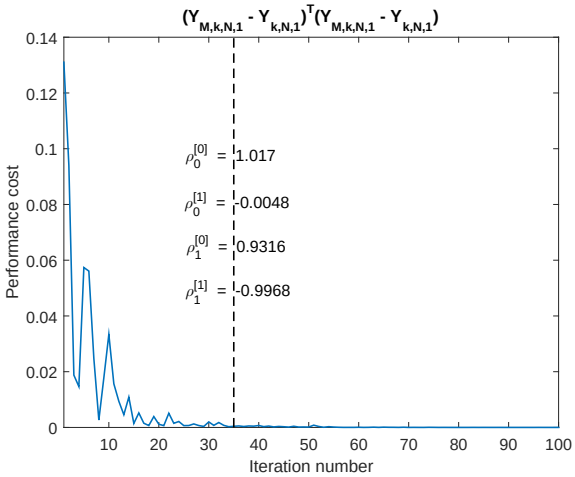


Figure 5.8: Iterative minimisation of the performance cost function.

IFT-LPV: THE MORPHING AEROFOIL REVISITED

While the previous section used a toy example for the purpose of comparison with a state-of-the-art data-driven LPV controller, this section revisits the morphing aerofoil LPV plant of Lee and Singh (2007), in this section considered to be an LPV system, as in the previous section, once again considering a lifting window of $N = 5$. The scheduling sequence is taken such that it is a superposition of a random signal with a sine of a random magnitude on an average wind speed of 6.5 m/s. In other words, the scheduling $\mu_k = V_k$ is given as:

$$V_k = 6.5 + 3\upsilon_k + 0.5\sin(\frac{2\pi k}{N} + \Psi_1), \tag{5.110}$$

where, as before, Ψ_1 is a random phase, and υ_k is a zero-mean white noise with unit covariance. As before, the system is connected in closed loop with an LPV PI controller, with controller parameters $K_p^{[0]}$, $K_p^{[1]}$, $K_i^{[0]}$ and $K_i^{[1]}$, as described in equation (5.99). The procedure laid out in Algorithm 2 is followed once more to tune these parameters using IFT-LPV. The reference signal is considered to be a step of magnitude 0.1, that stays constant over time, irrespective of the scheduling trajectory. The objective of the LPV controller is thus to reject the changes in the scheduling trajectory and command the same step response from the underlying LPV plant.

This optimisation problem does not admit a closed-form analytical solution. As such, in order to verify the optimality of the controller parameters, first, a four-dimensional grid search is performed to obtain the true optimal parameter bounds. It is seen that the cost function shows a clear local optimum, depicted using four two-dimensional graphs in Fig. 5.9. This figure shows the value of the logarithm of the cost function for each combination of PI-LPV parameters. In each graph, two of the parameters are held constant and the other two parameters are varied. The objective of IFT-LPV is to iterate, in a data-driven manner, to the local optimum found by the grid search in the four-dimensional space.

It can be seen in Fig. 5.10, that IFT-LPV is capable of iterating to the same optimal controller parameters as those obtained by the grid search. For ensuring a lower gradient approximation error, in this case, the threshold for the residual metric is taken to be $\epsilon_m^* = 0.1$. As a result of this, and because the scheduling sequence has a random white noise signal superposed on it, the number of iterations per experiment is higher than in the previous case. A histogram of the number of iterations per experiment is shown in Fig. 5.11. It can be seen that the most likely number of IFT experiments per iteration is 8, thus, the early termination reduces iteration time approximately by half.

Thus, this section shows by means of two case studies that IFT-LPV can approach the locally optimising fixed-structure LPV controller parameters to minimise the performance cost for a controlled LPV plant, in a direct data-driven manner. The concept of early termination serves to reduce the number of IFT experiments required at each iteration, and thereby reduces computational time.

5.5. CONCLUSIONS

Several industrial applications demand global LPV control for plants that show dynamic behaviour that can be approximated as LPV. IFT is an interesting candidate for optimising the gains of a low-order fixed-structure controller that is initiated from arbitrarily

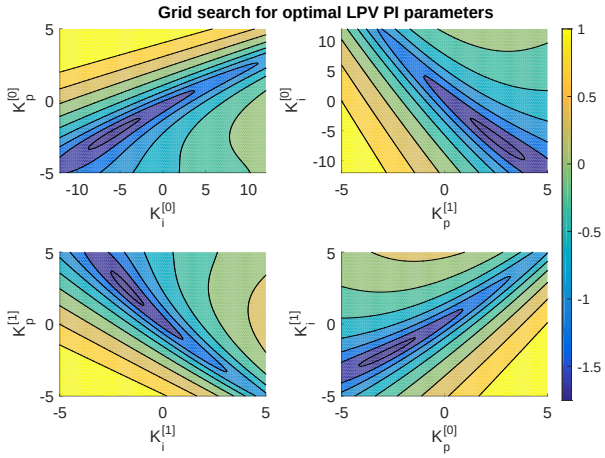


Figure 5.9: Grid search for the optimal PI-LPV parameters. The graphs show the logarithm of the cost function for the corresponding value of PI parameters.

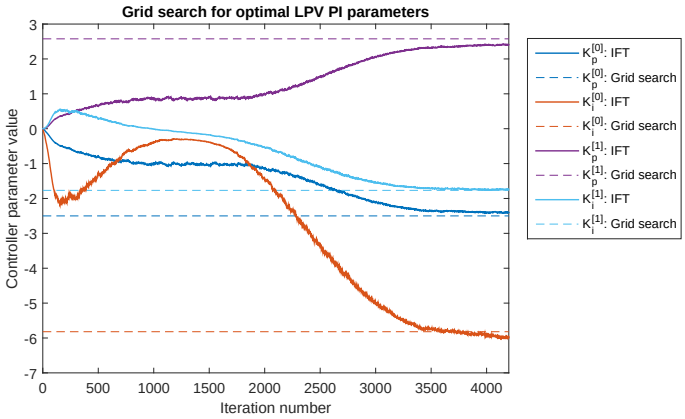


Figure 5.10: Comparison of IFT and Grid search for the optimal PI-LPV parameters.

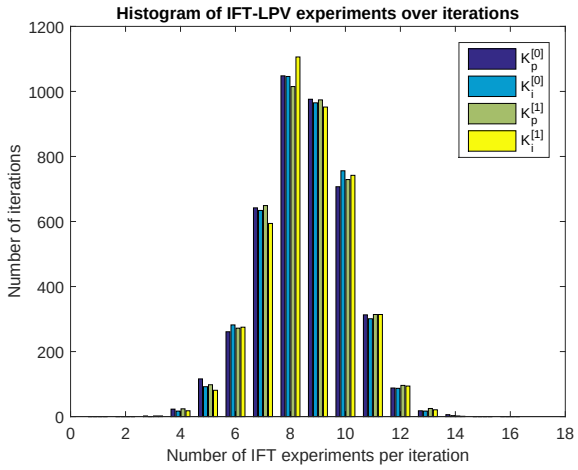


Figure 5.11: Histogram of the number of IFT experiments per iteration.

poor but stabilising initial values. This optimisation takes as argument a user-defined cost function, which is constrained to be such that its optimisation yields a stable closed loop. IFT uses input-output data to construct the gradient of the performance cost with respect to the controller parameters, which can be used in a gradient-based optimisation method to iterate to a locally optimising solution. For an adequately small step size in the gradient-descent method, IFT always reaches the local optimum of the cost function.

It is assumed here that the scheduling is exogenous, perfectly measurable and uncontrollable.

Conventional IFT is not applicable for the tuning of a global LPV controller for LPV plants, since the gradient estimates are then contaminated by the time-varying effect of the scheduling sequence. This effect can be compensated for by the iteratively optimising a gain schedule using IFT for the LPV plant, instead of the absolute value of the controller gains. However, this demands that the scheduling be approximated as constant during each iteration, this requirement may be too strict for practical applications.

For the case where the scheduling varies arbitrarily, the lifted system matrices can be decomposed into a known, time-varying factor that depends purely on the scheduling sequence, and an unknown, time-invariant factor characteristic of the system. With this factorisation, given an adequate number of experiments, the effect of the scheduling sequence can be compensated for, and the performance gradient can be estimated in an unbiased manner. For the case of tuning a feedforward controller for a plant LPV in the output matrices, the inner size of such a factorisation is linear in the lifting window size. Thus, although an increased number of experiments are needed for gradient estimation, this number increases linearly with the complexity of the problem. On the other hand, for the more general case of feedback tuning for a fully LPV plant, the inner size of the factorisation grow exponentially with the size of the lifting window, and the number of

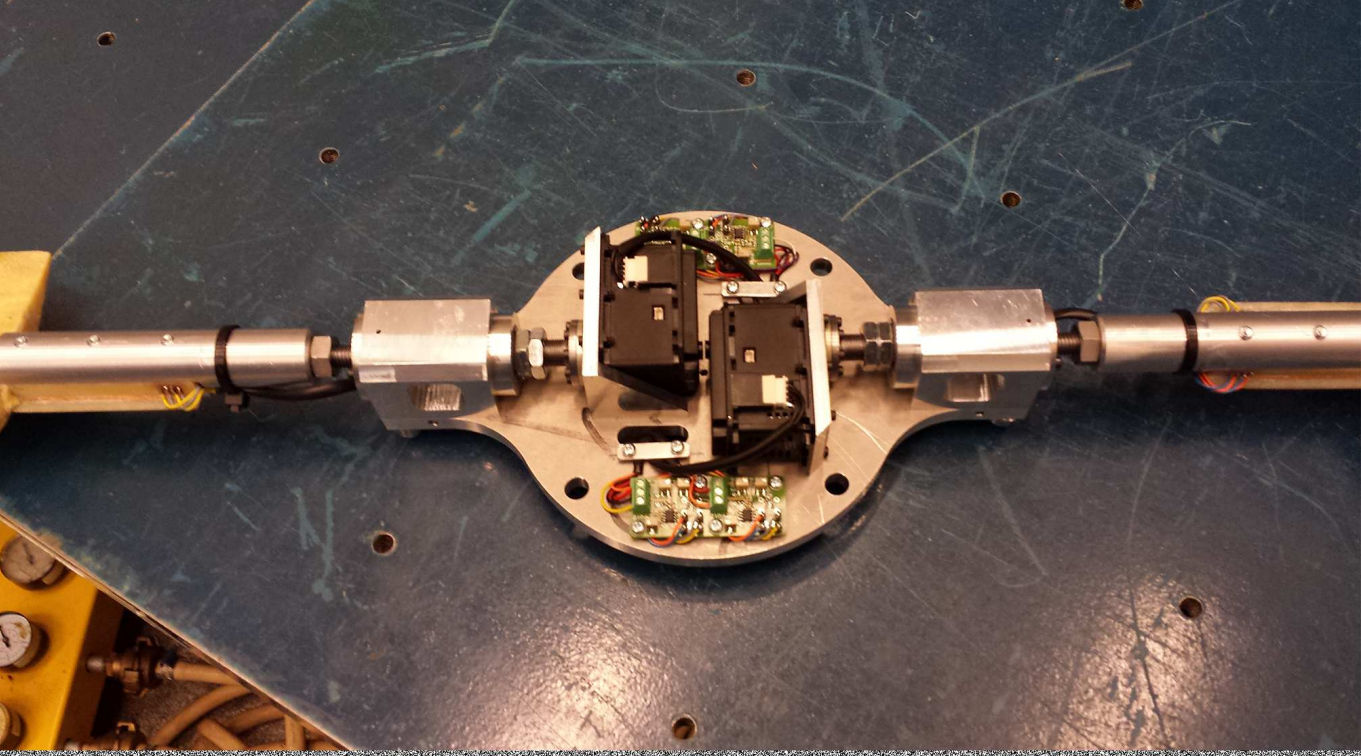
IFT experiments demanded may quickly grow infeasible for practical implementation. This phenomenon is the curse of dimensionality, well-known in the field of global LPV identification.

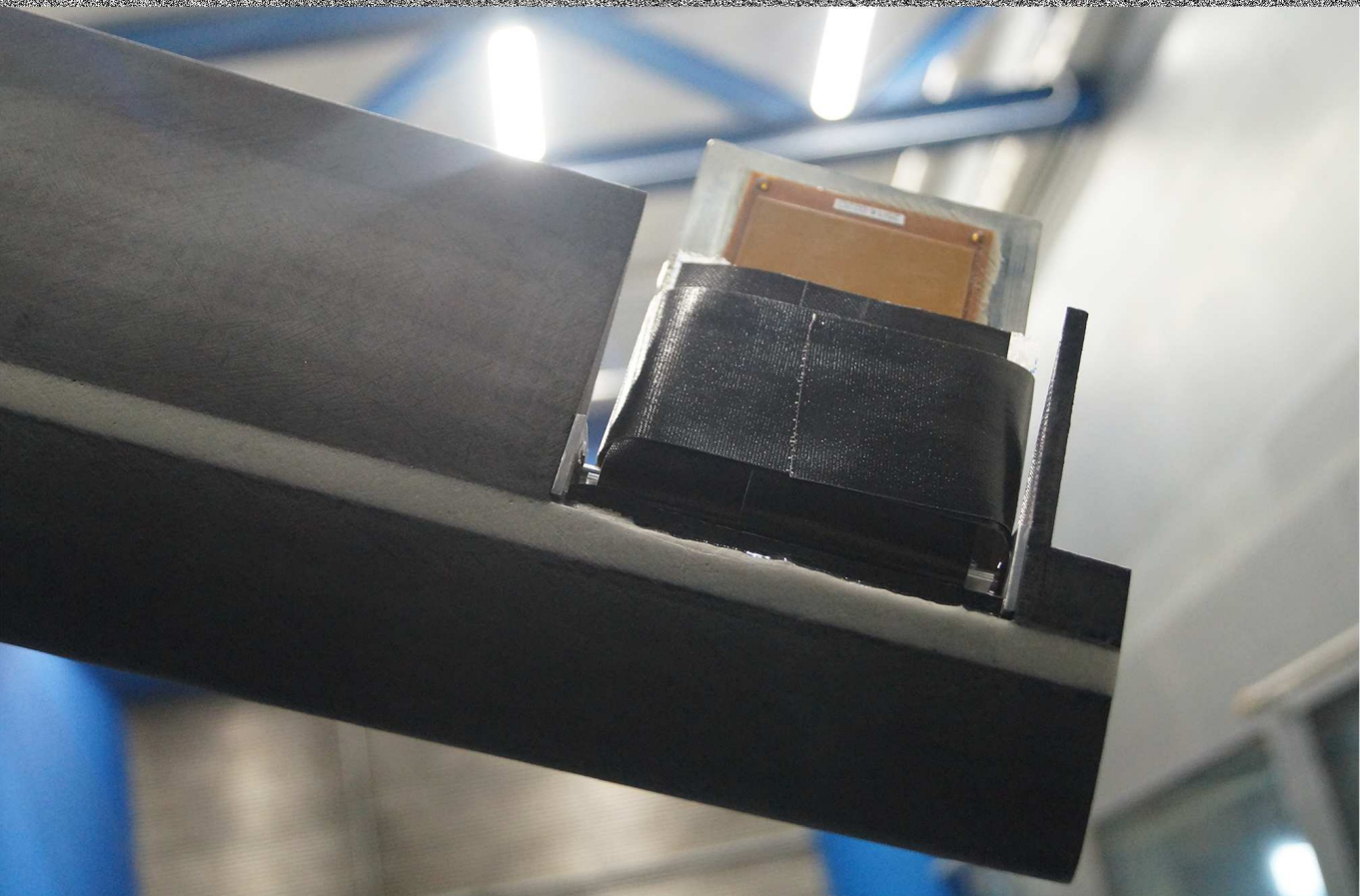
Two solutions have been proposed to overcome this curse of dimensionality. Firstly, for an LPV system (or scheduling sequence) that shows additional structure, such structure can be exploited to reduce the inner size of the factorisation. An example is taken for the case of a switched system, where the switching occurs once in every iteration. The approximate factorisation for this special case has a small inner size, and it is possible to tune the gains of a PI-LPV controller efficiently using IFT-LPV.

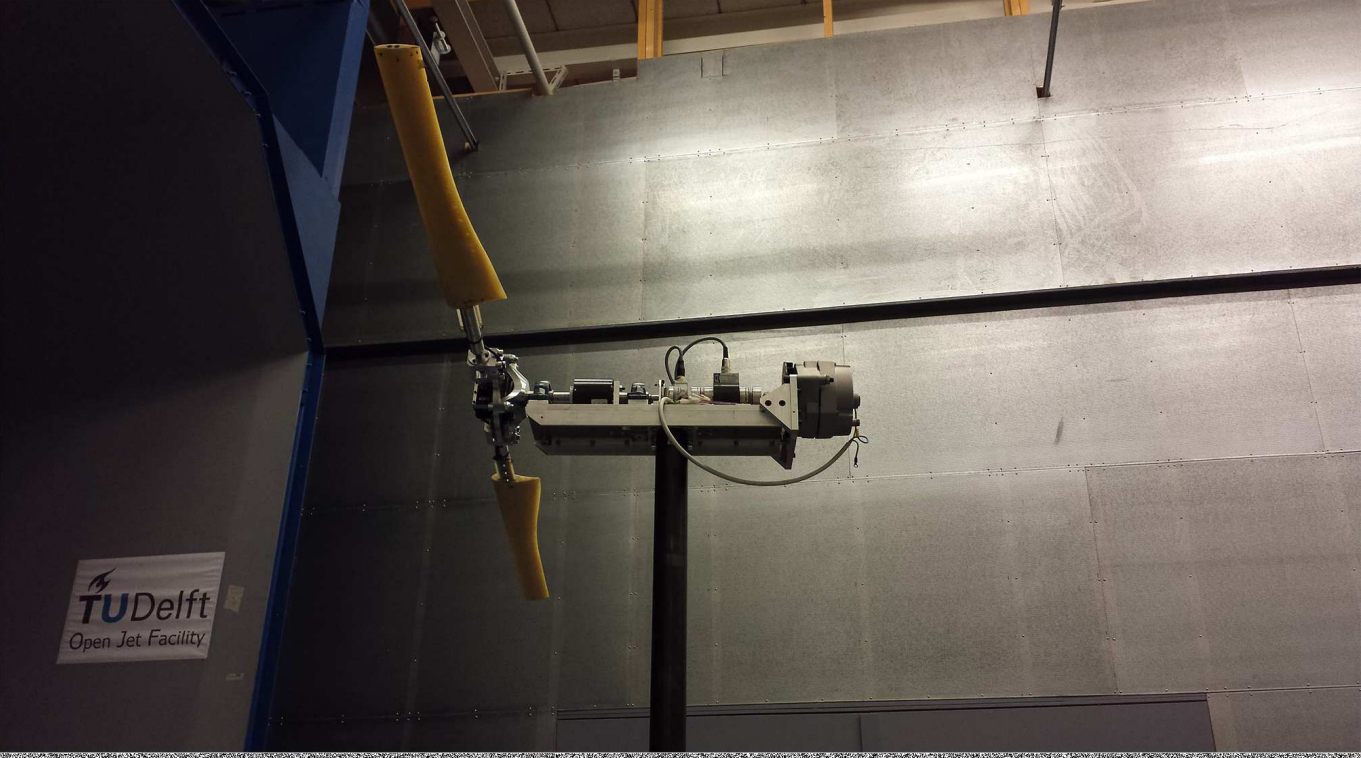
Secondly, it is possible to define a residual metric that can be thresholded to limit the number of IFT experiments in order to determine an approximation of the performance gradient estimate. This heuristic is a function of the closeness of the scheduling trajectories during the reference and gradient experiments. For the degenerate LTI or periodic scheduling case, this metric is identically 0 after the first gradient experiment, and only one more experiment is required for unbiased gradient estimation. Thus, this case is identical to IFT-LTI. As the complexity of the scheduling variation increases, the value of the metric becomes non-zero and multiple IFT experiments are required to obtain an acceptable approximation of the performance gradient.

IFT-LPV with early termination is described with the help of two case studies. In the first one, the performance of IFT-LPV is compared with the state-of-the-art VRFT-LPV algorithm, on the same numerical example. It is seen that while IFT-LPV demands more iterations, it is able to arrive at the exact values of the theoretically calculated optimal controller gains. Further, IFT-LPV is able to handle a greater scheduling complexity. Further, a more realistic LPV plant is considered, which concerns a fourth-order morphing aerofoil the dynamics of which vary strongly with wind speed, the scheduling variable. No theoretical solution exists for the optimal values of PI-LPV parameters for this model. It is found that IFT-LPV iterates to the same optimal controller values, as those found by using an exhaustive four-dimensional brute-force grid search. For this case, the implementation of early termination reduces the time of each iteration approximately by 50%.

Thus, this chapter develops and demonstrates a direct data-driven approach for optimally tuning the controller parameters for a global LPV controller, for minimising the performance cost of an LPV plant in closed loop. The developed strategy demands no more than a minimal amount of prior knowledge regarding the true system dynamics and the scheduling sequence. The second part of the thesis will validate the efficacy of the limited-complexity IFT-LPV algorithms in an experimental manner.






TUDelft
Open Jet Facility

II

For Flexible Wind Turbine Rotors

6

STATE OF THE ART

As yet, the wind is an untamed and unharnessed force;
and quite possibly one of the greatest discoveries hereafter made,
will be the taming and harnessing of it.

Abraham Lincoln, the USA (1809-1865),
describing the wind turbine control objectives of load alleviation and power capture.

This chapter reflects on the current state of the art of load control for modern wind turbines. First, the chapter explains the main components of the wind turbine and the control degrees of freedom, explaining the feedback control used in commercial wind turbines for nominal operation. Next, the concept of Individual Pitch Control (IPC) is discussed in terms of the advantages qua load reduction, and the increased actuator duty costs of its implementation. Finally, the efforts of the research community in the development of the concept of the 'smart' rotor is investigated, and the gaps in our current knowledge of the behaviour of such an advanced flexible rotor are identified.

6.1. INTRODUCTION

Modern wind turbines are massive, multi-megawatt machines; with rotor diameters exceeding 150 m, they form the largest rotating structures designed in the engineering world. This size is necessitated by the requirement to reduce the cost of energy to make wind power competitive with respect to conventional power sources: the larger the wind turbine, the lower the capital costs, as well as the reliability-driven operational costs. This progressive increase in rotor diameter has received further impetus from the move of wind energy offshore, in order to exploit the higher wind power potential and smoother wind fields. However, it has been estimated that this increase cannot be indefinitely sustained using the current level of technology: at a certain point, the increase in material costs as a result of rotor upscaling will not be offset by the corresponding increase in power capture.

Several advanced approaches are under research for delaying this point of peak rotor size, one of the most interesting of which is the development of flexible rotors with reduced material requirements. Such a rotor would require active and passive load control to limit the loading and ensure longevity of the turbine: wind turbine rotors are exposed to severe dynamic loads, with upto 10^8 to 10^9 loading cycles in their 20 year design lifetime, Nijssen (2006). Especially in the offshore environment, turbines also need to be designed to require a minimal amount of maintenance downtime. Load control of lightweight, flexible rotors would be beneficial for the turbine as a whole, as the load alleviation derived from such rotors would cascade downstream through the entire turbine supporting structure.

Loads on the wind turbine consist of deterministic components, like the effect of tower shadow and wind shear, and random components like atmospheric turbulence. As a result of the rotational nature of the system, the load spectra tend to show dominant peaks at the rotor speed (1P) and its harmonics (2P, 3P,...). Modern rotors tend to have rotational speeds of the order of 10 rpm, thus the most important exogenous load peak occurs at a frequency of the order of 0.1 Hz. Atmospheric turbulence acts to broaden these peaks; it also adds energy to the high-frequency region of the load spectrum.

Structural resonance modes show little influence on rotor loads[1] for conventional onshore turbines, since the first flexible rotor mode (flapwise) is usually of the order of 3.5P, and its modal energy is typically damped through the effect of aerodynamics. Such rotors are relatively stiff in torsion, and the flapwise and torsional degrees of freedom do not couple aeroelastically in the operational regime of traditional wind turbines. However, as rotors become more flexible, this assumption of modal independence is no longer valid, and the turbine rotor modes may couple in an unstable manner resulting in the highly destructive phenomenon of flutter. Such a phenomenon may also be caused by the introduction of advanced flow control devices along the blade; these same devices can also, in the closed loop, mitigate this phenomenon and stabilise the wind turbine in adverse wind conditions.

It is to be expected that control of these loads will be achieved, in the wind turbine of the future, by a combination of passive and active load control strategies. Passive load control corresponds to the use of elements that show fixed dynamic characteristics and typically do not require external power or control signals; these elements have seen widespread use in infrastructure, for instance, in the earthquake stabilisation of skyscrapers. An equivalent application in wind turbine rotors would however require relatively massive components, located within a rotating body, as described by Zhang et al. (2016) – these components are also typically designed to be responsive to one single modal frequency, and can often not react to frequency changes arising out of changes in the wind speed, or system structural properties. A different passive approach is to add sprung masses in the form of a blade trailing edge flap, Bottasso et al. (2016), which alleviates dynamic loads in response to wind speed variations. Such a component need not be physically massive, but it may still lack robustness to uncertainty or to variations in ambient environmental parameters. Passive approaches do not form part of the scope

[1] The first rotor lead-lag mode is relatively undamped, and it is found to be excited by periodic gravitational loading. However, this mode is relatively uncontrollable based on the current control degrees of freedom, and will not be explored in this thesis

of this thesis, however the author believes that such approaches are in many ways complementary to the active control methods studied herein.

An alternative, or extension, to passive control is the inclusion of active elements in the flexible rotor. One of the most direct methods of active control is full-span pitch, which can relieve loads on each blade independently, using the so-called 'Individual Pitch Control' (IPC) strategy. Other methods under research include the use of trailing edge flaps or tabs, or flow-modifying devices along the blade span, tailored for localised rotor load alleviation. Such methods typically require flow or load sensors along the rotor in order to operate in the closed loop. Active control methods can often provide more accurate load control than passive methods, however they require external power. As the combined aeroelastic effect of these devices is difficult to model, ensuring robust stability and performance for active wind turbine control can prove to be a challenging control problem.

This chapter first describes the main components and the baseline feedback controllers of a modern commercial wind turbine. Next, the use of IPC for controlling wind turbine loads will be discussed. Finally, attention is devoted to the 'smart' flexible rotor, i. e. a wind turbine rotor incorporating active load control. Special emphasis is laid on the concept of trailing edge flaps and morphing aerofoils, which show the highest degree of maturity qua practical concept realisation.

6.2. CONTROL OF COMMERCIAL WIND TURBINES

6.2.1. COMPONENTS OF A WIND TURBINE

A modern commercial wind turbine is a complex mechatronic system that involves the interplay of the energy capture, power conversion and control subsystems. The main components of the wind turbine, from the perspective of load alleviation, are described below, Manwell et al. (2002):

- **Rotor:** The rotor consists of a hub on which upto three blades are mounted; the rotor is free to rotate on a horizontal axis. The flow of the wind through the plane of the rotor is reponsible for the generation of aerodynamic torque, which is converted to electrical power by the downstream components of the turbine. The flow of the wind also generated aerodynamic loads that the load controller seeks to minimise.

- **Transmission:** The rotor is rigidly connected to the main shaft, that is supported on one or more main bearings. The transmission is connected to the wind turbine generator on the other side. A gearbox may be used to increase the speed of rotation.

- **Electrical subsystem:** The generator converts mechanical rotation into electrical power. Both the transmission and the generator are typically housed in the nacelle, located on top of the tower of the wind turbine. The generated power is conditioned by an electronic converter and a transformer such that it can be fed directly into the grid.

- **Support structure:** This term refers to the tower and the other structural elements

of the turbine, which are designed to withstand the operational loads of the turbine.

The control degrees of freedom of a conventional wind turbine are as follows:

- **Yaw control:** The nacelle is connected to the top of the tower through a yaw mechanism, which uses electrical or hydraulic motors for aligning the nacelle with the inflowing wind.
- **Torque control:** The generator speed can be controlled by using the electronic power converter to manipulate the electrical braking torque imposed on the transmission, and hence, the load driven by the wind turbine rotor.
- **Pitch control:** Each blade is equipped with electrical motors or hydraulic actuators such that it can be rotated about its longitudinal axis. While each blade can be pitched independently, baseline control only demands that the turbine blades be collectively pitchable.
- **'Smart' control:** Research is underway regarding the use of local flow control devices along wind turbine blades that could alleviate wind turbine loads.

A typical direct-drive commercial wind turbine is shown in Fig. 6.1. The various components of the turbine have been illustrated in the full and cross-sectional views. As a direct-drive turbine, the transmission of this machine does not include a gearbox, and is hence not susceptible to the concomitant reliability and maintenance issues. As a flipside, the generator is significantly larger, and usually requires large quantities of rare earth metals in its construction.

Depending on the wind speed, the operational regime of the wind turbine is divided into two main regions: the below-rated region, during which the wind speed is below the rated wind speed of the turbine, and the above-rated region, defined in the same way. In the below-rated region, the objective of wind turbine control is to maximise the energy capture, while the objective in the above-rated region is to regulate the turbine to rated power capture while avoiding rotor overspeed for safe operation. These control objectives relate mainly to the generation of power, however an increasingly important task of the wind turbine controller is the minimisation of loads, such that the turbine can survive its lifetime dynamic loading while minimising the costs of construction and maintenance.

6.2.2. BASELINE CONTROL OF A WIND TURBINE

Baseline wind turbine control has been depicted in Fig. 6.2. Additional advanced controllers can be connected in closed loop with this baseline controlled wind system. This section describes the main components of the baseline controller that are required to achieve speed regulation for optimal wind energy capture.

As stated before, the baseline control objective in the below-rated region is maximum energy capture, and in the above-rated region is speed regulation to the nominal operating speed of the turbine. Both objectives can be achieved by rotor speed trajectory tracking. In the below-rated region, torque control is used to maximise energy capture.

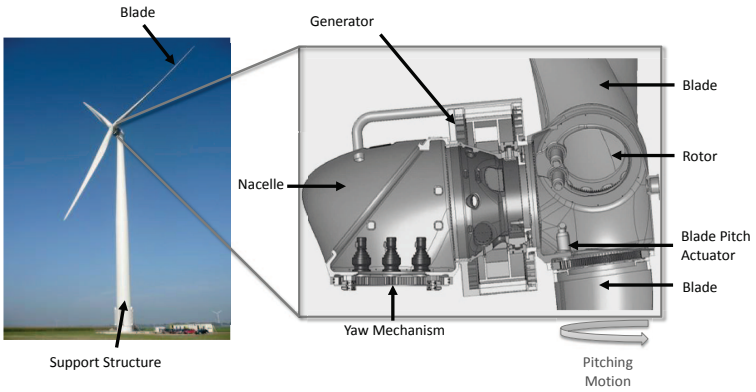


Figure 6.1: XEMC-Darwind XD115 wind turbine: full and cross-sectional views, XEMC-Darwind (2013).

The torque controller typically sets the generator torque to be proportional to the square of the generator speed, as per Manwell et al. (2002). In this region, the pitch of the blades is set to fine, which ensures that energy capture is possible in the most aerodynamically efficient manner. The torque control schedule is generally set up in such a manner that it minimises dwell time at the speeds close to the tower structural frequencies, in order to avoid undue excitation of tower modes.

In the above-rated region, the commanded generator torque is set to its nominally rated value. In the incoming aerodynamic torque is regulated by means of collectively pitching the wind turbine blades in such a manner that the generator speed remains constant at its rated value. This approach is called Collective Pitch Control (CPC), and is widely adopted by modern commercial wind turbines to regulate power extraction to nominal in the above-rated region.

Structural loads, typically fore-aft and side-side tower loads, and drive train torsional loads, are damped by the use of classically loop-shaped torque or CPC controllers that are connected in closed-loop with the baseline controller described in Fig. 6.2. Further, since a large proportion of turbine fatigue occurs from loading around the rated wind speed, peak-shaving is often implemented around this wind speed. Peak-shaving refers to a preponed implementation of collective pitch control, thus CPC is initiated even when the turbine is operating in the below-rated region (but close to the rated wind speed). This concept sacrifices some amount of power capture while ensuring that the turbine avoids the high peaks loads that can arise at these wind speeds. It should be noted that structural load control and peak-shaving is turbine- and model-specific, and has not been formalised or generalised for all wind turbines.

A full treatment of baseline closed-loop feedback control for wind turbines has been done by Bossanyi (2000). This reference describes the torque and pitch control required for following the desired turbine wind speed-power characteristics defined above. Further, the concept of structural modal damping has also been discussed, and guidelines

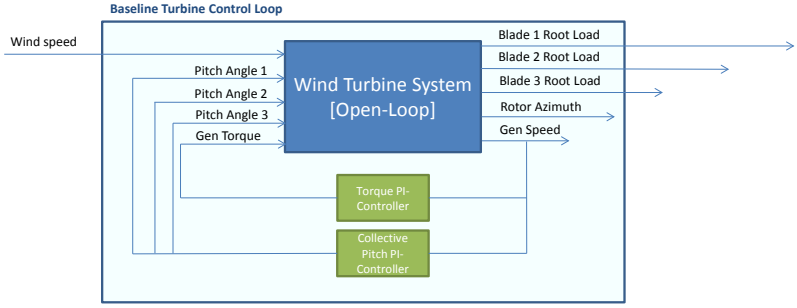


Figure 6.2: Wind turbine baseline control block diagram. [Turbine modal damping controllers not depicted]

have been set down for designing such controllers. Commercial wind turbines typically employ this approach to feedback control, and this reference is hence considered to represent 'baseline' control throughout the thesis, as recommended by Laks et al. (2009).

As mentioned by Bossanyi (2000), however, torque control and collective pitch control are not capable of addressing the load reduction issue at the dominant loading peaks of 1P, 2P and so on. This limitation arises out of the fact that the loads are asymmetric on the turbine blades, and hence cannot be attenuated by these symmetric control degrees of freedom. Further, localised loads that occur due to the interaction of structures in the wind with the increasingly large and flexible blades, can neither be sensed nor dissipated at the level of the global turbine torque and pitch controllers. This motivates the recent research into Individual Pitch Control (IPC) and the development of the concept of the 'smart' rotor.

6.3. INDIVIDUAL PITCH CONTROL (IPC)

For the purpose of baseline generator speed control, wind turbines today are equipped with pitch actuators that allow the independent pitching of each turbine blade, termed as Individual Pitch Control (IPC). The concept of IPC has been borrowed from the helicopter industry, Kessler (2011), wherein the rotating blades are pitched along their longitudinal axis for the purpose of load reduction. This control technique, applied to wind turbines, was found specifically useful for reducing periodic asymmetric loads, such as 1P and its harmonics, Bossanyi (2003). It is to be noted that due to reduced control authority, individual pitch cannot be used to best effect for fatigue load reduction in the below-rated region in most cases, Fischer et al. (2011). Since most of the fatigue-relevant loading occurs at the rated wind speed and higher, this is not necessarily a drawback. IPC can also be used to control tower and drive train structural loads. However, collective pitch or torque control may offer a more viable alternative to individual pitch for these loads, Fischer et al. (2012).

Thus, IPC is designed to attenuate individual periodic loads that act asymmetrically

on each blade, using feedback control. As such, it is necessary to obtain blade load measurements for each blade independently. Modern wind turbines are increasingly instrumented with blade root strain gauges to provide the feedback necessary for IPC. It may also be possible to reconstruct the blade loads using state estimation from rotor speed and pitch, however it has been shown in Stol and Balas (2003) that this severely restricts load reduction potential.

For a conventional wind turbine, IPC is hence required to be designed for synthesising three individual blade pitch control inputs based on the feedback from the three blade root load sensors, and the plant is thus inherently multivariable. Further, this system is non-linear in that it shows periodically varying dynamics, with base period 1P. As such, classical control design, while perfectly feasible for designing IPC controllers for wind turbines, stands to be simplified and optimised for enhanced load alleviation.

6.3.1. CONTROL APPROACHES: LTI AND PERIODIC CONTROL

In one of the earlier works on IPC for load reduction, the use of pitch control for reducing blade loads on a wing section model has been explored by Kallesøe (2006). Here, an LQ controller is designed to demonstrate load reductions. However, since the model is non-rotating, the azimuthal variations do not need to be accounted for, and linearising the model is straightforward. In some of the earlier works with full turbine models, exploring the concept of IPC, the system from individual pitch to blade and turbine loads has also been considered to be linear, for example Moriarty et al. (2001), Wright (2004). Similarly, in Stol (2003), the wind speed is modelled as an additive filtered disturbance, and the variations in the system due to change in azimuth are neglected. The controller is then designed as a multivariable controller using LQG techniques for torque and individual pitch control. However, although fatigue reduction is reported, it is concluded that the system is not linear even at constant wind speed, and the periodicity arising out of the azimuthal variation should be included to maximise load reduction. This has also been shown through field test results by Stol et al. (2006).

An optimal periodic control law is devised by Liebst (1985), while another implementation of periodic control is by linearising the periodic plant model at different azimuths and using LQG design, which shows better fatigue reduction capabilities, Stol and Balas (2003). Such a periodic law has been applied successfully to a floating offshore wind turbine in a simulation environment by Namik and Stol (2010), both for blade load reduction as well as for limiting the pitch motion of the turbine as a whole.

Many recent investigations of the technique of individual pitch control are extensions or variants of IPC with the Coleman or Multi-Blade Coördinate (MBC) transformation. This method is fundamentally based on the principle that when the MBC transformation is applied to the individual pitch system to bring it from a rotating frame of reference to a stationary frame of reference, we arrive at a linear decoupled system which is amenable for controller design via classical frequency domain techniques. In fact, it has been shown that a detailed design of an optimal periodic feedback law, although fundamentally more complex and accurate, will perform no better than the MBC transformation-based IPC, Stol et al. (2009).

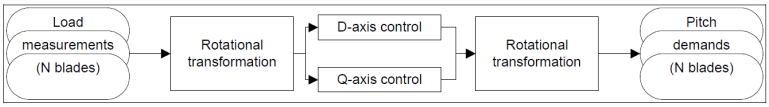


Figure 6.3: IPC with MBC transform for 1P load attenuation, Bossanyi (2003)

6.3.2. CONTROL APPROACHES: MULTI-BLADE COÖRDINATE TRANSFORM

The baseline controller for the implementation of IPC with the Multi-Blade Coördinate (MBC) transform remains similar to the classical turbine controller described in the previous section, in that there exists closed loop torque and collective pitch control for speed regulation. Further, individual pitch control is superposed on top of the collective pitch control in order to limit cyclic loading at 1P and its harmonics.

Commercial wind turbines typically have access to azimuth measurement, which is essential in this variation of IPC. Indeed, IPC can also be done purely via tracking the azimuthal position of the rotor blades; this is termed as 'cyclic pitch control' in the literature Bossanyi (2000), and derives from the helicopter industry.

Structurally, the controller is divided into three distinct parts, as can be seen in Fig. 6.3. Essentially, the load measurements are converted by the MBC transform from the rotating reference frame into a stationary frame of reference. This results in two orthogonal load signals, which have been named the d-(direct) and q-(quadrature) signal in this reference, in analogy with electrical machine theory.

In principle, in order to obtain the optimum control signals from these two load signals, a multivariable control problem needs to be set up, and an LQG controller could be designed. In Geyler and Caselitz (2008), a multivariable control problem has been set up, and an H_∞ controller is derived for IPC; it is shown that the performance is very similar to that of a simple PI controller. Using a simple analytic model of the wind turbine wtih limited degrees of freedom, it has been shown in Bossanyi (2003) that the performance of two decoupled PI controllers is similar to that of multivariable controllers, and PI controllers are easier in implementation. It is to be noted, however, for reliable safety margins, it may still become relevant to consider multivariable design methods for IPC controllers, Bossanyi et al. (2012b), Lu et al. (2015). Black-box controllers such as fuzzy logic and neural networks have also been proposed as alternatives by Bossanyi (2000), but no further work has been done in this field.

In most cases of prototype implementation in the literature, the two d-q axes load signals are considered decoupled from each other, and independent PI-controllers are tuned for them. An inverse MBC transform is then applied to the resulting control action from the d- and q-controllers, which gives the individual pitch action required for each blade.

If L_a, L_b, ... are the load signals in the rotating frames of reference, for B number of rotating blades, then the d- and q-loads, L_d and L_q are given by the equation as below:

$$\begin{bmatrix} L_d \\ L_q \end{bmatrix} = \frac{2}{B} \begin{bmatrix} \cos\Psi & \cos(\Psi + \frac{2\pi}{B}) & \cos(\Psi + \frac{4\pi}{B}) & \cdots \\ \sin\Psi & \sin(\Psi + \frac{2\pi}{B}) & \sin(\Psi + \frac{4\pi}{B}) & \cdots \end{bmatrix} \begin{bmatrix} L_a \\ L_b \\ L_c \\ \vdots \end{bmatrix} \tag{6.1}$$

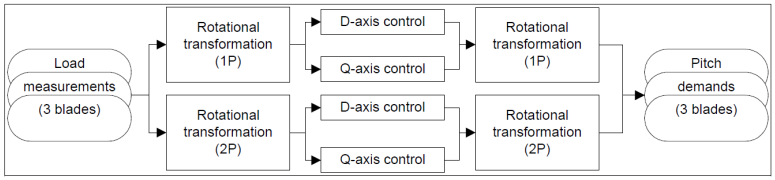


Figure 6.4: IPC with MBC transform for higher harmonics, Bossanyi (2003)

The output of the controllers will be pitch signals P_d and P_q along the d- and q-axes. This can be resolved into the blade pitch motions P_a, P_b, ... by performing an inverse MBC transform. In principle, the inverse transform can use the same value of azimuth Ψ as used for the forward transformation. In order to enhance load reduction, Buhl et al. (2007), the phase lag between pitch action and blade load, δ_{1P}, can be taken into account. Thus, the angle used for the inverse MBC transform is:

$$\Psi' = \Psi + \delta_{1P}, \tag{6.2}$$

and the transformation itself is given by

$$\begin{bmatrix} P_a \\ P_b \\ P_c \\ \vdots \end{bmatrix} = \begin{bmatrix} \cos\Psi' & \sin\Psi' \\ \cos(\Psi' + \frac{2\pi}{B}) & \sin(\Psi' + \frac{2\pi}{B}) \\ \cos(\Psi' + \frac{4\pi}{B}) & \sin(\Psi' + \frac{4\pi}{B}) \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} P_d \\ P_q \end{bmatrix}. \tag{6.3}$$

It can be seen that such a controller will reduce the periodic loads of frequency 1P for rotating components. When transformed to a fixed frame of reference for three-bladed turbines, the control action has a reducing effect on the low-frequency loading of the stationary components, most significantly the yaw bearing and yaw system, as also the tower base loads. In general, the loading of these components is dominated by a peak at frequency 3P, which is not addressed by the configuration above. Workarounds have been described for this – for instance, in Bossanyi (2005), a feedforward term has been included in parallel with the PI controller for the d- and q-axes, which targets 3P loads in the fixed reference frame. This is achieved by additional 1P and high frequency pitch actions.

A more effective and reliable alternative for this is to develop a similar additional controller with MBC transformation, but the transformation in this case is done for twice the azimuth angle. By superposing the additional pitch actions obtained from this controller to the one described above, 3P loads in the fixed frame, and, additionally, 2P loads for rotating components can also be reduced, in addition to the load reduction achieved by the previous controller. This is shown in Fig. 6.4. This paradigm can be extended for higher harmonic load reduction as well, Van Engelen (2006). It is shown that effect of higher harmonic control is more pronounced mainly in low turbulence conditions, Bottasso et al. (2013). Individual pitch control (IPC), implemented as described above, has been validated both in the simulation environment, as well as in the field.

6.3.3. IPC VALIDATION: AEROELASTIC TOOLS

Since IPC does not require additional control degrees of freedom, it can be directly validated using state-of-the-art wind turbine aeroelastic analysis tools. Further, IPC is generally designed to address low-frequency load components, and as such, a quasi-steady aerodynamic model may be considered adequate for investigating the load alleviation potential. For the sake of completeness, this section details the validation of IPC using different simulation environments.

- GH Bladed: This tool, developed by Garrad Hassan (now part of DNV-GL), has been used for the load analysis and certification of new wind turbine models. Bladed models the turbine structure using the multi-body approach, typically considering the blades and the tower to be flexible bodies. A fully turbulent 3D wind field can be simulated, and its interaction with the rotor disc is described using the Blade Element-Momentum (BEM) theory. This approach uses the quasi-steady aerodynamics approximation; corrections are made for dynamic stall, yawed inflow and 3D effects. Bossanyi (2003), Bossanyi (2005), Fischer et al. (2011), Fischer et al. (2012) performed fully turbulent simulations of a reference turbine using this software, and reported significant load reductions at the dominant load peaks using IPC.

- FAST: The software FAST, developed by NREL, couples a structural dynamics model that uses Kane's method of assumed modes, with a BEM model for aerodynamics, corrected for wake effects and dynamic stall. The tool is capable of handling fully turbulent wind fields. FAST has been used to compare MBC-based IPC with periodic control laws, Stol et al. (2009), Ozdemir et al. (2011),. This environment is adequately versatile to admit several extensions, for instance, IPC for floating turbines was explored by Namik and Stol (2010), Namik and Stol (2011).

- Cp-Lambda: This tool has been developed in Politecnico di Milano to provide a wind turbine simulation environment, it uses a finite-element multi-body formulation to describe structural dynamics. The aerodynamics utilises the lifting line concept; the BEM theory is used for describing aeroelastic interaction with root and tip losses, dynamic inflow and dynamic stall corrected for. An operational approach to the integration of IPC in the overall turbine load control strategy was explored by Bottasso et al. (2013), using Cp-Lambda for aeroelastic simulations.

- HAWC: HAWC is the aeroelastic analysis tool from the Danish Technical University used for the dynamic analysis of wind turbines. HAWC uses a prismatic finite-element formulation of the wind turbine structure. The quasi-steady BEM theory is used with a dynamic stall model in order to describe the aerodynamic loading of the structural elements. This code is used for evaluating IPC based on flow measurements by Larsen et al. (2005).

- TURBU and PHATAS: TURBU is a linearised aeroelastic analysis tool developed by ECN, meant for developing linearised models for controller development. This tool is programmed using a modular structure in the Matlab-Simulink environment. Along with a linear structural model, it uses BEM theory, extended with

Prandtl's correction for wake effects to describe the behaviour of the wind turbine. PHATAS is a non-linear simulation environment developed by ECN that uses a number of rigid and flexible structural degrees of freedom that interact with the wind flow in a manner described by the BEM theory with corrections for the tip effects, 3D effects and the dynamic wake. TURBU is used in Van Engelen (2007), Selvam et al. (2009) for generating linearised models for IPC controller synthesis.

In general, the IPC controllers implemented using aeroelastic tools assume simple models of the pitch system that issue a pitch angle command for each turbine blade. This may be further cascaded with the pitch controller to generate the appropriate pitch rate. It has be shown by Moriarty et al. (2001) that pitch actuator lag can obviate the load reductions obtained by pitch control, however it has been since observed in recent prototype implementations that this lag is not significant in practice. The field implementation of the concept of IPC is discussed next.

6.3.4. IPC VALIDATION: FIELD RESULTS

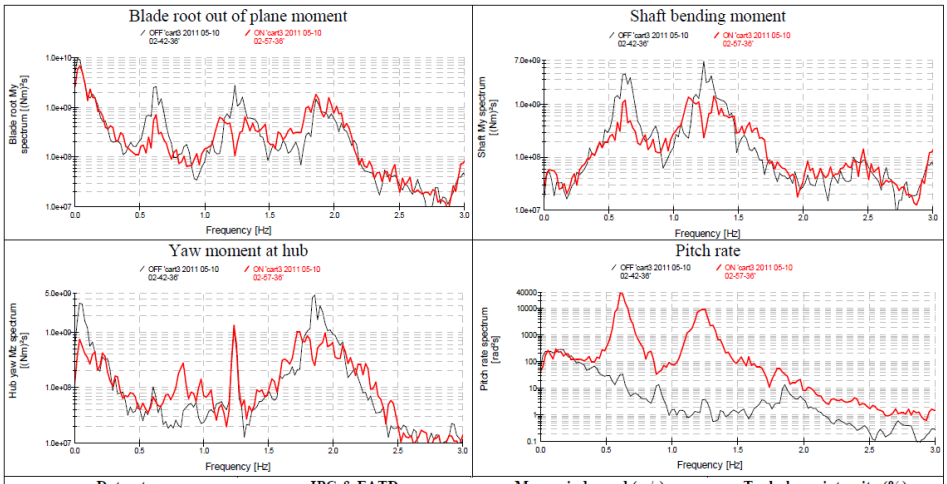
As a first step to its implementation on field turbines, IPC was further investigated experimentally at the NREL, located in the state of Colorado, USA. The prototype turbines on which this experiment was carried out, are called the CART-2, a two-bladed turbine with a lockable teeter hub, and the CART-3, a three-bladed turbine similar to modern commercial wind turbines. The CART turbines are both of 42 m diameter, and have a rated power of 660 kW.

Prior to the actual field testing, the CART-2 turbine was modelled in the software GH Bladed and the control algorithm was tested to estimate load reduction potential, Bossanyi and Wright (2009). The two-bladed turbine was run with a teeter hub lock on, to validate the necessity for a teeter hub.

It is seen that for the rotating components such as blade and hub, the stress peak at 1P is virtually eliminated. Similarly, on the non-rotating components such as the yaw bearing, the steady-state and 2P peaks are eliminated, keeping in mind that this is a two-bladed turbine. As can be expected, this requires individual pitch control action, mainly at a frequency of 1P. It is proved via simulations that the stress reduction attained by implementing IPC is so high as to eliminate the need for a teetered hub. It is seen that the power output in the above-rated region is not affected by IPC, and in fact the power output improves around the region of the knee, Bossanyi et al. (2012b). On the other hand, if IPC is used in the below-rated region, then reduction in power output is seen. However, since the loads in the below-rated region of the power curve are comparatively lower, it is recommended to bypass pitch control entirely in this region.

Field testing with the CART-2 turbine validates the simulation results, and stress frequency peaks at 1P on rotating components are indeed seen to be eliminated, Bossanyi (2005), Bossanyi et al. (2010). An analogous controller was designed for a three-bladed turbine; a model of the CART-3 turbine in GH Bladed, and similar simulation results were observed. That is to say, with 1P and 2P IPC, it was possible to eliminate cyclic loads with frequencies 1P and 2P on rotating components. At the same time, the steady-state and 3P loads on non-rotating components of the 3-bladed turbine could be reduced with individual pitch control, Bossanyi et al. (2010).

The simulation findings for three-bladed turbines are validated in general terms by



Dataset	IPC & FATD	Mean wind speed (m/s)	Turbulence intensity (%)
CART3 2011 05-10 02-42-36	OFF	12.13	10.82%
CART3 2011 05-10 02-57-36	ON	12.15	10.19%

Figure 6.5: Load spectra at wind speed 12 m/s, Bossanyi et al. (2012a)

field testing of the CART-3 turbine, as described in Bossanyi et al. (2012a). It is seen that the individual pitch PI controller is robust to parameter changes and gives very positive results for load reductions at 1P and 3P frequencies. The reduction in the load peaks can be seen in Fig. 6.5. Converting the spectra into the dynamic equivalent loads for fatigue life calculation for each wind bin, the reduction in equivalent blade loads as a result of the implementation of IPC can be seen in Fig. 6.6. There remain a few modifications that need to be made to compensate the physical limitations of the sensors and actuators. For instance, blade root strain gauges are used for measuring flapwise and edgewise bending loads which form the input to the individual pitch controller. The DC-level drift observed with strain gauges cannot be directly corrected via filtering, and hence it has to be calculated from first principles, starting from a good estimate of the wind speed. Further, it was found that the 1P loading is reduced only partially by individual pitch control. It is suggested that the reason for this is that the 1P loading is caused partially by mass imbalance, and partially by aerodynamic effects: it is further suggested that cyclic loading due to mass imbalance has not been reduced by using individual pitch control.

From the field testing of a 3-bladed turbine, it is concluded that individual pitch control can achieve fatigue load reductions in the above-rated region of 20-25% at the blade root, 30-35% for shaft bending and 16-22% for stationary components, Bossanyi et al. (2012a). At the same time, the pitch activity above-rated increases by a factor of around 4-5 times with IPC. Some further extensions to this basic concept of IPC are found in the literature, that make use of additional measurements to increase the capabilities of this load control strategy.

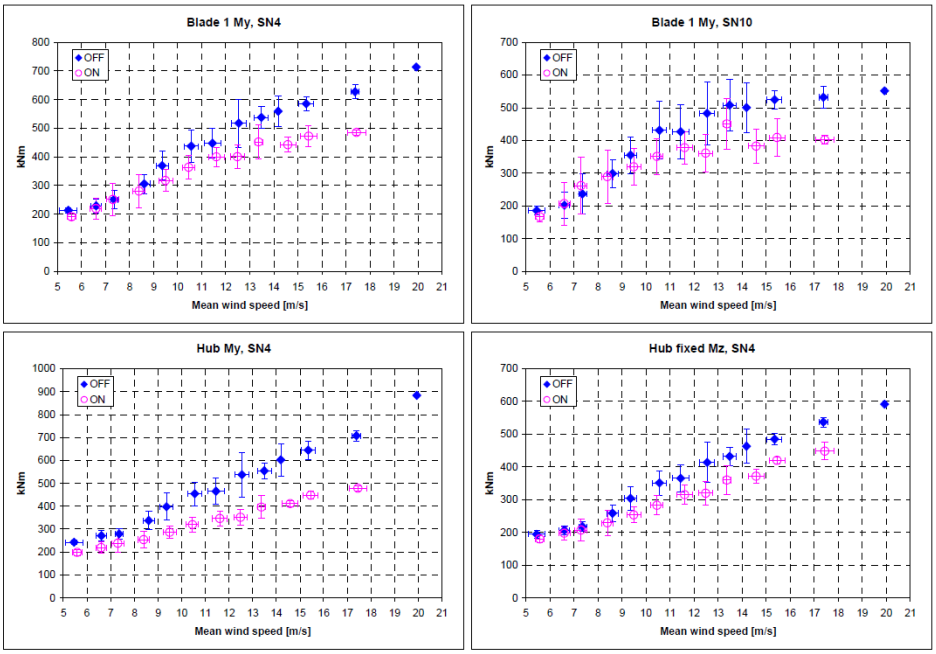


Figure 6.6: Reduction in blade loads by IPC, Bossanyi et al. (2012a)

6.3.5. IPC: EXTENSIONS

There are a few other approaches that have been explored with respect to IPC implementation. In Larsen et al. (2005), local blade flow measurements obtained from pitot tubes or other flow measuring devices, mounted on the blades, are used for reducing periodic loads. The pressure or flow measurements are used to calculate the instantaneous angle of attack and relative velocity at the blades. Individual pitch actions are calculated such that the instantaneous values of these flow properties deviate as minimally as possible from the average values over all the blades, with a correction for yaw error. A simple controller can be designed here since it is not necessary to use coördinate transforms to bring the signals to a stationary frame of reference. Flow measurements lead to a faster response as compared to load measurements, however high-frequency pitch actions are generally not deemed desirable. Also, flow measurements are usually done at discrete points, and they may be sensitive to very localised wind speed variations, the effect of which gets averaged out by load measurements. It is shown in the reference, and by Markou et al. (2011), via simulations, that this IPC technique shows results comparable to those of 1P IPC, and performs even better in case of a turbine in the wake of another turbine, as in a wind farm. However, the load reductions have not been compared to those achievable using higher-harmonic IPC.

An analogous approach to blade root moment measurement for IPC is blade tip deflection measurement, using an accelerometer. This has been investigated by Wright and Balas (2004), and found to be effective. However, the use of blade tip accelerometers

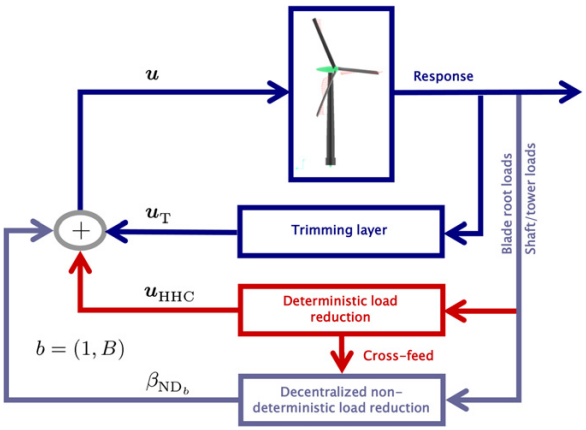


Figure 6.7: Multi-layer control of turbine loads, Bottasso et al. (2013)

is not considered optimal from a reliability and maintenance perspective.

In Bottasso et al. (2013), blade loads are demodulated and the corresponding control actions are remodulated independently of each other. It is shown that this approach, as opposed to the MBC transform approach, focusses more on reducing blade loads at the expense of reduced load alleviation on stationary components. Also, the reduction of other stochastic loads over and above the periodic loads is achieved here by the control methodology shown in Fig. 6.7. Here, the innermost loop corresponds to baseline power control with torque control in the below-rated and collective pitch control in the above-rated regions. The middle layer corresponds to reduction of periodic loads, using individual pitch. The outer layer attenuates the remaining load fluctuations that are stochastic in nature.

The possibility of controlling the pitch angle of the inboard part of the blade independently of the outboard part of the blade has also been investigated by Bossanyi et al. (2012b). It is seen that the added degree of freedom leads to at least 15% blade load reduction. However, this needs to be weighed against additional costs and reliability issues with a dual pitch mechanism. Similarly, in this reference, mention has been made of distributed load control concepts which use additional spanwise actuators to modify flow patterns around the blade with a view towards load reduction; this is currently an important topic under research.

It has been shown by Laks et al. (2011) that, if the wind speed signal were available, then it could be fed forward in parallel with the blade load feedback controller in order to increase load reduction capabilities. It has also been shown that blade local wind speed signals give superior performance than a rotor-averaged wind speed signal. However, inaccuracies in measurement can also degrade performance as compared to direct feedback IPC. It has also been shown that using a stable model-inverse feedforward controller with IPC in feedback shows increased load reduction Dunne et al. (2011). Feedforward also has the advantage of tighter power and speed control.

The wind speed signals for feedforward may either be estimated or directly measured. In Namik and Stol (2011), an estimated wind speed disturbance has been shown

to achieve better speed and power regulation for wind-dominated loading cases. Wind speed estimation and feed-forward in tandem with IPC with the MBC transform has been investigated in Selvam et al. (2009), in order to enhance load reduction potential. Direct measurements of wind speed using turbine-mounted sonic or cup anemometers are not useable by the controller due to poor dynamic resolution and the perturbing effect of the wind turbine rotor on the wind field.

It is interesting to note that the use of LIDAR for load mitigation in wind turbines also forms a related and important issue in current research. LIDAR instrumentation is capable of measuring the incoming wind velocity field simultaneously at different spatial points in terms of both magnitude and direction, leading to a vast improvement in wind speed estimation over the current industry anemometry standard. These estimates can be used as integrated feedforward signals to the current IPC controller, leading to enhanced load reduction capabilities.

The major shortcomings of the conventional IPC methodology described above are that the decoupling errors increase as turbines become larger and more flexible, and tricks like azimuth offset are required to enable SISO control, Bossanyi (2005). Also, there is little control over the frequency content of the commanded input signal. Further, considerable tuning efforts are required to ensure that the closed-loop system performs optimal rejection of disturbances with an adequate stability margin. These shortcomings will be addressed in the sequel using the SPRC and IFT-LPV methodologies developed in the first part of the thesis.

Individual pitch control, while effective in reducing blade root loads, cannot address the issue of localised variations in blade aerodynamic loads. Further, due to the massive pitch inertia of the blades, pitch control is typically used only for attenuation of the low-frequency components of the blade load spectrum. In order to achieve high-frequency localised aerodynamic control, additional actuators and instrumentation is required, motivating the development of the 'smart' rotor, delineated in the next section.

6.4. THE SMART ROTOR

Per definition, the smart rotor is a modern wind turbine rotor instrumented with actuators and sensors along the blade length, operating in closed loop with an advanced controller, such that the performance of the wind turbine is optimised, and the turbine lifetime loads are minimised. While generic sensors and control approaches can be tailored with relative ease for the development of the smart rotor, the actuators needed for local flow control require a greater amount of attention, in terms of both ensuring high control authority, as well as modularity and reliability. The actuators designed for the smart rotor have been reviewed in the literature by Barlas and Van Kuik (2010) and Bernhammer et al. (2012). These references describe in detail the various approaches taken towards the implemenation of such a rotor and their respective stages of maturity. A few of these approaches are described below:

- Trailing edge flaps/Morphing aerofoils: Trailing edge flaps refer to a deformable blade trailing edge that can be actuated to manipulate the pressure distribution over the aerofoil. Morphing aerofoil, on the other hand, is the terminology used

for the case where the chordwise shape or camberline of the aerofoil can be actively modified to achieve the same effect. The line of distinction between the two concepts is unclear, typically 'flaps' are used to refer to a design where the aerofoil deforms in a discrete manner, such that a clear distinction is observed between the static and the moving part of the aerofoil. Morphing aerofoils, on the other hand, change shape in a more continuous manner. Both approaches have their advantages and disadvantages, and show similar control authority. In this thesis, the two terms will be used interchangeably, and they will be discussed in detail in the following sections.

- Microtabs: Microtabs are small tabs that deploy perpendicular to the chord of the blade cross-section, Van Dam et al. (2007). These devices, on account of their low inertia and the use of smart materials, show high bandwidth capable of addressing high frequency loads in the spectrum. However, they typically only have on-off behaviour, and hence the control authority is limited. Further, microtabs show non-minimum phase behaviour, which limits their closed-loop performance, at the frequency 2P and higher.

- Synthetic jets and plasma actuators: These devices modify the boundary layer airflow in active manner. Synthetic jets use vibrating surfaces to extract and add momentum to the air flow around the blade, thereby manipulating the pressure distribution, Glezer and Amitay (2002). Plasma actuators, on the other hand, use high voltage discharge to ionise the air and thereby exert force on it, Corke et al. (2007). These methods are specifically useful for delaying boundary layer separation, and hence postponing the effect of stall. However, the overall effect in normal turbine operation is limited. Further, the highly nonlinear effect of these actuators makes overall actuator control design challenging.

- Active twist: The objective of an active twist actuator is to produce torsion along (a part of) the span of the blade such that aerodynamic loading forces are counteracted. Such an actuator typically comprises of smart material fibres embedded in the blade composite matrix such that they deform in response to an actuating signal, producing blade twist. Of the smart rotor concepts seen thus far, this concept demands the largest amount of power per unit deformation produced, and shows relatively low control authority. Rather than forming an add-on to existing blades, the active twist actuator demands a revision of the entire blade design paradigm. Such actively twisting blades have been considered in the helicopter industry, Chopra (2002), and developed for wind turbine blades, Lachenal et al. (2013).

The various smart rotor concepts described above have been compared in terms of their control authority in Fig. 6.8. It can be seen that trailing edge flaps and morphing aerofoils show the highest average ability to modify the blade cross-sectional lift properties, thus showing the highest control authority. Microtabs show a reasonable capacity of lift modification, while the boundary layer concepts and active twist control have the smallest effect on average lift manipulation. Further, the effect of the flaps on blade loads is approximately linear at low frequencies, easing controller development. These

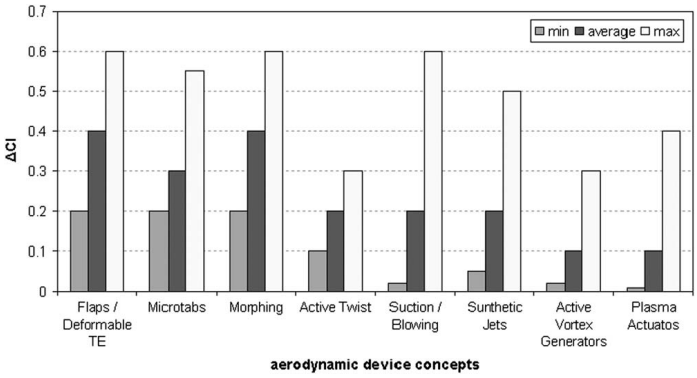


Figure 6.8: Comparison of smart rotor concepts in terms of control authority, Barlas and Van Kuik (2010)

attractive characteristics imply that a significant amount of research effort has been invested in exploring the characteristics of trailing edge flaps, and this concept possesses the highest level of technological maturity. The next sections will be devoted to exploring the most significant results of the studies in the use of flaps for smart rotors. First, a survey is made of the different designs of morphing aerofoils. Then, the load reductions achievable by using trailing edge flaps on wind turbines reported from simulation studies are discussed. Finally, the implementation on scaled wind turbines in wind tunnel tests and in the field is examined.

6.4.1. MORPHING AEROFOIL DESIGNS

A number of designs for morphing aerofoils stem from the aircraft industry, Barbarino et al. (2011), which has a much longer history of investigating variable shape deformable wings. However, the objective for morphing aircraft is different; research is focussed on increasing the efficiency under varying flight conditions, and on increased controllability and manœuvrability. Aircraft aerofoils differ markedly in that they operate under far higher speeds than wind turbine aerofoils, upto and exceeding transonic speeds, and they are per design much thinner than wind turbine aerofoils. However, it is expected as a first approximation that the lessons learnt from morphing aircraft can be transferred to smart wind turbine rotors.

A comparison has been made between a discrete trailing-edge flap and a smooth, continuous morphing aerofoil (the 'FishBone Active Camber' concept) by Woods et al. (2014). The two different approaches to a morphing blade cross-section can be visualised in Fig. 6.9. From wind tunnel tests, it is shown that the continuously morphing concept and the discrete trailing-edge flaps show a similar increase in the lift coefficient, reaching a maximum of 0.72. On the other hand, the discrete flap increases the drag to a much larger extent than the continuous FishBAC concept.

While morphing aerofoils, like the one above, have been designed using conventional actuators, such as servomotors, it is often found in practice that smart materials

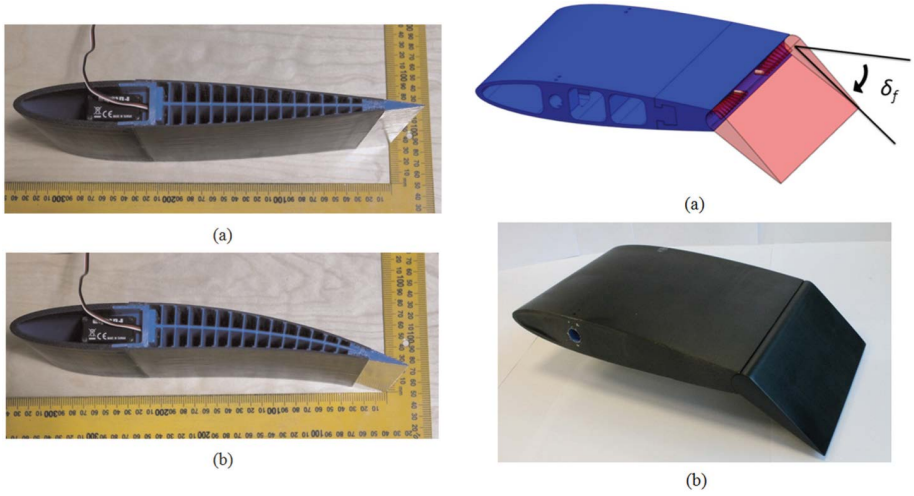


Figure 6.9: On the left, the continuous morphing aerofoil, the 'FishBAC'. On the right, a conventional discrete trailing-edge flap, Woods et al. (2014)

may prove a better option. One of the main advantages of an actuator made of smart materials is the increased power-to-weight ratio that simplifies its integration into the existing wind turbine blade. Another advantage is the high bandwidth typically associated with such actuators, that can help address the high frequency components of the blade load spectrum arising out of atmospheric turbulence. One such implementation of a smart material-based actuator is the use of Shape Memory Alloy (SMA) actuators for the application of the morphing aerofoil has been explored in Bil et al. (2013). These smart actuators are easy to manufacture and show suitable stroke length and load handling capability. However, the response of SMA actuators is non-linear due to asymmetric expansion and contraction properties, and they show relatively low bandwidth, Lara Quintanilla et al. (2013). Further, it is considered that SMA actuators have a limited fatigue life and may not be directly suitable for wind turbine load reduction, an application that demands several million fatigue cycles over the turbine lifetime.

Bilgen et al. (2010) make use of piezoelectrics in the form of Macro-Fibre Composite (MFC) actuators, that form a morphing trailing edge flap as seen in Fig. 6.10. With a large chordwise size, wind tunnel testing showed a maximal peak-peak change in the lift coefficient of 1.54. Modifications of this design such that the MFC's are integrated with a flexible aerofoil skin demonstrate in the wind tunnel that this control authority is retained at Reynold's numbers around 250,000, Bilgen and Friswell (2013). A similar approach by Debiasi et al. (2013) yields a lift coefficient change of upto 1; both camber and thickness control is possible with their morphing aerofoil design, Fig. 6.11.

A direct implementation of a piezoelectric flap, embedded in a soft foam matrix, was evaluated by Van Wingerden et al. (2008) for the purpose of load alleviation in a non-rotating wind turbine blade using wind tunnel experiments. A photograph of the blade and a cross-section at the flap location can be seen in Fig. 6.12. With artificial distur-

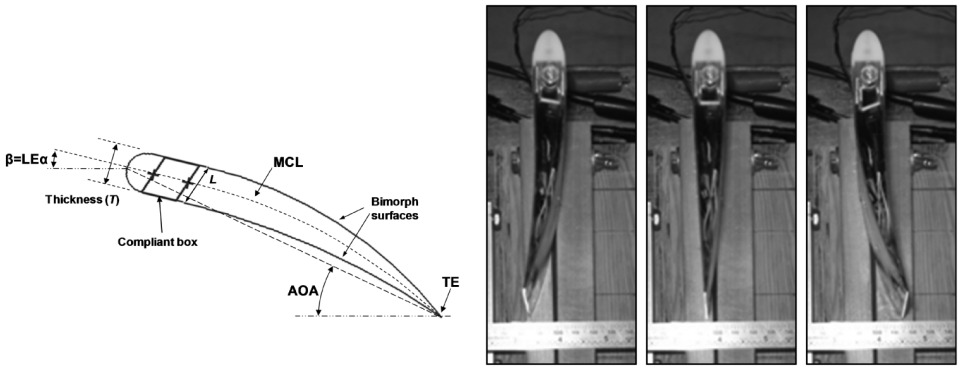


Figure 6.10: Morphing trailing edge of aerofoil, based on Macro-Fibre Composites, Bilgen et al. (2010)

Figure 6.11: Thickness and camber control of aerofoil, based on Macro-Fibre Composites, Debiasi et al. (2013)

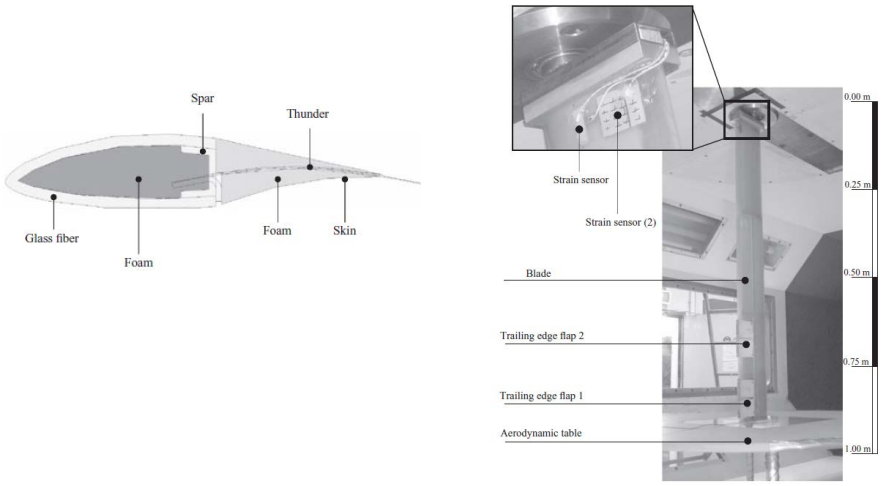


Figure 6.12: Load reduction with trailing edge flaps, non-rotating experiment, Van Wingerden et al. (2008)

bances induced on the blade by using one of the flaps as a disturbance generator, it was found that upto 90% of the loads at the first structural eigenfrequency could be reduced using this concept. Perfect load rejection was not possible due to the limited control authority of the trailing edge flaps.

In general, using direct trailing-edge flaps, it is seen that piezoelectric actuators can deliver high bandwidth but low stroke, adversely affecting the control authority of the flaps. In an effort to counter this limitation, the concept of the free-floating flap was developed by Heinze and Karpel (2006). Such a flap, as shown in Fig. 6.13 is free to rotate about its hinge axis, and is actuated by a small (piezoelectric) tab located at its trailing edge. The objective of such a design is that the low stroke of the tab can be aerodynamically amplified such that a large angular rotation of the flap is obtained. The concept of the free-floating flap has been investigated in a wind tunnel environment on a non-rotating experiment by Bernhammer et al. (2013), but it has not yet been demonstrated on a (scaled) wind turbine experimentally. The non-rotating study shows that such a flap can be completely autonomous in terms of the requirements of energy consumption, and can be fitted to a wind turbine blade as a plug-and-play device. This modularity simplifies operational maintenance requirements of the actuator, and thus enhances practical viability. However, it should be noted that the additional degree of freedom in the free-floating flap can couple aeroelastically with the flexible blade modes, leading to the manifestation of the unstable phenomenon of flutter are relatively low wind speeds. As such, this morphing aerofoil design demands extra attention from the control engineer, to ensure operational stability.

The concept of the trailing edge flaps, of different designs, have been validated in the simulation environment as well as in the field.

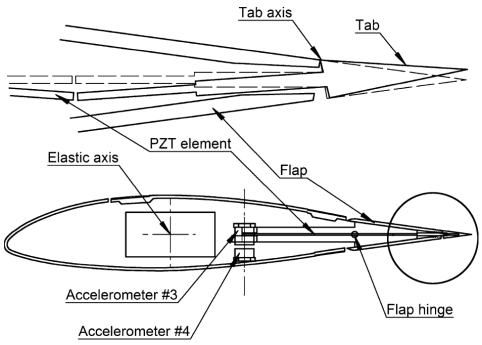


Figure 6.13: The concept of the free floating flap, Heinze and Karpel (2006)

6.4.2. FLAP CONTROL IN THE SIMULATION ENVIRONMENT

The earliest investigations into flap control for wind turbines were conducted in the simulation environment, in order to evaluate the load reduction potential of the trailing edge flaps. In numerical simulations, it is often not necessary to design the flap actuation mechanisms accurately, it often suffices to define the change in the quasi-steady aerodynamic characteristics of the blade as a function of flap deflection. As such, the results in the literature obtained in the simulation environment are often generic, and can apply to a number of different practical implementations of trailing edge flaps.

This section discusses the numerical investigations of wind turbine flap control, noting and comparing the different simulation environments and control approaches chosen.

- GH Bladed: The implementation of trailing edge flaps has been compared with IPC by Lackner and Van Kuik (2009), using GH Bladed. The flaps have a spanwise size of 20%, and a chordwise size of 10%; their aerodynamic influence is approximated by updating the quasi-steady aerofoil polars for the corresponding deflection angles. Simulations show that IPC achieves stronger reduction of 1P loads, especially at high wind speeds, as compared to flap control. However, flaps are highly effective at higher frequencies and low wind speeds, especially in the below-rated region. Further, flap control in tandem with pitch control has the capability to reduce power fluctuations and demands on pitch activity. This study uses the MBC transform also for controlling the flaps.

- FAST: Berg et al. (2009) investigate the use of trailing edge flaps using NREL FAST, which is extended by using modified lift and drag polars for different aerofoil configurations. This study compared discrete trailing edge flaps with morphing flaps that do not show discontinuities under deflection. With similar angles of deflection, it was seen that the load reduction potential of both varieties of trailing edge flaps was comparable. A simple PD controller was used to investigate flap-based load control.

- HAWC2: The multi-body wind turbine simulator HAWC2 from DTU, Henriksen et al. (2015), extended by including an unsteady potential-flow model for describing trailing edge flap dynamics, was used by Andersen et al. (2010) for optimising the location of a generic trailing edge flap, and for studying load reduction potential. The optimisation suggests that a flap sized 10% of the blade length should be located outboard along the blade span, near the belly of the spanwise lift curve. When a flexible rotor is modelled in place of a stiff one, the optimal location moves slightly inboard. Fatigue load reductions of 25% are obtained at rated wind speed operation for moderate flap activity with around 5° standard deviation. This work uses a simple proportional controller that acts on the measured blade strain. Finally, it is seen that increasing the number of flaps along the blade span yields diminishing returns in terms of the amount of load alleviated. Going further, Markou et al. (2011) show that, when the turbine is operating in partial wake conditions, load reductions upto 73% are achievable using trailing edge flaps.

- DU-SWAT: This aeroelastic code was developed in the Delft University of Technology, specifically for evaluating the behaviour of 'smart' flexible wind turbine rotors. It uses a multi-body formulation for structural dynamics, coupled with an unsteady aerodynamic model, including an unsteady trailing-edge flap model. This code was used to evaluate the fatigue and extreme behaviour of smart rotors by Bernhammer et al. (2016). Blade flapwise fatigue load reductions of upto 23.8% were reported, however it was also seen that blade torsional loads increased by 14%. Extreme blade loads were in general reduced by 6-8% in this analysis.

Flap control has also been investigated on scaled wind turbine prototypes.

6.4.3. FLAP CONTROL: EXPERIMENTAL INVESTIGATIONS

As from the previous sections, the numerical analysis of trailing edge flaps for wind turbine load alleviation has been given considerable attention in the literature. On the other hand, experimental investigations for wind turbine load control are somewhat limited in number. It should, however, be noted that the analogous concept of trailing-edge flaps for load control of rotorcraft blades has been demonstrated successfully using full-scale helicopter flight testing, King et al. (2014). For wind turbines, it is difficult to evaluate the effect of the trailing edge flaps on a small scale in the wind tunnel, since the Reynold's numbers encountered on utility-scale turbines are usually a few orders of magnitude higher than those achievable at small scale. Especially for local flow control at high frequencies, this discrepancy can fundamentally alter the flow phenomena that the trailing edge flap is expected to manipulate. However, in order to attain some degree of familiarity with the use of these advanced actuators before full-scale implementation, it is deemed necessary to perform scaled testing in the wind tunnel under controlled conditions.

The concept of the trailing edge flap for wind turbine load reduction has been demonstrated on a rotating scaled wind turbine of rotor diameter 2 m in the Open Jet Facility wind tunnel at the Delft University of Technology. The design of the trailing edge flap has been shown in Fig. 6.12, thus it is actuated by the means of piezoelectric actuators. Two trailing edge flaps of this design were integrated with the design of a

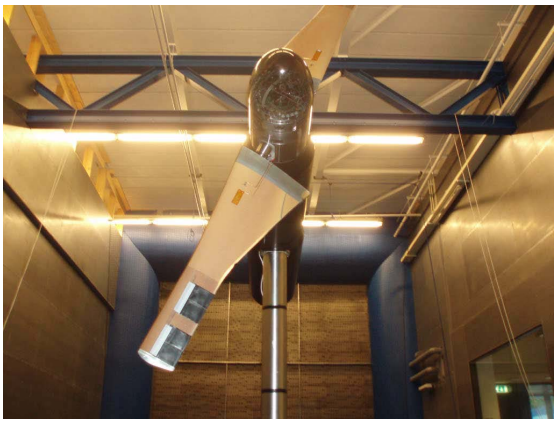


Figure 6.14: Rotating experiments with trailing-edge flaps, Van Wingerden et al. (2010b)

scaled wind turbine blade; the design of the scaled wind turbine has been described by Hulskamp et al. (2011), and the setup can be visualised in Fig. 6.14. Van Wingerden et al. (2010b) discuss the identification and control of this experimental turbine. Feedback control is achieved using an H_∞ design, while a model-based repetitive control strategy is used to further augment the performance of the controller. It is shown that despite the low stroke of the actuators, deterministic loads (tower shadow and yaw misalignment) can be reduced by nearly 90%. Comparison of the identified model with aeroelastic analysis shows that several aspects of flap aerodynamics are not yet modelled with adequate fidelity for robust prediction of flap performance, Barlas et al. (2013).

The first in-field implementation of the 'smart' rotor equipped with trailing-edge flaps was demonstrated by Berg et al. (2013) at the Sandia National Laboratories in the USA. Conventional actuators were used to control the flap degree of freedom, and this rotor was used to validate numerical models and demonstrate the control authority of these actuators. The first field demonstration of the load reduction potential of trailing edge flaps was carried out by Castaignet et al. (2014). Here, a Vestas V27 turbine, of 27 m rotor diameter, was retrofitted with active trailing edge flaps, of 5% of the blade span. A Model Predictive Control (MPC) strategy was used for the purpose of load control; frequency weighting was done of the MPC performance signal to target load alleviation at the 1P frequency. Conventional actuators were also used here; higher frequencies in the load spectrum were not targetted. A average load reduction of 14% was observed, primarily at the 1P peak. Comparing the modelling results from Flex5 with the experimental system identification results indicated a flap efficiency of 20%, however the authors do not comment on the source of this inefficiency.

Thus, it can be concluded that, while the load reduction potential of trailing edge flaps is clear, a significant gap still exists between our theoretical understanding of the dynamic effect of the flaps on wind turbine load performance, and the experimentally measured data.

6.5. CONCLUSIONS

For the purpose of converting wind energy into electrical power that can be fed directly into the grid, wind turbines today use sophisticated actuation and sensing mechanisms. The control objective of maximum power generation, capped at rated power, is achieved by a combination of electronic torque control, and blade pitch control, which together regulates the wind turbine speed along the appropriate reference trajectory. Such an approach, common to all commercial variable-speed variable-pitch, is defined as baseline control for the wind turbine. All advanced control approaches discussed in this thesis use the nominal baseline-controlled wind turbine as the plant, and are connected in an outer closed (or open) loop around this baseline plant.

Torque and collective pitch control have also been used, with a certain measure of success, to reduce structural loads occurring at modal frequencies, examples being the tower fore-aft motion damping, and drive train modal damping. However, these symmetric control degrees of freedom cannot affect the asymmetric exogenous periodic loading occurring at the rotor speed 1P and its harmonics, which dominate the dynamic lifetime loading of a turbine, and hence its fatigue life.

Individual Pitch Control (IPC), the asymmetric pitching of turbine blades, has been proposed in the literature to reduce these loads. The approach using the MBC transform is the most widely used IPC approach, and shows considerable load reduction; it has been validated both in the simulation environment as well as in the field. However, it often demands manual tuning of PI controller gains, which could lead to suboptimality or instability. Further, IPC requires high pitch actuator duty which could adversely affect the longevity of the pitch mechanism, especially if high frequency pitch motion is demanded by the simple PI controller.

Another limitation of IPC is that it is unable to address local blade loads occurring at high frequencies. To address such loads, the concept of the smart rotor has been investigated, which consists of rotor blades instrumented with local flow modifying actuators and load or flow measurement sensors. Different actuators have been investigated in the literature; it is found that trailing edge flaps have the highest technological maturity for the application. Several alternative designs of such flaps have been synthesised, with piezoelectrics providing low stroke but high bandwidth, and vice-versa for conventional actuators. The load alleviation potential of this approach has been proven numerically as well as experimentally. It is also found that further research is required into the unsteady aerodynamic behaviour of the flaps in order to bring numerical predictions closer to reality. As the state of the art stands today, considerable uncertainty remains in the aeroelastic prediction of the dynamic behaviour of the smart rotor.

7

AEROSERVOELASTIC SIMULATIONS

How harmful overspecialization is. It cuts knowledge at a million points; and leaves it bleeding.

Hari Seldon in 'Prelude to Foundation', authored by Isaac Asimov, the USA (1988), before travelling the world to synthesise a first-principles model of societal behaviour.

This chapter uses the aeroelastic simulation tool GH Bladed to explore the application of the theoretical concepts developed in the first part of the thesis on a high-fidelity non-linear numerical model of a modern wind turbine. First, the methodology of Subspace Predictive Repetitive Control (SPRC) will be used for the data-driven implementation of Individual Pitch Control (IPC) for load control. The simulation model will be extended with trailing-edge flaps, and the use of SPRC will be demonstrated for the case of these new actuators. Then, Iterative Feedback Tuning (IFT) will be explored for the purpose of turbine load control, in order to synthesise a global LPV controller for time-varying wind conditions. Finally, the concept of IPC for yaw control will be validated in the simulation environment, in order to introduce a new design variable for the trade-off of design loads in wind turbines. The chapter ends with conclusions drawn from the results of the aeroelastic simulations.

7.1. INTRODUCTION

As seen in the previous chapter, the load control of wind turbines is a periodic load attenuation problem, where the asymmetric blade loads show an exogenous forcing peak in the frequency spectrum at the rotor speed, 1P and its harmonics. Both Individual Pitch Control (IPC) and trailing-edge flap control can be used to reduce these loads. The control strategy conventionally used for load control involves a linearising transformation,

Parts of this chapter have been published in Mechatronics **24**, Navalkar et al. (2014), the Proceedings of the American Control Conference, Portland, USA, Navalkar et al. (2014c), the Proceedings of the IFAC World Congress, Cape Town, South Africa, Navalkar et al. (2014a), the Journal of Physics **524**, Navalkar et al. (2014b) and the Proceedings of the IFAC Workshop on Linear Parameter Varying Systems, Grenoble, France, Navalkar and Van Wingerden (2015).

called the MBC transform, that is required to be used for each frequency peak target. This transformation decouples the multivariable system into Single-Input Single-Output (SISO) systems, that can be controlled individually by using a hand-tuned PI controller.

For higher frequencies, distributed actuators, and increased system flexibility, the MBC transform is no longer capable of perfect decoupling, and hence it becomes once more necessary to treat the control problem in a fully multivariable manner. Further, it should be noted that the aerodynamic control authority of the actuators is difficult to model; it is also time-varying and changes with the ambient wind speed. In such a situation, where large system uncertainty and time-varying behaviour is observed, the concept of iterative data-driven control, explored in the first part of the thesis, may be considered interesting.

Another point of note is that the application of load control for wind turbines is highly sensitive to the actuator duty demanded by the controller. Both for IPC as well as trailing-edge flaps, it is desirable to constrain the smoothness of the actuator signal to a sufficient extent so as to extend the life of these actuators, and accordingly, limit the operational maintenance costs. As such, the use of basis functions for precise control over the shape and smoothness of the control signals may be considered to be an attractive component of the postulated control approaches.

Subspace Predictive Repetitive Control (SPRC), the theory of which is developed in Chapter 4, is considered a viable alternative to conventional IPC and flap control for these reasons: it is able to adjust the control law based on the estimated dynamics of the underlying plant, and hence does not require extensive system modelling. Further, since SPRC is formulated in the basis function subspace, it can provide a high degree of control over the smoothness of the actuator duty cycle. As the turbine dynamics change, as a result of a change in environmental parameters or due to aging, SPRC is able to adjust the control law optimally to these changes.

SPRC requires continuous reïdentification over the lifetime of its operation for a time-varying plant. On the other hand, for the special case of an Linear Parameter-Varying (LPV) plant, IFT-LPV, described in Chapter 5, can be used directly to tune the gains of a fixed-structure LPV controller such that they are (locally) optimal over the entire range of the scheduling sequence. Thus, for a wind turbine, the dynamics of which can be considered to be LPV such that the wind speed forms the scheduling variable, IFT-LPV can be used to tune a controller that is valid for all realisations of the wind speed signal. With this approach, IFT tuning can be done for a fixed period of time until the load reduction performance is adequate, and the controller parameters can then be frozen. This approach can also be extended to include basis functions, as done for SPRC.

In general, IPC and flap control are optimised to minimise blade loads, however these optima do not necessarily correspond to a case where the support structure loads are also minimised, Bottasso et al. (2013). The yaw degree of freedom forms the link, Stol (2003), between blade loads and support structure loads, however the yaw controller is typically designed independently of the IPC and flap controllers. Yaw control can be integrated with the conventional approach to IPC such that the joint minimisation of blade and support structure can be performed.

These concepts are validated using a numerical model of a utility-scale wind turbine

Table 7.1: XEMC-Darwind XD115 Wind Turbine specifications, XEMC-Darwind (2013)

Description	Symbol	Value
Rated power	P_{rated}	5000 kW
Rotor diameter	d_{ro}	115 m
Cut-in wind speed	v_{cutin}	4 m/s
Rated wind speed	v_{rated}	12 m/s
Cut-out wind speed	v_{cutout}	25 m/s
Rated rotational rotor speed	Ω_{ro}	18 rpm
Gearbox ratio	ν	1.0 [Direct-Drive]
Pitch-rate limit	$\dot{\theta}_{\text{limit}}$	6°/s

in the simulation environment GH Bladed. First, SPRC will be demonstrated using IPC and flap control. Next, the concept of IFT is investigated in this environment. Finally, the extension of IPC to yaw control is explored, and conclusions are drawn from the simulation results.

7.2. SPRC FOR WIND TURBINE PITCH CONTROL

In this section, the two-step data-driven control approach Subspace Predictive Repetitive Control (SPRC) will be used for IPC of a commercial wind turbine, modelled using the software GH Bladed, version 4.0. As described in the previous chapter, this software is used for load analysis and controller certification in the wind industry. It uses multi-body dynamics for structural modelling, and simulations are performed in this section considering the blades and the tower to be flexible bodies. Bladed uses a corrected Blade-Element-Momentum (BEM) theory for describing the interaction with a fully turbulent realisation of a wind field. Standard PI controllers can be defined for controlling the turbine; however it is also possible to use an externally compiled dll for defining the behaviour of the controller. This latter option is made use of, and the wind turbine controller, both baseline as well as IPC, are designed in a Matlab-Simulink environment and then compiled to a dll form that can be used with GH Bladed.

7.2.1. SIMULATION SETUP

The wind turbine modelled in this section is the commercial XEMC Darwind 5 MW machine, the characteristics of which are given in Table 7.1. The non-linear Bladed model has 69 states. Nominal operation, i. e., power production is achieved by using a baseline controller for torque and collective pitch control, following the guidelines set in the previous chapter. An SPRC controller is then designed to operate in an outer closed loop around this baseline-controlled plant such that it is able to minimise the asymmetric periodic loading of the rotor blades.

Typically, the certification of a wind turbine demands that the fatigue lifetime of the turbine, calculated as per the standards, IEC61400-1 (2005), exceed a value of 20 years. For the purpose of the calculations, an aeroelastic tool like GH Bladed is used to simulate the loading response of the wind turbine to turbulent inflow wind fields, for a number of different mean wind speeds. The typical duration of simulation is 600 seconds, and the

loading response is then extrapolated to obtain the total fatigue damage over 20 years.

In this section, simulations are carried out for a wind speed of 18 m/s. At this wind speed, the turbine operated in the above-rate mode, such that the commanded generator torque is at its rated value, and collective pitch control is active to ensure speed regulation. At this wind speed, it is expected that the IPC controller will be active. This wind speed corresponds to relatively high loading, and is expected to occur with a relatively high probability during the turbine lifetime.

Four wind field realisations, each with a different value of turbulence intensity, are simulated; turbulence intensity being defined as the ratio of the 10-minute wind speed standard deviation to the 10-minute mean wind speed. The four values simulated are 0%, 3.75%, 6% and 14%. While it is unlikely that turbulence levels of 0% will be observed in practice, this case enables the understanding of the behaviour of SPRC under ideal conditions. The higher levels of turbulence are more likely to occur in the field, and these simulations give more practically useful results for the implementation of SPRC.

From the load spectrum, it is apparent that a large part of the dynamic turbine loading occurs around the 1P and 2P peaks. As such, the basis functions used in the implementation of the SPRC control law are given by:

$$\phi_u = \phi_y = \begin{bmatrix} \sin(2\pi/N) & \cos(2\pi/N) & \sin(4\pi/N) & \cos(4\pi/N) \\ \sin(4\pi/N) & \cos(4\pi/N) & \sin(8\pi/N) & \cos(8\pi/N) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(2\pi) & \cos(2\pi) & \sin(4\pi) & \cos(4\pi) \end{bmatrix}^T . \tag{7.1}$$

Here, N is the length of the prediction horizon, taken to be equal to the inverse of the 1P frequency. The implementation of SPRC can be visualised in the block diagram, Fig. 7.1. Here, the loads measured by the strain gauges are stacked over the horizon N and cast into the basis function subspace by the projection matrix ϕ_y. Based on the control law adaptively synthesised at each time instant using the SPRC methodology from Chapter 4, the required pitch control signals are synthesised. It should be noted at this point that the basis functions are time-varying, and are locked to the phase or azimuth of the rotor. Thus, as opposed to conventional RC methodologies, this approach is entirely insensitive to variations in the rotor speed over time. This configuration is used to obtain SPRC implementation results in the next section.

7.2.2. SIMULATION RESULTS

In order to analyse the behaviour of SPRC, the model was simulated for different wind cases for a duration of 900 seconds each. For every simulation, the system Markov parameters were initialised to zero, that is to say, no prior knowledge was assumed regarding the true system dynamics. The implementation of SPRC proceeded as follows:

- For the first 40 periods, no IPC was active. The objective was to allow the baseline pitch and torque controllers to reach steady state operation without interference.
- For the next 60 periods, a filtered pseudo-random binary sequence, capped to within 3° of pitch activity, was superposed on top of the collective pitch demand. The purpose of this signal was to provide persistency of excitation, and ensure that

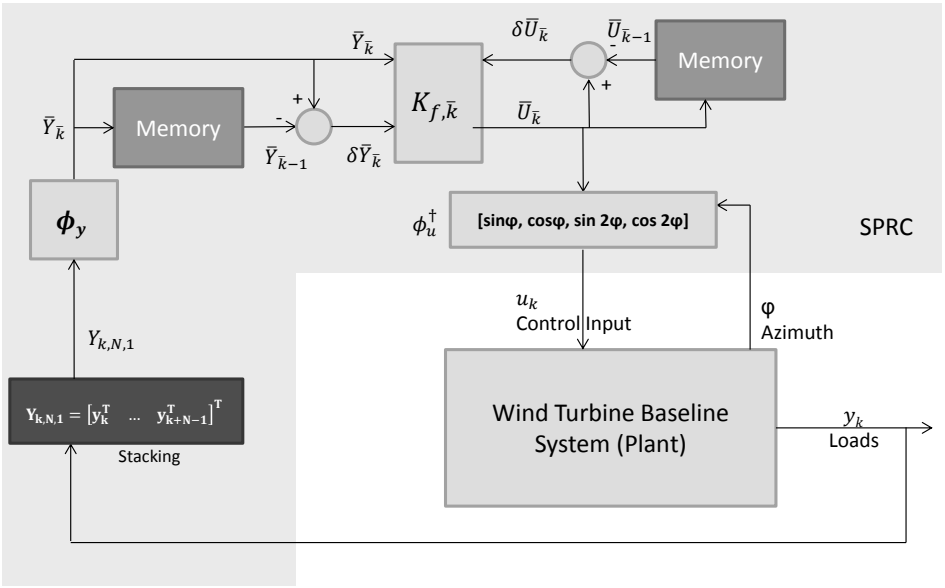


Figure 7.1: SPRC Implementation.

the identification step of the SPRC approach converges to a reasonably good estimate of the true system Markov parameters.

- At the end of this period, the SPRC feedback matrix $K_{f,\bar{k}}$ is synthesised from the identified Markov parameters, and the repetitive control law is implemented to achieve load reduction, in closed loop with the baseline-controlled plant.

In terms of time, this procedure implies that the turbine is considered to reach a steady state after 100 seconds of simulation time have passed, at which point identification is initiated. It is seen from the results that the identified parameter estimates require approximately 100 seconds to converge to relatively steady-state values. The identification is performed in this case in the time domain, using recursive PBSID without nuclear norm regularisation. The window size s is taken to be 20 samples, where the sampling frequency is 100 Hz. The distance between the true system Markov parameters, as obtained from offline batch identification, and the estimated Markov parameters, reduces over time, as can be visualised in Fig. 7.2. Identification starts after to 100 second-mark, and it is relatively smooth and rapid. At the end of 200 seconds, the variance accounted for (vaf) by the identification step of the two-step approach, exceeds 92%, at which point the Markov parameter estimates are considered adequately reliable for control law synthesis.

On the basis of these Markov parameter estimates, therefore, a repetitive control law is synthesised and implemented online. This control law iterates to a pseudo-feedforward sequence for minimising the deterministic periodic loads; the convergence to this sequence is also stable and rapid, as can be seen in Fig. 7.3. This figure plots the

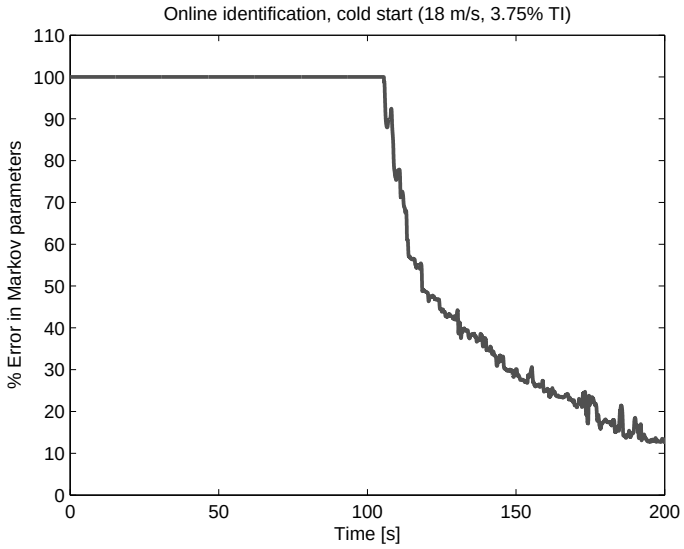


Figure 7.2: The error in online identification of Markov parameters, for a wind speed of 18 m/s and 3.75% turbulence intensity.

variation of the standard deviation of blade loads over time. Before repetitive control is activated, the blade loading is high, and shows variation due to the effect of the random signal used for identification, indicated by the square markers. After the SPRC control law is initiated, the error drops monotonically and rapidly to a low value, within 50 iterations. Since no turbulence intensity has been modelled in this case, the blade load variation drops by a factor of 17 after convergence; this value reduces as the level of turbulence rises.

The full table of results appears at the end of this section. First, a few selected results are presented graphically, to show the time- and frequency-domain behaviour of the designed controller. The following controllers are compared:

- Load simulations for the case where only the baseline torque and pitch controllers are in feedback with the wind turbine, and no IPC is implemented.
- Load simulations for the case where, alongwith the baseline controllers, IPC is implemented using the MBC approach for 1P and 2P load reduction, described in the previous chapter.
- Load simulations for the case where, alongwith the baseline controllers, IPC is implemented using SPRC with 1P and 2P phase-locked basis functions, described in Chapter 4.

Load reductions can be observed in Fig. 7.4 and Fig. 7.5. Fig. 7.4 shows the reduction in blade loads as a result of the use of IPC. It can be seen that the 1P and 2P loads are

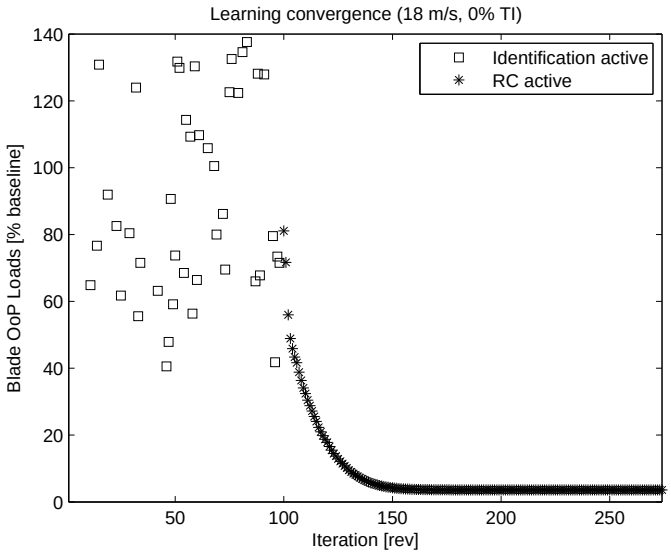


Figure 7.3: Repetitive Control convergence, for a wind speed of 18 m/s and 0% turbulence intensity.

eliminated by both IPC approaches. While the MBC approach applies to a broad band of frequencies, SPRC selectively targets the dominant peaks. Further, Fig. 7.5 shows the reduction in yaw bearing loads, where load reduction is observed at the 0P (DC-offset) and the 3P load peaks. The 1P peak observed arises out of mass/aerodynamic imbalance, and cannot be compensated for using the approaches described.

The control effort required by the two forms of IPC can be seen in Fig. 7.6 and Fig. 7.7, where the pitch angle time series and the pitch rate activity for Blade 1 have been depicted respectively. The low-frequency variation in the pitch activity is commanded by the collective pitch controller, which seeks to regulate generator speed to its rated value. Superposed on this are the IPC pitch demands, that appear as high-frequency, nearly sinusoidal variations. It can be seen that IPC-MBC demands greater pitch activity than SPRC. Further, the SPRC control signal shows a slight phase lead over the manually-tuned MBC control signal. From the frequency spectral comparison of Fig. 7.7, the advantage of using basis functions for SPRC becomes clear: the SPRC pitch activity is concentrated at the two frequencies 1P and 2P, corresponding to the basis function sinusoids. On the other hand, the MBC control signals show a much larger spread over the frequency spectrum, which could adversely affect actuator life.

All results are tabulated in Table 7.2, and they appear to follows the same trend. Similar levels of load attenuation are observed for both MBC and SPRC controllers, for different wind conditions. However, while pitch activity is nearly the same for 0% turbulence conditions, it increases much more modestly, as turbulence intensity rises, for the case of SPRC control than for MBC control.

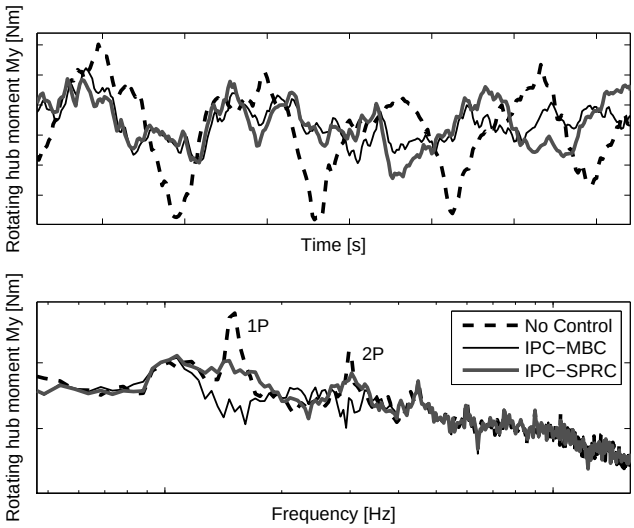


Figure 7.4: Blade out-of-plane bending moment, for a wind speed of 18 m/s and 3.75% turbulence intensity.

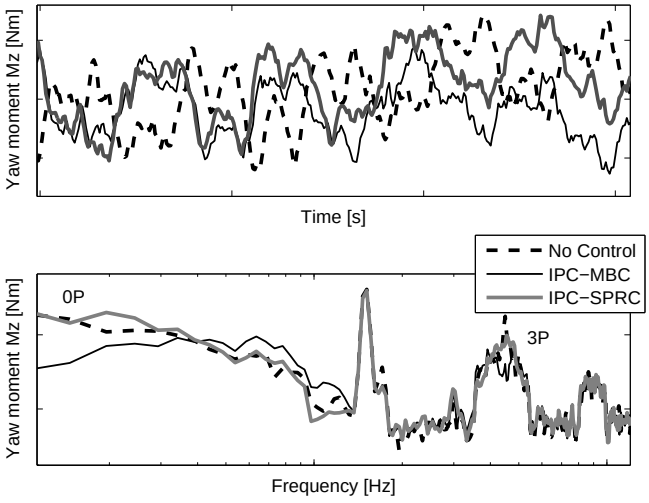


Figure 7.5: Yaw bearing moment, for a wind speed of 18 m/s and 3.75% turbulence intensity.

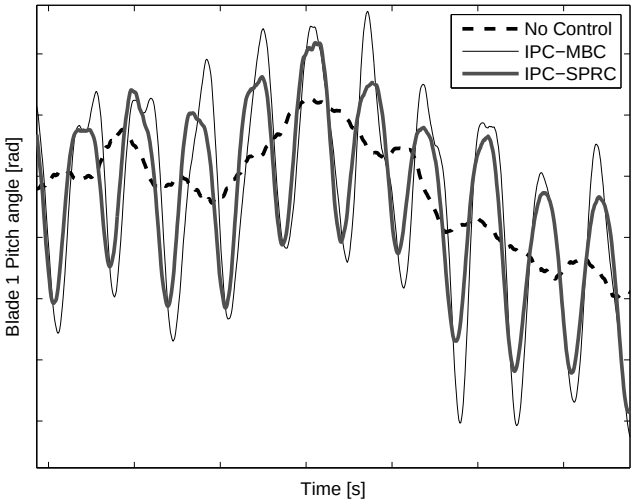


Figure 7.6: Pitch angle for Blade 1, for a wind speed of 18 m/s and 3.75% turbulence intensity.

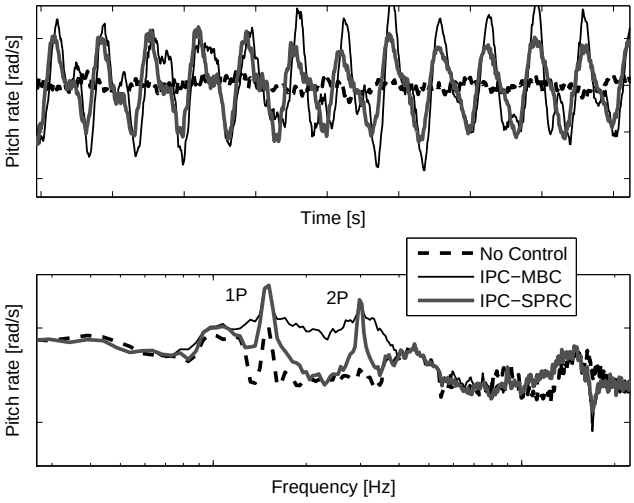


Figure 7.7: Pitch rate for Blade 1, for a wind speed of 18 m/s and 3.75% turbulence intensity.

	v_0 = 18 m/s			
I_T [%]	0	3.75	6	14
% Reduction over baseline in Standard deviation of Rotor Hub Out-of-plane loads				
IPC-MBC	88.59%	51.05%	43.36%	20.99%
IPC-SPRC	93.02%	44.35%	30.52%	23.01%
% Reduction over baseline in Standard deviation of Blade Flap loads				
IPC-MBC	17.26%	15.94%	16.57%	11.96%
IPC-SPRC	17.72%	15.64%	11.08%	16.14%
% Reduction over baseline in Mean of Yawing moment				
IPC-MBC	0.60%	10.57%	24.34%	23.26%
IPC-SPRC	5.83 %	4.29%	28.77 %	14.74%
% Reduction over baseline in Mean of Tower side-side moment				
IPC-MBC	19.57%	19.67%	19.59%	19.19%
IPC-SPRC	19.06%	19.44%	19.91%	22.37%
Standard deviation of Pitch Angle Rates [deg/s]				
IPC-MBC	1.354	1.51	1.76	2.334
IPC-SPRC	1.331	1.378	1.432	1.467

Table 7.2: Load Reduction Results

While the above simulations used time-domain identification, the next section describes results obtained using iteration-domain identification.

7.2.3. SIMULATION RESULTS: ITERATION-DOMAIN IDENTIFICATION

In order to reduce computation time, and to relax the requirement on the persistency of excitation, it is possible to perform system identification in the same lifted iteration domain as the one in which the repetitive control law is formulated, as described in Chapter 4. The reduction in computational complexity arises out of the projection of the stacked input-output data into the basis function space which is used to constrain the direction along which the control input energy lies.

It is seen in the previous section that the identification process demands a persistently exciting signal to be superposed on the pitch command, which can interfere with the collective pitch action and therewith impair speed regulation and power production of the wind turbine. In this section, we add the constraint that, at any instant of time, the summation of the IPC identification and control pitch actuation signals must equal to zero. With this constraint, the SPRC-IPC load reduction controller is entirely decoupled from the power production controllers and does not degrade their performance. This constraint can be enforced by mandating the basis functions, used for the identification and control signals, to satisfy the condition:

$$\phi_u \bar{\bar{I}} = 0. \tag{7.2}$$

Here, the term $\bar{\bar{I}} \in \mathbb{R}^{n_u N \times N}$ is defined as follows:

$$\bar{I} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T, \qquad \bar{I} \in \mathbb{R}^{n_u}, \tag{7.3}$$

$$\bar{\bar{I}} = \begin{bmatrix} \bar{I} & 0_{n_u \times 1} & \cdots & 0_{n_u \times 1} \\ 0_{n_u \times 1} & \bar{I} & \cdots & 0_{n_u \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{n_u \times 1} & 0_{n_u \times 1} & \cdots & \bar{I} \end{bmatrix}. \tag{7.4}$$

As it is known that the three blade pitch signals are approximately 120° out of phase, and the objective of IPC is to reduce the load peaks associated with the 1P and 2P frequencies, the basis functions for the three blade pitch inputs can be taken to be sinusoidal with frequencies 1P and 2P, 120° out of phase with each other. Thus, ϕ_u is taken to be:

$$\phi_u = \begin{bmatrix} \bar{\bar{\sigma}}_1 & \bar{\bar{\sigma}}_2 & \cdots & \bar{\bar{\sigma}}_N \\ \bar{\sigma}_1 & \bar{\sigma}_2 & \cdots & \bar{\sigma}_N \\ \bar{\bar{\sigma}}_2 & \bar{\bar{\sigma}}_4 & \cdots & \bar{\bar{\sigma}}_{2N} \\ \bar{\sigma}_2 & \bar{\sigma}_4 & \cdots & \bar{\sigma}_{2N} \end{bmatrix} \tag{7.5}$$

where:

$$\bar{\bar{\sigma}}_i = \begin{bmatrix} \sin(2\pi i/N) & \sin(2\pi i/N + 2\pi/3) & \sin(2\pi i/N + 4\pi/3) \end{bmatrix}, \qquad i = 1, 2, \ldots 2N, \tag{7.6}$$

$$\bar{\sigma}_i = \begin{bmatrix} \cos(2\pi i/N) & \cos(2\pi i/N + 2\pi/3) & \cos(2\pi i/N + 4\pi/3) \end{bmatrix}, \qquad i = 1, 2, \ldots 2N. \tag{7.7}$$

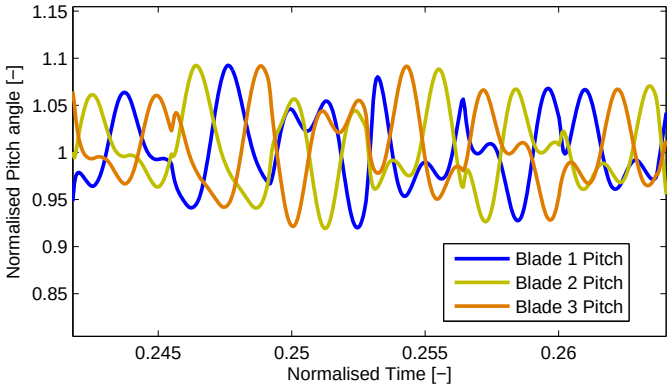


Figure 7.8: Persistently exciting signal in the basis function space, superposed on the pitch signal 18 m/s and 0% turbulence intensity.

It can be seen that the basis functions defined in this manner satisfy the equality constraint (7.2), and hence the identification and control steps of SPRC are entirely decoupled from the baseline power production controllers. Assuming the system to be relatively linear over a short period of time, sinusoidal inputs at specific frequencies should produce sinusoidal outputs at the same frequencies, hence the output basis functions are taken to be the same as the input basis functions, $\phi_u = \phi_y$.

Once again, the above-rated wind speed of 18 m/s is chosen for performing simulations; the duration of each simulation is taken to be 800 seconds. Two different turbulence cases are simulated:

- Zero turbulence: As before, in order to understand the behaviour of the algorithm in the ideal case with no turbulence and perfect periodicity of loads.
- Turbulence intensity 14%: This case with a moderately high value of turbulence makes for a more realistic simulation of SPRC with identification in the iteration domain.

As before, the identification phase of SPRC requires persistency of excitation. However, since in this section, identification is performed in the iteration domain projected into a basis function space, the algorithm only requires persistency of excitation in this reduced dimensional space. For this, a random signal is generated such that its energy lies purely along the directions defined by the basis vectors. Such a signal can be observed in Fig. 7.8. There are two advantages to using such a signal: for one, it is far smoother than the pseudo-random binary signal used in the previous section to ensure adequate persistency of excitation. Secondly, this additional signal for the three blades always sums instantaneously to zero, thereby avoiding interference with the collective pitch speed controller.

In the zero turbulence case, the identification algorithm is once again able to converge to the steady-state value of the iteration domain system Markov parameters within

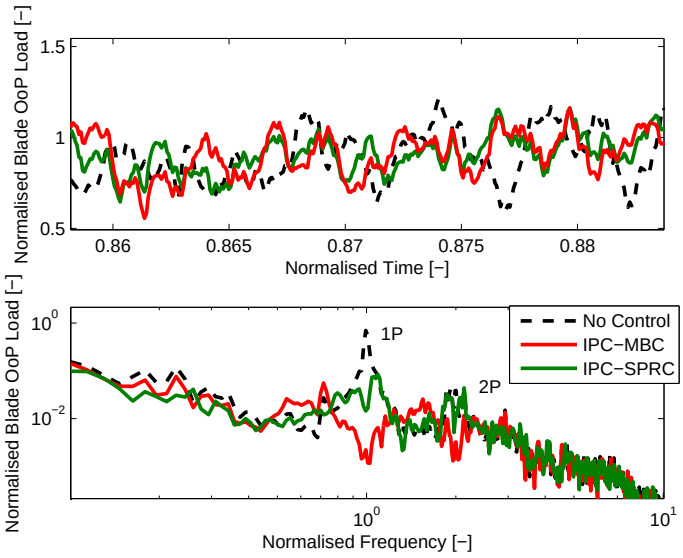


Figure 7.9: Blade load reductions at 18 m/s and 14% turbulence intensity.

100 seconds. At the end of this period, persistency of excitation is reduced to zero, and the identified parameters are frozen. A repetitive control law based on these identified parameters is able to reduce the deterministic loading by 95%, in this ideal, turbulence-free case.

Next, for the 14% turbulence case, a comparison of the load reductions and pitch activity is made once more for the case of IPC-MBC and IPC-SPRC. It can be seen from Fig. 7.9 that the periodic loads at 1P and 2P frequencies are reduced by the IPC controllers. Once again, the MBC controller addresses a broader spectrum of frequencies than the SPRC controller, yielding a blade load reduction of 27% as compared to the blade load reduction of 20% achieved by SPRC. On the other hand, as before, from Fig. 7.10, it is clear that the pitch activity of the SPRC controller is directed specifically at the 1P and 2P frequencies, while the MBC controller has a larger spread over the frequency spectrum. As compared to IPC-MBC, SPRC reduces pitch activity by more than 38%.

It may be concluded from this section that SPRC control, with both time-domain and iteration-domain identification, is able to achieve similar load reductions as those obtained with conventional IPC using the MBC transform. The advantage of using SPRC is that the algorithm does not require manual tuning, rather, it tunes itself autonomously to an optimal value in a data-driven manner. A further advantage is that this algorithm provides precise control over the shape and smoothness of the control signal. Finally, the advantage of the iteration-domain identification is that it decouples the IPC algorithm from nominal turbine power production, and increases the smoothness of the actuator duty cycle while reducing computational complexity.

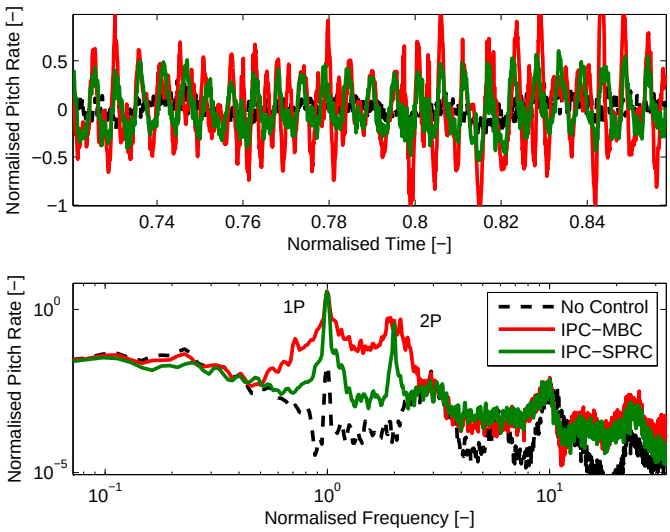


Figure 7.10: Pitch rate activity at 18 m/s and 14% turbulence intensity.

7.3. SPRC FOR TRAILING-EDGE FLAP CONTROL

The previous section discusses SPRC for pitch-based load control, while this section extends this application to trailing-edge flap control for a wind turbine equipped with a 'smart' rotor. The dynamics of the load response to trailing-edge flap control is far more difficult to model than the corresponding pitch actuation response. This modelling uncertainty is exacerbated by the requirement on flap control to be able to respond to high-frequency variations in the wind loading, which directly entails the coupling of unsteady aerodynamics with structural dynamics. As such, in order to tune the load controller appropriately to optimally utilise the control authority of the trailing-edge flaps, a data-driven approach, like SPRC, is considered suitable.

The simulation setup for the smart rotor is first described, then the simulations results are discussed.

7.3.1. SIMULATION SETUP

Once again, the simulation environment GH Bladed, version 4.0, was used to simulate the non-linear time domain behaviour of the wind turbine, and its response to smart rotor control. The wind turbine modelled in this case is the INNWIND 10 MW reference turbine, developed under the INNWIND.EU program, Bak et al. (2013). The major characteristics of this turbine are described in Table 7.3. The flaps are modelled such that they extend between 71 m and 81 m along the blade span, for all three blades. Further, the control authority of the flaps is modelled such that a lift increase of 10% occurs for every 3° increase in flap angle. This value of change in lift may be considered realistic, based on the literature survey performed in the previous chapter. The change in drag

Table 7.3: INNWIND D121 Reference Wind Turbine, Bak et al. (2013)

Description	Symbol	Value
Rated power	P_{rated}	10 MW
Rotor diameter	d_{ro}	178.3 m
Cut-in wind speed	v_{cutin}	4 m/s
Rated wind speed	v_{rated}	11.4 m/s
Cut-out wind speed	v_{cutout}	25 m/s
Rated rotational rotor speed	Ω_{ro}	9.6 rpm
Gearbox ratio	ν	50.0
Pitch-rate limit	$\dot{\theta}_{\text{limit}}$	10°/s

is not modelled, at above-rated wind speeds, drag mainly has an effect on the in-plane loading of the turbine; it is expected that collective pitch control will compensate for this effect. Unsteady effects are not included in the simulation, primarily because their coupling with structural dynamics cannot be modelled in this version of GH Bladed.

As per the previous chapter, a baseline controller is designed, which includes a torque controller and a collective pitch controller, to ensure that the wind turbine generator speed is regulated according to the desired reference trajectory. An individual pitch controller, and other forms of load controllers, are not operational in the current simulations. The major dynamic loading of the rotor occurs at rotor speed (1P) and its harmonics, 2P, 3P and so on. In order to limit the complexity of the flap controller, the objective of SPRC is restricted to the attenuation of the rotor loads occurring at 1P and 2P.

An SPRC controller is used to investigate the load reduction potential of the trailing edge flaps. The block diagram of the implementation of this controller is the same as in previous section, shown in Fig. 7.1, the key difference being that the control input in this section is the trailing-edge flap actuation command. Since the objective of the flap controller is to reduce the dynamic loading at the 1P and 2P frequencies, the basis functions used are the same as those in Section 7.2.1, i.e.:

$$\phi_u = \phi_y = \begin{bmatrix} \sin(2\pi/N) & \cos(2\pi/N) & \sin(4\pi/N) & \cos(4\pi/N) \\ \sin(4\pi/N) & \cos(4\pi/N) & \sin(8\pi/N) & \cos(8\pi/N) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(2\pi) & \cos(2\pi) & \sin(4\pi) & \cos(4\pi) \end{bmatrix}^T . \tag{7.8}$$

The simulations, set up as discussed, are then carried out for different wind field realisations.

7.3.2. SIMULATION RESULTS

For the case of evaluating SPRC for flap control, a mean wind speed of 18 m/s is chosen, as an adequately high above-rated wind speed. A zero turbulence wind field case is studied first, followed by a stochastic wind field realisation with 3.75% turbulence intensity. All cases are run for a simulation time of 800 seconds.

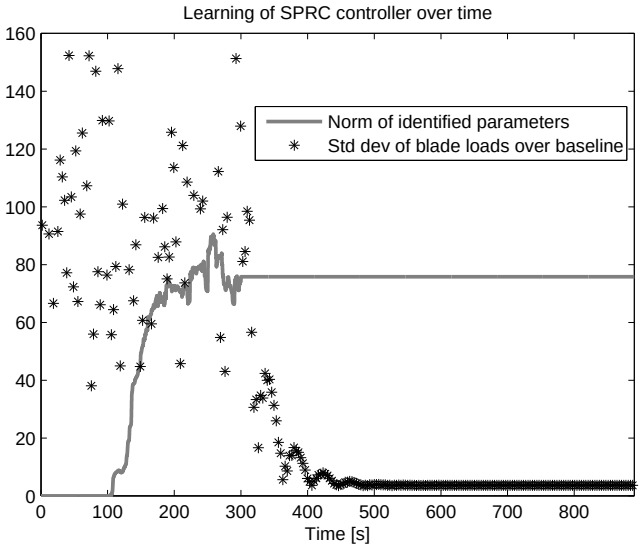


Figure 7.11: Identified Markov parameter estimate norm and performance cost convergence at 18 m/s and 0% turbulence intensity.

For the zero turbulence case, filtered white noise is superposed on the flap actuation signal to attain persistency of excitation. It is seen that the identification phase of the algorithm is able to converge to a reasonably high value of VAF within 120 seconds, or 19 iterations, starting from an initialisation of the system Markov parameter estimates to 0. The evolution of the Markov parameter estimates over time can be observed in Fig. 7.11. Once the value of VAF exceeds 90%, the identified Markov parameters are frozen, hence, the norm of the Markov parameters stays constant after a simulation time of 300 seconds. The repetitive control law synthesised based on the Markov parameter estimates is able to converge to a periodic control sequence that strongly reduces the blade loads, within 180 seconds or 28 iterations. After convergence, for this ideal case, the blade load standard deviation is reduced to 20% of its baseline value.

Next, simulations are carried out for the same wind speed, for a turbulence intensity of 3.75%. It is observed in this case that the action of the flap controller, as opposed to the IPC controller, also contains symmetric components, which shows a small degradation of the speed regulation performance of the collective pitch controller. The simulations demonstrate that blade load reduction is indeed possible, as can be observed in Fig. 7.12. While the load reductions are more modest as compared to IPC, a load reduction of 12.4% in the blade loads is still observed, primarily at the 1P and 2P frequency peaks, as per the constraints of the basis functions. Further, flap control, like IPC, also reduces loading in the support structure, as seen in Fig. 7.13, which shows side-side tower loading. Flap control reduces the mean of the side-side tower loads by 6.3%.

The flap actuator duty cycle can be observed in Fig. 7.14. As apparent, the flap actu-

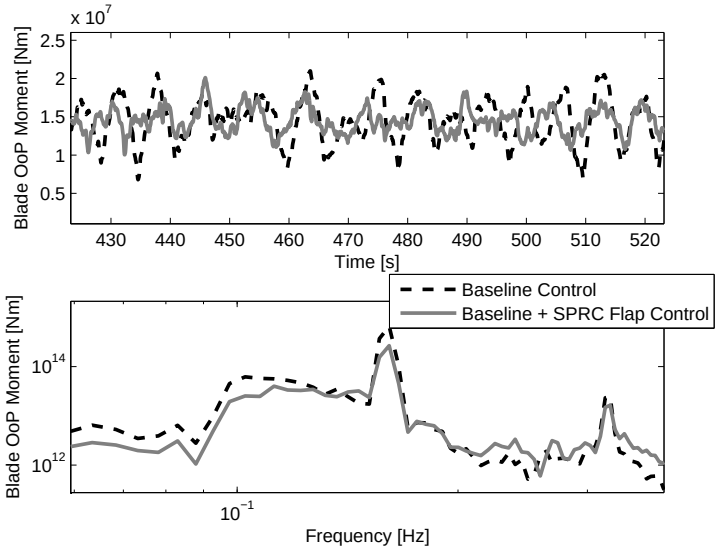


Figure 7.12: Blade load reductions using SPRC, 18 m/s and 3.75% turbulence intensity.

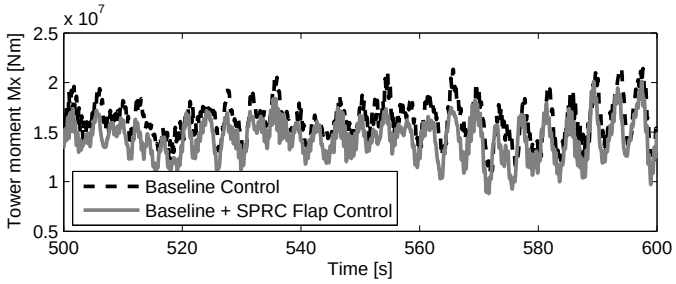


Figure 7.13: Tower side-side load reductions using SPRC, 18 m/s and 3.75% turbulence intensity.

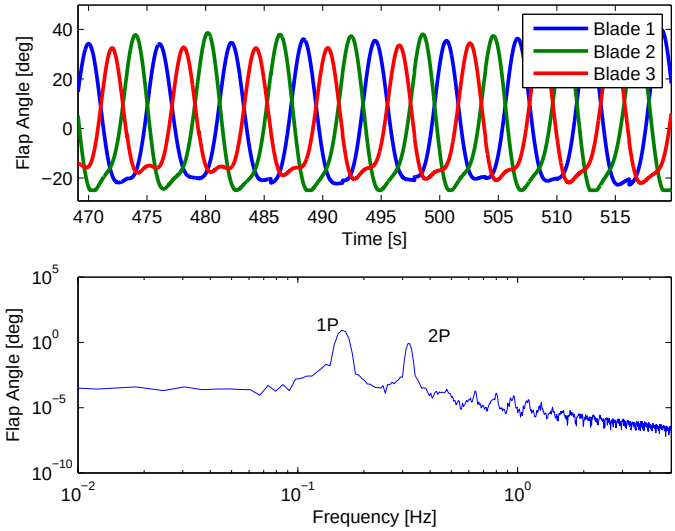


Figure 7.14: Flap actuation signal using SPRC, 18 m/s and 3.75% turbulence intensity.

ation signal only contains energy at the 1P and the 2P frequency, as per the constraints imposed by the basis functions. It can be seen that the flap activity is much higher than the corresponding IPC pitch activity required for achieving the same load reductions. It may be possible that achieving such high flap deflections may not be feasible, either structurally or in terms of ensuring that aerodynamic separation does not occur. Such constraints could in principle be incorporated when optimising the SPRC cost function over a finite hoirzon.

The collective pitch activity is slightly higher than baseline; this is because the symmetric forcing of the flap activity is required to be compensated by collective pitch action. However, it should be noted that the total pitch actuator duty is still an order of magnitude lower than that demanded by individual pitch control. Thus, although the use of trailing-edge flaps shows lower load reduction capabilities than IPC, pitch actuator duty is reduced significantly by using flaps to address asymmetric blade loading.

SPRC proves to be useful in addressing the unknown dynamic characteristics of the flap actuators; it also yields smooth control signals suitable for increased reliability of the actuators. The utility of SPRC may perhaps be greater for a simulation tool with greater fidelity, which is able to model the unsteady aeroelastic effects of the trailing edge flaps. Alternatively, it would be interesting to investigate this approach experimentally.

It should be noted that SPRC treats the system as instantaneously linear, and requires reïdentification for the case where the system dynamics vary with time. For instance, a wind turbine shows dynamics that are LPV (linear parameter-varying), and change with wind speed. In such a case, the optimal SPRC law will need to be recalibrated for different wind speeds, repeatedly over time. An alternative to this approach would be a fully LPV

data-driven controller, the gains of which converge to constant values over time. Such a fully LPV approach to wind turbine load control is considered in the next section.

7.4. IFT-LPV FOR WIND TURBINE PITCH CONTROL

In this section, the concept of IFT-LPV, developed in Chapter 5, is explored for the purpose of wind turbine IPC. The primary motivation behind the use of this LPV technique is that the underlying wind turbine pitch control system is LPV by nature, with the pitch control authority increasing in an approximately quadratic manner with an increase in inflow wind speed. As such, a wind turbine can be modelled as LPV with the wind speed forming the scheduling variable. For global wind turbine load rejection, irrespective of wind speed, it is hence desirable to synthesise an LPV IPC controller, which is also scheduled on wind speed. The direct data-driven approach of IFT affords the possibility to tune the gains of a low-order LPV controller to (locally) optimise the IPC performance, for all wind speeds, and is hence considered suitable for the application.

First, the simulation setup is described, and the specific form of the low-order LPV controller, to be tuned using IFT, is justified. Next, simulation results are discussed.

7.4.1. SIMULATION SETUP

As in the previous section, the simulation environment used is GH Bladed, version 4.0, and the turbine modelled in this environment is the INNWIND 10 MW reference turbine, with major characteristics given in Table 7.3. Bladed is a non-linear time-domain turbine simulator, and it uses multi-body dynamics, with a flexible tower and blades. The non-linear model herewith has 69 states, and the dynamics shows a variation with wind speed. Thus, the wind speed μ_k is the LPV scheduling variable; it satisfies the conditions of being exogenous, persistently exciting and uncontrollable. At this stage, it is assumed that the wind speed is also perfectly measurable, although this assumption has limited practical realisability. The transfer from pitch actuation to blade loads, linearised for three different wind speeds, can be observed in Fig. 7.15. It can be clearly observed that the control authority of pitch actuation increases with the wind speed, while the location of the poles remains relatively unaltered. As turbine rotors become more flexible, the coupling of aeroelastic modes will imply that the poles may also change location, however, the reference turbine shows a relative invariance of poles with a change in wind speed, that is true of most commercial wind turbines.

As before, nominal operation of the wind turbine is ensured through the use of baseline torque and collective pitch controllers, designed as in the previous chapter, that regulate generator speed to its reference value. The IFT-LPV controller that will be designed for the purpose of IPC for blade load alleviation, is connected in an outer loop around the baseline controlled plant, and is entirely independent of the baseline controller.

For the design of the controller, as blade loads occur in the frequency spectrum primarily at the 1P peak, the IFT-LPV controller is designed such that it also generates control action at this frequency. Further, the optimal IFT-LPV control action will be required to generate sinusoidal control action that changes magnitude and phase with changing wind speed, since the response of the underlying LPV system to control actions at different wind speeds will be different. Also, since it has been shown that the ideal control

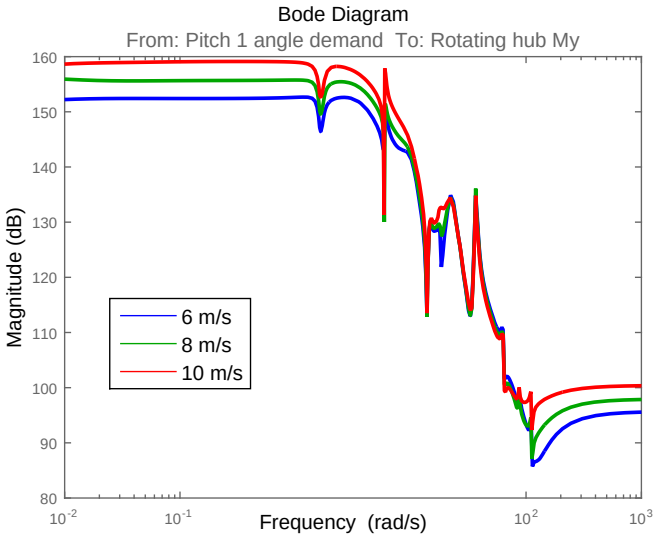


Figure 7.15: Linearised turbine dynamics from pitch actuation to blade loads, different wind speeds.

action for the three blades over each rotation is 120° out of phase with each other, this special structure can also be incorporated in the IFT-LPV control action in order to speed up the convergence of the algorithm by reducing the controller degrees of freedom.

It is seen from the linearised models made available from GH Bladed that the wind turbine system may be approximated as LPV in its input matrices (B taken LPV). It is then expected of the controller tuned using IFT-LPV to generate a control action that commands the three blade pitch angles, equally in magnitude, and 120° out of phase with each other, in response to the three blade loads that are also roughly equal in magnitude and 120° out of phase with each other. As such, the system can be considered to be pseudo-SISO, in that one single control action is to be synthesised in order to reduce the effect of three blade loads, rotated and averaged together. For this pseudo-SISO system, the LPV scheduling dependency can be moved from the input matrix B to the output matrices C and D, to yield a system LPV in its output matrices.

In the previous sections, it was seen that significant load control could be achieved by using SPRC to converge to a constant feedforward sequence scheduled on sinusoidal basis functions of the rotor azimuthal position. This same concept is used to define an LPV feedforward controller that is to be tuned using IFT; the feedforward signal forms linear combinations of sinusoidal basis functions scheduled on the measured rotor az-

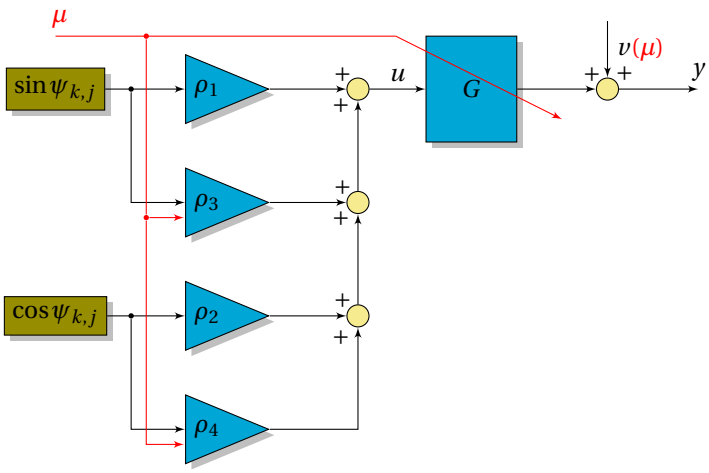


Figure 7.16: IFT implementation: wind turbine load control.

imuth φ_k, such that:

$$u_k^{(j)} = \rho_1 \sin\left(\varphi_k + \frac{2(j-1)\pi}{3}\right) + \rho_2 \cos\left(\varphi_k + \frac{2(j-1)\pi}{3}\right) + \rho_3 \mu_k \sin\left(\varphi_k + \frac{2(j-1)\pi}{3}\right) + \rho_4 \mu_k \cos\left(\varphi_k + \frac{2(j-1)\pi}{3}\right), \qquad j \in \{1,2,3\}, \tag{7.9}$$

j being the number of the blade. Further, the terms ρ_{j_ρ}, $j_\rho \in \{1,2,3,4\}$ are the four controller parameters that are sought to be optimised by using IFT. For each blade j, this implementation can be visualised in Fig. 7.16, where the synthetic feedforward reference signals are generated as 1P sinusoids scheduled on the measured blade azimuth. The azimuthal position $\varphi_{k,j}$ of each blade j is given by:

$$\varphi_{k,j} = \varphi_k + \frac{2(j-1)\pi}{3}. \tag{7.10}$$

Since the reference signal is phase-locked with the rotor azimuth, the control action produced will always be in the correct phase for load attenuation using IPC. It is seen that a specific fixed structure has been assumed for the controller; the gains of the basis functions are considered to increase (or decrease) in a linear manner with an increase in wind speed. This formulation may not be the most optimal usage of pitch control authority over the operational range of the wind turbine. For instance, a quadratic or exponential gain schedule may possibly provide better load control. The concept of IFT-LPV can, however, only find the most optimal controller within the set of controllers that admit the user-defined parameterisation.

It should be noted that the 'feedforward' reference signal in this case is a set of basis functions scheduled on the rotor azimuth. As such, the approach of IFT-LPV for feedforward controller tuning for systems LPV in the output matrices can be employed in this case, as described in Chapter 5, Section 3. It is notable that this formulation is not hampered by the curse of dimensionality typically found in data-driven LPV approaches.

Output load measurements are required in this case, they are however not used for performing feedback control. Instead, the output measurements form the performance criterion that is to be optimised by IFT by tuning the controller parameters. As such, at steady state, such a load controller will be a purely feedforward controller, and cannot render the system unstable even for a poorly chosen class of controllers.

The simulations in this section are carried out below rated wind speed, at which point the torque controller is active and the collective pitch controller is not active. Here, the largest variation in system dynamics is experienced, primarily since the generator speed is not regulated to a constant value, but it is allowed to vary in order to maximise power capture. As such, the wind speed is allowed to vary between 6 m/s and 10 m/s, which is below the rated wind speed value of 11.4 m/s for the reference turbine. It should be noted that the conventional MBC transform cannot be directly used for an IPC controller in this case, since it cannot account for the variations in system dynamics, and requires the design of LPV filters for shaping the input and output signals.

The results of IFT feedforward tuning for the LPV controller are described in the next section.

7.4.2. SIMULATION RESULTS

The wind turbine is simulated in GH Bladed such that it is subject to a wind field that varies between 6 and 10 m/s. Two specific controllers are tuned using the IFT-LPV methodology of Chapter 5:

- A controller that is Linear Time-Invariant (LTI), with ρ_3 and ρ_4 set identically to zero.
- A fully LPV controller where all four controller parameters are optimised using IFT.

The reduction in the performance cost with every iteration of IFT can be seen in Fig. 7.17. As the controller parameters are initialised to zero and updated using the cost gradients estimated from data using the IFT experiments, the performance cost reduces over successive iterations. While the LTI controller is also able to achieve reductions in performance cost, the LPV controller tuned using IFT outperforms the LTI controller tuned using the same technique. The convergence of the controller parameters ρ_{j_ρ} to their optimal values can be seen in Fig. 7.18. This rate of convergence depends upon the turbulence levels in the wind field, as the turbulence intensity doubles from 5% to 10%, the rate of convergence reduces by 35%.

Next, the tuned feedforward controllers are tested as regards to their load reduction capabilities by simulating a gust from 6 m/s to 10 m/s, occurring over a period of 50 seconds. As seen in Fig. 7.19, both LTI and LPV controllers designed using IFT are capable of achieving load reductions. While similar load reductions are achieved at low wind speeds, at higher wind speeds, the load reduction capability of the LPV controller is higher by 20%. The load reduction results are tabulated in Table 7.4. The pitch inputs demanded by the controllers can be observed in Fig. 7.20. In both cases, the control action has the correct phase for load reduction. With LPV control, the magnitude of the control action becomes optimally gain-scheduled. At higher wind speeds, since the pitch actuator has higher control authority, the magnitude of the pitch action demanded is lowered

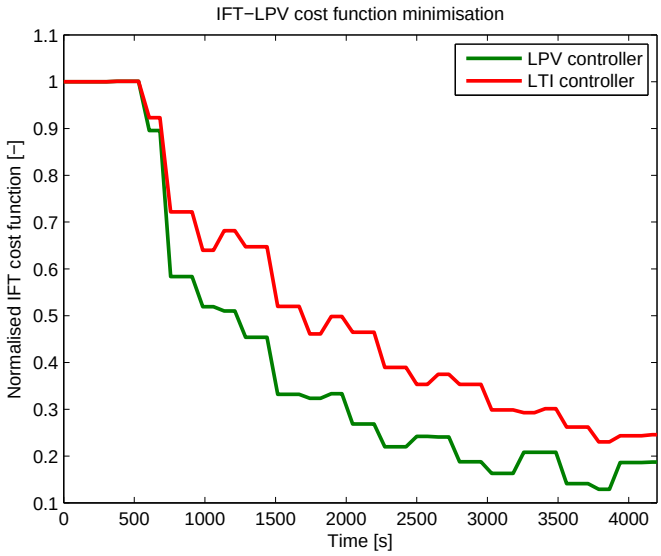


Figure 7.17: IFT-LPV cost after iterative tuning, wind speed between 6 m/s and 10 m/s.

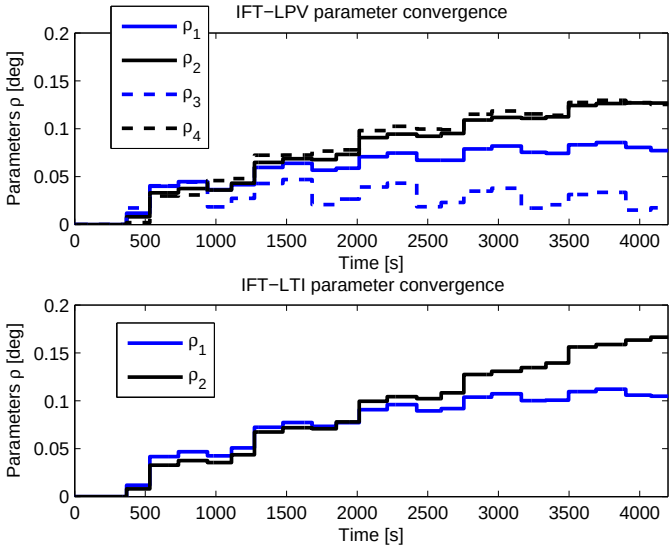


Figure 7.18: IFT-LPV and IFT-LTI parameter values after iterative tuning, wind speed between 6 m/s and 10 m/s.

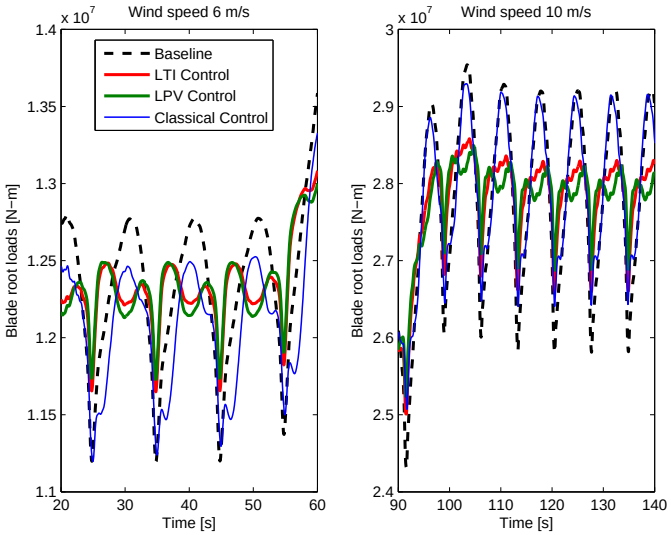


Figure 7.19: Blade load reductions using IFT-LPV and IFT-LTI controllers, wind speed between 6 m/s and 10 m/s.

Table 7.4: IFT Load Reduction Results

Load reduction [% below Baseline]	Wind Speed 6 m/s	Wind Speed 10 m/s
IFT-LTI	58.5%	58.2%
IFT-LPV	61.4%	69.5%

accordingly and greater load alleviation is achieved by the LPV controller with reduced control effort. For comparison, the results obtained using an IPC-MBC controller designed for above-rated operation have been depicted in the plots as the 'classical' controller. This controller is unable to compensate for changes in the rotor speed that occur in the below-rated region; it is also not able to recalibrate itself for the reduced control authority in this operational regime. As such, minimal load alleviation is observed when this controller is used in the below-rated region of operation.

Thus, it can be observed in this section that IFT-LPV can directly synthesise a feedforward IPC controller that achieves significant load reductions irrespective of the operating wind speed, without requiring the intermediate step of system identification. This controller is fixed-structure and hence low-order, it is locally optimal among the set of similarly parameterised controllers. As compared to an LTI controller designed in the same way, it achieves 20% more load reduction under off-design conditions.

Until this point, IPC controllers have been designed primarily for blade load reduction; support structure load alleviation is considered as a collateral benefit of IPC. How-

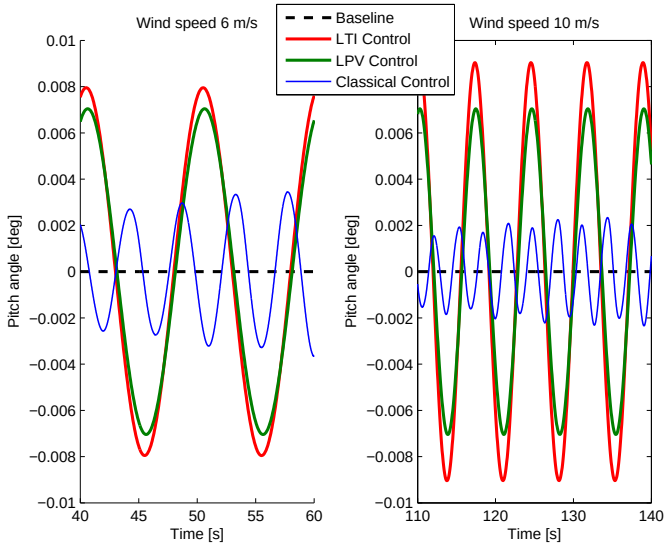


Figure 7.20: Pitch control action demanded by IFT-LPV and IFT-LTI controllers, wind speed between 6 m/s and 10 m/s.

ever, by combining IPC with yaw control, it is possible to consider the trade-off with loads in the non-rotating components of the wind turbine. This aspect of IPC for yaw control will be explored next using numerical simulations.

7.5. IPC FOR YAW CONTROL

The objective of yaw control is to keep the wind turbine rotor plane aligned perpendicular to the wind flow direction, in order to maximise power production. For a downwind turbine, such a configuration is stable, and yaw control can be achieved with minimal effort, Verelst et al. (2012). A modern commercial wind turbine, however, has an upwind configuration that is unstable in the yaw degree of freedom, and requires closed-loop feedback control for stabilisation. Such control actions are provided by a series of yaw motors that connect the nacelle with the (stationary) tower in such a manner that the yaw error between the attitude of the nacelle and the direction of wind flow is minimised. This yaw system, consisting of yaw motors, a gearing arrangement and a yaw bearing, includes some of the most highly stressed components in wind turbines, Pesmajoglous and Graham (1992), and directly influences the rotor and transmission loads transferred to the support structure, Ekelund (2000).

The dynamic behaviour of the yaw system has been studied, Maalawi (2007), and the effect of tuning structural parameters, like yaw stiffness, has been explored in the literature, for instance by Stubkier and Pedersen (2011). Further, the concept of using IPC to alter the yaw behaviour of wind turbines has been introduced in Burton et al. (2011). However, a detailed analysis of the effect of IPC for yaw control on the dynamic

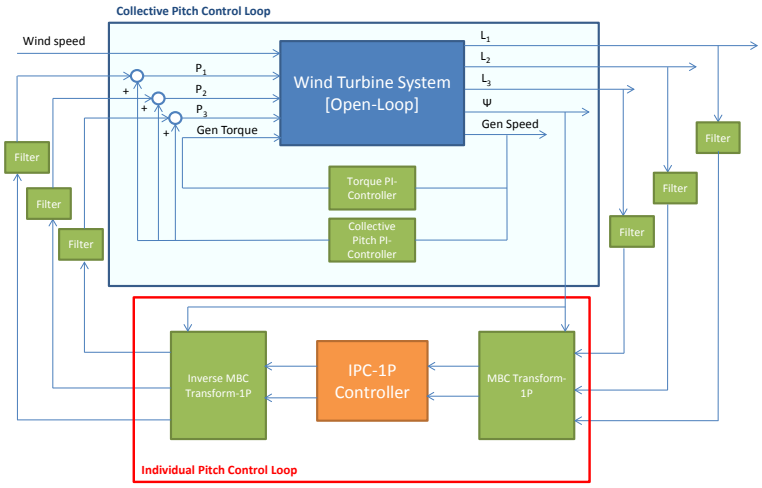


Figure 7.21: Conventional implementation of IPC-MBC.

loads experienced by upwind wind turbines has not yet been conducted.

In this section, the extension of IPC for wind turbine yaw control is investigated in the aeroelastic simulation environment GH Bladed. The effect on the support structure loads is discussed, and the collateral effect of reduced blade load alleviation and increased pitch activity is discussed. This section also shows that IPC can form a redundant yaw actuator that could serve as a backup in case of errors in the primary yaw control mechanism, or in case the yaw system duty cycle is to be limited from the point of view of reliability and operational maintenance.

7.5.1. IPC-Y: EXTENDING THE MBC TRANSFORM

This section explores how the conventional approach to IPC, using the Multi-Blade Coördinate (MBC) transform, can be extended to include yaw control as one of the objectives. The approach towards load control using IPC-MBC has been described in the previous chapter, and can be illustrated by using Fig. 7.21.

Thus, for a three-bladed turbine, based on the blade root measurements L_1, L_2 and L_3 in the rotating frame of reference, the MBC transform is used to synthesise the stationary frame-of-reference signals L_d and L_q,

$$\begin{bmatrix} L_d \\ L_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\Psi & \cos(\Psi + \frac{2\pi}{3}) & \cos(\Psi + \frac{4\pi}{3}) \\ \sin\Psi & \sin(\Psi + \frac{2\pi}{3}) & \sin(\Psi + \frac{4\pi}{3}) \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix}, \tag{7.11}$$

with Ψ being the measured rotor azimuth. The decoupled signals L_d and L_q are fed into SISO PI controllers to obtain the control action in the stationary frame of reference, P_d and P_q, on the basis of which the blade pitch commands P_1, P_2 and P_3 are synthesised

using the inverse MBC transform:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} \cos\Psi' & \sin\Psi' \\ \cos(\Psi' + \frac{2\pi}{3}) & \sin(\Psi' + \frac{2\pi}{3}) \\ \cos(\Psi' + \frac{4\pi}{3}) & \sin(\Psi' + \frac{4\pi}{3}) \end{bmatrix} \begin{bmatrix} P_d \\ P_q \end{bmatrix}, \tag{7.12}$$

with Ψ' being the rotor azimuth rotated through a small angle, for maximising load reduction. It has been shown by Van Engelen (2006) that the d- and q-axes in the stationary frame of reference can be physically interpreted as the tilt and yaw axes of the wind turbine. Thus, for instance, the load L_q obtained from the MBC transform provides a measure of the yaw load acting on the wind turbine rotor. When IPC with the MBC transform is used for a turbine fixed in yaw, the PI controllers act to minimise the yaw moment at the nacelle base. Conversely, if the turbine nacelle is free to yaw about the tower, IPC can be used to minimise the yaw error between the nacelle and the wind flow direction. As such, for a turbine free to yaw about the tower axis, IPC can be used to stabilise the turbine in yaw, and reject yaw error. For instance, for the case where the yaw error is χ, a simple PI controller can be used to generate an IPC control action $P_{q,\chi}$ (in the stationary frame of reference), as follows:

$$P_{q,\chi} = K_p \chi + K_i \int \mathrm{d}\chi, \tag{7.13}$$

with K_p and K_i the tunable parameters of the PI controller. This extra pitch action can now be summed with the IPC load control pitch action in the stationary frame of reference, in order to synthesise the desired pitch control actions as:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} \cos\Psi' & \sin\Psi' \\ \cos(\Psi' + \frac{2\pi}{3}) & \sin(\Psi' + \frac{2\pi}{3}) \\ \cos(\Psi' + \frac{4\pi}{3}) & \sin(\Psi' + \frac{4\pi}{3}) \end{bmatrix} \begin{bmatrix} P_d \\ P_q + P_{q,\chi} \end{bmatrix}. \tag{7.14}$$

This extension of the IPC-MBC controller can be visualised in Fig. 7.22. It can be observed that the load control pitch signal, along the yaw axis in the stationary frame of reference, is augmented with the yaw error control pitch signal, synthesised based on the feedback from the yaw sensor. This implementation of IPC replaces the red box in the conventional implementation of IPC, seen in Fig. 7.21.

Due to the non-linear effect of Coulombic friction in the yaw subsystem, low-magnitude yaw actuation signals will not have an effect on the yaw error. To avoid commanding such signals, a non-linear saturation block is used to shape the yaw error before it enters the PI yaw control block, so that the controller is insensitive to small yaw misalignments. The saturated yaw error χ' is shaped from the actual yaw error χ in such a manner that:

$$\chi' = \begin{cases} \chi \frac{\tanh(\chi+\alpha)-1}{2}, & \text{if } \chi \leq 0 \\ \chi \frac{\tanh(\chi-\alpha)+1}{2}, & \text{if } \chi > 0. \end{cases} \tag{7.15}$$

Here, the quantity α is the amount of yaw misalignment that can be tolerated before it becomes necessary for yaw control actuation to be implemented. The effect of the saturation block can be visualised in Fig. 7.23. The PI yaw controller then acts on the

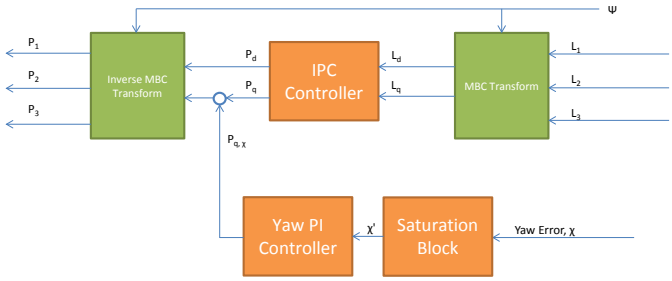


Figure 7.22: Implementation of yaw control using IPC-MBC.

saturated yaw input, as follows:

$$P_{q,\chi} = K_p \chi' + K_i \int \mathrm{d}\chi', \tag{7.16}$$

This IPC yaw controller is demonstrated in the next section using the GH Bladed simulation environment for the commercial Darwind XD115 wind turbine, the setup of which is described in Section 7.2.

7.5.2. SIMULATION RESULTS

The XEMC-Darwind commercial 5 MW turbine model, described in Section 7.2, is here modified such that the turbine nacelle is able to yaw freely about the tower. The potential of yaw control using IPC is demonstrated, and the effect on turbine loading is studied.

The yaw controller block from Fig. 7.22 is in this case implemented using a simple PI controller with fixed gains. While proportional action represents the effect of artificial yaw stiffness, integral action is necessary in this case to ensure zero steady-state error. Three different control configurations are investigated in terms of their effect on yaw stabilisation and turbine loads:

- A turbine operating in free yaw, where IPC is used for yaw control and load reduction, with collective pitch and torque control for baseline speed regulation.
- A turbine simulated with a constant structural yaw stiffness of 10^8 N-m, with IPC used only for load reduction, and collective pitch and torque control for baseline speed regulation.
- A turbine simulated with a constant structural yaw stiffness of 10^8 N-m, without IPC, but still with collective pitch and torque control for baseline speed regulation.

For the case of free yaw, it is considered that the yawing motion is opposed by a frictional force, that is taken to be 10% of the yaw moment experienced in the fixed-yaw case. While this is a conservative estimate, it allows us to explore the potential of IPC for yaw control.

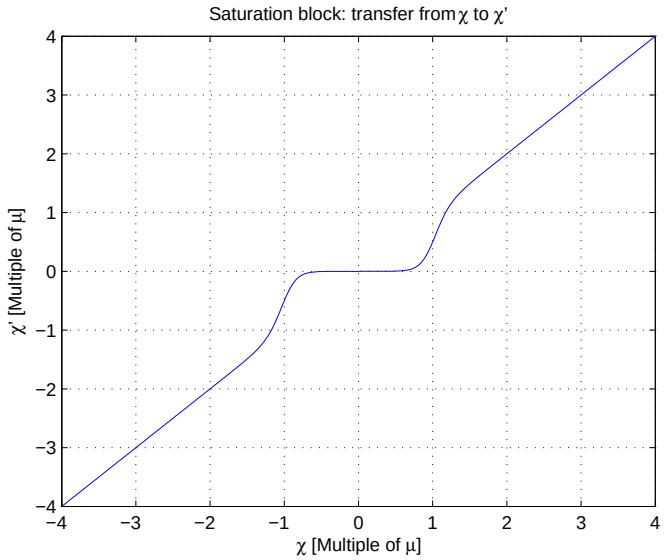


Figure 7.23: Saturation block to avoid correction for small yaw misalignments.

Under all operating conditions, simulation results show that it is possible to reject yaw error purely through the use of IPC. The trade-off between structural load alleviation and yaw error rejection can be tuned by adjusting the aggressiveness of the yaw controller, thus by increasing or reducing the gains of the PI controller in equation (7.16). Since this determines the bandwidth of the PI controller, this trade-off is essentially a function of the controller bandwidth.

Selected results are presented in this section to demonstrate the loading behaviour of this control approach. Specifically, in order to make these results comparable to the ones in the previous sections, the case with mean wind speed of 18 m/s, and a low turbulence intensity of 3.75% is chosen as a demonstration case.

With IPC for load control, the mean load on the yaw bearing reduces, as a collateral effect of reduced blade loads. However, this reduction is limited since the gains used in IPC for blade loads are not optimised for yaw load reduction. On the other hand, for free yaw, the only loads arising on the yaw bearing originate from yaw friction, and as such, the yaw loads effectively reduce by a factor of 15, as observed in Fig. 7.24. As expected, however, IPC is less effective in yaw error rejection, and the yaw error of the turbine increases by a factor of 14, Fig. 7.25. It should, however, be noted that in practical terms the magnitude of yaw error remains bounded to within a few degrees.

Free yaw effectively implies that the transfer of loads from the rotor to the support structure is strongly reduced, this effect can be seen in Fig. 7.26. With IPC for load control, the mean tower loads are reduced to some extent. The major component in tower load reduction, using IPC for yaw control, arises from the result that the 1P rotor mass/aerodynamic imbalance loads are attenuated strongly due to the decoupling ef-

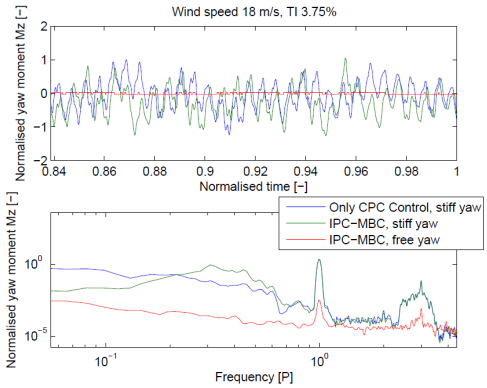


Figure 7.24: Yaw moment at yaw bearing for wind speed 18 m/s, turbulence intensity 3.75%.

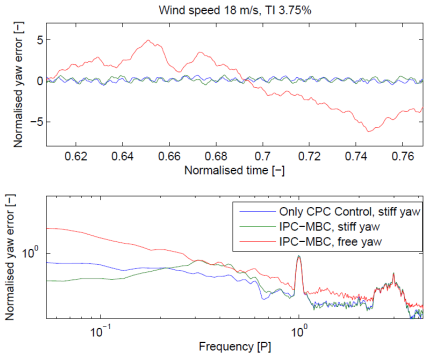


Figure 7.25: Turbine yaw error for wind speed 18 m/s, turbulence intensity 3.75%.

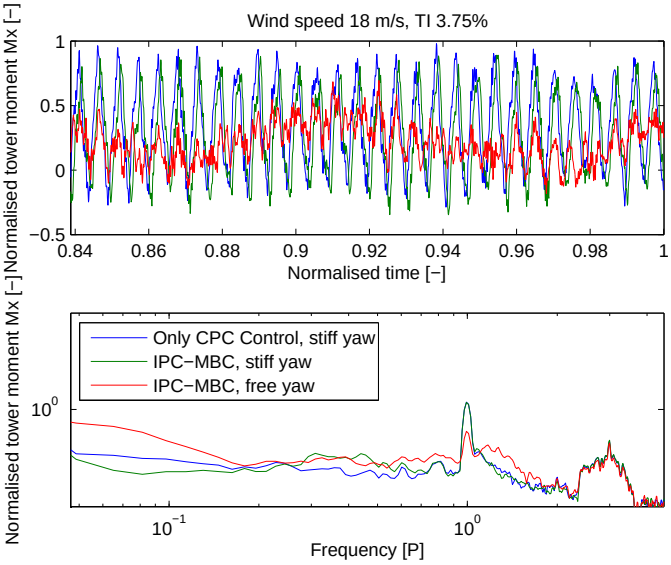


Figure 7.26: Tower side-side loads for wind speed 18 m/s, turbulence intensity 3.75%.

fect of free yaw. There is a small increase in the low-frequency part of tower loads due to the effect of yaw friction and increased yaw error. Overall, tower side-side loads reduce by more than 50% in this low-turbulence case.

The ability of IPC to reduce blade loads is impaired slightly due to its extension to include yaw control, as seen in Fig. 7.27. While conventional IPC can achieve upto 45% load reductions for this specific case, when IPC is used for yaw control, this number drops to 43%.

Finally, it can be observed from Fig. 7.28 that the pitch activity demanded by IPC does not increase significantly when yaw control is added as a control objective. However, the demanded pitch activity of the IPC controller is still markedly higher than that demanded by the collective pitch control, by a factor of nearly 5. Since only the 1P MBC transform has been used in this case, the pitch activity is concentrated to some extent around the 1P peak, however there is still a considerable spread of energy across the frequency spectrum. It is also interesting to see that there is a phase difference between the pitch control signals commanded by conventional IPC, and by IPC for yaw control, which is responsible for the yaw error rejection capability of the latter control approach.

If the controller bandwidth is increased, yaw error rejection is improved, however this causes a degradation in the blade load alleviation potential, as tabulated in Table 7.5. Further, the effect of turbulence on the yaw error rejection capabilities of IPC are tabulated in Table 7.6. It can be seen that the yaw error rejection performance is virtually unaffected by turbulence, while the load reduction potential for the support structure and for the blade reduces with increasing turbulence.

Thus, it can be concluded from the simulations in this section that it is indeed pos-

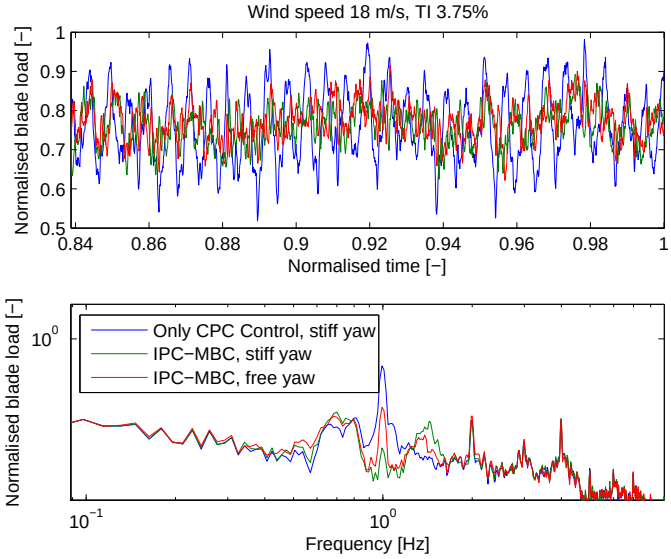


Figure 7.27: Blade root load reductions for wind speed 18 m/s, turbulence intensity 3.75%.

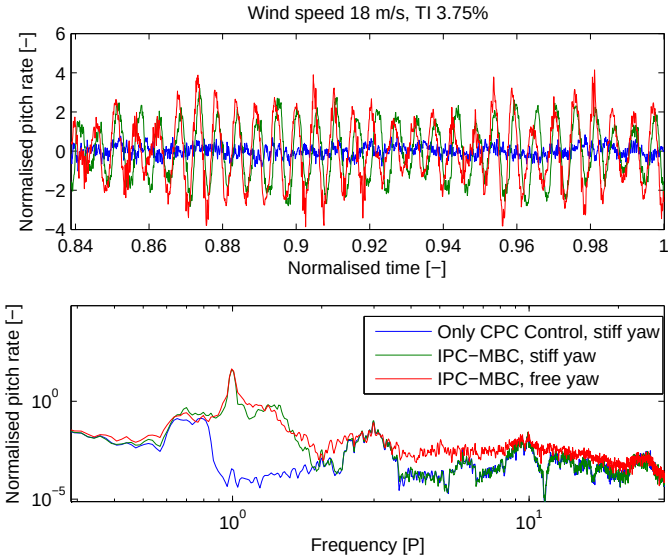


Figure 7.28: Blade pitch rate for wind speed 18 m/s, turbulence intensity 3.75%.

Table 7.5: Effect of changing yaw controller bandwidth, turbulence intensity 3.75%

Normalised quantity	No IPC	IPC	IPC	IPC	IPC
Yaw mode	stiff	stiff	free	free	free
Yaw controller bandwidth			0.0048P	0.0063P	0.0077P
Yaw bearing moment std dev	1	1.076	0.065	0.065	0.065
Yaw error std dev	1	1.076	14.30	13.37	12.20
Tower side-side std dev	1	0.990	0.474	0.470	0.463
Blade out-of-plane std dev	1	0.553	0.564	0.567	0.574
Pitch rate std dev	1	5.122	5.566	5.613	5.675

Table 7.6: Effect of changing turbulence intensity, yaw controller bandwidth 0.0077P

Normalised quantity	No IPC	IPC	IPC	IPC	IPC	IPC	IPC
Yaw mode	stiff	stiff	free	stiff	free	stiff	free
Turbulence intensity [%]	All	3.75	3.75	6	6	14	14
Yaw bearing moment std dev	1	1.076	0.065	1.019	0.047	0.972	0.029
Yaw error std dev	1	1.076	12.20	1.020	12.41	0.972	13.97
Tower side-side std dev	1	0.990	0.463	1.007	0.637	1.012	0.936
Blade out-of-plane std dev	1	0.553	0.574	0.657	0.697	0.884	0.904
Pitch rate std dev	1	5.122	5.675	4.613	5.319	3.402	4.477

sible to perform yaw control using IPC for a standard upwind turbine. Further, a simple parameterisable controller has been set up to achieve this control action. With the use of IPC for yaw control, of an appropriate bandwidth, it is shown possible to achieve the desired trade-off between blade loads and support structure loads; it thus shows synergy with the data-driven approaches to load control discussed in the previous sections. The collated conclusions of all the aeroelastic simulations carried out in this chapter will be discussed in the next section.

7.6. CONCLUSIONS

The iterative data-driven control theory developed in the first part of the thesis has been applied to a high-fidelity non-linear numerical model of a modern commercial turbine, using the loads and controller certification test bench software, GH Bladed. In general, the potential of achieving near-optimal load control starting from limited knowledge of the true system dynamics, has been demonstrated in a fully data-driven manner.

The two-step data-driven technique of Subspace Predictive Repetitive Control (SPRC) was found to achieve load reductions comparable to an optimally hand-tuned IPC controller, iterating to the best possible periodic signal that optimally minimised the deterministic component of the periodic rotor loads. Further, since SPRC is especially amenable to implementation via basis functions, it was shown that perfect control over the shape and smoothness of the SPRC control signal is possible. This feature, alongside

the reduction in pitch actuator duty, implies that SPRC is specifically interesting for the problem of achieving turbine load control while ensuring longevity for the pitch actuation system. If the identification step of SPRC is also performed in the iteration domain, in the basis function space, then, in addition to extremely smooth control actions, it is also possible to decouple IPC entirely from the nominal power production of the wind turbine.

Similar results were achieved using SPRC for trailing-edge flap control, where the data-driven approach of SPRC is attractive on account of the difficulty in modelling the unsteadiness in aerodynamic forces, and their interaction with structural dynamics. Significantly flap actuator duty was seen to be demanded in order to achieve load reductions, as compared to pitch activity: this is to be expected since flap actuators show lower control authority than full-span pitch. The implementation of flaps in Bladed is simplistic, it is expected that data-driven control would prove more valuable using aeroelastic simulation tools better able to capture fluid-structure interaction at the scale of the flap; it is also expected that data-driven control would be useful in the field.

SPRC considers the system as instantaneously linear, and when applied to time-varying systems, it requires continuous recalibration for optimal control behaviour. For an LPV system like a wind turbine, the dynamics of which can be considered to be a function of the operating wind speed, it could be more desirable to synthesis an LPV controller, the optimally tuned gains of which converge to constant values over time. Such a controller is synthesised by using the IFT-LPV approach, which tunes an LPV PI controller using input-output data generated using IFT experiments. This controller is set up as a feedforward controller, thus with frozen gains, the controller cannot destabilise the system. Further, by recasting the system as pseudo-SISO, and LPV in its output matrices, it is possible to circumvent the curse of dimensionality associated with data-driven LPV approaches, and limit the number of IFT experiments required per iteration. This controller is able to achieve load reductions superseding those achieved by using an optimal LTI controller across the operating wind speed range. The PI-nature of this controller is an advantage in that the order of the controller is far lower than that of the underlying system; however, the IFT approach can only iterate to the optimal controller within this restricted set of parameterised controllers. As such, the performance achieved may not be the maximal load attenuation performance possible for this specific system.

While these approaches consider a wind turbine with an independent yaw mechanism, it has been shown in this chapter that it is possible to achieve yaw control using IPC, and herewith stabilise an upwind turbine that is free to yaw about the tower. The conventional IPC approach from the previous chapter, which uses the MBC transform, is extended with a simple PI controller that commands IPC such that yaw error is suppressed. Simulations show that support structure loads are reduced at the expense of increased blade loads, and a trade-off can be achieved by tuning the bandwidth of the PI-IPC yaw controller. It is postulated that this degree of freedom could also be extended in a data-driven manner.

Thus, data-driven control approaches to wind turbine load alleviation have been investigated using aeroelastic simulations in this chapter, and shown to possess several attractive characteristics. The next two chapters will focus on studying this approach experimentally in the wind tunnel.

8

WIND TUNNEL EXPERIMENTS: PITCH CONTROL

Für das können gibt es nur einen Beweis:
das Tun.

Marie von Ebner-Eschenbach, Austria (1830-1916),
on experimental work.

This chapter validates the data-driven control approach described in the first part of the thesis on an experimental setup of a pitch-controlled wind turbine, in a controlled wind tunnel environment. First, the experimental setup and the testing environment is described. Then, the implementation results of the two-step data-driven approach Subspace Predictive Repetitive Control (SPRC) are described, both for constant as well as changing wind conditions. The concept of IPC for yaw control is then demonstrated for the first time on an upwind turbine using the same setup. This approach uses the PI controllers, the gains of which can be tuned using Iterative Feedback Tuning (IFT). The use of IFT demonstrates the need for gain scheduling for this control approach. The chapter ends with lessons learnt from the wind tunnel experiments.

8.1. INTRODUCTION

It has been discussed in Chapter 6 that wind tunnel testing affords an interesting opportunity for evaluating load control strategies under controlled experimental conditions to understand the possibilities and limitations of their real-time implementation. Wind tunnel testing of smart rotors with trailing-edge flaps has been conducted in Van Wingerden et al. (2010b). A linear approach to Individual Pitch Control (IPC) for two-bladed wind turbines was also tested recently in the wind tunnel environment in Van Solingen et al. (2014).

Parts of this chapter have been published in the IEEE Transactions on Control Systems Technology **23**, Navalkar et al. (2015).

It is considered that the control paradigm of Subspace Predictive Repetitive Control (SPRC) would be suitable for the real-time implementation of load control for wind turbines because of the following reasons:

- As a repetitive control methodology, SPRC exploits the periodic nature of wind turbine loads to maximise load reduction potential. Further, as a fully multivariable approach, it is easily extendable to new actuators, like trailing-edge flaps, and new sensors, such as LIDAR.
- As shown using aeroelastic simulations, the use of basis functions for identification and/or control ensures that the control input signals are precisely controlled in terms of their shape and smoothness. Herewith, it is possible to constrain the pitch actuator duty cycle and increase actuator life.
- Since online system identification is an integral part of SPRC, this control methodology is able to adapt to changes, either in wind conditions or in turbine properties, and thereby maintain optimal performance.
- As a predictive control methodology, the control input optimisation step can be performed over a finite horizon, enabling the inclusion of constraint handling capabilities in the algorithm.

Conventional IPC-MBC, as described in Chapter 6, is fundamentally limited by the assumption that the rotor is perfectly balanced, both in terms of mass distribution as well as aerodynamic performance. For practical rotors, this assumption may not always be true, due to manufacturing limitations or asymmetric aging. This asymmetric load can strongly influence the periodic turbine loads, adversely affecting its fatigue life. Since SPRC makes no assumption regarding symmetry, it is suitable for correcting for imbalances in non-ideal rotors.

This chapter first describes the scaled wind turbine and the wind tunnel environment used for performing the experiments. Then, the experimental results obtained from the implementation of SPRC are discussed, for both constant and varying wind conditions.

The previous chapter described the potential for achieving yaw control using IPC. Such a controller would be interesting from the point of view of redistributing turbine loads, i.e. achieving the desired trade-off between rotor and support structure loads. This concept, demonstrated in the simulation environment in the previous chapter, is explored experimentally here using the same setup and wind tunnel conditions. For this new control approach, a simple PI controller is used in the MBC domain, as described in the previous chapter. Iterative Feedback Tuning (IFT) is used to explore the behaviour of the ideal controller gains, with respect to the operating wind speed.

The next section focusses on the design and layout of the experimental setup used for pitch control in this chapter, and extended to include flap control in the next chapter.

8.2. EXPERIMENTAL SETUP

The experimental setup has been designed in such a manner that it can be used for investigating the proposed control strategy in a realistic setting. The prototype turbine can

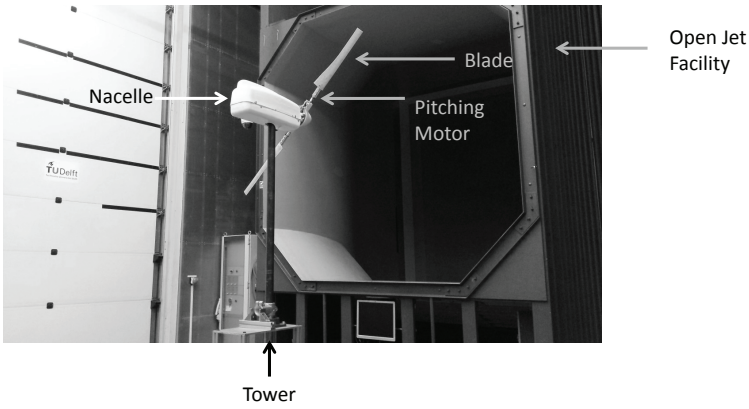


Figure 8.1: Photograph of wind turbine (rotor diameter 1.6 m) at the mouth of the open jet of the wind tunnel.

Table 8.1: Parameter comparison of the scaled turbine

	Reference turbine, Bak et al. (2013)	Scaled rotor
Rated wind speed (m/s)	11.4	6.5
Tip speed ratio (-)	7.86	3
Rated rotational speed (rpm)	9.6	230
Fore-aft tower mode (Hz)	0.25	20.73
Side-side tower mode (Hz)	0.25	20.73
First flapwise mode (Hz)	0.56	15
Ratio 1st blade freq. to 1P (-)	3.5	3.913

be visualised in Fig. 8.1, it is two-bladed machine of rotor diameter 1.6 m. The main characteristics of the turbine are compared against those of the INNWIND 10 MW reference turbine, Bak et al. (2013), in Table 8.1. The primary ratio sought to be retained constant in the scaling process is the ratio of the first blade eigenfrequency to the 1P frequency (= the turbine rotor speed).

The main objective of tower design is the avoidance of structural resonance; since the experimental turbine is two-bladed, rather than the conventional three-bladed configuration of the reference turbine, it is essential to avoid the exogenous tower excitation frequencies which occur at the even multiples of the rotor speed, thus the 2P frequency and its harmonics. For the reference turbine, the forcing frequencies occur instead at 3P and its harmonics. As such, since the forcing frequencies of the experimental turbine are lower than those of the reference turbine, the tower is designed such that its modes fall in the stiff-stiff zone, with the first mode being fore-aft, lying well beyond the frequency 4P. The second tower mode, which is the side-side mode, is at a similar fre-

quency. Although it shows lower damping, it is also excited to a lower extent. Practically, this stiff-stiff approach is conservative, and leads to greater material requirements. However, in this manner, it is ensured that the blade and tower modes do not couple and it is more straightforward to interpret rotor load attenuation results.

The turbine hub is equipped with two blades that are able to pitch longitudinally along their axes, with the use of servomotors. These actuators form the primary actuators that will be used for load control in this chapter. The main components of the experimental setup are herewith the wind tunnel itself, the hub with the instrumented pitchable blades, the instrumented nacelle, the support structure, and the real-time control system. Each component is described in detail in the next sections.

8.2.1. WIND TUNNEL

The experiments are conducted in the Open Jet Facility (OJF) at the Delft University of Technology. Scaled wind turbine testing for the study of wind turbine load alleviation has been conducted in this wind tunnel previously, see Van Wingerden et al. (2010b), Van Solingen et al. (2014), and Verelst et al. (2012). This wind tunnel has a closed-circuit, recirculating design. The open jet of the nozzle has an effective diameter of 3 m, while the test cross-section has a size of 6 m × 6.5 m. As such, the largest rotor diameter of scaled turbines that can be tested in this wind tunnel is approximately 2 m. The fan of the wind tunnel is rated at 500 kW, and it is able to furnish wind speeds of upto 3 5 m/s. The wind flow shows negligible values of turbulence, although work is underway to be able to manipulate the turbulence characteristics of the wind tunnel air flow. The air flow past the test cross-section is collected through turbulence screens and fed back to the fan. The heat generated is extracted from the air flow with the use of heat exchangers. The experimental turbine is fixed rigidly to the wind tunnel table in the test cross-section such that it points directly upwind into the open jet of the wind tunnel.

8.2.2. BLADES AND HUB

The blades of the scaled wind turbine were designed for a previous experiment by Verelst et al. (2012) in order to evaluate the behaviour of a downwind turbine in free yaw. The blades consist of a wooden spar that provides structural support to a foam matrix with a smooth aerodynamic outer surface. The blade is 555 mm in length, with a chord that tapers from 100 mm at the blade root to 60 mm at the blade tip. For ensuring the ideal angle of attack along the blade span, the blade is twisted along its length, with a maximum twist of -9.562° twist at the root. The blade cross-section forms two distinct aerofoils, with the NREL S822, of thickness 16% used at the blade tip, while the NREL S823, of thickness 21% is used elsewhere. A detailed analysis of the aerodynamic design of the blade can be found in Verelst et al. (2012).

Each of the spars is instrumented with two strain gauges, one located at the root of the blade, and another located 30% outboard along the blade. Although both of these sensors can be used for feedback, in practice, since full-span pitch control is tested, only the sensors at the blade root are used for feedback.

Each of the blades is able to pitch longitudinally about its axis. The actuation necessary herefor is achieved through a Dynamixel MX-106 motor each, the shaft of which is connected rigidly to the respective blade roots. The housing of the motors is connected

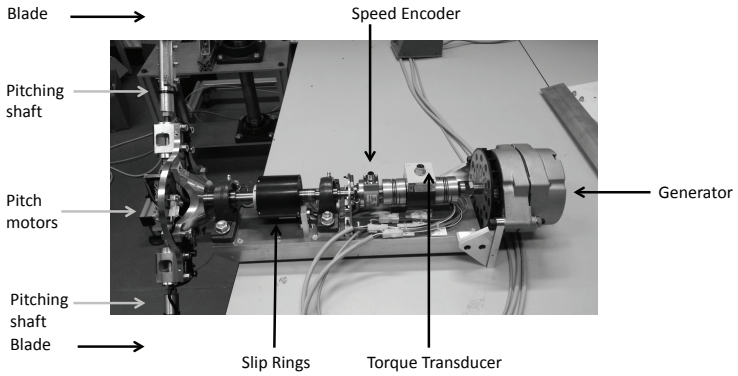


Figure 8.2: Photograph of wind turbine hub-nacelle assembly with the nacelle cover and hub cap removed.

rigidly to the hub; thus the blades are able to pitch about their axis relative to the hub. The servomotor has a relatively high bandwidth of 15 Hz, which is considered sufficiently for rejecting the 1P blade loads. The bandwidth degrades minimally under wind loading. While the blades and the pitching mechanisms were designed and manufactured to be as symmetric as possible, it can be seen from the results section that there exists still some degree of imbalance in the blade-hub assembly and pitch control authority. The connection between the blades and the hub, and the connection between the hub and the nacelle can be visualised in Fig. 8.2.

8.2.3. NACELLE AND TOWER

The stationary part of the turbine, located on top of the tower, to which the rotating hub-blade assembly is attached, is termed as the nacelle, an exploded view of the nacelle of the experimental turbine can be seen in Fig. 8.2. Principally, the nacelle consists of an alumnium bedplate that provides structural support to the turbine rotor and the generator, as well as the transmission system. The generator used is the Delco 12SI dynamo, able to operate in a speed range corresponding to that of the rotor. As such, a gearbox is not required, and the rotor is connected directly through the instrumented main shaft to the generator. The mainshaft is supported on the bedplate by means of two main bearings.

The main shaft is instrumented with a torque transducer and a speed encoder, such that the operational condition of the wind turbine can be tracked. These sensors can be seen in Fig. 8.2. Also seen in the figure is a set of slip rings, which are meant to transfer the strain sensor signals and the pitch actuation signals between the rotating part of the wind turbine, i. e. the hub and blades to the stationary part of the turbine, which is the nacelle.

The nacelle is connected rigidly to the top of the tubular steel tower. Unlike modern wind turbines, the nacelle cannot yaw relative to the tower. However, the tower as a whole is supported on its base by means of two bearings, and thus the entire turbine is

able to yaw about its tower base. The turbine can be locked in yaw such that the rotor plane makes an angle with the wind inflow; herewith, it is possible to simulate the loads associated with yawed inflow with this experimental setup. For the purpose of these experiments, the turbine is locked in a yaw position such that the rotor plane is almost perfectly normal to the direction of wind flow.

8.2.4. CONTROL AND THE REAL-TIME ENVIRONMENT

As discussed in Chapter 6, a conventional turbine requires the use of generator torque control and pitch control in order to ensure nominal operation. For the current set of experiments, these two control degrees of freedom are commanded independently.

It is required to control the torque imposed upon the turbine generator in order to limit turbine speed-up. For this scaled turbine, the torque is controlled indirectly by means of connecting the generator in series with a dump load of variable resistance. By manipulating the value of this resistance in real time, it is possible to command generator current and hence the electrical torque imposed upon the generator. For the current set of experiments, the value of the resistance is kept constant for the duration of each experiment, and hence, the turbine is under constant load operation. As such, torque control is kept decoupled from pitch control. However, the setup offers in principle the potential to perform torque and pitch control simultaneously. It should be noted that the dump load also serves the purpose of dissipating the energy extracted from the air stream during the experiment.

Pitch control is performed with the objective of reducing turbine rotor loads, as measured by the strain gauges[1]. The pitch control loop, that acts on the feedback received from the strain gauges, is hence decoupled from the torque control loop that provides constant load speed regulation. Control is implemented physically by the use of the xPC Target environment provided by Simulink, Mathworks (2014b). The controller is first designed in the Matlab, Mathworks (2014a), and Simulink environment and then compiled to a real-time target PC. The target PC reads sensors signals from the wind turbine and issues commands to the pitch and torque actuators. The input-output communication is achieved through the use of the National Instruments data acquistion system PCI-6259.

Fig. 8.3 shows the block diagram of the experiemental setup. As can be seen, the generator speed is controlled by setting a constant value for the resistance of the dump load. The strain gauges measure rotor loads; these signals is used as feedback by the IPC controller that commands the blade pitch signals individually in such a manner that the blade loads are minimised. The controller used for synthesising individual pitch actions is encapsulated within the light grey box. As motivated in the introduction, the SPRC methodology developed in Chapter 4 will be used for generating IPC control actions.

The results of using SPRC for performing IPC, on this realistic scaled wind turbine setup, are described in the next section.

[1]Although pitch control can also be used for speed regualtion, this application is not explored in the current set of experiments and this chapter studies the exclusive use of pitch control for rotor load reduction.

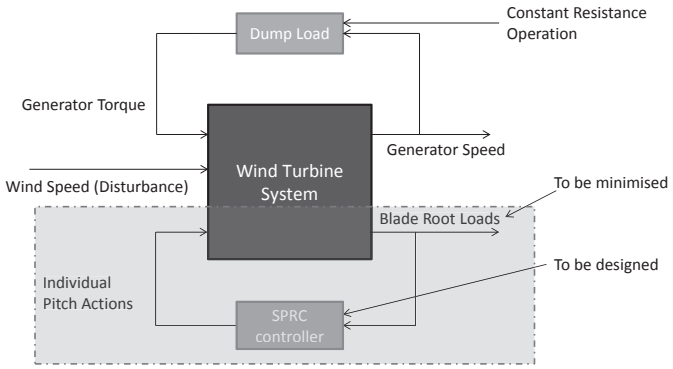


Figure 8.3: Block diagram of the experimental setup.

8.3. EXPERIMENTS: IPC WITH SPRC

The objective of IPC is to reduce rotor loads by pitching the wind turbine blade independently of each other. The primary dynamic loads on the scaled wind turbine originate from tower shadow; they are periodic with fundamental period 1P, equal to rotor speed. The dominant load peaks in the baseline-controlled case for the scaled wind turbine can be seen in Fig. 8.4.

For a field turbine, as seen in the aeroelastic simulations from the previous chapters, the dynamic loading at 1P and its harmonics is also contributed to by the effects of wind shear, yawed inflow and the rotational sampling of turbulence, none of these effects are present to any significant extent in the current wind tunnel experiments. Further, the effect of turbulence is to broaden the loading peaks in the frequency spectrum, increasing energy across a wider range of frequencies. However, even for field turbines, the primary dynamic loading components in the blade root load spectrum occur at the 1P and the 2P frequencies. Hence, the SPRC control law used in this section for rotor load control will use the same 1P and 2P basis functions as before. Not only will this achieve more targetted load control, it will also ensure that the computationally expensive data-driven algorithm is still tractable in the real-time environment qua computational complexity. As such, the basis functions used in this section are the same as those used in Chapter 7, Section 2:

$$\phi_u = \phi_y = \begin{bmatrix} \sin(2\pi/N) & \sin(4\pi/N) & \cdots & \sin(2\pi) \\ \cos(2\pi/N) & \cos(4\pi/N) & \cdots & \cos(2\pi) \\ \sin(4\pi/N) & \sin(8\pi/N) & \cdots & \sin(4\pi) \\ \cos(4\pi/N) & \cos(8\pi/N) & \cdots & \cos(4\pi) \end{bmatrix} \otimes I_2. \tag{8.1}$$

It should be noted that since the scaled turbine is under constant load operation, its rotor speed is not constant, and hence the period N varies over time. It should however be noted that due to the incorporation of the speed encode on the main shaft of the test turbine, the rotor azimuth and the rotational speed are both quantities directly measurable by the controller. As such, the basis functions are scheduled on the value of the azimuth instead of on time. Such time-varying phase-locked basis functions achieve load con-

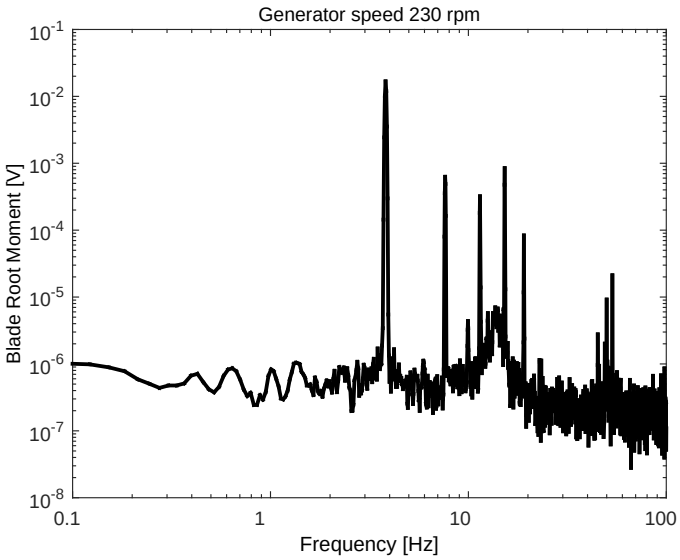


Figure 8.4: Blade root load spectrum at nominal operation, baseline control.

trol with smooth actuator signals while ensuring robustness to changes in the rotational period of the turbine. This rotor-locked implementation circumvents one of the largest practical issues in the implementation of repetitive control strategies, i.e. small changes in the period of the disturbance often drastically deteriorate conventional repetitive control performance if time-invariant basis functions are used.

The implementation of SPRC for rotor load alleviation using these basis functions is studied using the experimental setup described in the previous section. The mean operational wind speed during the current set of experiments is varied between 5.8 m/s and 10.5 m/s. Since the turbine is under constant load operation, its rotor speed varies linearly with the wind speed. As such, the mentioned wind speed range corresponds to a rotor speed range of 166 rpm to 400 rpm.

The behaviour of SPRC was tested for different wind conditions, and representative results are presented in this section. First, the effect of SPRC under constant wind conditions is studied. The tuning of the convergence of the SPRC algorithm is then presented. After this, the response of the controller to changing wind conditions is described. Finally, the advantage of a controller, that is able to adapt to the variation in system dynamics, is demonstrated experimentally.

8.3.1. CONSTANT OPERATING CONDITIONS

The SPRC load controller is tested under different constant operating wind speeds, within the range given in the previous section, and it is found to behave satisfactorily and deliver load reductions at all oeprating speeds. This section presents representative results for a constant operating wind speed of 6.5 m/s, which corresponds to a

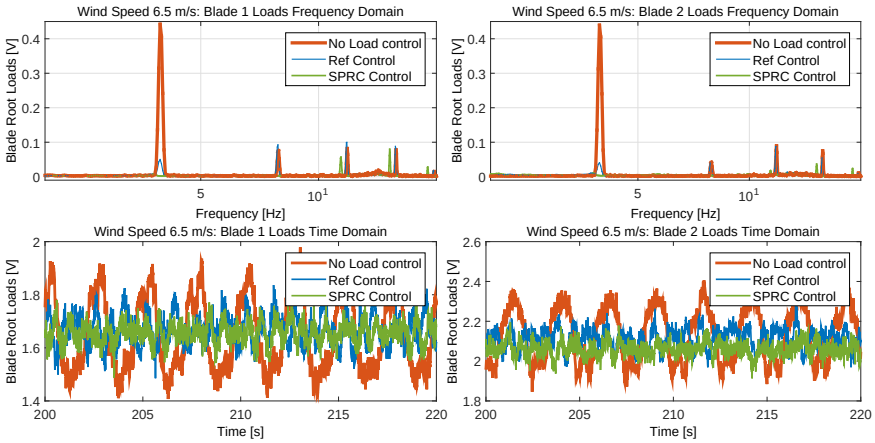


Figure 8.5: Blade root load reduction through the action of SPRC.

rotor speed of 230 rpm. For the purpose of comparison, the performance of the SPRC controller is presented along with the load control performance of an H_∞ controller used in conjunction with a linear coördinate transform, Van Solingen et al. (2014). This controller is defined as the reference controller (Ref), and it should be noted that it is designed to only target the load peaks at the 1P frequency.

When the SPRC controller is implemented, both the identification and controller synthesis steps converge within a matter of seconds, and substantial load reductions are observed, with the 1P and 2P frequency peaks almost entirely eliminated. This can be seen in Fig. 8.5. The reduction in blade loads is to an extent of 65% for Blade 1 and 74.5% for Blade 2. Since the tunnel wind speed is controlled manually, it is not exactly identical across all time series, and as such, the 1P peaks and its harmonics move slightly around their nominal position in the frequency spectra. This variation is considered to be small such that the conclusions derived from the results in this section should still hold in the case where the wind speed is exactly repeatable across all time series.

It can be seen that even under these constant operating conditions, the SPRC controller outperforms the linear controller. This is a consequence of performing load control purely at the 1P and 2P frequencies, independently for the two blades. The other frequencies in the load spectrum either get attenuated, such as the 3P peak for Blade 2, or remain unaffected. By extending the basis functions to include sinusoidal basis functions at 3P, 4P and higher frequencies, it could be possible to achieve even greater load reductions. However, since these load peaks are relatively unimportant in the load spectrum of field turbine blades, such an extension is not attempted. Load reduction is hence limited to that at the 1P and 2P frequencies in this section.

The pitch actuation signals can be seen in Fig. 8.6, It can be seen that the pitch actuation spectrum contains energy only at the 1P and 2P frequencies; SPRC thereby ensures precise control over the smoothness of actuator signal, as compared to a PI controller,

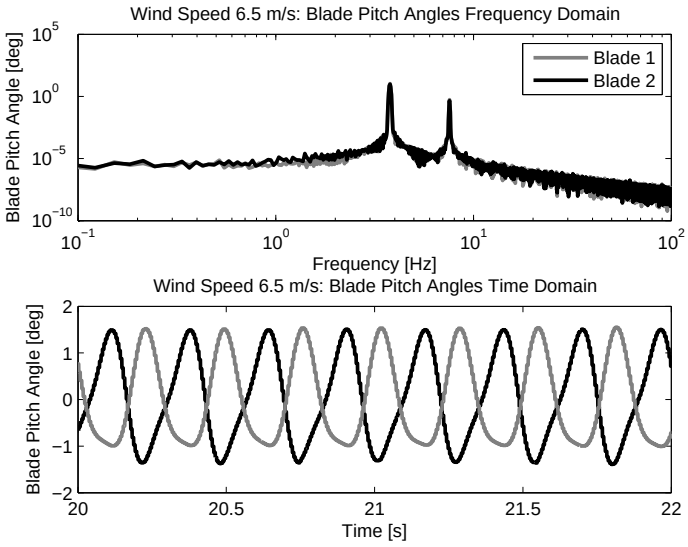


Figure 8.6: Blade pitch actuation signals commanded by SPRC.

the actuation commanded by which contains energy across a large range of the spectrum.

Further, it is noteworthy that, since the manufactured turbine blades and pitch mechanisms are not perfectly symmetric, SPRC accounts for this asymmetry by generating asymmetric control actions. In conventional IPC with the MBC transform, the IPC actions are constrained to be perfectly antisymmetric for the two blades, however, this is not a constraint with SPRC, and it is able to achieve greater load reductions even in the case where there exists a small but finite amount of mass or aerodynamic imbalance.

The convergence of the SPRC algorithm is a function of the signal-to-noise ratio, and it can be tuned by the user as shown in the next section in a manner best suited for the level of external noise in the data.

8.3.2. CONVERGENCE TUNING

In order to ensure that SPRC convergence is adequately fast and stable, it is possible to add a tuning parameter $0 < \beta < 1$ to the SPRC control update law in the manner shown below:

$$\theta_j = \theta_{j-1} - \beta K_{f,j} \begin{bmatrix} \bar{Y}_{j-1} \\ \delta\theta_{j-1} \\ \delta\bar{Y}_{j-1} \end{bmatrix}. \tag{8.2}$$

Here, the stacked control input $\bar{U}$, projected in the basis function space, has been replaced by the angle θ, which essentially corresponds to the amplitude of pitch activity at the 1P and 2P frequencies. Further, the period index j replaces the iteration index k, since one iteration is considered to correspond to one period. Finally, the term β can

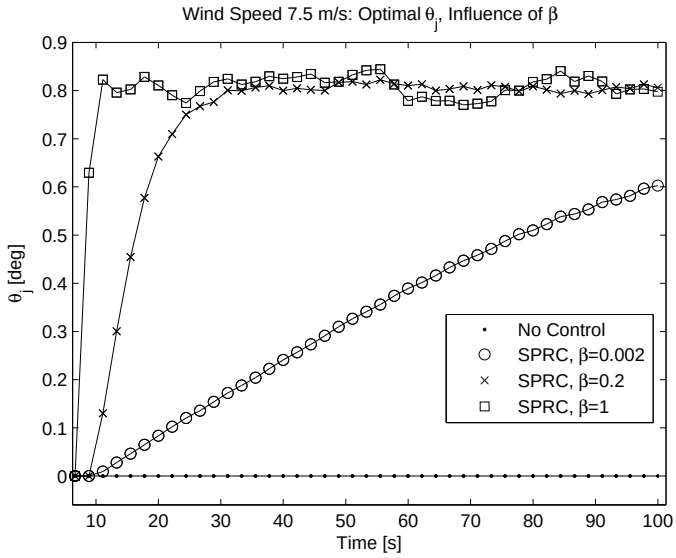


Figure 8.7: Convergence of pitch activity by adjusting tuning factor β.

be manipulated by the user to achieve the desired SPRC convergence. A low value reduces convergence speed, and responsiveness to changes in the system dynamics. On the other hand, it also reduces the sensitivity of the algorithm to noise. Conversely, a high value speeds up convergence but increases sensitivity to noise. The effect of changing the value of β are depicted in this section. Experiments are carried out at a constant wind speed of 7.5 m/s. Three different values of β are tested, and in each experiment, the initial estimate of the system Markov parameters is set to 0, to evaluate the convergence behaviour of SPRC. Load reduction results can be seen in Fig. 8.7 and Fig. 8.8. For the low value of $\beta = 0.002$, convergence is slow but smooth, while for the value of $\beta = 1$, it is rapid but vacillatory. In all cases, the system stably converges to the optimal load control law. Further, the steady-state value of load alleviation is independent of the value of β.

While these results have been obtained for constant operating conditions, the response to variable wind conditions are studied next.

8.3.3. VARYING WIND CONDITIONS: NONADAPTIVE SPRC

For a wind turbine under normal operation, there may occur a sudden and large change in the mean wind speed or wind distribution across the rotor plane. Since flow-modifying actuators, like the pitch control actuators, possess control authority that strongly depends on the operating wind speed, it may be necessary to adjust the control strategy according to the changed operating conditions. As such, the response of both adaptive and non-adaptive SPRC to changed operating conditions is explored in this section.

For the experimental wind turbine, a change in the wind speed leads to a change

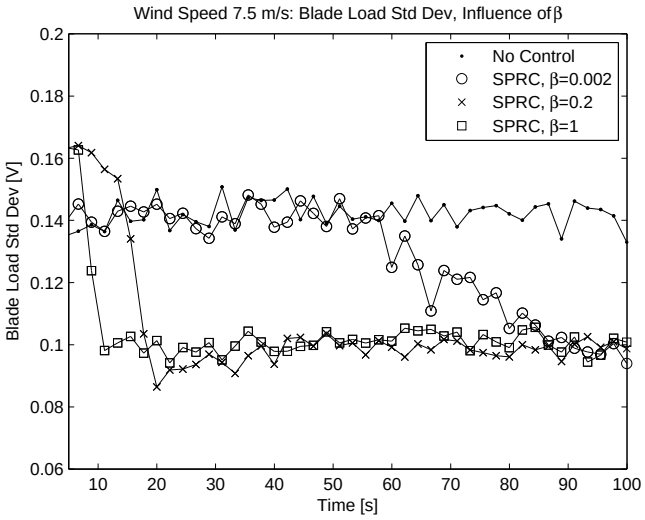


Figure 8.8: Convergence of load reduction by adjusting tuning factor β.

in the rotor speed, due to the constant load nature of operation. This behaviour is not seen in commercial wind turbines in above-rated operation. However, the objective of this section is to explore the effect of changed control authority on the load control algorithm. Further, a change in turbine rotor speed mainly impacts the mean turbine loads, while the load variation is relatively insensitive to the rotor speed. Hence, in this case, the changes in the structural loading arising out of variations in rotor speed, are considered negligible.

8

In this section, an offline, nonadaptive version of the SPRC law was generated. Specifically, system identification was done based on data collected at 6.5 m/s, and a repetitive control law was synthesised. The objective of the current experiments was to evaluate whether such an LTI controller is able to handle changes in operating conditions. In the first experiment, the system was allowed to stabilise at a wind speed of 6.5 m/s, with the non-adaptive SPRC law active. As it was designed for this wind conditions, it is able to achieve good load reductions. Then, the wind speed was reduced to 5.8 m/s, within a time period of 40 seconds. In the pitch activity signal, Fig. 8.9 and the blade load signal, Fig. 8.10, the wind speed change causes a transient between 200 and 240 seconds. The initial transient in these signals is the start-up transient.

It should be noted that the wind speed signal is not exactly identical for the two datasets shown in these figures, since the wind speed is manually controlled. However, the difference in wind speed is not expected to alter the conclusions drawn from these results.

The non-adaptive SPRC control strategy is still able to achieve load reductions in the changed operating conditions. At reduced wind speeds, the control authority of the actuators reduces, and hence more control effort is required, as seen in Fig. 8.9. However,

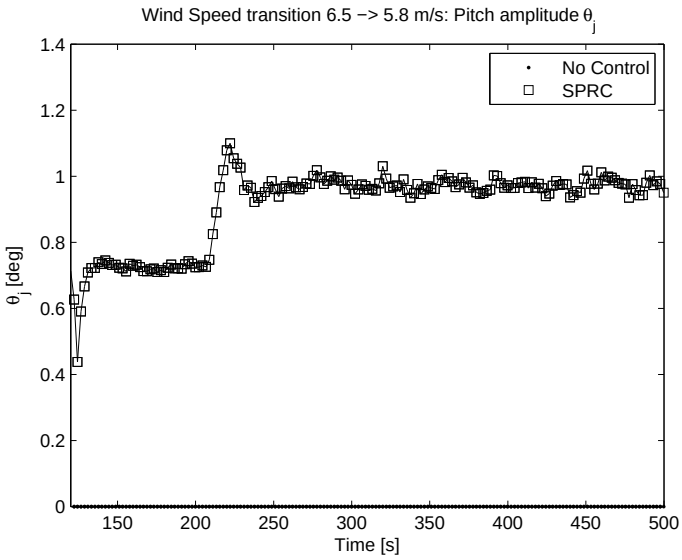


Figure 8.9: Change in pitch activity, non-adaptive SPRC, 6.5 m/s to 5.8 m/s between 200-240 seconds.

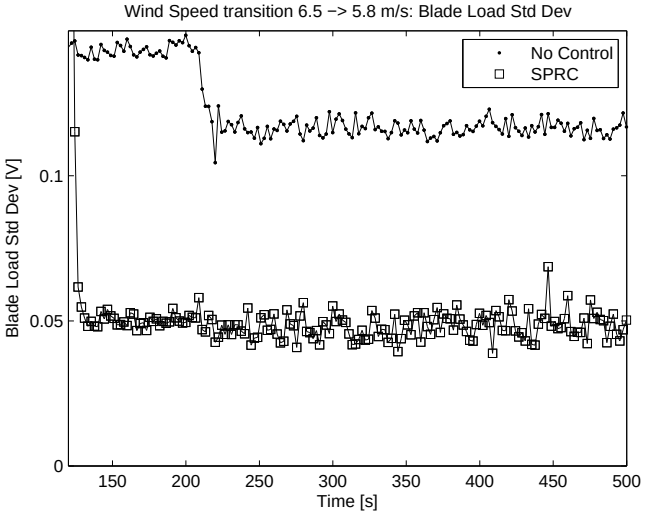


Figure 8.10: Change in blade loads, non-adaptive SPRC, 6.5 m/s to 5.8 m/s between 200-240 seconds.

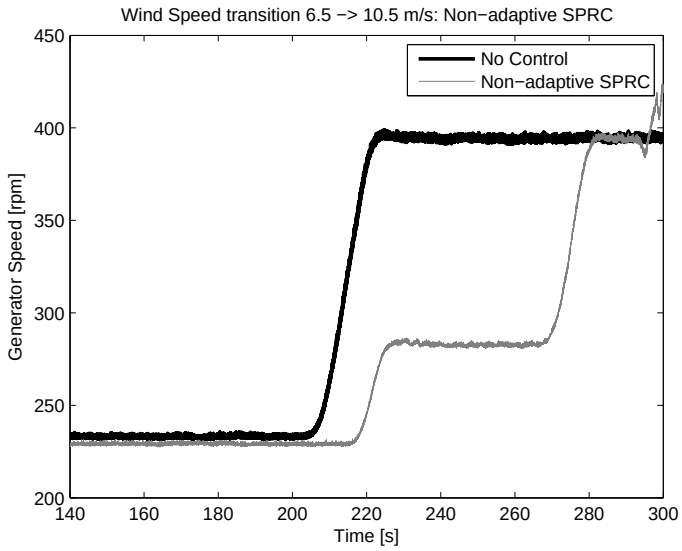


Figure 8.11: Change in wind turbine rotor speed, non-adaptive SPRC, 6.5 m/s to 10.5 m/s.

due to a mismatch in the plant and the controller, the percentage reduction in loads is clearer lower at the off-design wind speeds, as in Fig. 8.10.

In the next experiment, the scaled turbine, operating at a steady wind speed value of 6.5 m/s, is subject to a sudden 60% increase in wind speed from 6.5 m/s to 10.5 m/s. Such a rapid increase in wind speed may be caused by extreme wind gusts in the field. For the scaled wind turbine, this corresponds to an increase in rotational speed from 230 rpm to 400 rpm. The same non-adaptive SPRC controller is used for load control; at the end of the wind speed transition, the system is no longer stable, and the pitch control input and load signals grow unboundedly.

Further, even when the experiment is conducted with two smaller steps in the wind speed signal (230 rpm to 280 rpm, 280 rpm to 400 rpm), the system still becomes unstable at the higher wind speed. To ensure that the setup does not undergo permanent damage, the experiments are aborted through emergency wind tunnel shutdown and pitching the turbine blades rapidly to feather. The aborted experiment can be visualised in Fig. 8.11 and Fig. 8.12.

Thus, the non-adaptive version of the algorithm cannot stabilise the system when there occurs a sudden and large change in system dynamics.

8.3.4. VARYING WIND CONDITIONS: ADAPTIVE SPRC

In this section, a fully adaptive version of the SPRC algorithm will be explored, which involves continuous system identification. Simultaneously, a repetitive control law is synthesised and implemented at every time step, based on the system Markov parameters obtained from the recursive identification step.

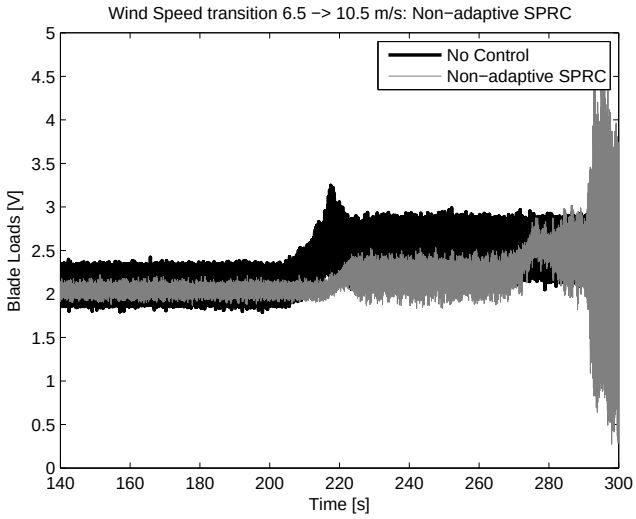


Figure 8.12: Change in blade loads, non-adaptive SPRC, 6.5 m/s to 10.5 m/s.

Three experiments are conducted, in order to test the response to the same wind speed transition, i. e. from 6.5 m/s to 10.5 m/s. Three different control cases are tested: first, no load controller is incorporated in the loop, and only baseline control is active. In the second experiment, two non-adaptive SPRC controllers, tuned respectively for the two different wind speeds of 6.5 m/s and 10.5 m/s, are switched on manually, before and after the wind speed transitions. Finally, a fully adaptive SPRC controller is used for load control throughout the experiment, and it is implemented without manual intervention.

The effect of the change in wind speed on the regulated generator speed can be visualised in Fig. 8.13. For all three experiments, the effect of the change in wind speed on the change in rotor speed is virtually identical, because the baseline speed controller is the same. As the wind speed increases by 60% from 6.5 m/s to 10.5 m/s within 30 seconds, the rotor speed also almost doubles from 230 rpm to 400 rpm, in a manner similar to the experiments of the previous section. As the wind speed in the tunnel is manually controlled, the time series are close but not identical.

From Fig. 8.14, it can be seen that the two non-adaptive SPRC controllers, designed for the two different operating wind speeds of 6.5 m/s and 10.5 m/s, are able to achieve significant load reduction when switched on after the wind turbine has stabilised to its new operating points. This result is not particularly unexpected, since the non-adaptive SPRC controllers are essentially LTI, tuned for those specific operating points. It is more interesting to note that the fully adaptive SPRC controller is able to recalibrate itself autonomously after the wind speed transition, and achieve load reductions both before and after the change in operating conditions, without manual intervention. This can be compared with the previous section, where an LTI non-adaptive SPRC controller was unable to retain closed-loop stability under the off-design conditions.

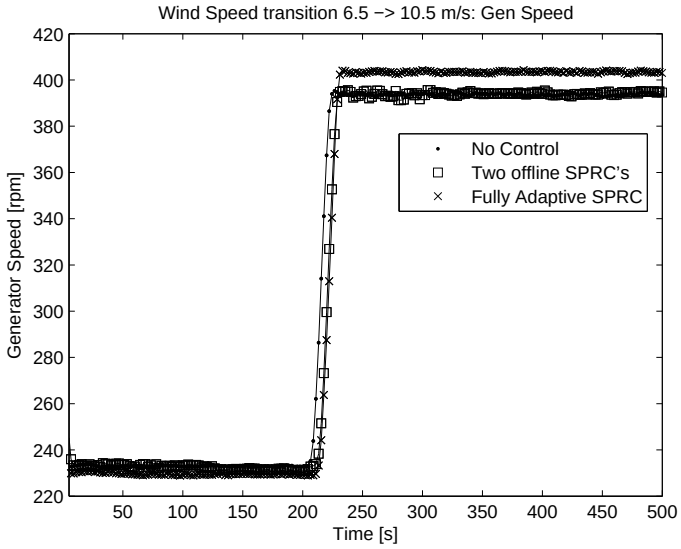


Figure 8.13: Change in wind turbine rotor speed, 6.5 m/s to 10.5 m/s.

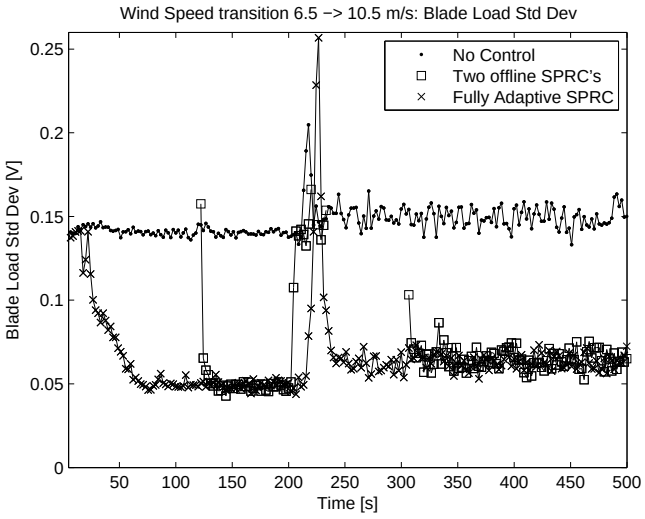


Figure 8.14: Change in blade loads, 6.5 m/s to 10.5 m/s.

Further, it is interesting to note that the level of load reduction achieved by the fully adaptive SPRC controller is identical to the load reductions achieved by the two LTI non-adaptive controllers that were optimised for their respective operating conditions. Thus, although adaptive SPRC also requires additional persistency of excitation superposed on the control signal, it is still able to adjust to changes in system dynamics without suffering a degradation in performance. Overall, load reductions of more than 60% are achieved with the fully adaptive SPRC controller.

Thus, this section shows that SPRC is able to reject the 1P and 2P blade loads entirely, starting with very limited system knowledge, both under constant and varying wind conditions. Further, load reductions are achieved with near-perfect control over the frequency content of the actuator signal, and asymmetry in control authority can be compensated for, as opposed to traditional IPC-MBC methods. Finally, unlike an LTI controller that shows instability under off-design conditions, SPRC is able to adjust to changed system dynamics and recalibrate itself within a matter of a few iterations, to achieve stable load alleviation.

8.4. EXPERIMENTS: IPC FOR YAW CONTROL

While the previous section explores the use of IPC for load control, the concept of IPC for yaw control, demonstrated using aeroelastic simulations in the previous chapter, will be explored experimentally here for the first time on a physical wind turbine. The setup described in Section 8.2 is also used for performing this set of experiments, with a few adjustments. The tower, and along with it, the nacelle and the blades, are allowed to yaw freely about the tower base; stops are installed to limit the yaw error to within ±20°. A photograph of this system can be seen in Fig. 8.15.

Below the lower tower yaw bearing, a small metal extension is attached rigidly to the yawing tower. A fixed laser sensor points directly at this extension, which approachs towards and recedes from laser sensor as the tower yaws. As such, the output of the laser sensor provides a measure of the yaw error of the turbine. For small yaw angles, the relationship between the yaw angle and the laser output may be assumed linear. Further, the output of the laser sensor is calibrated such that the zero coincides with the turbine position with rotor plane exactly perpendicular to the wind flow direction.

The output of this sensor is used as feedback for the IPC yaw controller. As described in the previous chapter, the yaw error χ is first fed to a PI controller, with manually tuned gains, K_p and K_i, to obtain the pitch action in the stationary frame of reference, $P_{q,\chi}$. Then, this pitch action in the stationary frame of reference is converted into blade pitch action using the inverse MBC transform from the previous chapter. The deadband saturator described in the previous chapter was found to be unnecessary, and hence it was not used during the experiments. It should be noted that the turbine used in the previous chapter was a three-bladed commercial turbine, while the scaled turbine is a two-bladed machine. Accordingly, the inverse MBC transform is required to be adjusted:

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} \cos\Psi & \sin\Psi \\ \cos(\Psi+\pi) & \sin(\Psi+\pi) \end{bmatrix} \begin{bmatrix} 0 \\ P_{d,\chi} \end{bmatrix}. \tag{8.3}$$

The pitch commands P_1 and P_2 are then issued to blade 1 and 2, based on the measured blade 1 azimuth Ψ. These control signals then act to produce a yawing moment

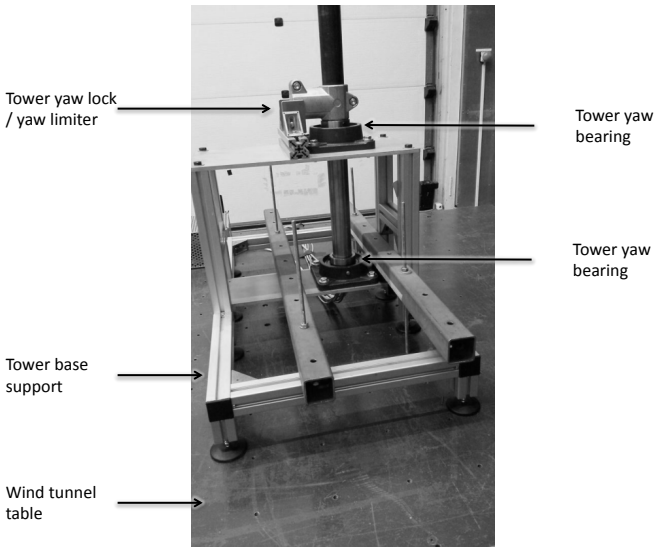


Figure 8.15: Photograph of the yawable tower and its connection to the fixed tower base.

about the tower base that rejects yaw error and keeps the turbine aligned with the wind flow.

First, the results obtained from using a manually-tuned PI controller are discussed. Then, IFT-LTI described in Chapter 2 will be applied to this system at a constant wind speed, to optimally tune the K_p gain of a proportional controller for IPC-yaw control. Then, IFT will be used to determine if it is necessary to gain-schedule this system for different wind speeds, and differential pitch control authority.

8.4.1. IPC FOR YAW CONTROL: MANUALLY TUNED PI CONTROL

First, in order to validate the concept of IPC for yaw control, the PI controller, defined in Equation (7.16), is tuned manually. The wind turbine is operated in free yaw under wind conditions of 7.5 m/s, and it is seen that when the IPC controller is not active, the turbine is not stable upwind in the yaw degree of freedom. The turbine tends to move downwind, and it is held in its extreme position of $\pm 20^\circ$ by the yaw angle limiters. However, when the IPC controller is active and connected in closed loop with the feedback measurements of the yaw error sensor, the system is stable and rejects yaw error such that the rotor plane faces upwind and is perpendicular to the direction of inflow. Herewith, for the first time, IPC for yaw control has been demonstrated experimentally.

Next, it is shown that the IPC for yaw control is able to follow a wind direction change. Since the direction of wind flow in the wind tunnel cannot be altered, this capability of IPC is demonstrated by commanding a reference yaw trajectory that the IPC controller is required to follow. Such a reference trajectory is artificially created, and consists of a square wave between the extreme yaw angles of $\pm 8^\circ$, with a time period of 10 seconds. This reference trajectory is well within the maximum yaw limits of $\pm 20^\circ$. Further,

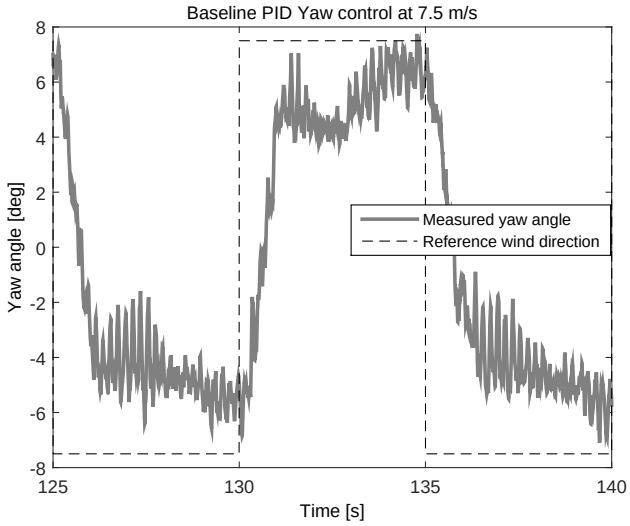


Figure 8.16: Yaw reference trajectory tracking using IPC with PI controller.

it should be noted that the yaw inertia of the turbine is large relative to the other degrees of freedom, and the frequency of 0.1 Hz is seen to be close to the bandwidth of the IPC-yaw actuator. The response of the turbine to this yaw command can be seen in Fig. 8.16, and it can be seen that although controller performance is not optimal, reference yaw trajectory tracking is possible with IPC for yaw control.

The pitch activity required to produce this yaw response can be seen in Fig. 8.17. There are several issues of note here: firstly, for the two blades, the pitch activity is exactly 180° out of phase. This is by design; since the MBC transform for two-bladed turbines is used, it is constrained to produce exactly antisymmetric control actions. Further, it can be seen that for positive yaw errors, there occurs a large positive transient in Blade 1, and a large negative transient in Blade 2; the signs of these transients are flipped when the sign of the yaw error changes abruptly. Finally, it can be observed that even when the turbine is to be held in one constant yaw position, fairly substantial IPC activity is still required to reject yaw error. This is a direct result of the fact that the turbine is not stable in any upwind position in the yaw degree of freedom.

In this section, the PI controller used is manually-tuned. The gains of the fixed-structure PI controller can also be tuned using Iterative Feedback Tuning (IFT), described in Chapter 2.

8.4.2. IPC FOR YAW CONTROL: IFT AT CONSTANT WIND SPEEDS

The modelling of the system from individual pitch control to the yaw degree of freedom is still an open question. System identification experiments showed that, for very small deviations in yaw, the system at constant wind speeds can be approximated as a first-order or overdamped second-order system. However, for most datasets, the VAF ob-

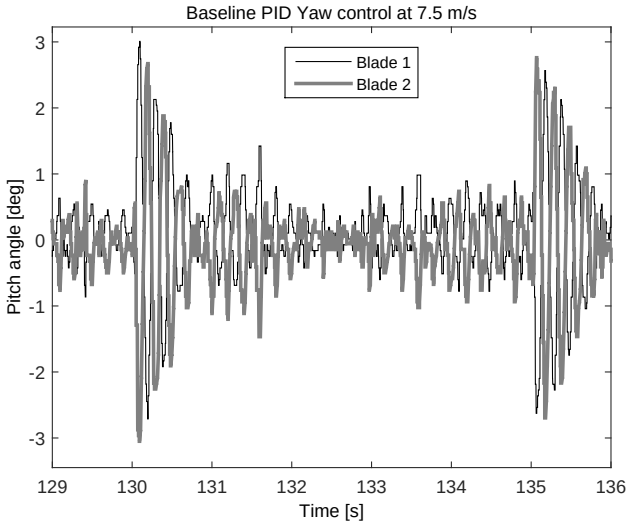


Figure 8.17: IPC pitch activity for yaw reference trajectory tracking with PI controller.

tained from system identification was poor, and it is postulated that the system is highly non-linear. Further, it is expected that the control authority of the pitch actuators will change with wind speed.

To understand, in a data-driven manner, the effect of wind speed and the need for gain scheduling, the direct approach of IFT is used to tune the gains of the PI controller, specifically, the proportional gain K_p. IFT experiments for an LTI system were conducted as per the established theory described in Chapter 2. The same reference trajectory was used as in the previous section, with one square wave cycle reperesenting one iteration. It can be seen in Fig. 8.18 that the K_p gains converge to value 60% higher within 600 seconds, and the integrated yaw error decays over these iterations to stabilise at a lower value.

The square wave response tuned by IFT is seen in Fig. 8.19. It can be seen that the rise time decreases after 18 iterations, and the step response is closer to the reference square wave. For further optimising the performance, it would be necessary to also tune the K_i gain in a similar manner.

This section assumes that the same controller gains can be used for different wind speeds. Further, it is assumed that the control action for both blades is required to be exactly antisymmetric. Both of these assumptions can be validated using IFT.

8.4.3. IPC FOR YAW CONTROL: CONTROLLER GAIN SENSITIVITY

The first-principles modelling of the IPC-yaw system is considered out of scope at present. Hence, the manner in which the system dynamics change as a function of wind speed is unknown. IFT can be used to determine the need for a gain scheduled controller by tuning LTI controller parameters for different constant wind speeds. This

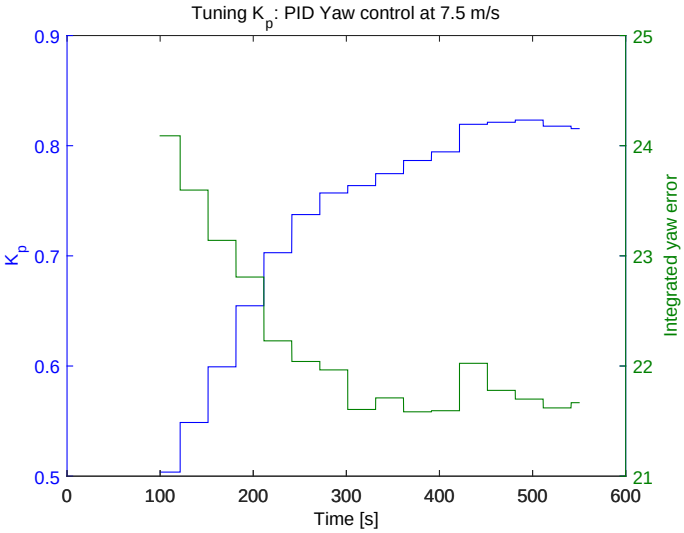


Figure 8.18: IFT tuning of K_p for IPC-yaw PI controller.

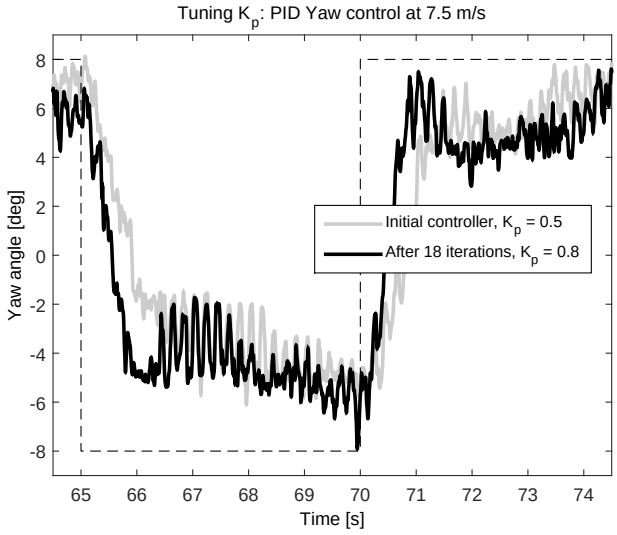


Figure 8.19: Square wave response after IFT tuning of K_p for IPC-yaw PI controller.

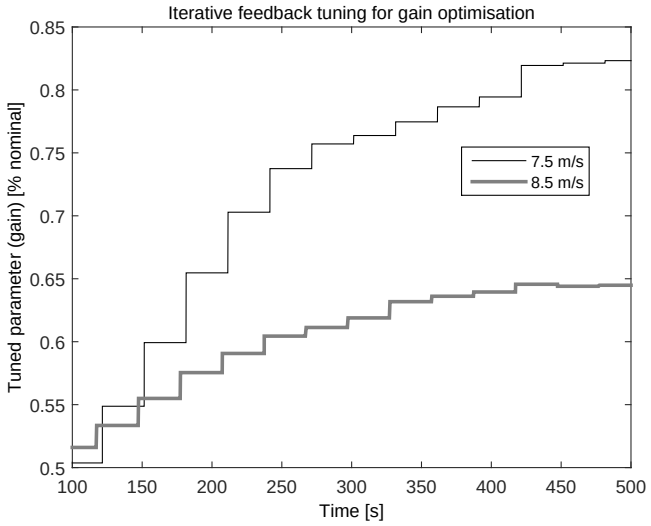


Figure 8.20: IFT tuning of K_p for two different wind speeds for the IPC-yaw PI controller.

procedure is followed for two different wind speeds in Fig. 8.20. It can be directly seen that for the two wind speeds, the ideal value of K_p to which IFT converges is different. For the higher wind speed, a lower value of K_p is required. Since the control authority of the IPC pitch actuators is higher at higher wind speeds, this trend is to be expected. It is also interesting to note that the ratio between the converged gains is approximately equal to the inverse of the square of the ratio between the wind speeds:

$$\frac{K_{p,IFT}|_{V_1}}{K_{p,IFT}|_{V_2}} \approx \frac{V_2^2}{V_1^2}. \tag{8.4}$$

With this equation, it is possible to determine a gain schedule for the IPC-yaw controller.

Next, the assumption that both blades require the equal but opposite K_p gains is challenged by using IFT to tune the gains of the two blades independently. Once again, a constant operating wind speed of 7.5 m/s is used, and the same reference trajectory as before is used. It can be seen from the Fig. 8.21 that the two blades do not have identical control authority, and a larger controller gain is required by Blade 2 than by Blade 1. The asymmetry in the pitch mechanism of commercial blades is likely to be lower, but IFT can still be used to tune the gains individually in order to achieve better control.

Thus, it has been shown experimentally in this section that IPC can stabilise an upwind turbine in yaw using a PI controller, the gains of which can be optimised using IFT. It is also shown that the PI controller gains should be gain scheduled for optimal performance at different wind speeds, and that using different gains for the two blades can increase performance.

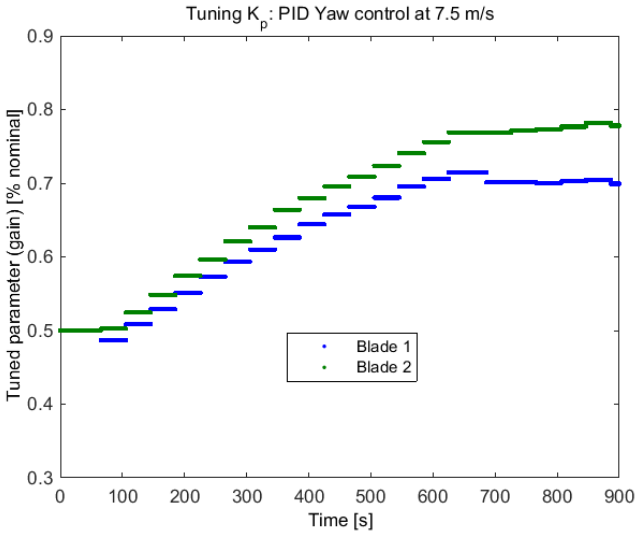


Figure 8.21: IFT tuning of two different K_p values for the two blades, for the IPC-yaw PI controller.

8.5. CONCLUSIONS

Overall, the experimental setup of the scaled, two-bladed turbine, described in this chapter, was shown capable of evaluating individual pitch control strategies, both for blade load reduction, as well as for turbine yaw control. This setup is the first to facilitate blade pitch control for wind turbines in the controlled, low-turbulence wind tunnel environment. It also provides the same control degrees of freedom, and demonstrates loading characteristics similar to a modern commercial wind turbine.

Subspace Predictive Repetitive Control (SPRC), a two-step data-driven approach to control, was demonstrated in real-time on this setup. SPRC, aimed specifically at minimising periodic loads, was shown to be suitable for the application of wind turbine load reduction: it was capable of nearly eliminating the 1P and 2P peaks from the blade load spectrum. This load alleviation was achieved in spite of the asymmetric control authority of the two blades, which was compensated for by the data-driven approach; a comparable LTI controller was unable to achieve the same degree of load control. Such asymmetric individual pitch has not been demonstrated in the wind industry before.

The algorithm incorporates an easy-to-tune convergence parameter, which can be increased to increase responsiveness to changes in sysem dynamics, or decreased to robustify the algorithm against noise, depending upon the application-specific noise levels. This tuning parameter is far simpler to manipulate than PI controller gains or MPC controller parameters.

It is most interesting to note that SPRC is able to adjust to sudden and rapid changes in system dynamics. Starting from arbitrarily poor system Markov parameter estimates, the system can converge to a repetitive control law that optimally rejects loads under constant wind conditions. When wind conditions change abruptly, the SPRC controller

is able to recalibrate itself and resume load alleviation operation rapidly, in a completely stable manner.

The performance of fully adaptive SPRC, which incorporates continuous reïdentification and control law synthesis, is identical to the performance achieved by optimal LTI controllers designed for specific operating points. Thus, the set of experimental results in this chapter validate the proof of concept of a fully adaptive control law on a real-time setup of a scaled wind turbine.

The other set of experiments described using this setup demonstrate, for the first time on an upwind turbine, the potential for using IPC for the yaw control of such a turbine in free yaw. This principle behind this concept has been discussed in the literature, and demonstrated on downwind turbines, which are stable in free yaw. However, for an upwind turbine, the aeroelastic simulations from the previous chapter show that the yaw degree of freedom is unstable, but stabilisable, if the turbine is in free yaw. This principle has been validated experimentally, and it has been shown that a simple, manually tuned PID controller is able to follow a specific yaw trajectory with relatively fair performance, with a bandwidth of about 0.1 Hz.

Iterative feedback tuning was performed to understand the sensitivity of the IPC-yaw controller to changes in operating conditions. It is seen that the pitch actuators gain more control authority at higher wind speeds, and IPC-yaw needs to be gain scheduled in order to compensate for this effect. The optimal controller gains appear to decrease with an increase in the operating wind speed. Finally, it is also shown via IFT that the two blades of the turbine have asymmetric load control authority, and different controller gains should be used for the two blades to maximise yaw control performance.

This chapter specifically explores wind turbine pitch control, which, for modern wind turbines, is typically restricted to low-frequency control. To extend the load alleviation capabilities to the higher end of the frequency spectrum, the concept of trailing-edge flaps is explored experimentally in the next chapter.

9

WIND TUNNEL EXPERIMENTS: FLAP CONTROL

...te weeg te brengen, dat de Roeden, bij eene onvoorziene verheffing van Wind; haare fnelheid uit zichzelven maatigen zullen, wil hij, dat men de hekfcheeden aan den voorkant, dat is, die in plaats der borden gefteld zullen zijn, merkelijk dunner en buigzaamer maake dan de anderen, ten einde zij zouden kunnen doorbuigen, als de wind onvoorziens zich verheft; waar door zij, in dat geval, een kleiner hoek zullen verkrijgen, en dus zal 'er dan de wind met minder voordrijvende kracht op werken; ...

Reinier Arrenberg, the Netherlands (1779)
explaining bend-twist coupled rotor blades that produced passive load control.

As per the chapter on state of the art for flexible rotor design, trailing-edge flaps have been shown, numerically and experimentally, to possess considerable load reduction potential complementary to full-span pitch control. The current chapter specifically describes the aeroelastic analysis and LPV modelling of trailing-edge, free-floating flap for blades redesigned for integration with the experimental setup of the previous chapter. The manufacturing of these blades is discussed, and then the blades are assembled on to the scaled wind turbine. Recursive identification using the nuclear norm is carried out under varying wind conditions in the wind tunnel, using a nominally stabilising collocated controller in closed-loop, in order to detect the onset of flutter, and the stable operating range. Finally, IFT is used to optimally tune a feedforward controller that deploys 1P pitch action and 2P flap action to minimise rotor loads.

9.1. INTRODUCTION

Conventional trailing-edge flaps show good localised load control potential, as demonstrated using aeroelastic simulations in Chapter 7. Their practical implementation, how-

Parts of this chapter have been published in the Proceedings of the Science of Making Torque from Wind, Navalkar et al. (2016), and are under review in the IEEE Transactions on Control Systems Technology, Navalkar and Van Wingerden (2016) and in Wind Energy Science, Navalkar et al. (2016).

ever, poses a challenge. On the one hand, using conventional actuators, like electric motors, can be straightforward, given their high level of maturity. These actuators show large stroke, and suffer minimal degradation of performance under aerodynamic loading. However, they have low power-to-weight ratios, and their bandwidth is typically limited. On the other hand, 'smart' actuators like piezoelectrics have high power-to-weight ratios and high bandwidth, but show limited stroke under external loading. As a solution, the concept of the free-floating flap was discussed in Chapter 6, wherein the motion of a piezoelectric tab is amplified aerodynamically to achieve large trailing-edge deflections.

This concept of the free-floating flap, which is free to rotate about a hinge axis, can thus aerodynamically increase the camber control authority provided by a piezoelectric bender. However, this additional degree of freedom also couples with the flexible blade modes, to yield a low wind speed form of flutter, Bernhammer et al. (2013). This unstable phenomenon is required to be mitigated; however operating close to the flutter limit of the blades maximises flap control authority. As such, this form of trailing-edge flaps are both interesting and challenging in that closed-loop operation is required for stability. Further, since the system is highly non-linear (and possibly uncertain), designing a controller that guarantees both stability and performance can be a non-trivial task.

From numerical investigations in the literature, and from experimental evidence on non-rotating blades, Bernhammer et al. (2013), it has been concluded that the system dynamic response varies strongly with wind speed. Beyond the flutter speed, the poles of the system cross into the right half-plane. As such, it is postulated that the system can be described as a Linear, Parameter-Varying (LPV) system, where the wind speed forms the scheduling variable. However, no attempts have been made yet to formulate a first-principles model of this system. Such an LPV model, preferably low-order, could facilitate the development of robustly stabilising controllers for this system.

While non-rotating blades with free-floating trailing edge flaps have been investigated experimentally in the reference, a full implementation on a scaled wind turbine in a wind tunnel environment is still missing. With the help of numerical and analytical LPV modelling, this chapter explores the redesign and manufacture of the wind turbine blades of the previous chapter, instrumented with these flaps. It is the objective of the chapter to perform identification and control experiments on these augmented blades to investigate their load reduction potential under realistic loading conditions. This chapter thus reports, for the first time, on the implementation of free-floating flaps on a scaled wind turbine, and on the first experimental implementation of flap control combined with individual pitch control.

Although an attempt will be made to model the system, it is expected that both parametric uncertainty as well as unmodelled dynamics will cause discrepancies between predicted and observed behaviour of the turbine. In order to adjust control actions according to the changes in system dynamics, and to keep track of the operating regime of the blades, it would be desirable to perform recursive identification of the transfer between flap activity and the influence on blade loads. For this, the identification approach from Chapter 3, which augments closed-loop recursive identification with the nuclear norm, appears to be suitable on account of its reduced sensitivity to measurement noise. It should, however, be noted that the measurements are corrupted by ex-

ogenous forcing signals with fundamental frequency 1P (rotor speed), these frequencies are specifically accounted for in the identification procedure.

Finally, it would be desirable to use a data-driven controller for load control, that optimally tunes itself to maximise load reductions, combined with a baseline stabilising controller. Due to the highly non-linear nature of the control problem, the data-driven controller would be more suitable as a feedforward controller, so that it does not affect the stability of the closed loop. Iterative Feedback Tuning (IFT) could be used to update the gains of the feedforward controller, since, as discussed in Chapter 2, it possesses the ability to iterate to the (locally) optimal gains of a low-order fixed structure controller like the one in the current application. However, IFT has typically been implemented to tune LTI controllers, usually for the case where the plant dynamics is also approximated as LTI. For the pitch- and flap-control of flexible rotor loads, as the system is LPV with wind speed-dependent dynamics, such an LTI controller would be suboptimal over the entire wind speed range. Hence, it would be more interesting to use IFT to synthesise a gain schedule for the feedforward load controller, as described in Chapter 5. Such an IFT implementation has not yet been explored in practice.

As such, this chapter reports on a number of novelties:

- The first experimental demonstration of combined pitch and flap control is described.
- This chapter also reports on the first demonstration of the load alleviation potential of free-floating flaps on a physical wind turbine. It is also shown for the first time that free-floating flaps cause a low wind speed form of flutter in flexible wind turbine rotors.
- IFT is used for the first time to tune a gain schedule iteratively for a plant, the dynamics of which vary strongly with the scheduling variable wind speed.

Accordingly, this chapter first describes the design of wind turbine blades instrumented with free-floating flaps. Next, the blade dynamics are modelled using numerical tools as well as an analytical approach to synthesise an LPV state-space realisation of the system. Results of the identification and control experiments are then reported on. First, the concept of recursive closed-loop identification with the nuclear norm is compared with batchwise identification. Then, feedforward controller gain schedule tuning using IFT is investigated, and the chapter ends with conclusions drawn from these results.

The next section focusses on the design of blades equipped with free-floating flaps that can be retrofitted on the experimental setup of the previous chapter.

9.2. BLADE DESIGN AND MANUFACTURE

To recapitulate, the scaled experimental turbine from the previous chapter is a two-bladed machine, with rotor diameter less than 2m, designed to run at a tip speed ratio of 3.7, and wind tunnel testing is done in the Open Jet Facility of the Delft University of Technology, described in the previous chapter. In this section, the blades are redesigned using the same aerodynamic and structural design principles, following the blade design described by Van Wingerden et al. (2010b). The aerodynamic and structural details

of these blades can be found in Hulskamp et al. (2011). One key difference is that, as distinct from the experiments of Van Wingerden et al. (2010b), this set of experiments combines pitch with flap control. For reducing the torsional inertia imposed on the pitch actuation system, the blade design is hence modified such that the root chord is scaled down by 30%. Overall, the chord at the root has a length of 200 mm, and it tapers to a tip size of 120 mm. The total blade length is 750 mm, and the total blade twist is 12°.

In order to ensure light weight and structural integrity, the blade was stereolithographically 3D printed and then bonded with carbon fibre. Such an approach has been followed before by Bauer et al. (2014) to manufacture small wind turbine blades, and different 3D printing techniques have been compared for this application by Karutz (2015). In these references, the blades have been 3D printed in mutliple sections, that are then welded together. In order to avoid solid plastic-plastic joints, the blades for the current set of experiments were 3D printed in one piece, and then bonded with carbon fibre.

Three different plastics that can be 3D printed – ABS M30, PC-ABS and nylon – were evaluated in terms of their ability to bond with carbon fibre. Three rectangular coupons of size 200 mm × 30 mm, of thickness 3 mm, were 3D printed using each of the three plastic materials. These coupons were bonded on the top and bottom with carbon fibre, in a single unidirectional layer of thickness 0.14 mm, impregnated with epoxy resin. The three coupons were then tested to failure in a four-point bending machine to study the fracture mechanism, with intersupport distance 140 mm, and distance between the points of force application equal to 40 mm. The results can be seen in Fig. 9.1 and Fig. 9.2. For low loads, the response of the material is nearly linear, while at higher loads, small kinks appear in the force-deformation diagram. Each kink corresponds to the snapping of one or more carbon fibre strands in compression. Finally, a large decrease in strength is observed when the carbon fibre delaminates from the substrate, for the materials ABS M30 and nylon. For the material PC-ABS, brittle fracture occurs before delamination, which implies that the bond between this materials and the carbon fibre is strongest of the materials tested. Further, since PC-ABS retains its strength over a larger range of deformations, and shows good mechanical workability, the decision is made to use this material for blade manufacture.

The blade is 3D printed using the material PC-ABS as a shell of thickness 3 mm, such that its outer surface has the desired aerodynamic shape. The shell is supported by an inner rectangular box printed using the same material; this box is not designed to withstand aerodynamic loads, but it is present to prevent warping. The blade shell was designed with a rectangular slot of chordwise thickness 10 mm along the blade length; one each on the top and bottom sides. A 0.14 mm layer of carbon fibre was bonded to the plastic substrate within these slots, in order to achieve the desired structural stiffness. Crushed glass fibre in an epoxy matrix was then used to fair the slots with the rest of the aerodynamic surface.

The stiffening effect of the carbon fibre, predicted by the Solidworks model, is compared against the measured effect of carbon fibre stiffening, shown in Fig. 9.3. The predicted increase in stiffness is 17.2%, while the measured increase in blade stiffness is 16.6%.

With the new blades, the control-relevant characteristics of the scaled turbine are compared against the INNWIND 10 MW reference turbine as according to Table 9.1. A

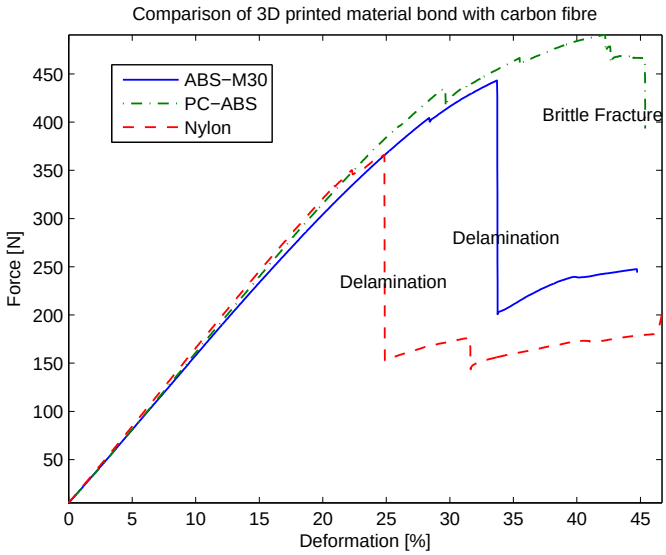


Figure 9.1: Structural behaviour of the bond between 3D printed plastic substrate and carbon fibre spar.

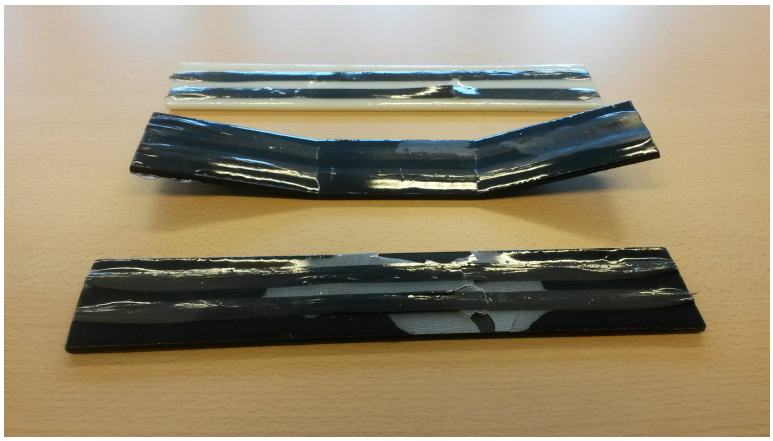


Figure 9.2: 3D printed samples post fracture. Top: ABS M30, Middle: PC-ABS, Bottom: Nylon.

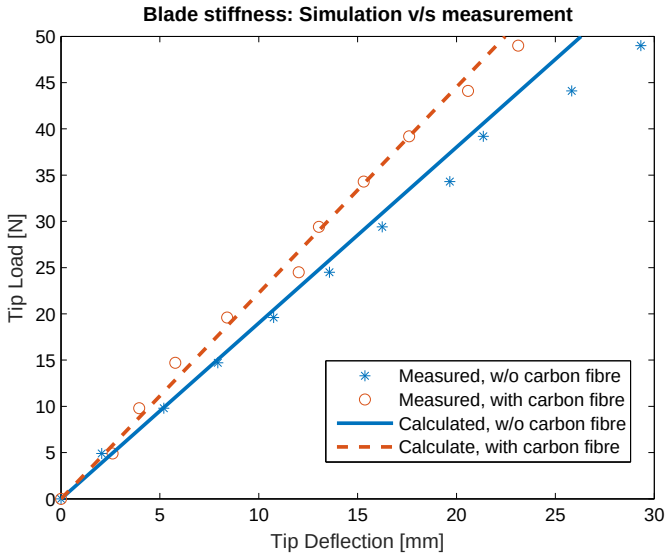


Figure 9.3: Calculated stiffness characteristics compared with measured stiffness characteristics.

Table 9.1: Parameter comparison of the scaled turbine

	Reference turbine Bak et al. (2013)	Scaled rotor Chapter 8	Scaled rotor Chapter 9
Rotor diameter (m)	178.3	2	2
Rated wind speed (m/s)	11.4	6.5	4.5
Tip speed ratio (-)	7.86	3	5.35
Rated rotational speed (rpm)	9.6	230	230
Fore-aft tower mode (Hz)	0.25	20.73	20.73
First flapwise mode (Hz)	0.56	15	14.4
Ratio 1st blade freq. to 1P (-)	3.5	3.913	3.75

photograph of the scaled turbine in the Open Jet Facility is given in Fig. 9.4, it can be observed that, apart from the redesigned blades, the setup is identical to the one used in the previous chapter.

The operational speed of the turbine is designed to be 230 rpm, yielding an average Reynold's number of roughly 200,000. Since this speed is lower than that of the reference turbine of Van Wingerden et al. (2010b), the blade flapwise eigenfrequency was also scaled down. The structural stiffness of the blade was tuned by adjusting the thickness of the carbon fibre spars such that the first eigenmode of the blade has a frequency of approximately 20 Hz. The Solidworks design of the blade can be seen in Fig. 9.5, while a photograph of the manufactured blade is shown in Fig. 9.6.

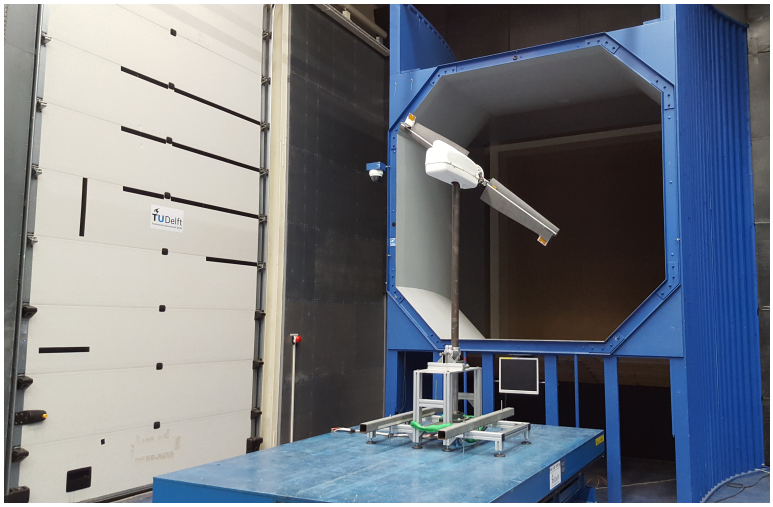


Figure 9.4: Photograph of the wind turbine located close to the open jet of the wind tunnel. The rotor diameter is 2 m.

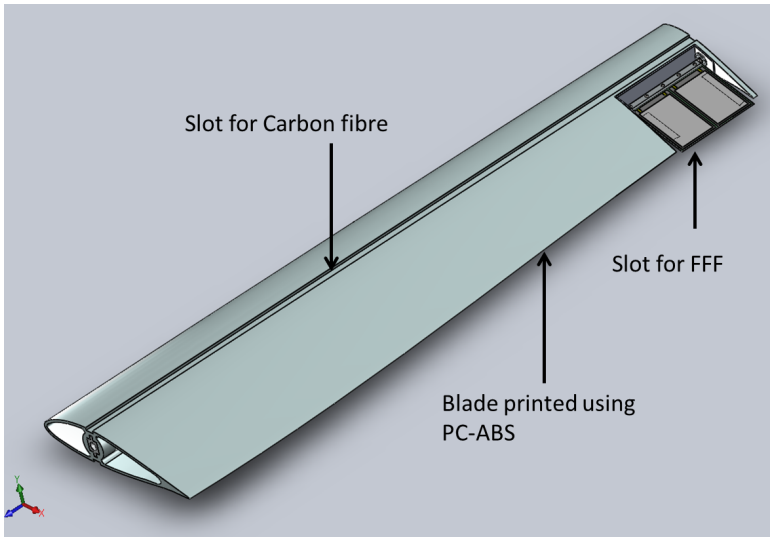


Figure 9.5: Blade CAD model.

Figure 9.6: Photograph of blade.

The free-floating flap fits into a slot located near the tip of the blade, and a cross-section of the flap can be seen in Fig. 9.7. The flap can rotate freely about a hinge axis, the rotation of the flap is measured by a collocated contactless angular position encoder. The rotation of the flap is limited by making use of limit stops. The flap itself consists of a 0.15 mm thick plate of spring steel, with piezo Macro-Fibre Composite actuators (MFC M8557-P1) bonded to the top and the bottom surfaces. Imposing voltage within the range of +500 V to -500 V on the piezobenders produces a change in the curvature of the flaps. In order to avoid aerodynamic discontinuities, the flap is encapsulated by a fairing shroud, held in place by using low-rigidity foam. This configuration causes a sudden change of curvature just downstream of the spar, but for the current application, the loss of aerodynamic efficiency is considered to be made up for by the increased control authority of the free-floating flap.

Since it is known from Bernhammer et al. (2013) that the addition of such free-floating flaps alters the aeroelastic behaviour of the wind turbine blades, the blades are modelled numerically and analytically and their behaviour for a range of operating wind speeds is explored in the next section.

9.3. AEROELASTIC BLADE ANALYSIS

First, the blade is analysed using the finite-element software MSC Nastran, Rodden and Johnson (1994), coupled with an aerodynamic panel code. Then, in order to obtain an analytical LPV model suitable for controller design, a potential-flow method will be combined with a modal representation of the blade structure.

9.3.1. NUMERICAL ANALYSIS

In the software MSC Nastran, the experimental blade is modelled as a cantilever beam, composed of 1D CBar finite elements, each connected rigidly with a flat aerodynamic panel. The flap is modelled in a similar manner, and it is connected to the main blade

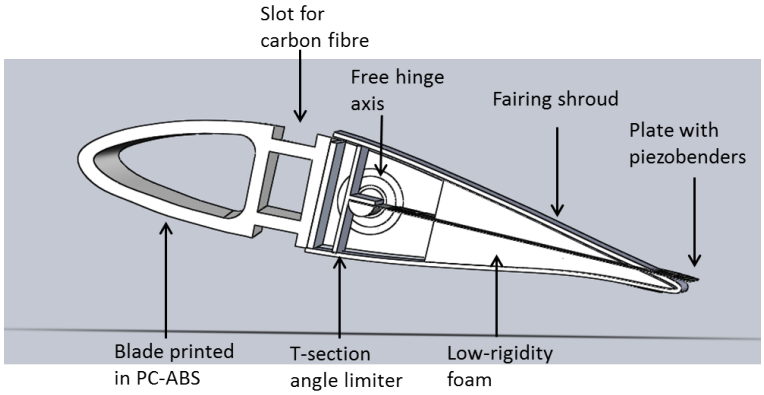


Figure 9.7: Flap cross-section, chordwise size 120 mm, shell thickness 3 mm. The flap forms 40% of the chord.

at a far outboard location, in such a manner that it able to rotate freely about its axis. The wind speed is considered to be constant along the blade, however, this assumption is not valid in practice; the total relative incident wind speed increases linearly from the root towards the tip. However, the error introduced by this assumption is somewhat mitigated by the fact that the aeroelastic, contribution of the inboard parts of the blade is lower due to limited structural motion. As such, the constant wind speed imposed on the blade in the numerical environment is taken to be approximately equal to the total relative air speed incident at the blade tip location.

There are two limitations of this approach of aeroelastic blade modelling. Firstly, the experimental blade needs to be analysed in a rotating frame of reference, with a spanwise variation of wind speed and centrifugal stiffening. This implementation is, however, not straightforward in MSC Nastran and hence analysis is done in a stationary frame of reference. Wake effects and the effects of inertia forces are not accounted for in this analysis. As such, this analysis forms a first-order approximation of the true system dynamics of the experimental blade, and it stands to be improved by including corrections to account for the effects mentioned.

First, a modal analysis is carried out in Nastran at zero wind speed to identify the purely structural blade modes and their associated frequencies; these are tabulated in Table 9.2. As expected, the rigid-body flap deflection mode occurs at 0 Hz. Some of the modes have been depicted in Fig. 9.8 to Fig. 9.11. Similar modal frequencies are obtained from the 3D Solidworks model:

- 1^{st} flapwise frequency: 18.97 Hz (Solidworks), 19.44 Hz (Nastran)
- 1^{st} edgewise frequency: 78.37 Hz (Solidworks), 76.67 Hz (Nastran)
- 2^{nd} flapwise frequency: 84.8 Hz (Solidworks), 87.88 Hz (Nastran).

These frequencies are higher than the modal frequencies measured experimentally.

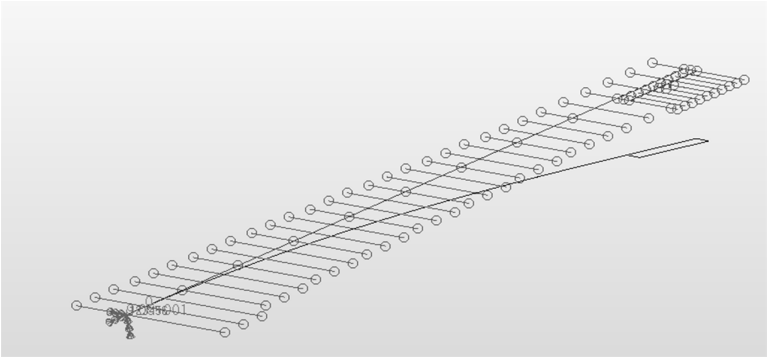


Figure 9.8: Second mode: First blade out-of-plane, frequency 19.44 Hz.

This is because the experimental frequencies correspond to the rotor, and hence also include the effect of blade connection stiffness, motor stiffness and hub flexibility and are per definition lower than the blade frequencies obtained numerically. This difference between modelled and measured modal frequencies can be considered to be a form of parametric uncertainty, and provides motivation for a data-driven control approach.

It can be seen that the first blade out-of-plane mode is the flexible mode with the lowest frequency, at 19.44 Hz. This mode is likely to couple aeroelastically with the rigid-body flap motion, and undergoes excitation through the axial thrust force exerted by the wind on the rotor. The lead-lag and torsional degrees of freedom have much higher frequencies; this is also observed in commercial blades, which are stiffer in these directions. As there is limited excitation along these degrees of freedom, they may be considered to contribute little to the dominant aeroelastic behaviour of the experimental blade. Modelling and analysis of the blade in the aeroelastic sense will hence concentrate purely on the out-of-plane flexible degree of freedom and its interplay with the rigid-body flap motion.

Table 9.2: Structural modes of the blade at zero total air speed

Mode description	Modal frequency	Mode description	Modal frequency
Rigid-body flap mode	0 Hz	1st Out-of-plane mode	19.44 Hz
1st In-plane mode	76.67 Hz	2nd Out-of-plane mode	87.88 Hz
3rd Out-of-plane mode	223.9 Hz	2nd In-plane mode	291.3 Hz
1st Torsional mode	361.6 Hz	4th Out-of-plane mode	449.6 Hz

Next, a flutter analysis is carried out in MSC Nastran to understand the change in the dynamics of the system over the operational wind speed range of the experimental wind turbine. The results of this flutter analysis are presented in Section 9.3.3, and compared with the results of analytical modelling described in the next section.

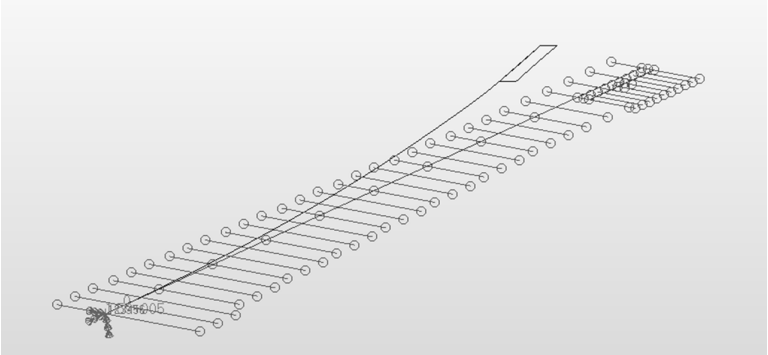


Figure 9.9: Third mode: First blade in-plane, frequency 76.67 Hz.

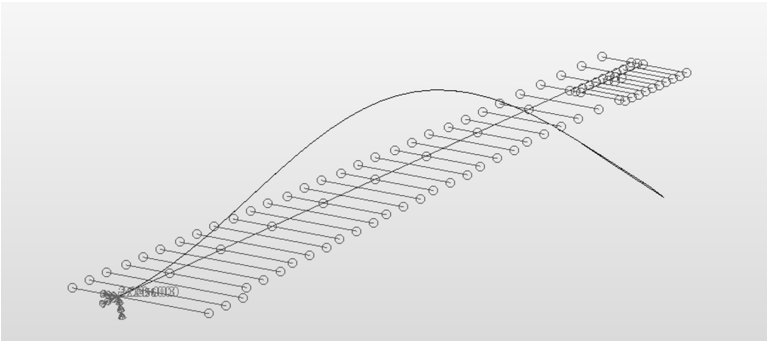


Figure 9.10: Fourth mode: Second blade out-of-plane, frequency 87.88 Hz.

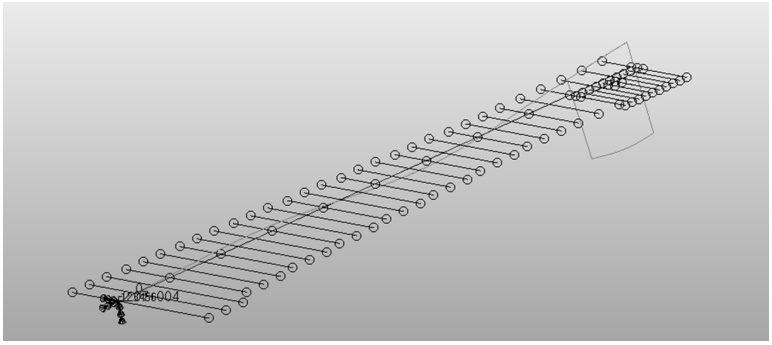


Figure 9.11: Seventh mode: First blade torsional, frequency 361.6 Hz.

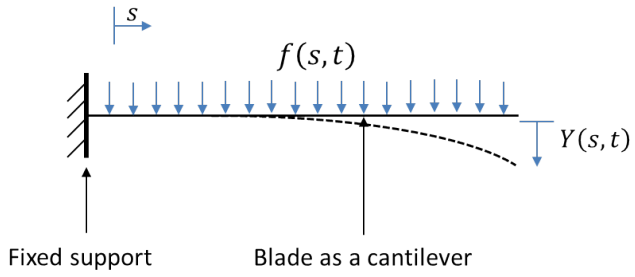


Figure 9.12: Blade front view: pure out-of-plane motion.

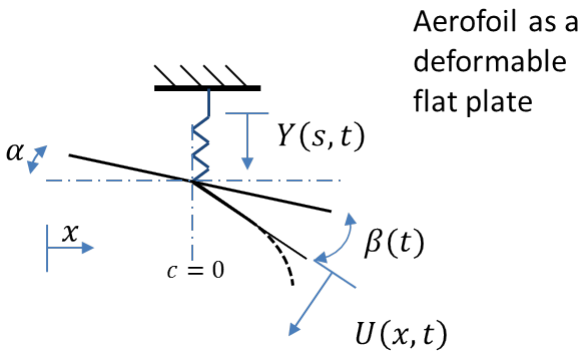


Figure 9.13: Blade cross-section at flap location.

9.3.2. ANALYTICAL LPV MODELLING

The objective of this section is to synthesise a low-order LPV model of the system that captures the dominant system dynamics, such that it can be used directly for controller synthesis. Such a model can be formulated by coupling a low-order linear structural model with a potential flow-based unsteady aerodynamic model. It should be noted that the mathematical symbols used in this section have been redefined for the sake of legibility, and do not retain the connotation specified in Part I of the thesis.

STRUCTURAL MODEL

For the scaled wind turbine, from the previous chapter, it can be concluded that the load peaks of interest lie in the range of 0-16 Hz, or 0-4P, where 1P is the operational rotor speed of 230 rpm. From the numerical analysis, it can be concluded that the most important structural degrees of freedom are the rigid-body flap motion, and the first flapwise flexible blade mode. Further, the bending produced in the camberline of the free-floating flap forms a control degree of freedom. The structural model can be visualised in Fig. 9.12 and Fig. 9.13.

As according to Fig. 9.12, the blade has been modelled as a cantilever beam con-

strained to deform in its first flapwise mode. Thus, if $Y(s,t)$ is the flapwise deformation at spanwise location s at time t, then it can be decomposed as:

$$Y(s,t) = \psi(s)\xi(t), \tag{9.1}$$

such that $\psi(s)$ describes the first flexural mode shape of a standard cantilever beam, as described by Hodges and Pierce (2011):

$$\psi(s) = \cosh(1.8751 s/\ell) - \cos(1.8751 s/\ell) - 0.7341(\sinh(1.8751 s/\ell) - \sin(1.8751 s/\ell)), \tag{9.2}$$

with ℓ the length of the blade. This definition assumes homogeneous blade structural properties along the span; it is not strictly true since the blade tapers from root to tip. As shown in Fig. 9.12, the aerodynamic forces $f(s,t)$ act along the length of the blade.

The flap is located between spanwise coördinates ℓ_f and ℓ. A chordwise cross-section of the blade between these radial stations is shown in Fig. 9.13. It can be seen that the angle of attack of the blade is α, which can be taken to be the same as the full-span pitch angle of the blade. While this can be considered a control degree of freedom, since the turbine can use individual pitch control, for the purpose of this analysis, it has been considered constant. Further, the turbine blade is twisted so as to form a constant angle of attack with the incoming wind. Since the incoming wind has been assumed constant in a spanwise direction, the angle of attack is also taken constant over the blade span.

The out-of-plane motion of the blade, $Y(s,t)$ represents the heave motion of the aerofoil in Fig. 9.13. Further, the flap is able to rotate freely as a rigid body about its point of connection to the blade, which is considered to be at a distance $c = 0$ from the centre of the aerofoil. This rigid-body angular motion is denoted by the flap angle $\beta(t)$. Finally, the flap camberline deformation, imposed by the piezobenders is considered to be represented by $U(x,t)$, where x is the chordwise coördinate. It is assumed that the piezobenders produce camberline deformation that has energy only along the first cantilever bending mode shape. As such, the control input $U(x,t)$ can also be decomposed as:

$$U(x,t) = \psi(x)u(t). \tag{9.3}$$

It should be noted that the connection of the free-floating flap to the main blade structure can be modelled with higher fidelity as a pin connection instead of a clamped connection. However, the difference in the first flexural mode shapes is not considered to add error to any significant extent to the modelling results. In the above equations, the chordwise coördinate x is dimensionless, and normalised by the half-chord length $b(s)$. Overall, the structural deformation $y(x,s,t)$ of any arbitrary point on the blade defined by spanwise location s and chordwise location x at time t can thus be represented as:

$$y(x,s,t) = \sum_{i=0}^{3} y_i(x,s)\phi_i(t), \tag{9.4}$$

with the terms y_i and ϕ_i defined in Table 9.3.

At this point, it is possible to set up the equations of motion for each structural degree of freedom. For the out-of-plane motion of the beam, the equation of motion becomes:

$$M\ddot{\xi}(t) + B\dot{\xi}(t) + K\xi(t) = \Xi(t). \tag{9.5}$$

Table 9.3: Camberline deformation terms

Description	Spatial term	Temporal term
Angle of attack	$y_0(x,s) = -\alpha b(s)x$	$\phi_0(t) = 1$
Out-of-plane motion	$y_1(x,s) = \psi(s)$	$\phi_1(t) = \xi(t)$
Flap rigid-body motion	$y_2(x,s) = -b(s)x$ for $s \geq \ell_f, x \geq c$	$\phi_2(t) = \beta(t)$
Control camberline deflection	$y_3(x,s) = \psi(x)b(s)$ for $s \geq \ell_f, x \geq c$	$\phi_3(t) = u(t)$

The structural parameters M, B and K are the generalised mass, damping and stiffness terms, respectively, associated with the first flexural mode of the blade. The term $\Xi(t)$ is the generalised aerodynamic force associated with this mode. Using the principle of virtual work, defined in Hodges and Pierce (2011), these quantities can be calculated as follows:

$$M = \int_0^\ell m(s)\psi(s)^2 \mathrm{d}s, \qquad K = \int_0^\ell EI(s)\psi''(s)^2 \mathrm{d}s, \qquad B = b_\xi K, \tag{9.6}$$

$$\Xi(t) = \int_0^\ell f(s,t)\psi(s)\mathrm{d}s. \tag{9.7}$$

The term $m(s)$ is the mass per unit length of the blade at spanwise location s, obtained from Nastran. Further, the term $EI(s)$ is the flexural stiffness of the blade, measured at location s, also obtained from the numerical model of the previous section. Finally, b_ξ is the blade structural damping. Next, the equation of motion for the free-floating flap degree of freedom is given by:

$$I\ddot{\beta}(t) + K_\beta \beta(t) = \Xi_m(t). \tag{9.8}$$

Here, the term I refers to the inertia of the flap around its hinge. The term K_β is the stiffness to flap rotation, identically zero for a free-floating flap, and ∞ for a conventional flap. Finally, the term $\Xi_m(t)$ refers to the aerodynamic moment on the flap around its hinge axis, and it can be determined in the same manner as $\Xi(t)$.

The next section discusses the aerodynamic model used for synthesising the generalised aerodynamic forces and moments.

AERODYNAMIC MODEL

Aerodynamic modelling is performed in accordance with the principles laid out by Gaunaa (2010). This approach is valid for thin two-dimensional aerofoils of arbitrary geometry, at low angles of attack. It is a potential-flow method, and considers that the circulation change is produced by a series of bound vortices distributed along the chord of the aerofoil. As such, this approach does not consider the 3D influence of the bound vortices on neighbouring blade elements, this simplification introduces an error that requires correction, as will be done in equation (9.20) below. It also does not include wake effects or stall behaviour. However, it is here considered appropriate for arriving at a simple LPV model suitable for controller design, that still capture the dominant dynamics of the blade with free-floating flaps.

For the blade with cross-section as shown in Fig. 9.13, immersed in an airflow with instantaneous wind speed V, the aerodynamic force per unit length at spanwise location s and at time t is given by:

$$f(s,t) = \frac{\rho b(s)^2}{\pi} \sum_{i=0}^{3} (\ddot{\phi}(t) G_{y,i}(-1) + V\dot{\phi}(t) G_{\mathrm{d}y\mathrm{d}\epsilon,i}(-1)) + 2\pi\rho b(s) V Q_c(t), \tag{9.9}$$

with ρ the density of air. The geometric terms $G_{*,i}$ are defined in the following manner:

$$G_{*,i}(c) = \int_c^1 g_{*,i}(x)\mathrm{d}x, \tag{9.10}$$

where:

$$g_{y,i}(x) = \int_{-1}^{1} y_i(x_1) \ln\left(\frac{(x-x_1)^2 + (\sqrt{1-x^2} - \sqrt{1-x_1^2})^2}{(x-x_1)^2 + (\sqrt{1-x^2} + \sqrt{1-x_1^2})^2} \right) \mathrm{d}x_1, \tag{9.11}$$

$$g_{\mathrm{d}y\mathrm{d}\epsilon,i}(x) = \int_{-1}^{1} y_i'(x_1) \ln\left(\frac{(x-x_1)^2 + (\sqrt{1-x^2} - \sqrt{1-x_1^2})^2}{(x-x_1)^2 + (\sqrt{1-x^2} + \sqrt{1-x_1^2})^2} \right) \mathrm{d}x_1. \tag{9.12}$$

The term $y_i'(x)$ represents the slope of the camberline, given by $y_i'(x) = \frac{1}{b(s)} \frac{\mathrm{d}y_i(x)}{\mathrm{d}x}$. In equation (9.9), the term $Q_c(t)$ is the circulatory force, defined in terms of the aerodynamic states $z_1(t)$ and $z_2(t)$ as:

$$Q_c(t) = Q(t)(1 - \tilde{A}_1 - \tilde{A}_2) + z_1(t) + z_2(t) \tag{9.13}$$

$$Q(t) = -\frac{V}{2\pi} \sum_{i=0}^{3} \phi_i(t) H_{\mathrm{d}y\mathrm{d}\epsilon,i} - \frac{1}{2\pi} \dot{\phi}_i(t) H_{y,i}. \tag{9.14}$$

Once again, the geometric terms $H_{*,i}$ are defined as:

$$H_{y,i} = -2\int_{-1}^{1} \frac{y_i(x)\sqrt{1-x^2}}{x-1}\mathrm{d}x, \qquad H_{\mathrm{d}y\mathrm{d}\epsilon,i} = -2\int_{-1}^{1} \frac{y_i'(x)\sqrt{1-x^2}}{x-1}\mathrm{d}x. \tag{9.15}$$

The evolution of the aerodynamic states is described by empirically fitting the indicial response of the aerofoil in the following manner:

$$\dot{z}_*(t) + \frac{\tilde{b}_* V z_*}{b(s)} = \frac{\tilde{b}_* \tilde{A}_* V Q(t)}{b(s)}. \tag{9.16}$$

For a flat plate, the empirical coefficients $\tilde{b}_*$ and $\tilde{A}_*$ can be obtained from Jones (1939).

In a similar manner, the aerodynamic moment over the flap, $\Xi_m(t)$ is given by:

$$\Xi_m(t) = \int_{\ell_f}^{\ell} M_p(s)\mathrm{d}s, \tag{9.17}$$

$$
\begin{aligned}
M_p(s) &= \frac{\rho b(s)^2 V^2}{\pi}\sum_{i=0}^{3}\phi_i(t)G_{\mathrm{dy}d\epsilon,i}(c) + \rho b(s)^2 V^2\frac{K_1(c)}{2\pi}\sum_{i=0}^{3}\phi_i(t)H_{\mathrm{dy}d\epsilon,i} \\
&- \frac{\rho b(s)^3 V}{\pi}\sum_{i=0}^{3}\dot{\phi}_i(t)(I_{\mathrm{dy}d\epsilon,i}(c) - cG_{\mathrm{dy}d\epsilon,i}(c)) + \frac{\rho b(s)^2 V}{\pi}\sum_{i=0}^{3}\dot{\phi}_i(t)G_{y,i}(c) \\
&+ \rho b(s)^2 V\frac{K_1(c)}{2\pi}\sum_{i=0}^{3}\dot{\phi}_i H_{y,i} - \frac{\rho b(s)^3}{\pi}\sum_{i=0}^{3}\ddot{\phi}_i(t)(I_{y,i}(c) - cG_{y,i}(c)) \\
&+ 2\rho b(s)^2 V((c+\frac{1}{2})\cos^{-1}c - (1+\frac{c}{2})\sqrt{1-c^2})Q_c(t).
\end{aligned}
$$

with the term $K_1(c) = \cos^{-1}c - c\sqrt{1-c^2}$. The terms $I_{*,i}$ are geometric and given as:

$$
I_{y,i}(c) = \int_c^1 x g_{y,i}(x)\mathrm{d}x, \qquad I_{\mathrm{dy}d\epsilon,i}(c) = \int_c^1 x g_{\mathrm{dy}d\epsilon,i}(x)\mathrm{d}x. \tag{9.18}
$$

Thus, the aerodynamic forces can be described using a second-order differential equation.

AEROELASTIC MODEL

Coupling the structural models from equations (9.5) and (9.8) and aerodynamic models in the equations (9.9) and (9.17) described in the previous sections, it is possible to describe the full aeroelastic model as follows:

$$
\begin{aligned}
&\mathcal{M}_{11}\ddot{\xi}(t) + \mathcal{B}_{11}V\dot{\xi}(t) + K\dot{\xi}(t) + \mathcal{M}_{12}\ddot{\beta}(t) + \mathcal{B}_{12}V\dot{\beta}(t) + \mathcal{K}_{12}V^2\beta(t) + Q_1 V z_1(t) + Q_1 V z_2(t) \\
&\qquad = \mathcal{M}_{13}\ddot{u}(t) + \mathcal{B}_{13}V\dot{u}(t) + \mathcal{K}_{13}V^2 u(t) + N, \\
&\mathcal{M}_{21}\ddot{\xi}(t) + \mathcal{B}_{21}V\dot{\xi}(t) + \mathcal{I}\ddot{\beta}(t) + \mathcal{B}_{22}V\dot{\beta}(t) + \mathcal{K}_{22}V^2\beta(t) + Q_2 V z_1(t) + Q_2 V z_2(t) \\
&\qquad = \mathcal{M}_{23}\ddot{u}(t) + \mathcal{B}_{23}V\dot{u}(t) + \mathcal{K}_{23}V^2 u(t), \\
&\mathcal{G}\dot{z}_1(t) + \mathcal{B}_{41}V\dot{\xi}(t) + \mathcal{B}_{42}V\dot{\beta}(t) + \mathcal{K}_{42}V\beta(t) + Q_3 V z_1(t) = \mathcal{B}_{43}V\dot{u}(t) + \mathcal{K}_{43}Vu(t), \\
&\mathcal{G}\dot{z}_2(t) + \mathcal{B}_{51}V\dot{\xi}(t) + \mathcal{B}_{52}V\dot{\beta}(t) + \mathcal{K}_{52}V\beta(t) + Q_4 V z_1(t) = \mathcal{B}_{53}V\dot{u}(t) + \mathcal{K}_{53}Vu(t).
\end{aligned} \tag{9.19}
$$

Here, N is the steady-state normal force that produces structural deformation but does not contribute to its dynamic behaviour. In these equations, the deflection of the piezobenders, $u(t)$ is taken to be the control input. Since piezodynamics are typically orders of magnitude larger than aerostructural dynamics, the behaviour of the piezobender is not modelled. Instead, it is assumed that, at any instant of time, the desired camberline shape can be imposed on the free-floating flaps by the piezobenders.

All coefficients in the aeroelastic equations above are constant and depend only on the geometry of the system. From the differential equations, the dependence of the dynamics on the ambient wind speed is evident. For the special case where the chordwise size of the flap is exactly 50% of chord length, we have that $c = 0$, and the coefficients of the above equations are tabulated in Table 9.4. It should be noted that the term $\bar{A}$ is defined such that $\bar{A} = 1 - \tilde{A}_1 - \tilde{A}_2$.

In the next section, the aeroelastic behaviour of this analytical model is compared with that of the numerical model from the previous section.

Table 9.4: Aeroelastic model coefficients

$\mathcal{M}_{11} = M + \pi\rho\int_0^\ell b(s)^2\psi(s)^2\mathrm{d}s$	$\mathcal{M}_{12} = -\frac{2}{3}\rho\int_{\ell_f}^\ell b(s)^3\psi(s)\mathrm{d}s$
$\mathcal{M}_{21} = -\frac{2}{3}\rho\int_{\ell_f}^\ell b(s)^3\psi(s)\mathrm{d}s$	$\mathcal{I} = I + 0.2573\rho\int_{\ell_f}^\ell b(s)^4\mathrm{d}s$
$\mathcal{M}_{13} = -0.9629\rho\int_{\ell_f}^\ell b(s)^3\psi(s)\mathrm{d}s$	$\mathcal{M}_{23} = 0.3988\rho\int_{\ell_f}^\ell b(s)^4\mathrm{d}s$
$\mathcal{B}_{11} = 2\pi\rho\bar{A}\int_0^\ell b(s)\psi(s)^2\mathrm{d}s$	$\mathcal{B}_{12} = (1.571 + 3.571\bar{A})\rho\int_{\ell_f}^\ell b(s)^2\psi(s)\mathrm{d}s$
$\mathcal{B}_{21} = -0.4292\rho\bar{A}\int_{\ell_f}^\ell b(s)^2\psi(s)\mathrm{d}s$	$\mathcal{B}_{22} = (0.8929 + 0.2439\bar{A})\rho\int_{\ell_f}^\ell b(s)^3\mathrm{d}s$
$\mathcal{B}_{13} = -(2.878 + 6.346\bar{A})\rho\bar{A}\int_{\ell_f}^\ell b(s)^2\psi(s)\mathrm{d}s$	$\mathcal{B}_{23} = (1.896 + 0.4335\bar{A})\rho\int_{\ell_f}^\ell b(s)^3\mathrm{d}s$
$\mathcal{B}_{41} = \tilde{b}_1\tilde{A}_1\int_0^\ell \psi(s)\mathrm{d}s$	$\mathcal{B}_{42} = -0.5683\tilde{b}_1\tilde{A}_1\int_{\ell_f}^\ell b(s)\psi(s)\mathrm{d}s$
$\mathcal{B}_{51} = \tilde{b}_2\tilde{A}_2\int_0^\ell \psi(s)\mathrm{d}s$	$\mathcal{B}_{52} = -0.5683\tilde{b}_2\tilde{A}_2\int_{\ell_f}^\ell b(s)\psi(s)\mathrm{d}s$
$\mathcal{B}_{43} = -1.945\tilde{b}_1\tilde{A}_1\int_{\ell_f}^\ell b(s)\psi(s)\mathrm{d}s$	$\mathcal{B}_{53} = -1.945\tilde{b}_2\tilde{A}_2\int_{\ell_f}^\ell b(s)\psi(s)\mathrm{d}s$
$\mathcal{K}_{12} = -5.142\rho\bar{A}\int_{\ell_f}^\ell b(s)\psi(s)\mathrm{d}s$	$\mathcal{K}_{13} = -12.22\rho\bar{A}\int_{\ell_f}^\ell b(s)\psi(s)\mathrm{d}s$
$\mathcal{K}_{22} = (0.182 + 0.3513\bar{A})\rho\int_{\ell_f}^\ell b(s)^2\mathrm{d}s$	$\mathcal{K}_{23} = -(0.931 + 0.8348\bar{A})\rho\int_{\ell_f}^\ell b(s)^2\mathrm{d}s$
$\mathcal{K}_{42} = -0.8183\tilde{b}_1\tilde{A}_1(\ell_f - \ell)$	$\mathcal{K}_{43} = -1.01\tilde{b}_1\tilde{A}_1(\ell_f - \ell)$
$\mathcal{K}_{52} = -0.8183\tilde{b}_2\tilde{A}_2(\ell_f - \ell)$	$\mathcal{K}_{53} = -1.01\tilde{b}_2\tilde{A}_2(\ell_f - \ell)$
$Q_1 = -2\pi\int_0^\ell b(s)\psi(s)\mathrm{d}s$	$Q_2 = -0.4292\rho\int_{\ell_f}^\ell b(s)^2\mathrm{d}s$
$Q_3 = \tilde{b}_1\ell$	$Q_4 = \tilde{b}_2\ell$
$\mathcal{G} = \int_0^\ell b(s)\mathrm{d}s$	

9.3.3. MODEL COMPARISON

An aeroelastic flutter analysis was performed using the analytical and numerical models described in the previous sections, and the resultant flutter diagram can be seen in Fig. 9.14 and Fig. 9.15. It should be noted that in all flutter diagrams, the abscissæ correspond to total air speed, defined as the resultant of the inflow wind speed and the rotor speed at the blade tip. Further, as discussed in the previous chapter, the speed regulation trajectory of the wind turbine is linear, so that rotor speed increases with wind speed at the rate of 51.1 rpm/(m/s). It can be seen that the variation of the frequency of the poles with total air speed is predicted in a similar manner by both models. The frequency of the rigid-body flap mode rises from 0 Hz at 0 total air speed, to approximately 12 Hz at a total air speed of 50 m/s, in a nearly linear fashion. On the other hand, the frequency of the flexible out-of-plane blade mode remains nearly constant at around 19 Hz over the air speed range investigated.

However, there is a noticeable difference between the damping variation predicted by the numerical and analytical models. Specifically, the Nastran model predicts a larger variation in damping over wind speed, with the system becoming unstable at a flutter speed of 33 m/s. However, according to the analytical model, flutter does not occur within the investigated wind speed range. Since subsequent experimental investigations, described in the next sections, found that the phenomenon of flutter occurs at a total operating air speed of close to 30 m/s, it can be concluded that the numerical

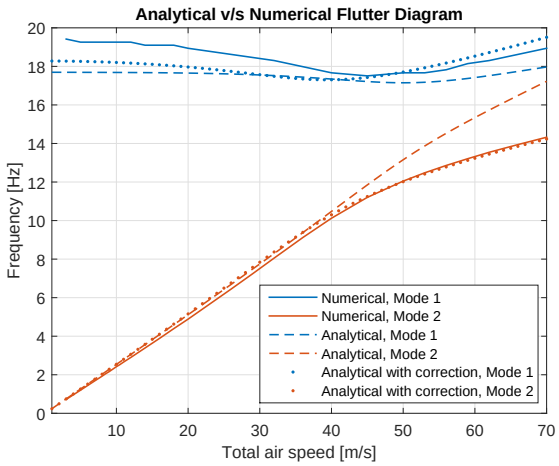


Figure 9.14: Flutter diagram of blade: Frequency.

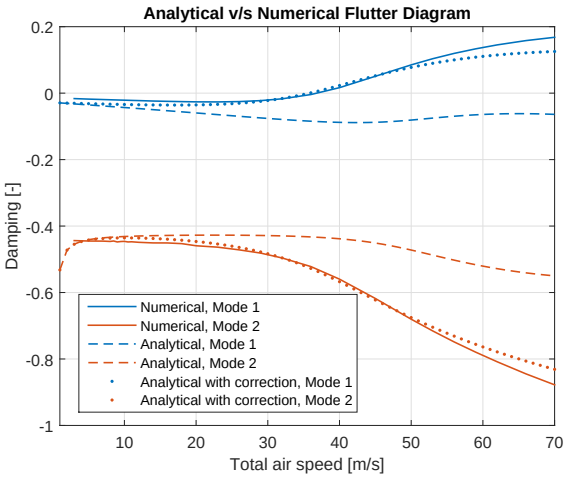


Figure 9.15: Flutter diagram of blade: Damping.

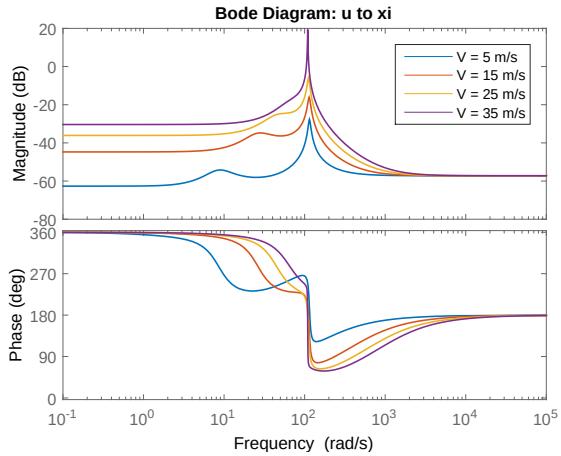


Figure 9.16: Transfer function from control input to blade out-of-plane motion.

Nastran model is more accurate than the potential flow model.

When the Nastran and the potential-flow models are compared, it is found that the latter analytical model underpredicts the mass cross-coupling terms in the aeroelastic equations. Since the objective of this section is to derive a low-order LPV model suitable for controller design, these cross-mass terms $\mathcal{M}_{12}$ and $\mathcal{M}_{21}$ in the analytical model are corrected by an empirical factor $K_{cc} = 8$, tuned such that the flutter speed predicted by the analytical model matches that of the numerical model:

$$\mathcal{M}_{12} = \mathcal{M}_{21} = -\frac{2K_{cc}}{3}\rho \int_{\ell_f}^{\ell} b(s)^3 \psi(s) \mathrm{d}s. \tag{9.20}$$

This correction term is necessary because the analytical model assumes the independence of blade elements, and hence neglects the aerodynamic effect of the bound vortices of each blade element on its neighbours. This reduces the predicted aerodynamic cross-influence of the structural degrees of freedom on each other. While it has here been chosen to use an aerodynamic correction to tune the mass terms in the aeroelastic equations, a similar tuning factor could also be used to adjust the cross-stiffness or cross-damping terms, to achieve the same model behaviour. To avoid the need for a tuning factor, greater modelling fidelity would be required to relax the assumption of blade element independence, but it is not known whether the model would then admit a simple LPV state-space realisation.

If this correction is used, the predicted analytical flutter diagram is very close to the one obtained numerically using Nastran, and it has been shown in Fig. 9.15 using a dotted line.

The set of differential equations (9.19) can be converted to a set of transfer functions that describe the dynamics of the system from the control degree of freedom $u(t)$ to the structural degrees of freedom $\xi(t)$ and $\beta(t)$. The Bode diagrams of these transfer

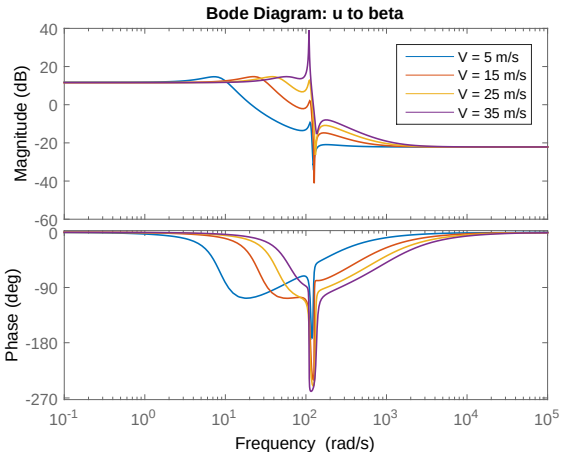


Figure 9.17: Transfer function from control input to flap rigid-body motion.

functions, at different wind speeds, are depicted in Fig. 9.16 and Fig. 9.17. It can be seen that the peak corresponding to the heavily damped flap rigid body mode moves towards higher frequencies with increasing wind speeds, while the lightly-damped peak corresponding to the flexible blade out-of-plane mode remains at the same frequency. It can also be seen that the damping corresponding to the flexible mode reduces with an increase in wind speed, and at a wind speed of 35 m/s, the system is unstable.

In equation (9.8), it has been assumed that for the current case of a free-floating flap, the flap is connected to the main structure of the blade by a virtual rotary spring of stiffness $K_\beta = 0$, which affords resistance to the rigid body flap motion, denoted by $\beta(t)$. For an equivalent conventional trailing-edge flap, the rotary stiffness would be $K_\beta = \infty$. However, it is also possible to use a physical spring to provide resistance to the rigid-body motion of the trailing-edge flap. Such a case can be modelled using the aeroelastic equations (9.19), for a finite, non-zero value of K_β. The presence of such a spring would alter the dynamic behaviour of the blade; transfer functions are derived for different values of K_β, and the Bode diagrams of these transfer functions are plotted in Fig. 9.18. It can be seen that there is an intermediate, finite value of K_β for which the sprung trailing-edge flap shows higher control authority than either the free-floating flap or the conventional trailing-edge flap. A similar effect can be seen in Fig. 9.19, which plots the variation of the flutter speed with spring stiffness. However, as control authority increases, the flutter speed also drops, reaching a minimum of 23 m/s for a spring stiffness of 11.9 N-m/rad. It is interesting to note that as the spring stiffness increases beyond this value, the flutter speed increases exponentially to a point where it no longer occurs within the operational range of the turbine.

Herewith, it can be concluded that the simple analytical model cannot directly capture all dynamic effects predicted by the numerical model. However, with a simple empirical correction, this LPV analytical model can describe the same blade aeroelastic be-

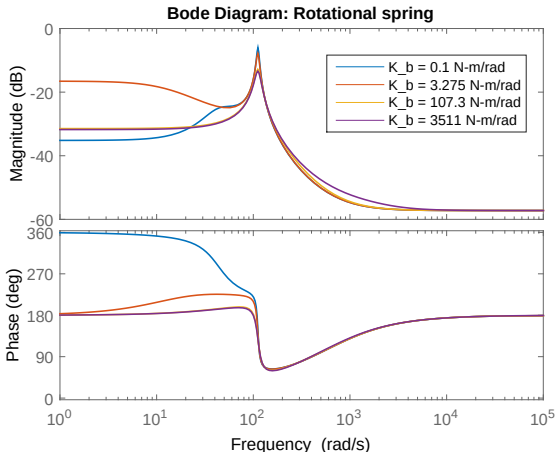


Figure 9.18: Transfer function from control input to blade out-of-plane motion, flap with rotational spring, wind speed 25 m/s.

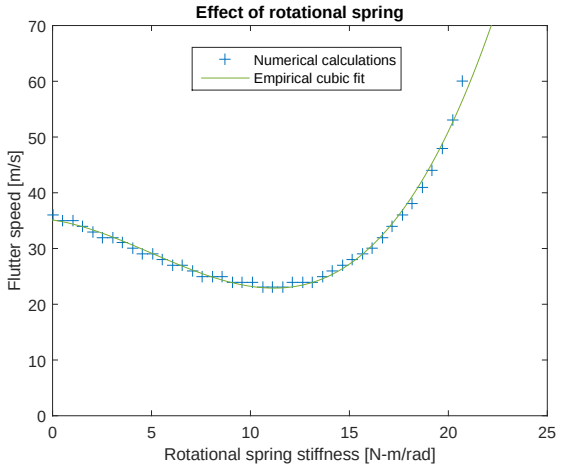


Figure 9.19: Effect of rotational spring on flutter speed.

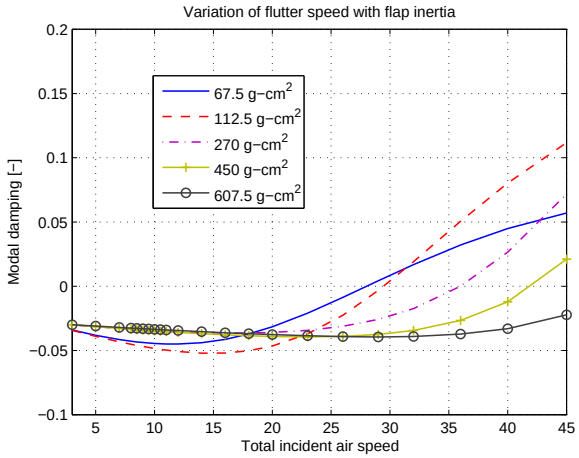


Figure 9.20: Variation of flutter speed with flap inertia.

haviour as the Nastran model. Based on this analytical model, it is shown possible to determine the system transfer functions, suitable for controller design. It is also possible to use the model to explore parametric sensitivity of system dynamic behaviour, especially to analyse the effect of structural parameters on the flutter limit of the blade. For instance, by varying the flap inertia I in the model, it can be seen that the flutter speed can be tuned, as shown in Fig. 9.20. With increasing flap inertia, the flutter speed increases until flutter no longer occurs within the operational range of the turbine. However, since this is also accompanied by a reduction in control authority, it is desired to tune the flutter speed to occur just beyond the turbine operational envelope. Hence, a flap inertia value of 270 g-cm^2 is chosen, which corresponds to a flutter speed of 36 m/s.

Since the modelling procedure above gives a qualitative estimate of system behaviour, which may in practice deviate from the true system behaviour on account of parametric uncertainty and unmodelled dynamics, the next section explores recursive identification of the new flexible rotor integrated with the scaled turbine setup, described in Section 9.2.

9.4. RECURSIVE SYSTEM IDENTIFICATION

The blades are assembled on the scaled turbine, and experiments are conducted to understand system behaviour in the wind tunnel environment, for different operating wind conditions, in order to understand the effect of the free-floating flaps on blade loads. For this purpose, the recursive identification algorithms delineated in Chapter 3 will be utilised.

For operational control, the pitch actuators, for the current of experiments, command constant fine pitch for maximal power capture, and maximal thrust loading. As before, the turbine is operated in constant load mode, such that the rotor speed varies

linearly with the wind speed. The nominal operating speed is taken to be 230 rpm; for the current set up, this occurs at a wind speed of 4.5 m/s. This linear variation in rotor speed does not occur in the above-rated operational region of a standard wind turbine, instead, the speed is regulated to a constant value by using collective pitch control. However, the reason for implementing linear speed variation is threefold: it enables investigations into overspeed behaviour that may induce flutter, it emulates below-rated turbine behaviour, and it enables the validation of an adaptive controller that has to adjust its behaviour to account for large changes in system dynamics and exogenous forcing frequencies.

A cautious ramp up of wind speed in the wind tunnel shows that the rotor enters flutter at a wind speed of 6 m/s, corresponding to a rotor speed of 315 rpm. In order to ensure stable operation at and beyond this wind speed, a nominally stabilising classically-designed PID controller, acting on feedback from the flap angle sensor and issuing flap actuation commands, is connected in feedback with the wind turbine system.

The objective of flap control, like the objective of pitch control in the previous chapter, is the minimisation of blade loads as measured by strain sensors located at the root of each blade. The blade root load spectrum can be visualised in Fig. 9.21. It cana be seen that the dominant load peaks occur at the frequencies 1P, equal to rotor speed, and 2P, its first harmonic. These loading peaks arise out of tower passage, and are hence exogenous by nature. As such, these frequencies have to be compensated for in the algorithms of Chapter 3 by considering an exogenous periodic disturbance signal d_k, defined as:

$$d_k = \begin{bmatrix} \sin\Psi_k \\ \cos\Psi_k \\ \sin 2\Psi_k \\ \cos 2\Psi_k \end{bmatrix}, \tag{9.21}$$

where Ψ_k is the measured azimuth of the rotor. As the azimuth is a direct measurement from the speed encoder located in the turbine nacelle, the artificial signal d_k is always phase-locked with the true blade load peaks 1P and 2P, irrespective of the operating wind speed. As such, this signal can be used for system identification even when the wind speed, and hence the rotor speed, varies over time.

Since the system dynamics vary with wind speed, an online recursive identification method would be useful to track the changes in the dynamics. The primary transfer function of interest is the system from flap action to blade loads. The flap angle sensors form an auxiliary degree of freedom, the transfer function between flap actuation and flap angular motion is also of interest. Specifically, to ensure stability, a classically-designed controller K is connected in feedback between this sensor and the flap actuators:

$$K = \underbrace{0.0001}_{\text{Static gain}} \underbrace{\frac{s/0.001+1}{s/10+1}}_{\text{High-pass}} \underbrace{\frac{s^2+0.001s*50/2\pi+(50/2\pi)^2}{s^2+0.1s*50/2\pi+(50/2\pi)^2}}_{\text{notch for 50 Hz electrical back-coupling artefact}} \underbrace{\frac{1}{2\pi s/100+1}}_{\text{Low-pass}}. \tag{9.22}$$

As such, the recursive identification has to be performed online, in closed loop. Hence, a closed-loop identification technique, like Recursive PBSID (RPBSID), described in Chapter 3, is suitable for this application.

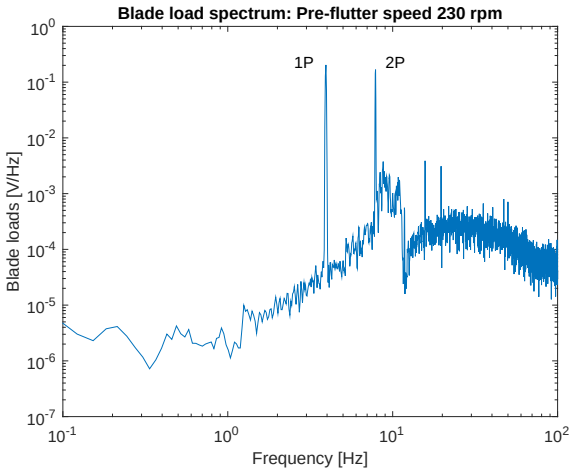


Figure 9.21: Blade load spectrum at the pre-flutter rotor speed of 230 rpm.

In order to validate the results of RPBSID, first, batchwise system identification is performed using standard PBSID, for a number of different constant wind speed conditions, with resulting variance accouted for (vaf) exceeding 60% in each system identification experiment. The transfer functions from the flap actuation to the flap angular position, and from the flap actuation to the blade loads, are represented by the Bode diagrams in Fig. 9.22 and Fig. 9.23, respectively. It can be seen that the transfer involves two primary low-frequency modes. The rigid-body flap mode has relatively high damping, and its frequency increases in an approximately linear fashion with wind speed, reaching nearly 15 Hz close to the flutter speed. On the other hand, the flexible blade out-of-plane mode has a constant frequency slightly higher than 15 Hz, that does not change with wind speed. The damping of this mode reduces progressively with increasing wind speed, and the mode becomes unstable at 290 rpm, corresponding to 5.8 m/s inflow wind speed, or 31.4 m/s total inflow air speed at the rotor tip. These results corroborate closely the numerical results of the previous section.

Recursive identification will be explored next, first under constant operating conditions and then under varying wind conditions.

9.4.1. CONSTANT OPERATING CONDITIONS

For the first set of experiments, the turbine is operated at a constant wind speed of 4.5 m/s, which corresponds to a constant rotor speed of 230 rpm, corresponding to the pre-flutter regime of operation. The stabilising collocated controller is switched off, thus the plant is operated nominally in an open-loop setting. A white noise signal low-pass filtered with cut-off frequency 50 Hz, is imposed on the flap actuators, to provide persistency of excitation. The recursive identification algorithm parameters are all initialised to zero, in other words, the recursive algorithm is cold started. A low value of 0.997 is used for the forgetting factor in the resursive least squares estimation problems to be

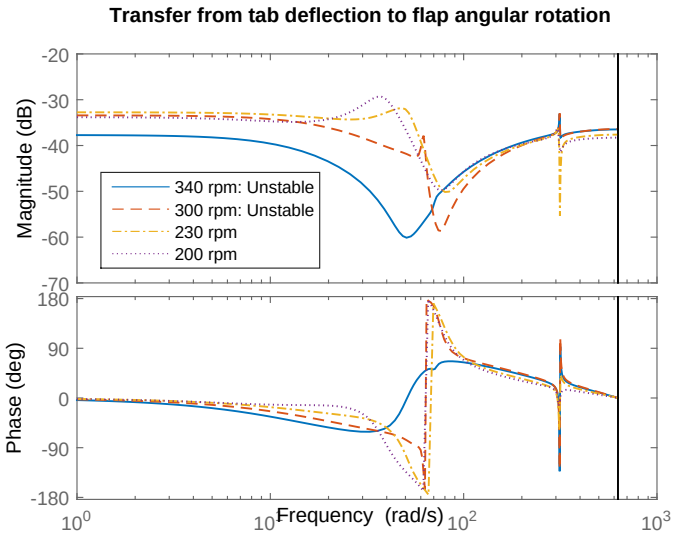


Figure 9.22: Transfer from flap actuators to flap angle, different wind speeds.

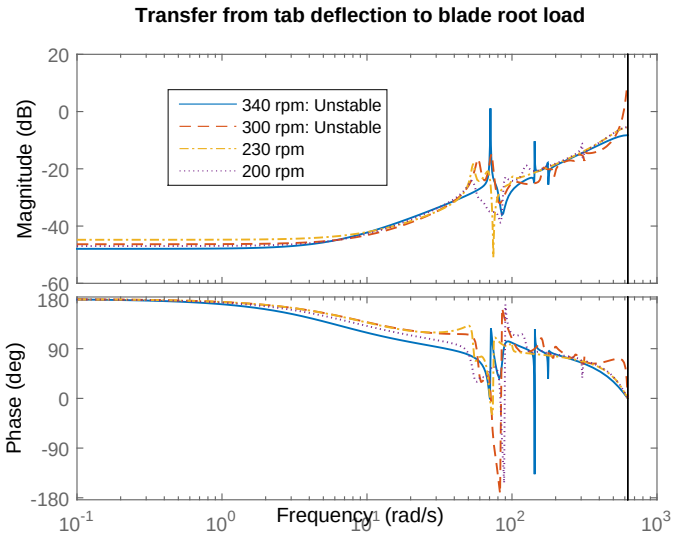


Figure 9.23: Transfer from flap actuators to blade root loads, different wind speeds.

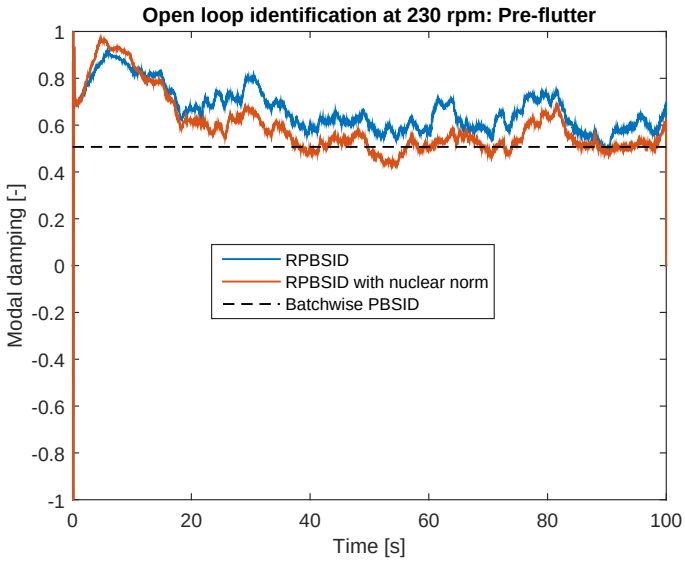


Figure 9.24: Open-loop pole tracking at pre-flutter speed, comparison of recursive and batch algorithm.

solved for recursive identification. From numerical modelling, an approximate state order 4 is chosen. Although the real-time system has sampling frequency 2000 Hz, the data is resampled to 500 Hz, such that the identificaiton algorithm focusses on low-frequency dynamics. The objective of identification is to estimate the transfer from flap activity to blade loads.

The estimation of the damping of the poles obtained by applying the recursive algorithms of Chapter 3 is shown in Fig. 9.24. It can be seen that the conventional recursive PBSID converges within 40 seconds to a pole estimate close to that obtained using the batchwise algorithm. On the other hand, the nuclear-norm enhanced version is able to converge faster and with lower variance than its standard counterpart.

Next, the turbine is operated at a wind speed of 6.5 m/s, which corresponds to a rotor speed of 340 rpm. This wind condition corresponds to the post-flutter regime of operation. Since the system is then open-loop unstable, the stabilising controller from flap angle rotation to flap actuator is switched on. The white noise signal is superposed on this control signal to ensure persistency of excitation. Once again, the recursive algorithms are initialised to zero, and identification is carried out with a low value of forgetting factor. The identification results can be seen in Fig. 9.25. Both recursive algorithms are able to converge to a fairly accurate estimate of the (unstable) modal damping within 40 seconds. Once again, the nuclear norm-enhanced algorithm shows faster convergence and lower variance than the standard algorithm.

The effect of the nuclear norm can be observed by considering the Markov parameters estimated by the two algorithms in Fig. 9.26. It can be seen that estimates of the

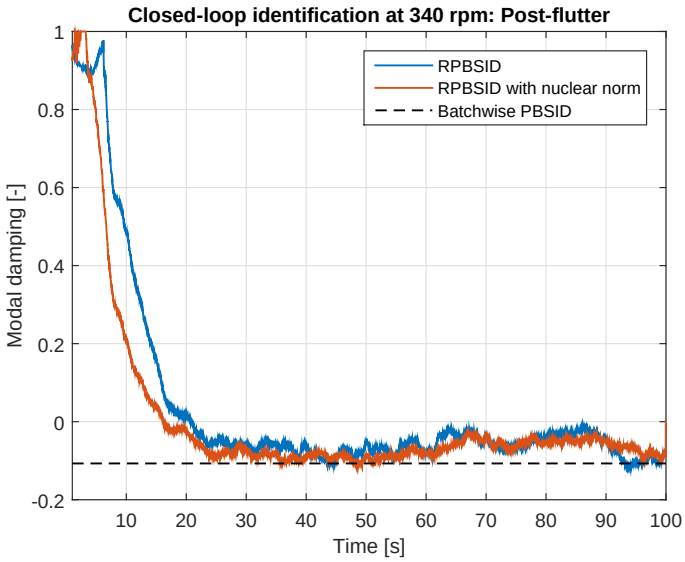


Figure 9.25: Closed-loop pole tracking at post-flutter speed, comparison of recursive and batch algorithm.

low-order Markov parameters, for instance CK and CB, with and without the use of the nuclear norm, are very close to each other. The effect of the nuclear norm is to suppress the magnitude of the higher-order Markov parameters. Since these higher-order parameters are usually corrupted to a larger extent by measurement and process noise, the effect of the nuclear norm is to make system parameter estimation less sensitive to noise.

The variation of Markov parameter estimates over time, with and without the use of the nuclear norm, can be seen in Fig. 9.27. The behaviour of the estimated Markov parameter CB is representative for the lower-order Markov parameters, while that of the parameter CA^5B is representative for the higher-order Markov parameters. It can be seen that, with the use of the nuclear norm, there is a clear bias between the estimates of CB, although their variation over time is similar. On the other hand, the both the variance and the mean of the Markov parameter CA^5B are significantly reduced by the use of the nuclear norm. As the variance in parameter estimation is caused by noise, for this case of constant operating conditions, the overall estimate of system dynamics is less sensitive to the effect of noise when the nuclear norm is used.

The objective of a recursive identification method is to track variations in system dynamics over time, and hence the response of the algorithms under varying operating conditions is explored in the next section.

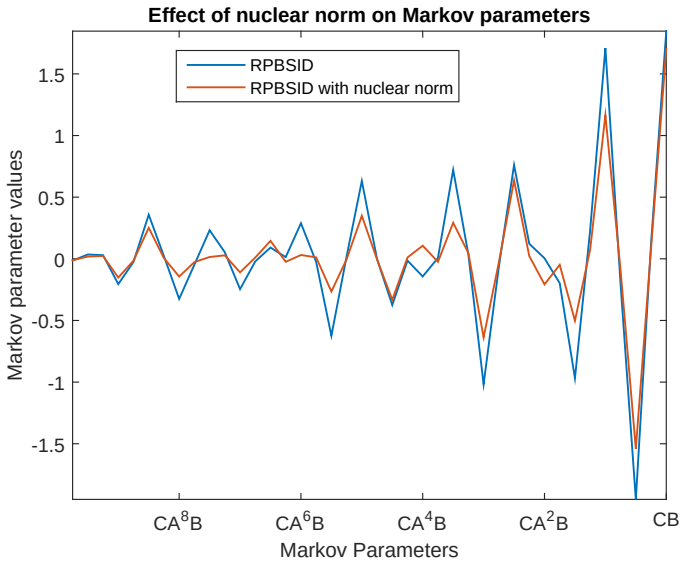


Figure 9.26: Effect of using the nuclear norm on estimated Markov parameters.

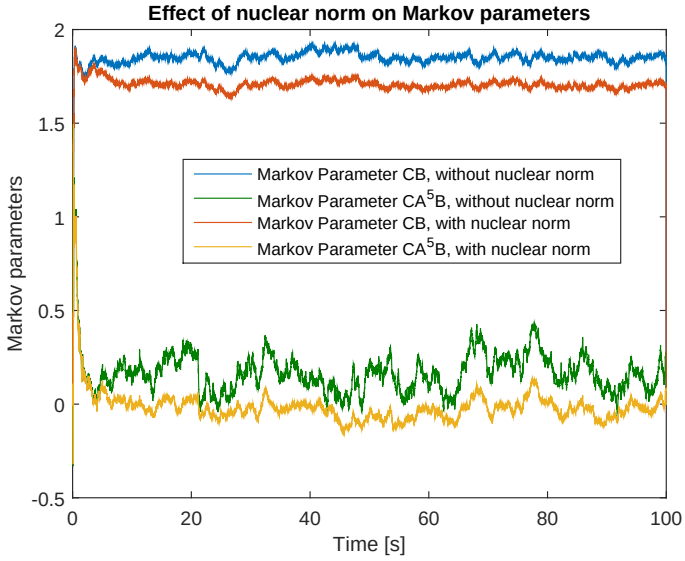


Figure 9.27: Effect of using the nuclear norm on estimated Markov parameters.

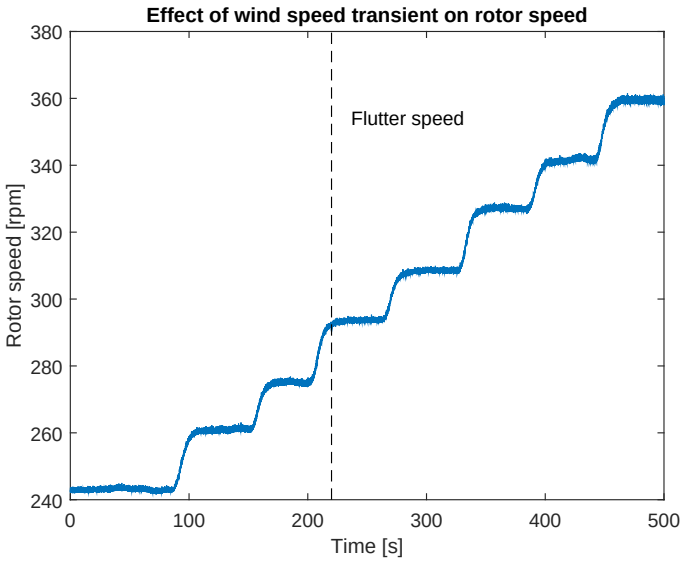


Figure 9.28: Changes in the rotor speed as a consequence of varying the wind speed in a stepwise manner in the wind tunnel.

9.4.2. TIME-VARYING OPERATING CONDITIONS

For this case, the wind speed in the tunnel is increased in a stepwise manner from a pre-flutter wind speed of 4.5 m/s to a post-flutter wind speed of 7 m/s. Accordingly, the rotor speed also increases in a stepwise manner from 230 rpm to 360 rpm, as seen in Fig. 9.28. The turbine crosses the flutter limit at a rotor speed of 290 rpm. This is indicated in the figure by a vertical broken line at a time instant of 220 s.

Since the turbine is expected to operate both in the stable pre-flutter regime as well as the unstable post-flutter regime in this set of experiments, the stabilising collocated controller is connected in closed loop with the plant, as such, closed-loop identification is performed in this experiment set. A filtered white noise signal is superposed on the output of this controller in order to ensure persistency of excitation. At time $t = 0$, the parameters of the identification algorithm are initialised to zero. However, they are not reïnitialised for every change in wind speed that occurs within the experiment, and thus, the algorithm is warm started using parameter values from the previous time instants in order to speed up convergence.

The results obtained by using the identification algorithms can be seen in Fig. 9.29. It can be seen that the estimates of the modal damping stabilise temporarily at the same values as those obtained from the corresponding identification experiments conducted at constant wind speed. Herewith, it can be concluded that the system is not just time-varying, but also parameter-varying, with wind speed as the scheduling variable.

The figure also shows that the use of the nuclear norm makes the algorithm more responsive; it converges faster and with lower variance to a good approximation of the

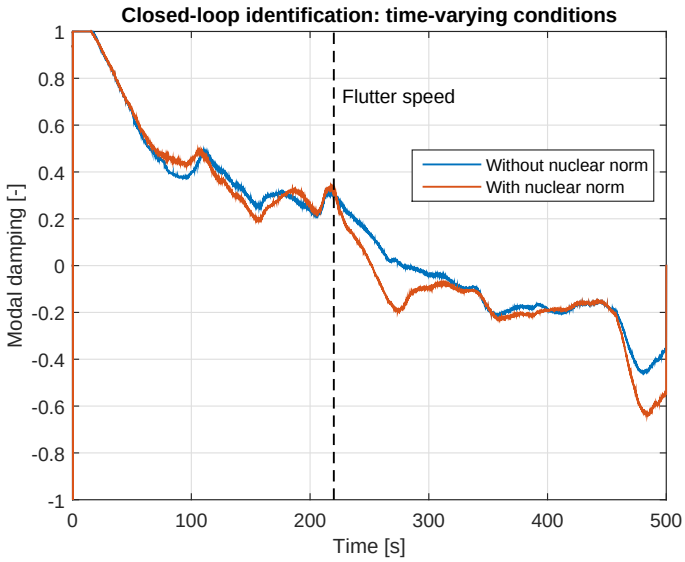


Figure 9.29: Closed-loop pole tracking for a stepwise increase in ambient wind speed.

value of modal damping. It can also be seen that the onset of flutter is detected by the nuclear norm-enhanced algorithm within 32.4 seconds. On the other hand, the conventional algorithm requires 60.8 seconds to detect the onset of flutter.

Thus, it can be concluded that both algorithms described in Chapter 3 are able to estimate system parameters with a high degree of accuracy. The effect of the nuclear norm is to suppress the value and the influence of noise on the higher-order Markov parameters, and thereby to render the algorithm faster in terms of convergence and to reduce the variance of parameter estimates. With the system dynamics and flap control authority validated, the combined use of flap and pitch control to reduce blade loads is explored in the next section.

9.5. ITERATIVE FEEDFORWARD TUNING FOR COMBINED PITCH AND FLAP CONTROL

In this section, the blade pitch and flap actuators are used simultaneously in order to reduce blade loads. As described in the previous section, the blade load spectrum is dominated by the 1P and 2P peaks and their harmonics, caused by tower shadow rotor loading. On a field turbine, the effect of turbulence would be to broaden the peaks, and add energy to the high-frequency part of the spectrum. However, even for a wind turbine in the field, the primary load peaks occur at 1P and 2P. As such, the load controller tested in this section primarily target load alleviation at these frequencies.

In order to ensure safe operation at post-flutter wind speeds, the nominally stabilising collocated controller is kept operational in closed loop between the flap angle feed-

back and the flap actuation mechanism. To decouple the load alleviation objective from the system stabilisation objective, a feedforward load controller is tuned using IFT, in such a manner that input-output data is used to improve the load control performance iteratively over time.

The optimal load controller parameters depend upon the operating wind conditions, and hence an LPV controller would be ideally suited for this application, where the wind speed is considered to be the scheduling variable. However, as explored in Chapter 5, the full LPV implementation of IFT demands significant computational complexity, and hence, the simplifying assumption is made that the wind speed remains approximately constant over the duration of each IFT experiment. With this quasi-LPV approach, the plant can be assumed to be LTI for the duration of each experiment, and IFT-LTI can be used to tune optimal controller parameters for different operating wind speeds. However, these parameters will be optimal only for one specific wind speed, and they will be sub-optimal over the operating range of the wind turbine. Hence, IFT-LTI has to be perfomed for a number of different operating wind speeds, or, alternatively, the approach of IFT has to be used to optimally tune a gain-schedule for varying wind speeds for the feedforward load controller.

The implementation of the feedforward load controller can be seen in Fig. 9.30. As in the previous chapters, the reference signal for the feedforward controller is taken to be a set of phase-locked sinusoidal basis functions, and thus this load control approach is similar to an adaptive implementation of cyclic pitch and flap control. For the pitch actuators, the sinusoidal basis functions have frequency 1P, while for the flap actuators, the sinusoidal basis functions have frequency 2P. As such, pitch and flap activity is decoupled in the frequency domain, and the actuators are expected to perform load control strictly at the corresponding frequencies, such that the cross-influence of actuation efforts is minimised. The controller parameters ρ are essentially the weights of these basis functions that are to be optimally tuned to maximise load reduction; each of the controller parameters is considered to be an affine function of the ambient wind speed. For pitch control, hence, 4 controller parameters are to be optimised, denoted as $\rho_{s,IPC}$, $\rho_{sV,IPC}$, $\rho_{c,IPC}$ and $\rho_{cV,IPC}$, and similarly for flap control.

The procedure of IFT for optimal gain schedule tuning, described in Section 5.2, is carried out under different wind conditions in order to synthesise a locally optimal load controller. First, IFT is used at constant wind speed, in both the pre- and post-flutter speed regimes. Then, an optimal gain schedule is trained using IFT for varying wind conditions.

9.5.1. CONSTANT WIND SPEED: PRE-FLUTTER

Selected results regarding the use of IFT in the pre-flutter regime are presented here; the behaviour of the algorithm is similar for different wind speeds. First, IFT is used for tuning the feedforward load controller at a constant wind speed of 4.5 m/s, which corresponds to a rotor speed of 230 rpm. The evolution of the controller gains and the cost criterion over iterations can be seen in Fig. 9.31 and Fig. 9.32. The gains converge to their steady-state values over a period of ten minutes, and achieve load reduction. The blades and actuator duty after convergence can be seen in Fig. 9.33 and Fig. 9.34. It can be seen that the actuator signal commanded is sinusoidal for both the pitch and flap

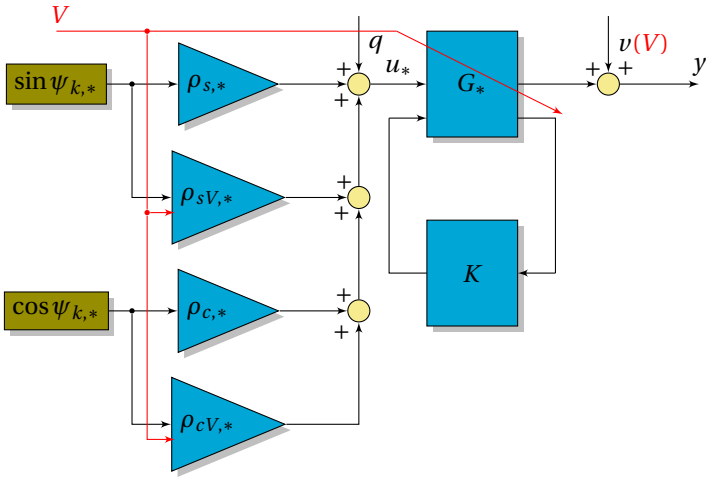


Figure 9.30: IFT implementation: wind turbine load control. $*$ stands for IPC or IFC.

actuators, with frequency 1P and 2P respectively. Due to this control action, the 1P and 2P loads in the blade spectrum are almost entirely eliminated, and IFT is successful in tuning the feedforward controller optimally.

It is interesting to note that the optimal control action commanded from the actuators of the two blades is not exactly antisymmetric. This effect is especially pronounced in the flap actuator signal. The reason for this unequal duty cycle is the difference in the manufacturing of the blades. Indeed, as per the numerical modelling, a change of a few grammes in the weight of each flap can strongly alter the aeroelastic behaviour of the blade, and it can even prepone flutter.

These manufacturing discrepancies are expected to be lower for a commericial turbine. A conventional controller, as described by Bossanyi (2003), would demand antisymmetric pitch activity and identical flap activity for load alleviation. However, such a controller would not be able to account for asymmetric actuator control authority arising out of aging issues or external wind conditions, and hence, it would not be able to reach the same degree of load alleviation achievable by the load controller described above. Similar results are observed at wind speeds in the post-flutter regime.

9.5.2. CONSTANT WIND SPEED: POST-FLUTTER

The wind turbine is run at a series of post-flutter wind speeds, and selected load alleviation results, for a post-flutter rotor speed of 330 rpm, are described here. For this set of experiments, the nominally stabilising controller K is connected in closed loop between the flap angle sensor and flap actuation mechanism. This controller is not optimised for load reduction, its only purpose is to stabilise the system. IFT then tunes the feedforward controller that shapes the loads acting on this nominally stabilised baseline closed-loop plant. It can be seen in Fig. 9.35 and Fig. 9.36 that, even under the challenging unstable post-flutter conditions, the load controller tuned using IFT is still able to achieve load reductions. From the actuation signals, it can be seen that pitch activity is no longer close

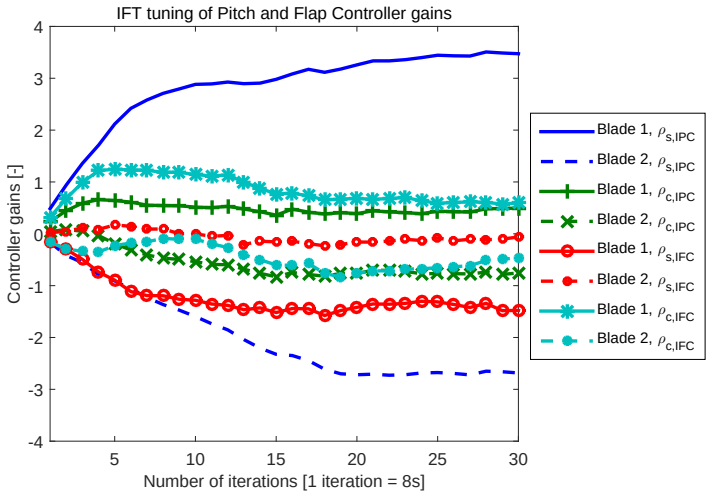


Figure 9.31: Convergence of controller gains over iterations.

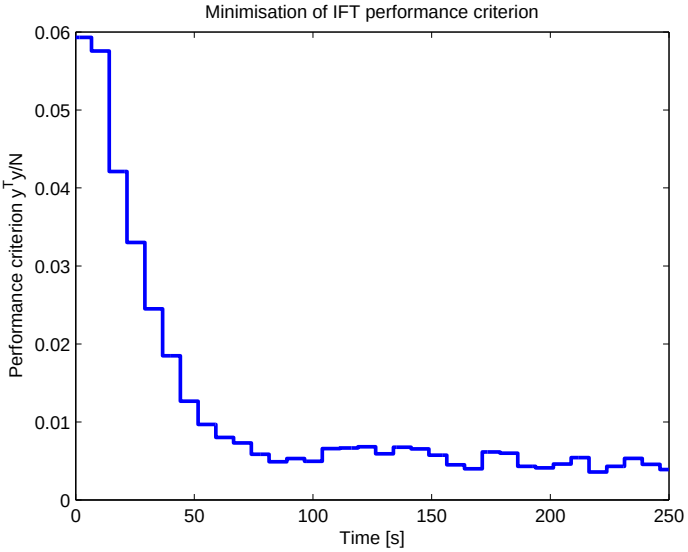


Figure 9.32: Minimisation of IFT cost criterion over iterations.

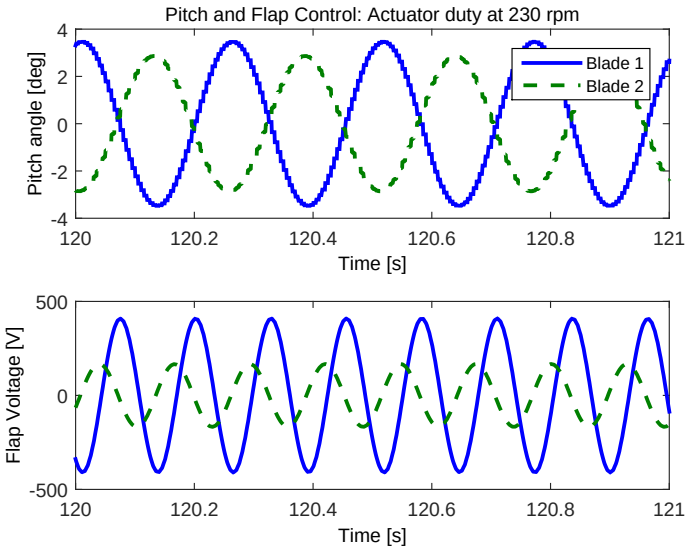


Figure 9.33: Actuator duty cycles of optimised controller (Pre-flutter).

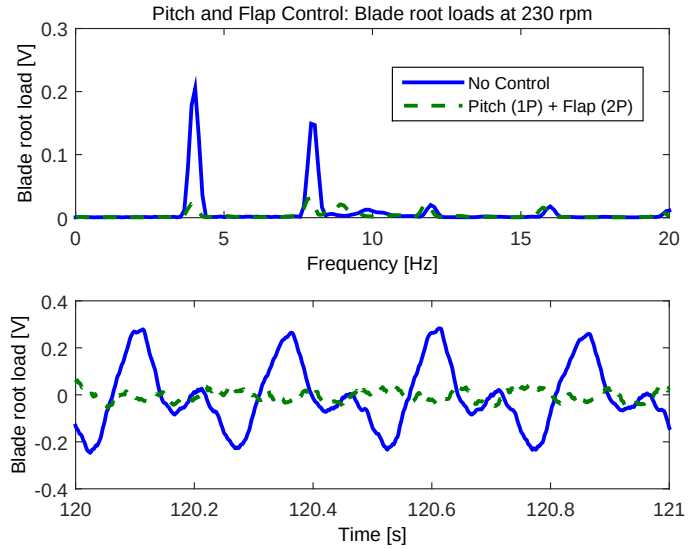


Figure 9.34: Blade root load reduction of optimised controller (Pre-flutter).

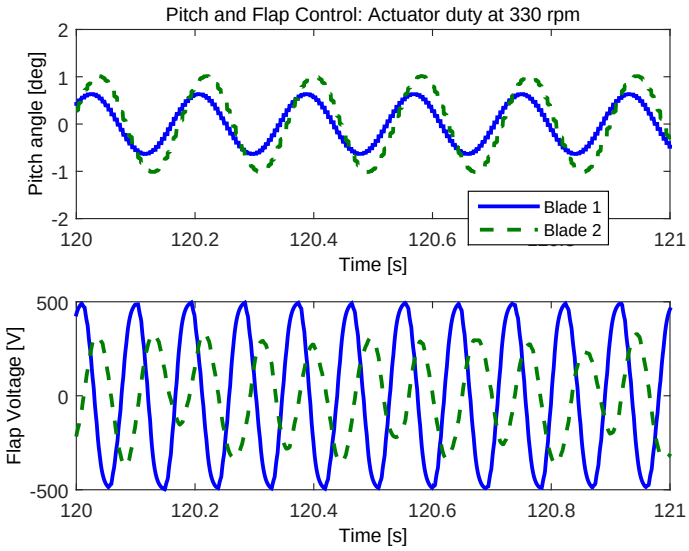


Figure 9.35: Actuator duty cycles of optimised controller (Post-flutter).

to antisymmetric, and a conventional IPC controller would not be able to achieve load reductions under these conditions. Further, it can also be seen that flap activity is close to its maximal limits of ±500 V, beyond which it is not possible to use the flaps to either stabilise the system or to achieve load reduction.

Since the wind speed experienced by the turbine in the field cannot be considered to be constant, the next section describes results obtained with varying wind speeds.

9.5.3. VARYING WIND SPEED

In the case of varying wind speed, IFT is used to determine a gain schedule for the controller gains instead of constant controller gain values. The algorithm of Section 5.2 is used to estimate an ideal linear gain schedule that is optimal over the entire operating range of the wind turbine. As such, IFT is used to iteratively tune the slope $\rho^{[1]}$ and intercept $\rho^{[0]}$ of the optimal controller gain schedules in this section. A varying wind speed profile is generated in the wind tunnel; it can be visualised in Fig. 9.37. The optimal gain schedule parameters estimated can be seen in Fig. 9.38 and Fig. 9.39. Since the wind speed is nearly constant for the first 100 seconds, a good gain schedule cannot be obtained due to a lack of persistency of excitation in the scheduling variable (wind speed). However, once the wind speed begins to change, the gain schedule rapidly converges to its steady state value. The steady-state optimally tuned linear gain schedules are compared against the optimal controller gains for constant operating wind speeds in Fig. 9.40 and Fig. 9.41. It can be directly observed that the optimal gain schedule for the pitch controller is a good fit of the optimal controller gains tuned for constant operating wind speeds. On the other hand, it can also be observed that the variation of the optimal

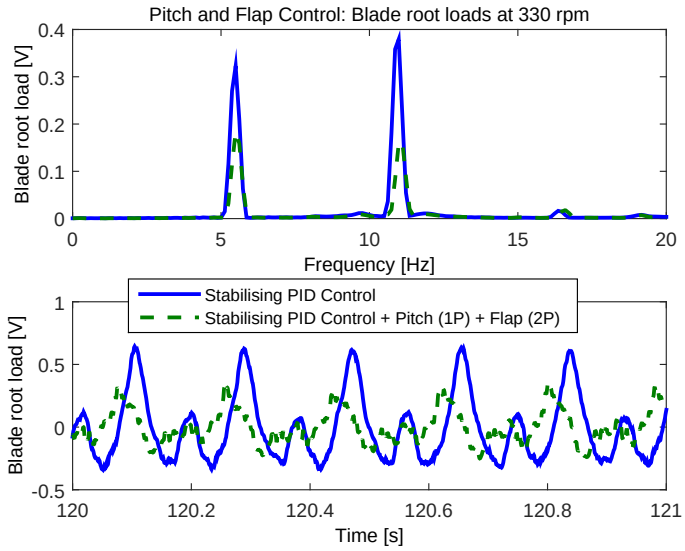


Figure 9.36: Blade root load reduction of optimised controller (Post-flutter).

flap controller gains does not vary linearly with wind speed, and a linear gain schedule is suboptimal for a large part of the operating range.

Thus, combined pitch and flap control is experimentally shown capable of achieving significant rotor load reductions. Further, the extension of IFT, capable of tuning gain-scheduled controllers, has also been demonstrated experimentally.

9.6. CONCLUSIONS

This chapter provides experimental proof of concept of the load alleviation potential of free-floating flaps for wind turbines. It also demonstrates combined pitch and flap control of rotor loads, for the first time on an experimental wind turbine.

The blades of the scaled turbine described in the previous chapter are redesigned such that they can be instrumented with trailing-edge free-floating flaps. A new manufacturing method is used for blade production: the aerodynamic shell is first 3D printed using the plastic material PC-ABS, and then bonded with an external carbon fibre spar for structural stiffness. This combination of materials was experimentally found to show good bonding properties. The free-floating flap was fit into a slot at an outboard blade section; it is free to rotate about its hinge axis, while its camberline can be modified by piezoelectric actuation. The free rotation of the flap is measured by an angle encoder.

The dynamic behaviour of the blade was analysed numerically, and a low-order Linear Parameter-Varying (LPV) analytical model was set up to facilitate controller synthesis. With a correction factor to compensate for the cross-influence of blade elements, the analytical model could be tuned to match the dynamic behaviour of the numerical

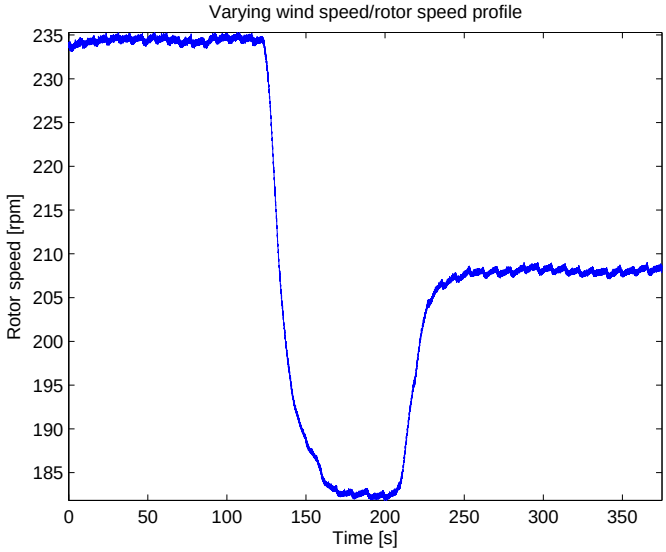


Figure 9.37: Varying operational speed for optimisation of gain schedule.

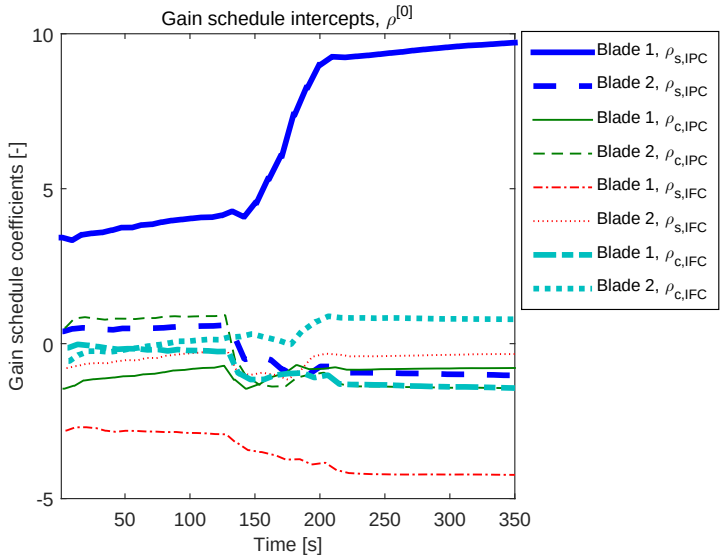


Figure 9.38: Optimisation of gain schedule intercepts for varying wind speed conditions.

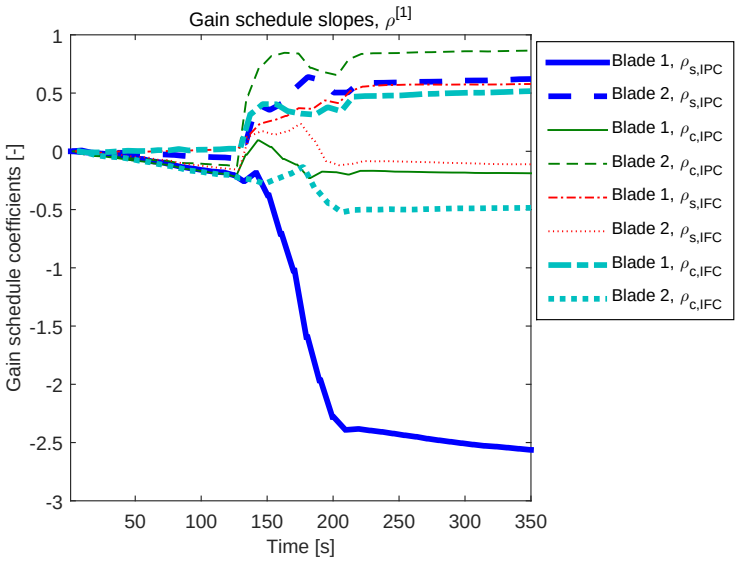


Figure 9.39: Optimisation of gain schedule slopes for varying wind speed conditions.

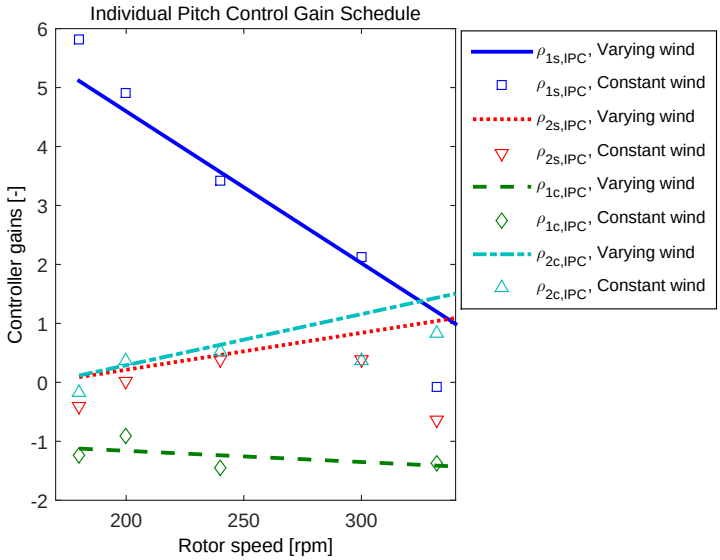


Figure 9.40: Gain schedule at varying wind speeds versus optimal gains at constant wind speeds: Pitch control.

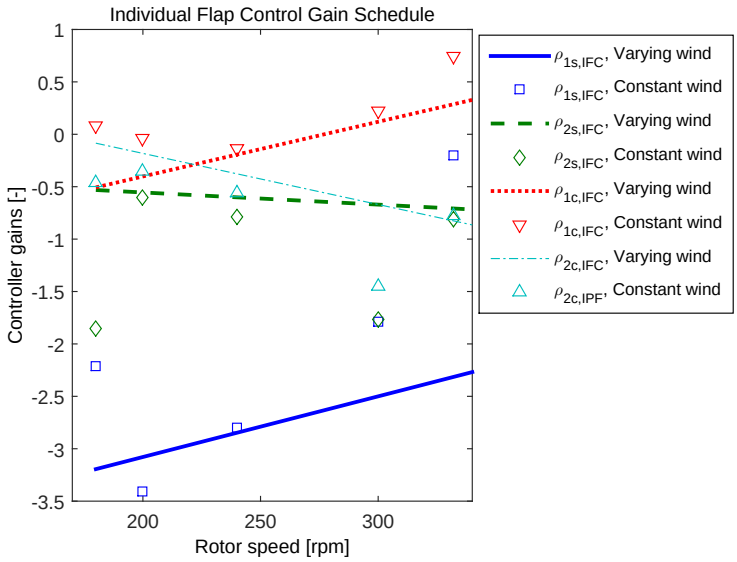


Figure 9.41: Gain schedule at varying wind speeds versus optimal gains at constant wind speeds: Flap control.

model. Both models predict that the flap rigid-body motion couples with the flexible flapwise motion of the blade for increasing wind speeds, and the system becomes open-loop unstable beyond the flutter speed. This flutter speed could be tuned by adjusting the inertia properties of the flap, and it is designed to occur just beyond the operational regime of the wind turbine, in order to maximise control authority while retaining open-loop stability. A theoretical exercise was conducted to explore the effect of a rotary spring of finite thickness, connecting the trailing edge flap to the main blade structure. It was found that control authority increases for a specific optimal value of the stiffness of the spring, but it also causes a lowering of the flutter speed.

The setup was assembled and operated in the wind tunnel, and it was found that the system goes into the unstable flutter mode at an inflow wind speed of 6 m/s. To ensure structural integrity, a classical controller was connected in closed loop between the collocated sensor and the flap actuation mechanism. Identification experiments were carried out, both in the pre-flutter and the post-flutter regimes of operation. Batchwise identification using PBSID corroborated the results obtained from numerical analysis in terms of the behaviour of modal frequency and damping over the wind speed range. Recursive algorithms, discussed in Chapter 3, were also applied to the setup under constant and varying wind conditions. The variation of modal damping could be tracked with reasonable accuracy by these algorithms, and they could also detect the onset of flutter in varying wind conditions. The advantage of using the nuclear norm was clear: it suppressed the effect of noise and hence increased responsiveness and reduced variance estimates. With the nuclear norm, flutter could be detected within half the time

that it would otherwise take the conventional algorithm.

The load reduction potential of combined pitch and flap control was then investigated. The nominally stabilising controller was allowed to run in closed loop; over and above this controller, a feedforward load controller was designed, the gains of which were tuned using IFT. The reference to this feedforward controller was an artificial sinusoidal signal with frequency 1P for pitch actuation and 2P for flap actuation, thus achieving frequency separation of the two actuation mechanisms. With the use of IFT, the gains of these controllers were optimised at constant wind speeds such that maximal load reduction could be achieved. Under pre-flutter conditions, near-optimal load control was shown to be possible, with the 1P and 2P peaks virtually eliminated from the load spectrum. Load alleviation was also shown to be possible even at post-flutter wind speeds, however the load attenuation was limited by the control authority of the flaps. It was interesting to note that the commanded actuator duty cycles, especially in the post-flutter regime, were not (anti-)symmetric, as commanded by conventional load control strategies. It is postulated that this extra control degree of freedom, which does not impose symmetry on the actuator control signals, is responsible for the increased load attenuation achievable using IFT. Finally, IFT was shown capable of tuning an optimal gain schedule in varying wind conditions. While a linear gain schedule was shown to be near-optimal for pitch control, flap control shows a non-linear variation of optimal controller gains with wind speed, and further work is required to understand the best manner to achieve optimal flap control behaviour throughout the operating range.

Thus, the experiments reported in this chapter form a first step towards understanding the advantages and challenges associated with implementing free-floating flaps on commercial wind turbines.

III

Conclusions and Recommendations

CONCLUSIONS AND RECOMMENDATIONS

असतो मा सद्गमय।
तमसो मा ज्योतिर्गमय॥

The Upanishad, attributed to multiple authors, India (circa 7^{th}-6^{th} century BC), describing the recommended direction of motion for society.

This final chapter puts together the conclusions drawn from the results obtained in Part II of the thesis, based on the theoretical work of Part I of the thesis. The lessons learnt are described, and recommendations are made for future theoretical work and experimental validation in the field of improved load control for flexible wind turbine rotors.

CONCLUSIONS

The main research goal of this thesis was to investigate if load control for wind turbines could be achieved in a data-driven manner such that the turbine-specific control law is updated iteratively over time, optimising performance under different wind conditions. From the theoretical and experimental investigations conducted, it can be concluded that such a form of adaptive load control can be made practically tractable, if inherent structure in the problem is exploited. The two-step convex data-driven approach of Subspace Predictive Repetitive Control, developed in Chapter 4, can provide a first approximation for the structure of the ideal control law for load alleviation. A fixed-structure control law can be iteratively updated for all operating conditions using Iterative Feedback Tuning for Linear Parameter-Varying Systems (IFT-LPV) developed in Chapter 5.

The theoretical and experimental research subquestions raised in the introduction have been answered in the foregoing chapters of the thesis:

1. *How can recent advances in low-variance system identification be modified to fit a closed-loop, online environment?*

   Conventional closed-loop recursive identification has been extended by using the nuclear norm in the optimisation cost function. For recursive identification, this reduces estimate variance for low values of the forgetting factor. This is achieved by the suppression of higher-order system Markov parameters, which are heavily influenced by measurement noise, as such, the nuclear norm reduces the sensitivity of online identification to noise in the data.

2. *How can online system identification be combined with controller synthesis to minimise periodic loads, with precise control over the shape and smoothness of the actuator commands?*

   System parameter estimates from online identification has been used in Subspace Predictive Repetitive Control (SPRC), designed specifically for attenuating periodic

disturbances. For this purpose, the control law is synthesised in a lifted domain, with the size of the lifting window equal to an integral multiple of the fundamental period of turbine rotation. Computational complexity is strongly reduced by forcing the control input to remain within a subspace defined by basis vectors, for increased control over actuator signal attributes.

3. *How can the gains of low-order fixed-structure LPV controllers be optimally tuned for LPV systems?*

   The direct, data-driven concept of Iterative Feedback Tuning (IFT) is extended in Chapter 5 to apply to the LPV case. By decomposing the system matrices into a time-varying, known, scheduling-dependent factor and a time-invariant, unknown factor, it is possible to compensate for the perturbing effect of varying wind conditions such that the performance gradient with respect to the controller parameters can be estimated in an unbiased manner. Structure can be exploited for limiting the increase in the number of experiments required to estimate performance gradients.

4. *How do the control strategies discussed in Part I behave in a fully non-linear simulation environment for commercial wind turbines?*

   The theoretical predictions of the behaviour of SPRC and IFT-LPV have been validated for wind turbine pitch and flap control using aeroelastic simulations. Further, the concept of pitch control for yaw stabilisation is proposed, for an expanded turbine load control design space.

5. *How do the control strategies discussed in Part I affect the loading behaviour of a scaled prototype of a pitch-controlled wind turbine, under controlled, wind tunnel conditions?*

   An experimental setup has been designed for operation in the Open Jet Facility wind tunnel at the Delft University of Technology. The two-step data-driven approach of SPRC has been validated, for challenging operating conditions, and for asymmetric blade behaviour, and it is found suitable for real-time implementation. Also, on this setup, for the first time, individual pitch control for yaw stabilisation has been demonstrated experimentally.

6. *How can a pitch-controlled turbine be extended to include trailing-edge flaps, and how should the control strategies of Chapter 8 be modified to achieve optimal load control for the wind turbine?*

   The concept of free-floating flaps was demonstrated on a wind turbine for the first time experimentally, and, combined with pitch control, was shown to possess significant load reduction potential. It was also shown that this form of a flexible rotor is susceptible to a low wind speed form of flutter. However, stable operation and load rejection was achieved even under varying post-flutter operating conditions.

Thus, overall, the thesis proves the potential and feasibility of the concept of iterative data-driven control, both numerically and experimentally, via the two-step as well as the direct approach, for the case of load control of flexible rotors with both conventional as well as advanced actuation methods.

RECOMMENDATIONS AND FUTURE WORK

Practical implementation of structure-exploiting data-driven rotor load controllers is herewith feasible and yields good performance. These control approaches can be considered directly applicable, in a feedforward sense, for load control.

Of the two different data-driven approaches, it would be recommended that Subspace Predictive Repetitive Control (SPRC) be used in high-fidelity turbine simulations, in a variety of different representative wind conditions, including extreme wind gusts and extreme direction changes. After extensive controller validation, the actual controller implemented on the wind turbine should be a simple, low-order, parameterised controller, the parameters of which can then be tuned in the field using Iterative Feedback Tuning for Linear Parameter-Varying systems, such that, for each turbine, the locally optimising controller within the restricted set of parameterised controllers can be found. Since there is as yet no globally satisfactory proof of stability for these adaptive approaches, caution has to be exercised during the iterative optimisation process.

For the new concepts of Individual Pitch Control for yaw stabilisation, and free-floating flaps for blade load mitigation, preliminary steps have been taken to demonstrate the proof of concept. Both of these concepts require further investigation at a theoretical level: experiments using data-driven control can help identify practical operating issues that require both aeroelastic and controller redesign to achieve good performance. A few detailed recommendations for future work are explained in the following subsections.

1. *Identification:* System identification using the nuclear norm improves the quality of the estimate of the system parameters, however, without regularisation, the identification cost function has a convex optimum for the trivial system with all Markov parameters zero. As such, a good (data-driven) trade-off has to be made in weighting the nuclear norm.

2. *Subspace Predictive Repetitive Control (SPRC):* SPRC in the evaluated case required continuous reïdentification because linear time-invariant identification was no longer adequate for the wind turbine system. With a better model structure, such as LPV or Wiener/Hammerstein, the requirement for rapid high-accuracy identification could be relaxed.

3. *Iterative Feedback Tuning (IFT) for LPV systems:* The curse of dimensionality encountered by full IFT-LPV is still required to be addressed by compactifying the LPV factorisation, by using techniques like tensor regression, Gunes et al. (2015). Another desirable extension of the approach is for the case where the scheduling sequence is not exogenous.

4. *Advanced Actuators:* The dynamic modelling of free-floating flaps has several assumptions: independence of blade elements, non-rotating nature, no wake effects, and so on. These effects need to be studied and the analytical model needs to be updated or corrected, while still maintaining its LPV nature.

Herewith, the thesis is concluded with a list of recommendations for facilitating the translation of the theoretical and experimental work, reported herein, into a practical implementation of an optimally-controlled ‘smart’ flexible rotor.

BIBLIOGRAPHY

Åström, K. J. Adaptive feedback control. *Proceedings of the IEEE*, 75:185–217, 1987.

Åström, K. J. and Wittenmark, B. On self-tuning regulators. *Automatica*, 9:185–199, 1973.

Andersen, P. B., Henriksen, L., Gaunaa, M., Bak, C., and Buhl, T. Deformable trailing edge flaps for modern megawatt wind turbine controllers using strain gauge sensors. *Wind Energy*, 13:193–206, 2010.

Annergren, M., Hansson, A., and Wahlberg, B. An ADMM algorithm for solving ℓ_1 regularised MPC. *Proceedings of the American Control Conference*, Montréal, Canada: 4486–4491, 2012.

Arimoto, S., Kawamura, S., and Miyazaki, F. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1:123–140, 1984.

Arnold III, W. F. and Laub, A. J. Generalised eigenproblem algorithms and software for algebraic riccati equations. *Proceedings of the IEEE*, 72:1746–1754, 1984.

Arrenberg, R. Verhandelingen van het bataafsch genootschap der proefondervindelyke wysbegeerte te rotterdam. *Vierde deel*, 1779.

Bak, C., Zahle, F., Bitsche, R., Kim, T., Yde, A., Henriksen, L. C., Natarajan, A., and Hansen, M. Description of the DTU 10 MW reference wind turbine. *DTU Wind Energy*, Report-I-0092, 2013.

Bamieh, B., Pearson, J. B., Francis, B. A., and Tannenbaum, A. A lifting technique for linear periodic systems with applications to sampled data control. *Systems and Control Letters*, 17:79–88, 1991.

Barbarino, S., Bilgen, O., Ajaj, R. M., Friswell, M. I., and Inman, D. J. A review of morphing aircraft. *Journal of Inteliigent Material Systems and Structures*, 22:823–877, 2011.

Barlas, T. K. and Van Kuik, G. A. M. Review of the state of the art in smart rotor control research for wind turbines. *Progress in Aerospace Sciences*, 46:1–27, 2010.

Barlas, T. K., Van Wingerden, J. W., Hulskamp, A. W., Van Kuik, G. A. M., and Bersee, H. E. N. Smart dynamic rotor control using active flaps on a small-scale wind turbine: aeroelastic modelling and comparison with wind tunnel measurements. *Wind Energy*, 16:1287–1301, 2013.

Bauer, S., Czakalla, M., Göppele, F., Grandy, M., Hammer, A., Hofmann, L., Jenter, T., Kaus, S., Pullman, T., A., R., Samol, D., Schlesiger, J., Schwenk, J., Beyer, F., and Cheng, P. W. Abschlussbericht Windenergieprojekt WS 13/14, Windenergie aus dem 3D-Drucker. *Universität Stuttgart*, 2014.

Berg, D. E., Wilson, D. G., Resor, B. R., Barone, M. F., Berg, J. C., Kota, S., and Ervin, G. Active aerodynamic blade load control impacts on utility-scale wind turbines. *Sandia National Laboratories*, Albuquerque, USA, 2009.

Berg, J. C., Barone, M. F., and Resor, B. R. Field test results from the Sandia Smart rotor. *AIAA Aerospace Sciences Meeting including the New Horizons forum*, Grapevine, USA, 2013.

Bernhammer, L. O., Van Kuik, G. A. M., and De Breuker, R. How far is smart rotor research and what steps need to be taken to build a full-scale prototype? *Journal of Physics: Conference Series*, 555:012008, 2012.

Bernhammer, L. O., De Breuker, R., Karpel, M., and Van der Veen, G. J. Aeroelastic control using distributed floating flaps activated by piezoelectric tabs. *Journal of Aircraft*, 50 (3):732–740, 2013.

Bernhammer, L. O., Van Kuik, G. A. M., and De Breuker, R. Fatigue and extreme load reduction of wind turbine components. *Journal of Wind Engineering and Industrial Aerodynamics*, 154:84–95, 2016.

Bil, C., Massey, K., and Abdullah, E. J. Wing morphing control with shape memory actuators. *Journal of Intelligent Material Systems and Structures*, 24(7):879–898, 2013.

Bilgen, O. and Friswell, M. I. Piezoceramic composite actuators for a solid-state variable-camber wing. *Journal of Intelligent Material Systems and Structures*, pages 1–12, 2013.

Bilgen, O., Kochersberger, K. B., and Inman, D. J. Novel, bidirectional, variable-camber aerofoil via macro-fibre composite actuators. *Journal of Aircraft*, 47(1):303–314, 2010.

Bitmead, R. R., Gevers, M., and Wertz, V. Adaptive optimal control: The thinking man's GPC. *Prentice Hall Publications*, 1990.

Bolder, J., Oomen, T., and Steinbuch, M. Exploiting rational basis functions in iterative learning control. *Proceedings of the IEEE Conference on Decision and Control*, Florence, Italy, 2013.

Bossanyi, E. A. The design of closed loop controllers for wind turbines. *Wind Energy*, 3: 149–163, 2000.

Bossanyi, E. A. Individual blade pitch control for load reduction. *Wind Energy*, 6(2): 119–128, 2003.

Bossanyi, E. A. Further load reductions with individual pitch control. *Wind Energy*, 8: 481–485, 2005.

Bossanyi, E. A. and Wright, A. D. Field testing of individual pitch control on the NREL CART-2 wind turbine. *Proceedings of the European Wind Energy Conference*, Marseille, France, 2009.

Bossanyi, E. A., Wright, A. D., and Fleming, P. A. Controller field tests on the NREL CART-2 turbine. *Upwind Work package deliverable report 11593/BR/08*, 2010.

Bossanyi, E. A., Fleming, P. A., and Wright, A. D. Field test results with individual pitch control on the NREL CART-3 wind turbine. *American Institute of Aeronautics and Astronautics*, 1019:1–10, 2012a.

Bossanyi, E. A., Savini, B., Iribas, M., Hau, M., Fischer, B., Schlipf, D., Van Engelen, T., Rossetti, M., and Carcangiu, C. E. Advanced controller research for multi-MW wind turbines in the UPWIND project. *Wind Energy*, 15:119–145, 2012b.

Bottasso, C. L., Croce, A., Riboldi, C. E. D., and Nam, Y. Multi-layer control architecture for the reduction of deterministic and non-deterministic loads on wind turbines. *Renewable Energy*, 51:159–169, 2013.

Bottasso, C. L., Croce, A., Gualdoni, F., and Montinari, P. Load mitigation in wind turbines by a passive aeroelastic device. *Journal of Wind Engineering and Industrial Aerodynamics*, 148:57–69, 2016.

Bristow, D. A., Tharayil, M., and Alleyne, A. G. A survey of iterative learning control. *IEEE Control Systems Magazine*, pages 96–114, 2006.

Buhl, T., Thomsen, K., and Markou, H. Design guidelines for integrated aeroelastic control of wind turbines. *STABCON Task-12 report, EU Contract ENK5-CT-2002-00627*, 2007.

Burton, T., Sharpe, D., Jenkins, N., and Bossanyi, E. A. Wind energy handbook. *John Wiley Publications*, page Chichester, 2011.

Cai, J. F. Fast singular value thresholding without singular value decomposition. *UCLA CAM Report*, 2010.

Campi, M. C. and Savaresi, S. M. Direct non-linear control design: the virtual reference feedback tuning (VRFT) approach. *IEEE Transactions on Automatic Control*, 51(1):14–27, 2006.

Castaignet, D., Barlas, T. K., Buhl, T., Poulsen, N. K., Wedel-Heinen, J. J., Olesen, N. A., Bak, C., and Kim, T. Full-scale test of trailing edge flaps on a Vestas V27 wind turbine: active load reduction and system identification. *Wind Energy*, 17:549–564, 2014.

Chen, T. and Francis, B. A. Optimal sampled-data control systems. *Springer*, 1995.

Chiuso, A. Rôle of vector auto-regressive modelling in predictor-based subspace identification. *Automatica*, 43(6):1034–1048, 2007.

Chopra, I. Review of the state of art of smart structures and integrated systems. *AIAA Journal*, 40:2145–2187, 2002.

Corke, T., Post, M., and Orlov, D. SDBD plasma enhanced aerodynamics: concepts, optimisation and applications. *Progress in Aerospace Sciences*, 43:193–217, 2007.

Debiasi, M., Bouremel, Y., Lu, Z., and Ravichandran, V. Deformation of the upper and lower surfaces of an aerofoil by macro fibre composite actuators. *Proceedings of the AIAA Applied Aerodynamics Conference*, San Diego, USA, 2013.

Dijkstra, B. G. and Bosgra, O. H. Extrapolation of optimal lifted system ILC solution, with application to a waferstage. *Proceedings of the American Control Conference*, Anchorage, USA, 2002.

Dong, J. and Verhaegen, M. On the equivalence of closed-loop subspace predictive control with LQG. *Proceedings of the IEEE Conference on Decision and Control*, Cancún, Mexico:4085–4090, 2008.

Dong, J. and Verhaegen, M. Cautious $\mathcal{H}_2$ optimal control using uncertain Markov parameters identified in closed loop. *Systems and Control Letters*, 58:378–388, 2009.

Dong, J., Verhaegen, M., and Holweg, E. Closed-loop subspace predictive control for fault tolerant MPC design. *Proceedings of the 17th World Congress*, Seoul, South Korea: 3216–3221, 2008.

Dong, J., Kulcsar, B., Van Wingerden, J. W., and Verhaegen, M. Closed-loop subspace predictive control for linear parameter varying systems (i) - the nominal case. *Proceedings of the European Control Conference*, Budapest, Hungary:2009–2014, 2009.

Dougherty, D. and Cooper, D. A practical multiple model adaptive strategy for single-loop MPC. *Control Engineering Practice*, 11:141–159, 2003.

Doyle, J. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, 1978.

Dunne, F., Pao, L. Y., Wright, A. D., Jonkman, B., and Kelley, N. Adding feedforward blade pitch control to standard feedback controllers for load mitigation in wind turbines. *Mechatronics*, 21:682–690, 2011.

Ekelund, T. Yaw control for reduction of structural dynamic loads in wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 85:241–262, 2000.

Emedi, Z. and Karimi, A. Fixed-order LPV controller design for LPV systems by convex optimisation. *Proceedings of the IFAC Symposium SSC*, Grenoble, France, 2013.

Favoreel, W. and De Moor, B. SPC: Subspace predictive control. *Proceedings of the 14th IFAC World Congress*, H(International Federation of Automatic Control):235–240, 1999.

Favoreel, W., De Moor, B., and Van Overschee, P. Model-free subspace-based LQG-design. *Proceedings of the American Control Conference*, San Diego, USA:3372–3376, 1999.

Fazel, M., Hindi, H., and Boyd, S. A rank minimisation heuristic with application to minimum order system approximation. *Proceedings of the American Control Conference*, pages 4734–4739, 2001.

Fischer, T., Rainey, P., Bossanyi, E. A., and Kühn, M. Study on control concepts suitable for mitigation of loads from misaligned wind and waves on offshore wind turbines supported on monopiles. *Wind Engineering*, 5:561–574, 2011.

Fischer, T., De Vries, W., Rainey, P., Schmidt, B., Argyriadis, K., and Kühn, M. Offshore support structure optimisation by means of integrated design and controls. *Wind Energy*, 15:99–117, 2012.

Formentin, S. and Savaresi, S. M. Virtual reference feedback tuning for linear parameter-varying systems. *IFAC Proceedings Volumes*, 18:10219–10224, 2011.

Formentin, S., Piga, D., Tóth, R., and Savaresi, S. M. Direct data-driven control of linear parameter-varying systems. *Proceedings of the IEEE Conference on Decision and Control*, Firenze, Italy:4110–4115, 2013.

Frueh, J. A. and Phan, M. Q. Linear quadratic optimal learning control (LQL). *International Journal of Control*, 10:832–839, 2000.

Gaunaa, M. Unsteady, two-dimensional potential-flow model for thin variable geometry aerofoils. *Wind Energy*, 13:167–192, 2010.

Gevers, M. A decade of progress in iterative process control design: From theory to practice. *Journal of Process Control*, 12(4):519–531, 2002.

Geyler, M. and Caselitz, P. Robust multivariable pitch control design for load reduction on large wind turbines. *Journal of Solar Energy Engineering*, 130:1–12, 2008.

Glezer, A. and Amitay, M. Synthetic jets. *Annual Review of Fluid Mechanics*, 34:503–529, 2002.

Goodwin, G. C., Ramadge, P. J., and Caines, P. E. Discrete-time multivariable adaptive control. *IEEE Transactions on Automatic Control*, 25(3):449–456, 1980.

Gunes, B., Van Wingerden, J. W., and Verhaegen, M. Tensor regression for LPV subspace identification. *Proceedings of the 17th IFAC Symposium on System Identificaiton, Beijing, China*, 2015.

Halko, N., Martinsson, P. G., and Tropp, J. A. Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 2011.

Hallouzi, R. and Verhaegen, M. Fault-tolerant subspace predictive control applied to a Boeing 747 model. *Journal of Guidance, Control and Dynamics*, 31(4):873–883, 2008.

Heinze, S. and Karpel, M. Analysis and wind tunnel testing of a piezoelectric tab for aeroelastic control applications. *Journal of Aircraft*, 43(6):1799–1804, 2006.

Henriksen, L. C., Tibaldi, C., and Bergami, L. HAWCStab2 user manual. *DTU Department of Wind Energy*, 2015.

Hjalmarsson, H. Iterative feedback tuning: an overview. *International Journal of Adaptive Control and Signal Processing*, 16:373–395, 2002.

Hodges, D. H. and Pierce, G. A. Introduction to structural dynamics and aeroelasticity. *Cambridge University Press*, 2011.

Houtzager, I., Van Wingerden, J. W., and Verhaegen, M. Recursive predictor-based subspace identification with application to the real-time closed-loop tracking of flutter. *IEEE Transactions on Control Systems Technology*, 20(4):934–949, 2012.

Hulskamp, A. W., Van Wingerden, J. W., Barlas, T. K., Champliaud, H., Van Kuik, G. A. M., and Verhaegen, M. Design of a scaled wind turbine with a smart rotor for dynamic load control experiments. *Wind Energy*, 14:339–354, 2011.

IEC61400-1. International electrotechnical commission, wind turbines part 1: Design requirements. *Geneva, Switzerland*, 2005.

INNWIND. www.innwind.eu. *Geneva, Switzerland*, 2012.

Jones, R. T. The unsteady lift of a wing of finite aspect ratio. *NACA Report 0681, Langley Memorial Aeronautical Laboratory*, 1939.

Kadali, R., Huang, B., and Rossiter, A. A data-driven subspace approach to predictive controller design. *Control Engineering Practice*, 11:261–278, 2003.

Kallesøe, B. S. A low-order model for analysing effects of blade fatigue load control. *Wind Energy*, 9:421–436, 2006.

Karutz, R. Untersuchung Generativer Fertigungsverfahren im Rotorblattbau von Kleinwindkraftanlagen. *Universität Stuttgart*, Bachelor Thesis, 2015.

Kessler, C. Active rotor control for helicopters: motivation and survey on higher harmonic control. *CEAS Aeronaut J*, 1:3, 2011.

King, F. A., Maurice, J. B., Fichter, W., Dieterich, O., and Konstanzer, P. In-flight rotorblade tracking control for helicopters using active trailing-edge flaps. *Journal of Guidance, Control and Dynamics*, 37(2):633–643, 2014.

Koumboulis, F. N., Tzamtzi, M. P., and Chamilothoris, G. E. Multivariable iterative feedback tuning – a step-wise safe switching approach. *Proceedings of the IEEE International Conference ETFA*, Patras, Greece:1354–1363, 2007.

Kwon, W. H. and Pearson, A. E. On feedback stabilisation of time-varying discrete linear systems. *IEEE Transactions on Automatic Control*, 23(3):479–481, 1978.

Lachenal, X., Daynes, S., and Weaver, P. M. A zero torsional stiffness twist morphing blade as a wind turbine load alleviation device. *Smart materials and structures*, 22, 2013.

Lackner, M. A. and Van Kuik, G. A. M. A comparison of smart rotor control approaches using trailing edge flaps and individual pitch control. *AIAA Aerospace Sciences Meeting including the New Horizons Forum*, Orlando, USA, 2009.

Laks, J., Pao, L. Y., Wright, A. D., Kelley, N., and Jonkman, B. The use of preview wind measurements for blade pitch control. *Mechatronics*, 21:668–681, 2011.

Laks, J. H., Pao, L. Y., and Wright, A. D. Control of wind turbines: past, present and future. *Proceedings of the American Control Conference*, St. Louis, USA, 2009.

Lara Quintanilla, A., Hulskamp, A. W., and Bersee, H. E. N. A high rate shape memory alloy actuator for aerodynamic load control on wind turbines. *Journal of Intelligent Material Systems and Structures*, 24(15):1834–1845, 2013.

Larimore, W. E. Canonical variate analysis in identification, filtering and adaptive control. *Proceedings of the 29th Conference on Decision and Control*, Honolulu, USA:596–604, 1990.

Larsen, T. J., Madsen, H. A., and Thomsen, K. Active load reduction using individual pitch, based on local blade flow measurements. *Wind Energy*, 8:67–80, 2005.

Lee, K. W. and Singh, S. N. Global robust control of an aeroelastic system using output feedback. *Journal of Guidance, Control and Dynamics*, 30(1):271–275, 2007.

Leith, D. J. and Leithead, W. E. Survey of gain-scheduling analysis and design. *International Journal of Control*, 73:1001–1025, 2000.

Liebst, B. S. A pitch control system for the KaMeWa wind turbine. *Journal of Dynamic Systems, Measurement and Control*, 107:47–52, 1985.

Ljung, L. System identification: Theory for the user. 1987.

Ljung, L. and Gunnarsson, S. Adaptation and tracking in system identification – a survey. *Automatica*, 26(1):7–21, 1990.

Ljung, L. and Söderström, S. Theory and practice of recursive identification. *Cambridge, Massachussetts, USA*, The MIT Press, 1983.

Longman, R. W. Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73:950–954, 2000.

Lovera, M. Recursive subspace identification based on projector tracking. *Proceedings of the IFAC Symposium on System Identification*, Rotterdam, the Netherlands, 2003.

Lovera, M., Bergamasco, M., and Casella, F. LPV modelling and identification: an overview. *Lecture Notes in Control and Information Sciences*, 437:3–24, 2013.

Lu, Q., Bowyer, R., and Jones, B. L. Analysis and design of coleman transform-based individual pitch controllers for wind turbine load reduction. *Wind Energy*, 18(8):1451–1468, 2015.

Maalawi, K. Y. A model for yawing optimisation of a wind turbine structure. *International Journal of Mechanical Sciences*, 49:1130–1138, 2007.

Manwell, J. F., McGowan, J. G., and Rogers, A. L. Wind energy explained: theory, design and application. 2002.

Markou, H., Andersen, P. B., and Larsen, G. C. Potential load reductions on megawatt turbines exposed to wakes using individual-pitch compensator and trailing-edge flaps. *Wind Energy*, 14:841–857, 2011.

Mathworks, I. Matlab, http;//www.mathworks.com/products/matlab. Natick, USA, 2014a.

Mathworks, I. Simulink, http;//www.mathworks.com/products/simulink. Natick, USA, 2014b.

Moriarty, P. J., Eggers Jr., A. J., Chaney, K., and Holley, W. E. Scale and lag effects on control of aerodynamic power and loads on a hawt rotor. *Journal of Solar Energy Engineering*, 123:339–345, 2001.

Mosca, E. Optimal, predictive and adaptive control. 1995.

Namik, H. and Stol, K. A. Individual blade pitch control of floating offshore wind turbines. *Wind Energy*, 13:74–85, 2010.

Namik, H. and Stol, K. A. Performance analysis of individual blade pitch control of offshore wind turbines on two floating platforms. *Mechatronics*, 21:691–703, 2011.

Narendra, K. S. and Parthasarthy, K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, 1990.

Navalkar, S. T. Iterative data-driven load control for flexible wind turbine rotors. *PhD Dissertation, Delft University of Technology*, 2016.

Neto, A. T., Dion, J. M., and Dugard, L. On the robustness of LQ regulators for discrete time systems. *IEEE Transactions on Automatic Control*, 37(10):1564–1568, 1992.

Nijssen, R. P. L. Fatigue life prediction and strength degradation of wind turbine rotor blade composites. *PhD Thesis, Delft University of Technology*, 2006.

Nishihara, R., Lessard, L., Recht, B., Packard, A., and Jordan, M. I. A general analysis of the convergence of ADMM. *Proceedings of the International Conference on Machine Learning*, Lille, France, 2015.

Ogata, K. Modern control engineering. 1997.

Oh, T. H., Matsuhita, Y., Tai, Y. W., and Kweon, I. S. Fast randomised singular value thresholding for nuclear norm minimisation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA:4484–4493, 2015.

Oomen, T., Van de Wijdeven, J., and Bosgra, O. Suppressing intersample behaviour in iterative learning control. *Automatica*, 45:981–988, 2009.

Ozdemir, A. A., Seiler, P. J., and Balas, G. J. Performance of disturbance augmented control design in turbulent wind conditions. *Mechatronics*, 21:634–644, 2011.

Pesmajoglous, S. and Graham, J. M. Yaw loads on horizontal axis wind turbines using an unsteady vortex lattice method. *Proceedings of the British Wind Energy Association Conference*, Nottingham, UK:109–115, 1992.

Ragheb, M. Wind generators history. 2014.

Rodden, M. and Johnson, E. MSC Nastran aeroelastic analysis user's guide. MSC Sotware Corp., Santa Ana, USA., 1994.

Rodrigues, M., Theilliol, D., Adam-Medina, M., and Sauter, D. A fault detection and isolation scheme for industrial systems based on multiple operating models. *Control Engineering Practice*, 16(2):225–239, 2006.

Rodrigues, S., Restrepo, C., Katsouris, G., Texeira Pinto, R., Soleimanzadeh, M., Bosman, P., and Bauer, P. A multi-objective optimisation framework for offshore wind farm layouts and electric infrastructures. *Energies*, 9(3):216, 2016.

Sayed, A. H. Fundamentals of adaptive filtering. 2003.

Selvam, K., Kanev, S., Van Wingerden, J. W., Van Engelen, T., and Verhaegen, M. Feedback-feedforward individual pitch control for wind turbine load reduction. *International Journal of Robust and Nonlinear Control*, 19:72–91, 2009.

Shamma, J. S. An overview of LPV systems. *Control of linear parameter-varying systems with applications*, Springer, US:3–26, 2012.

Shi, Y., Longman, W., and Nagashima, M. Small gain stability theory for matched basis function repetitive control. *Acta Astronautica*, 95:260–271, 2014.

Skogestad, S. and Postlethwaite, I. Multivariable feedback control. 1996.

Stol, K. A. Disturbance tracking control and blade load mitigation for variable-speed wind turbines. *Journal of Solar Energy Engineering*, 125:396–401, 2003.

Stol, K. A. and Balas, M. J. Periodic disturbance accommodating control for blade load mitigation in wind turbines. *Journal of Solar Energy Engineering*, 125:379–385, 2003.

Stol, K. A., Zhao, W., and Wright, A. D. Individual blade pitch control for the Controls Advanced Research Turbine (CART). *Journal of Solar Energy Engineering*, 128:498–505, 2006.

Stol, K. A., Moll, H. G., Bir, G., and Namik, H. A comparison of multi-blade coördinate transformation and direct periodic techniques for wind turbine control design. *Proceedings of the AIAA Aerospace Sciences Meeting including the New Horizons Forum*, Orlando, USA, 2009.

Stubkier, S. and Pedersen, H. C. Design, optimisation and analysis of a hydraulic soft yaw system for a 5 MW wind turbine. *Wind Engineering*, 35:529–550, 2011.

Tousain, R. and Van Casteren, D. Iterative learning control in a mass product: light on demand in DLP projection systems. *Proceedings of the American Control Conference*, New York, USA, 2007.

Uppal, F. J. and Patton, R. J. Neuro-fuzzy uncertainty decoupling: a multiple-model paradigm for fault detection and isolation. *International Journal of Adaptive Control and Signal Processing*, 19(4):281–304, 2005.

Van Dam, C. P., Chow, R., and Nakafuji, D. Y. Computational investigation of small deploying tabs and flaps for aerodynamic load control. *Journal of Physics: Conference Series*, 75:012027, 2007.

Van de Wijdeven, J. and Bosgra, O. H. Hankel iterative learning control for residual vibration suppression with MIMO flexible structure experiments. *Proceedings of the American Control Conference*, New York, USA, 2007.

Van de Wijdeven, J. and Bosgra, O. H. Using basis functions in iterative learning control: analysis and design theory. *International Journal of Control*, 83:661–675, 2010.

Van der Veen, G. J., Van Wingerden, J. W., Bergamasco, M., Lovera, M., and Verhaegen, M. Closed-loop subspace identification: an overview. *IET Control Theory and Applications*, 7(10):1339–1358, 2013.

Van der Velden, B. J. C. H., Oomen, T., and Heertjes, M. F. Constrained iterative feedback tuning for robust high-precision motion control. *Proceedings of the IFAC World Congress*, Cape Town, South Africa, 2014.

Van Engelen, T. G. Design model and load reduction assessment for multi-rotational mode individual pitch control (higher harmonics control). *Proceedings of the European Wind Energy Conference*, Athens, Greece, 2006.

Van Engelen, T. G. Control design based on aero-hydro-servo-elastic linear models from TURBU (ECN). *Proceedings of the European Wind Energy Conference*, Milan, Italy, 2007.

Van Solingen, E., Navalkar, S. T., and Van Wingerden, J. W. Experimental wind tunnel testing of linear individual pitch control for two-bladed wind turbines. *Journal of Physics: Conference Series*, 524: 012056, 2014.

Van Wingerden, J. W., Hulskamp, A., Barlas, T. K., Marrant, B., Van Kuik, G. A. M., Molenaar, D. P., and Verhaegen, M. On the proof of concept of a 'smart' wind turbine rotor blade for load alleviation. *Wind Energy*, 11:265–280, 2008.

Van Wingerden, J. W., Gebraad, P. M. O., and Verhaegen, M. LPV identification of an aeroelastic flutter model. *Proceedings of the IEEE Conference on Decision and Control*, Atlanta, USA, 2010a.

Van Wingerden, J. W., Hulskamp, A., Barlas, T. K., Houtzager, I., Bersee, H. E. N., Van Kuik, G. A. M., and Verhaegen, M. Two-degree of freedom active vibration control of a prototyped smart rotor. *IEEE Transactions on Control Systems Technology*, pages 284–296, 2010b.

Verelst, D. R., Larsen, T. J., and Van Wingerden, J. W. Wind tunnel tests of a free yawing downwind wind turbine. *Proceedings of the Science of Making Torque from Wind*, Oldenburg, Germany, 2012.

Verhaegen, M. and Hansson, A. N2SID: Nuclear norm subspace identification of innovation models. *Automatica*, 72:57–63, 2016.

Verhaegen, M. and Verdult, V. Filtering and system identification: A least squares approach. H, 2003.

Versteijlen, W. G., Metrikine, A. V., and Van Dalen, K. N. A method for identification of an effective Winkler foundation for large-diameter offshore wind turbine support structures based on in-situ measured small strain soil response and 3d modelling. *Engineering Structures*, 124:221–236, 2016.

Wang, L., Gawthrop, P., Owens, D. H., and Rogers, E. Switched linear predictive controllers for periodic exogenous signals. *International Journal of Control*, 83:661–675, 2010.

Woodley, B. R., How, J. P., and Kosut, R. L. Subspace based direct adaptive $\mathcal{H}_\infty$ control. *International Journal of Adaptive Control and Signal Processing*, 15:535–561, 2001.

Woods, B. K. S., Bilgen, O., and Friswell, M. I. Wind tunnel testing of the fishbone active camber morphing concept. *Journal of Intelligent Material Systems and Structures*, pages 1–14, 2014.

Wright, A. D. Modern control design for flexible wind turbines. *NREL/TP-500-35816*, 2004.

Wright, A. D. and Balas, M. J. Design of controls to attenuate loads in the controls advanced research turbine. *Proceedings of the ASME Wind Energy Symposium*, 2004.

XEMC-Darwind, B. V. Website XEMC-Darwind, www.xemc-darwind.com. 2013.

Ye, Y. and Wang, D. DCT basis function learning control. *IEEE/ASME Transactions on Mechatronics*, 10:449–454, 2005.

Zhang, Z., Staino, A., Basu, B., and Nielsen, S. R. K. Performance evaluation of full-scale tuned liquid dampers (TLDs) for vibration control of large wind turbines using real-time hybrid testing. *Engineering Structures*, 126:417–431, 2016.

SUMMARY

Wind energy has reached a high degree of maturity: for wind-rich onshore locations, it is already competitive with conventional energy sources. However, for low-wind, remote and offshore regions, research efforts are still required to enhance its economic viability. While it is possible to reduce the cost of energy by upscaling wind turbines, it is believed that we may be approaching a plateau in turbine size. Beyond this plateau, the material costs associated with the high dynamic turbine loads would outweigh the benefits of increased energy capture. To postpone this plateau, research is currently being carried out in the active control of loads for lightweight, flexible rotors.

Traditional control for wind turbines involves the use of fixed-structure low order controllers, the gains of which are often hand-tuned separately for each turbine class. However, for the increasingly multivariable plant, such time-invariant approaches may no longer yield good performance. As such, the thesis focusses specifically on data-driven control for these flexible turbines.

First, different data-driven approaches in the literature are evaluated and categorised as two-step approaches; which involves distinct online identification and control steps; and direct approaches, which uses data to iteratively tune fixed-structure controller gains. The approaches need to be modified to be made tractable in real time for implementation on wind turbines.

For time-varying plants, such as wind turbines, it is often interesting to perform identification repeatedly over time for the two-step data-driven approach. Conventional recursive identification is extended in this thesis through the use of the nuclear norm. The benefit of the nuclear norm is evident in that it increases responsiveness of the algorithm, through the suppression of the effect of external noise. Identification can be readily combined with repetitive control for reducing periodic loads in the Subspace Predictive Repetitive Control (SPRC) technique. SPRC can be performed in a restricted basis function subspace, thus reducing the computational complexity and providing smooth control signals. The control law is stabilising and performs well as long as the identification converges to relatively good estimates, and the system dynamics change slowly.

For varying wind speed, the approach above would require continuous reïdentification. As an alternative, a direct data-driven approach, Iterative Feedback Tuning (IFT) has been extended to gain-schedule tuning and for designing a Linear Parameter-Varying (LPV) controller for an LPV plant. This requires an exponential increase in the number of tuning experiments per iteration; however, structure can be used to reduce computational complexity. IFT-LPV converges to a locally optimal low-order controller.

These data-driven approaches are evaluated for the load control of flexible rotors. A review of the state of the art shows that, for the low-frequency region of the load spectrum, full-span pitch control has demonstrable control authority. For higher frequencies, among the new actuators, it is found that trailing-edge flaps have the highest level of technological maturity.

Aeroservoelastic simulations are carried out to show the potential of the data-driven approaches. SPRC is able to adaptively tune itself to achieve average blade load reductions close to those achieved by conventional approaches under similar conditions. For these load reductions the actuator duty is roughly half of that with the conventional approach. IFT-LPV has been used to tune a feedforward controller that works on similar basis functions scheduled on the azimuth. It can provide the correct control action irrespective of wind conditions. To expand the load control design space, pitch control is designed to stabilise an upwind turbine in yaw, without the yaw drive. This approach enables a trade-off between blade and support structure loads.

SPRC is then investigated with wind tunnel experiments for pitch control of a scaled wind turbine. It reduces deterministic loads by over 60% with strict control over the pitch activity, and can also compensate for asymmetric blade control authority and changed operating conditions adaptively. Further, on this setup, the concept of IPC has been shown to perform yaw stabilisation for an upwind turbine for the first time. The setup blades are then redesigned to include free-floating trailing-edge flaps. First-principles models are set up for the system, and it is found that the system shows a low wind-speed form of flutter; this is validated experimentally. Recursive identification, using the nuclear norm, is able to track the unstable mode damping, and detect flutter twice as fast as conventional methods. Finally, a feedforward controller is tuned using IFT for combined pitch and flap control; the load peaks at 1P and 2P are almost entirely attenuated. IFT is also able to tune an linear gain schedule for operation across a range of wind speeds.

It is concluded that iterative methods for data-driven control perform well for the highly uncertain control problem of flexible rotor load alleviation. For this, use has to be made of the structure of the problem. The two-step approach (such as SPRC), with combined recursive identification and control law synthesis, provides a convex first approximation of the desired controller. With the help of direct approaches, (like IFT-LPV), the controller structure can be reduced and fine-tuned to improve the control performance. Such quasi-feedforward data-driven approaches can complement the existing turbine control structure and achieve enhanced load control performance for flexible rotors.

SAMENVATTING

Windenergie heeft op dit moment al een hoge mate van volwassenheid bereikt: windenergie vanuit windrijke locaties op land is nu al competitief met conventionele energiebronnen. Echter, er zijn nog steeds onderzoeksinspanningen nodig om de economische levensvatbaarheid van windenergie te verbeteren voor locaties met minder wind, op afgelegen plekken, en op zee. Hoewel het mogelijk is om de energiekosten te verminderen door de windturbines verder op te schalen, is het mogelijk dat de groei van de turbines in de toekomst gaat afvlakken. Op dat moment zou de hogere energieopbrengst van grotere turbines niet meer opwegen tegen de kosten van het extra benodigde materiaal om de hogere dynamische belasting op de turbine aan te kunnen. Om deze afvlakking uit te stellen wordt nu onderzoek gedaan naar de actieve regeling van belastingen op lichtgewicht, flexibele rotors.

Voor commerciële windturbines worden regelaars gebruikt met een vaste structuur en een lage orde. De versterkingsfactoren van deze regelaars worden vaak van tevoren handmatig afgesteld, apart voor elke turbineklasse. Voor meer multivariabele systemen, zoals flexibele turbines, levert een dergelijke tijdsinvariante benadering mogelijk geen goede prestaties. Daarom richt dit proefschrift zich op datagestuurde regelaars.

Allereerst zijn datagestuurde benaderingen in de literatuur onderverdeeld in twee categorieën. De methodes in de eerste categorie zijn tweetrapsmethodes, met een online identificatiestap en een regelingsstap. De methodes in de tweede categorie gebruiken een directe aanpak waarin de data iteratief wordt gebruikt om een regelaar met een vaste structuur af te stellen. Voor beiden methodes geldt dat ze verder dienen te worden aangepast om ze realtime te kunnen uitvoeren op een windturbine.

Voor systemen die variëren in de tijd, zoals windturbines, is het bij de tweetrapsmethodes voor datagestuurde regeling vaak profijtelijk om de identificatiestap herhaaldelijk uit te voeren. In dit onderzoek wordt deze conventionele identificatie uitgebreid met het gebruik van een nucleaire norm. Hiermee kan het reactievermogen van het algoritme worden verhoogd door de invloed van externe ruis te onderdrukken. In de zogenaamde Subspace Predictive Repetitive Control (SPRC) techniek is de identificatie rechtstreeks gecombineerd met repetitieve regeling voor het verminderen van periodieke lasten. SPRC kan worden uitgevoerd met een beperkte basisfunctiedeelruimte. Zo kan de rekenkundige complexiteit worden verminderd en kunnen de stuursignalen gladder en regelmatiger worden gemaakt. De resulterende besturingsregels zijn stabiliserend en presteren goed zolang de identificatie convergeert tot een relatief goede schatting, en de systeemdynamica slechts langzaam verandert.

Voor een veranderlijke windsnelheid zou deze aanpak vereisen dat de identificatie continu herhaald wordt. Er is een alternatieve methode ontwikkeld door een directe benadering genaamd Iteratieve Feedback Tuning (IFT) uit te breiden met mogelijkheden om een regelaaraanpassingsschema af te stemmen, of een Lineair Parameter Variërend (LPV) regelsysteem te genereren. Hiervoor is een exponentiële toename van het aantal

afstemmingsexperimenten per iteratie nodig. Echter, de structuur van het systeem kan worden uitgebuit om de rekenkundige complexiteit te verminderen. IFT-LPV convergeert naar een plaatselijk optimale regelaar.

Deze datagestuurde benaderingen worden beoordeeld door de regulering van belastingen op flexibele rotors in ogenschouw te nemen. Een overzicht van de stand van de techniek laat zien dat bladhoekregeling over de volledige lengte van het blad effectief is in het laagfrequente deel van het spectrum van de belastingen. Een beschouwing van nieuwe soorten aandrijvingen die zich richten op regeling van het hoogfrequente deel van de belastingen laat zien dat technieken met flappen aan de achterrand van het turbineblad in het meest volwassen stadium van ontwikkeling zijn.

Er zijn aeroservoelastische simulaties uitgevoerd om het regelingspotentieel aan te tonen. SPRC is in staat om zichzelf adaptief af te stellen, en benadert daarbij de gemiddelde bladbelastingverlagingen die een conventionele methode in vergelijkbare omstandigheden behaalt. Daarbij is de arbeidscyclus van de actuator ongeveer de helft van die van de conventionele methode. IFT-LPV werd gebruikt om een voorwaartsregelaar af te stellen die werkt met basisfuncties die van de azimuthoek afhangen. Deze aanpak kan de juiste regelactie geven ongeacht de windomstandigheden. Om de ontwerpruimte van de belastingregelaar uit te breiden is een bladhoekregelaar gemaakt die een bovenwindse windturbine met een gierhoekafwijking kan stabiliseren, zonder daarbij de giermotor te gebruiken. Met deze aanpak kan een afweging worden gemaakt tussen belastingen op de bladen en die op de ondersteunende constructie.

Vervolgens worden windtunnelexperimenten gedaan met SPRC bladhoekregeling op een geschaalde windturbine. Deze aanpak vermindert deterministische belastingen met meer dan 60%, waarbij de actuatoractiviteit strikt gereguleerd wordt, en er adaptief gecompenseerd kan worden voor een asymmetrische bladregelingseffectiviteit en voor veranderde werkomstandigheden. Verder is er voor het eerst individuele bladregeling voor gierhoekstabilisatie van een bovenwindse windturbine gedemonstreerd. De bladen zijn vervolgens opnieuw ontworpen met vrijzwevende flappen aan de achterrand van het blad. Uit natuurkundige modellen blijkt dat het blad bij een lage windsnelheid flutter vertoont, en dit is ook experimenteel gevalideerd. Recursieve identificatie met de nucleaire norm kan flutter tweemaal sneller opsporen dan conventionele methoden. Tenslotte wordt IFT gebruikt om een voorwaartsregelaar af te stellen die zowel de bladhoek als de flap regelt. Deze regelaar kan de belastingpieken bij 1P en 2P vrijwel geheel uitdempen. IFT kan ook een lineair aanpassingsschema voor de versterkingsfactoren genereren voor een bepaald variatiebereik van de windsnelheid.

De conclusie is dat iteratieve methoden voor datagestuurde regeling goed presteren op het belastingreguleringsprobleem voor flexibele rotors, ondanks dat de onzekerheden in dit probleem groot zijn. Hierbij dient de structuur van het probleem te worden uitgebuit met behulp van basisfuncties. Een tweetrapsbenadering (zoals SPRC) met een combinatie van recursieve identificatie en regelingswetsynthese biedt een convexe eerste benadering van de gewenste regelaar. Met behulp van directe methoden (zoals IFT-LPV) kan de structuur van de regelaar worden verkleind, en vervolgens kan de regelaar verder worden afgesteld om de prestaties te verbeteren. Dergelijke datagestuurde benaderingen met quasi-voorwaartskoppeling kunnen de bestaande turbineregeling aanvullen en de prestaties van belastingregulering voor flexibele rotors verbeteren.

LIST OF PUBLICATIONS

JOURNAL PAPERS

Bernhammer, L. O., Navalkar, S. T., De Breuker, R., and Karpel, M. Experimental and numerical investigation of an autonomous flap for load alleviation. *Journal of Aircraft*, Article in Press, 2016.

Navalkar, S. T. and Van Wingerden, J. W. Nuclear norm-based recursive subspace identification for wind turbine flutter detection. *IEEE Transactions on Control Systems Technology*, Submitted for review, 2016.

Navalkar, S. T., Van Wingerden, J. W., Van Solingen, E., Oomen, T., Pasterkamp, E., and Van Kuik, G. A. M. Subspace predictive repetitive control to mitigate periodic loads on large scale wind turbines. *Mechatronics*, 24(8):916–925, 2014.

Navalkar, S. T., Van Solingen, E., and Van Wingerden, J. W. Wind tunnel testing of subspace predictive repetitive control to variable pitch wind turbines. *IEEE Transactions on Control Systems Technology*, 23(6):2101–2116, 2015.

Navalkar, S. T., Bernhammer, L. O., Sodja, J., Van Kuik, G. A. M., and Van Wingerden, J. W. Wind tunnel tests with combined pitch and free-floating flap control: data-driven iterative feedforward tuning. *Wind Energy Science*, Under review, 2016.

Smit, J., Bernhammer, L. O., Navalkar, S. T., Bergami, L., and Gaunaa, M. Sizing and control of trailing edge flaps on a smart rotor for maximum power generation in low fatigue wind regimes. *Wind Energy*, 19(4):607–624, 2016.

CONFERENCE PAPERS

Barlas, T. K., Jost, E., Pirrung, G., Tsiantas, T., Riziotis, V., Navalkar, S. T., Lutz, T., and Van Wingerden, J. W. Benchmarking aerodynamic prediction of unsteady rotor aerodynamics of active flaps on wind turbine blades using ranging fidelity tools. *Proceedings of the Science of Making Torque from Wind*, Munich, Germany, 2016.

Bernhammer, L. O., Navalkar, S. T., Sodja, J., De Breuker, R., and Karpel, M. Experimental investigation of an autonomous flap for laod alleviation. *Proceedings of the AIAA/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Kissimmee, USA, 2015a.

Bernhammer, L. O., Navalkar, S. T., Sodja, J., De Breuker, R., and Karpel, M. Experimental and numerical study of an autonomous flap. *Proceedings of the International Forum on Aeroelasticity and Structural Dynamics*, Saint Petersburg, Russia, 2015b.

Macquart, T., Navalkar, S. T., and De Breuker, R. Application of a new decoupling loop shaping control strategy for load alleviation of smart rotor blades equipped with multiple trailing edge flaps. *Proceedings of the International Conference on Adaptive Structures and Technologies*, Kobe, Japan, 2015.

Navalkar, S. T. and Van Wingerden, J. W. Iterative feedback tuning of an LPV feedforward controller for wind turbine load alleviation. *Proceedings of the IFAC Workshop on Linear Parameter Varying Systems*, Grenoble, France:207–212, 2015.

Navalkar, S. T. and Van Wingerden, J. W. Nuclear norm-enhanced recursive subspace identification: closed-loop estimaiton of rapid variations in system dynamics. *Proceedings of the American Control Conference*, Boston, USA, 2016.

Navalkar, S. T., Van Wingerden, J. W., and Oomen, T. Subspace predictive repetitive control with lifted domain identification for wind turbine individual pitch control. *Proceedings of the IFAC World Congress*, Cape Town, South Africa, 2014a.

Navalkar, S. T., Van Wingerden, J. W., and Van Kuik, G. A. M. Individual blade pitch for yaw control. *Journal of Physics 524 (1), the Science of Making Torque from Wind Conference*, Copenhagen, Denmark, 2014b.

Navalkar, S. T., Van Wingerden, J. W., Van Solingen, E., Oomen, T., and Van Kuik, G. A. M. Subspace predictive repetitive control for wind turbine load alleviation using trailing edge flaps. *Proceedings of the American Control Conference*, Portland, USA, 2014c.

Navalkar, S. T., Oomen, T., and Van Wingerden, J. W. IFT-LPV: Data-based tuning of fixed structure controllers for LPV systems. *Proceedings of the IFAC Symposium on System Identification*, Beijing, China:721–726, 2015a.

Navalkar, S. T., Van Wingerden, J. W., Fleming, P. A., and Van Kuik, G. A. M. Integrating robust lidar-based feedforward with feedback control to enhance speed regulation of floating wind turbines. *Proceedings of the American Control Conference*, Chicago, USA: 3070–3075, 2015b.

Navalkar, S. T., Bernhammer, L. O., Sodja, J., Slinkman, C. J., Van Wingerden, J. W., and Van Kuik, G. A. M. Aeroelastic design and LPV modelling of an experimental wind turbine blade equipped with free-floating flaps. *Proceedings of the Science of Making Torque from Wind*, Munich, Germany, 2016.

Smit, J., Bernhammer, L. O., Navalkar, S. T., Bergami, L., and Gaunaa, M. Sizing and control of trailing edge flaps on a smart rotor for maximum power generation in low fatigue wind regimes. *Proceedings of the ASME Wind Energy Symposium*, National Harbour, USA, 2014.

Van Solingen, E., Navalkar, S. T., and Van Wingerden, J. W. Experimental wind tunnel testing of linear individual pitch control for two-bladed wind turbines. *Journal of Physics: Conference Series*, 524(1), 2014.

Visser, M., Navalkar, S. T., and Van Wingerden, J. W. LPV model identification for flutter prediction: a comparison of methods. *Proceedings of the IFAC Workshop on Linear Parameter Varying Systems*, Grenoble, France:121–126, 2015.

CURRICULUM VITÆ

Sachin Navalkar was born on 27 November 1986 in Mumbai, India. He studied Mechanical Engineering at the University of Mumbai, acquiring his Bachelor's degree in 2008, graduating in the top 0.3% of the university with a final thesis on the design and manufacture of a powered knee-foot-ankle orthosis for assisted locomotion.

In 2009-2010, he studied Sustainable Energy Technology at the Delft University of Technology, the Netherlands. He received his Master's degree cum laude in this specialisation with honours in the field of wind turbine load control. His Master thesis was entitled 'Subspace predictive control for smart wind turbine rotors'. In this period, he was a founding member of the TU Delft student organisation the 'Energy Club', energyclub.nl.

In 2010-2012, he worked for the wind turbine manufacturer Vestas Wind Systems, as a development engineer focussed on rotating components. In this period, he lived in Chennai, India, Århus, Denmark, and Hamburg, Germany, for a period of eight months each. His work was primarily centred around the dynamic modelling of drive train components, and around root cause analysis of their behaviour based on field data.

In 2012, he started his PhD work in the Delft University of Technology, investigating the viability of data-driven approaches to load attenuation in modern wind turbines, under the supervision of dr. ir. Jan-Willem van Wingerden and Prof. dr. ir. Gijs van Kuik. The primary objective of this work has been the extension of data-driven control theory to real-time implementation using wind tunnel experiments. Specific emphasis has been laid on exploiting the structure in the problem and focussing on a pseudo-feedforward approach to obviate stability considerations. The results of this PhD work are presented in this thesis.

Since late 2016, he has been working for Siemens Wind Power in Copenhagen, Denmark, as a Control Engineer dedicated to controller optimisation for new wind turbine fleets.

ACKNOWLEDGEMENTS

And there you were: the primordially ugly littly boy, ensaddened because you were knocked out for a duck, and because you'd just failed your arts and crafts exam. And you ran to your mother.
"Don't worry, handsome," she said. "You can do anything if you put your mind to it."
So you did. You found your passion in bricklaying, and you got good at it. You could even stack ten layers on top of each other, before they fell down again.
And then you saw the world. The Great Wall, Eiffel's Tower, the Palace in the Lake. Awestruck and intimidated, you wondered how many years, nay how many lifetimes it would take you to stack bricks that high and that beautiful.
And then you saw the world again; but this time you saw the people. People who used mortar to hold bricks together, people who used steel in place of brick. People who didn't lay bricks, but conjured castles in their dreams: some of which materialised; most of which didn't, but made the world that much better for just having been.
And you were awestruck once more, awestruck and inspired. And you ran to your mother.
"Go on, handsome," she said. "You can do anything if you don't go it alone."

Dear acknowledgee,

Thank you for helping me compile this book, but thank you first for being awesome. In the last four years, every time I've looked around, I've found reason to believe that we're moving in the right direction; I've found reason to believe that it's because of you.

To start off, I'd like to thank Jan-Willem, for the chance that you gave me to work on this cool topic. But then also for the fact that when I came to you, my PhD problems automatically became 'our' problems. And also for helping me realise (and digest) that not every problem has a solution. I'd like to thank Tom for thinking along with me, and challenging me to get my story straight every time we met. And also Gijs, for providing the much-needed wind energy insights to keep me on track and helping me stick to the big picture view. It's amazing to talk with someone who's spent so much time and understands so much about the evolution of wind energy, over its first decades.

My initial years into the PhD were greatly aided by the data-driven meetings, where Michel helped me find my bearings. I'll always be grateful for the insight provided by the windcontrol guys, Gijs, Patricio, Yue, Edwin, Pieter and Hildo. Edwin, I think getting the inordinate number of things to fall in place for a few days of wind tunnel testing would have proved utterly impossible but for your help. And I'd like to repeat my thanks to Pieter for his can-do-itiveness and the sheer number of times I've asked for his help. And now I'm sure that Yasin, Sebastiaan, Sjoerd and Bart will hold the standard high.

The wind tunnel experiments would not even have taken off the ground if it were not for Kees. Thank you for not compromising on quality, and for tolerating me when I came up with the absurdest of plans that you somehow worked into our turbine; all the while managing to pull together one of the most polished and professional setups I've

ever seen. And thank you to Will, for that last push before the big experiments, where we all three sat together trying to figure out which way a turbine should rotate.

Thanks are due to Kiran, Heleen, Marieke and Kitty, for being so fantastically efficient and friendly. And thank you to Sylvia for making me fit into the Wind Energy group.

I've great respect for the DCSC guys; it's a whole new mountain to derive abstract mathematical proofs and propositions that hold in general. And I also have great respect for their foosball skills. We've had awesome times together: with Subbu, Yashar, Mohamad, Noortje, Renshi, Sadegh at badminton, Winterberg and Luxembourg; the IFAC trip and the board game nights with hey Hildo, Elisabeth, Ruxandra, Arne (where's Subbu?) and then again with Hans, Dean, Paolo, Laurens, Vishal, Reinier and Baptiste. And I'll always remember hurtling down an Italian ski slope with my toes clenched to keep my balance, chased by shouts of "Stop skiing like a n00b!" I've had fun with my cabinmates, Yu Hu, Yue and Nico. And had to be kept on my toes by the greatest Master students ever, Jeroen, Mart, Bhagyashri and Dylan.

The Wind Energy group at Aerospace has always impressed me with the sheer diversity of work that they carry out: Gerard, Carlos, Wim, Nando, Uwe, Michiel - I think you've had a tremendous influence on spreading wind energy knowledge, especially to students (including myself). I'd like to thank Lars for letting me in on his work, and helping me take it forward. It's amazing that you can always find a different take on a wind turbine when you talk any of these guys, so I would like to thank Ricardo, Etana, Maarten, René, Ye, Daniel, Ashim, Gaél, Zi, and the new crop of younglings, Sebastian, Sebastian, Tom, Mikko, Carlos, Navneeth, Vinit and Juan. I'd like to thank Thijs for your patience in helping me get used to CFD and your lead in the Smart Rotor meetings. Jurij, it's been fantastic working with you: it's rare to see a combination of the craziest and wildest ideas combined with the ability to work with your hands. And thanks to Roeland, Noud and Paul for the help (and interest) in my project.

Talking of unsung heroes, I think the guys at aQysta deserve a round of applause: Jaime, Lennart, Tim, Pratap, Fred, Rahul, Tei, Andhika. It's incredible and refreshing the passion that you put into your work, and I'm glad I got a chance to work with you – here's to seeing a Barsha pump along every river stretch in the near future.

The Spartans have been a great part of the journey. Bart, it's amazing that with your goofy grin you're able to motivate us into doing the craziest stuff: be it a kilometre of lunging or planking in the hail, you actually made an obstacle run look like a piece of cake. And kudos to the other Spartans for making it tolerable: Paul, Koushik, Else, Lucas, Lukas, Richard, Kristen, Eduardo, Xavier.

And to wrap up, these four years would have been incomplete without AvS: TT, Tiggy, Chocka, Reshu, Subbu, Saish, Sid, Kalsi, Srijith, and the various hangers-on; Yash, Raj, Prakhar. I'm not just talking about the food, although I've to admit that getting back to AvS after spending 12 hours in the university, and smelling the most incredible बास at the end of the corridor feels almost like coming home. It was equally as much about listening to your drivel wash over my head at the end of the day while playing Temple Run: there's no group of people I'm happier being completely asocial with.

There are people I don't mention; they go above and beyond these four years. But I'm certain they don't need reassurance that they're as much part of me as I am of them.

Thanks guys!